Mixtures of Heterogeneous Experts

Taiwo Gabriel Omomule

Dissertation presented for the degree of Doctor of Philosophy
in the Computer Science Division, Faculty of Science
at Stellenbosch University

Supervisor: Prof. A. P. Engelbrecht

December 2022

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2022

Abstract

This research considers the problem of the *No-Free-Launch-Theorem*, which states that no one machine learning algorithm performs best on all problems due to algorithms having different inductive biases. Another problem is that the combinations of experts of the same type, referred to as a mixture of homogeneous experts, do not capitalize on the different inductive biases of different machine learning algorithms. Research has shown that mixtures of homogeneous experts deliver improved accuracy compared to that of the base experts in the mixture. However, the predictive power of a homogeneous mixture of experts is still limited by the inductive bias of the algorithm that makes up the mixture of experts. Therefore, this research proposes the development of mixtures of heterogeneous experts through the combination of different machine learning algorithms to take advantage of the strengths of the machine learning algorithms and to reduce the adverse effects of the inductive biases of the different algorithms.

A set of different machine learning algorithms are selected to develop four different types of mixtures of experts in the research. Empirical analyses are performed using non-parametric statistical tests to compare the generalization performance of the ensembles. The comparison is carried out to investigate the performance of the homogeneous and heterogeneous ensembles in a number of modelling studies examined on a set of classification and regression problems using selected performance measures. The problems represent varying levels of complexity and characteristics to determine the characteristics and complexities for which the heterogeneous ensembles outperform

homogeneous ensembles.

For classification problems, the empirical results across six modelling studies indicate that heterogeneous ensembles generate improved predictive performance compared to the developed homogeneous ensembles by taking advantage of the different inductive biases of the different base experts in the ensembles. Specifically, the heterogeneous ensembles developed using different machine learning algorithms, with the same and different configurations, showed superiority over other heterogeneous ensembles and the homogeneous ensembles developed in this research. The ensembles achieved the best and second-best overall accuracy rank across the classification datasets in each modelling study.

For regression problems, the heterogeneous ensembles outperformed the homogeneous ensembles across five modelling studies. Although, a random forest algorithm achieved competitive generalization performance compared to that of the heterogeneous ensembles. Based on the average ranks, the heterogeneous ensembles developed using different machine learning algorithms where the base members consist of the same and different configurations still produced better predictive performance than a number of heterogeneous ensembles and homogeneous ensembles across the modelling studies.

Therefore, the implementation of a mixture of heterogeneous experts removes the need for the computationally expensive process of finding the best performing homogeneous ensemble. The heterogeneous ensembles of different machine learning algorithms are consistently the most or one of the most accurate ensembles across all classification and regression problems. This is attributed to the advantage of capitalizing on the inductive biases of the different machine learning algorithms and the different configurations of the base members in the ensembles.

Opsomming

Hierdie navorsing oorweeg die probleem van die *No-Free-Launch-Theorem*, wat aandui dat geen masjienleer algoritme die beste op alle probleme presteer nie, as gevolg van algoritmes wat verskillende induktiewe vooroordele het. Nog 'n probleem is dat die kombinasies van kundiges van dieselfde tipe, waarna verwys word as 'n mengsel van homogene kundiges, nie munt slaan uit die verskillende induktiewe vooroordele van verskillende masjienleer algoritmes nie. Navorsing het getoon dat mengsels van homogene kundiges verbeterde akkuraatheid lewer in vergelyking met dié van die basis kundiges in die mengsel. Die voorspellings krag van 'n homogene mengsel van kundiges word egter steeds beperk deur die induktiewe vooroordeel van die algoritme waaruit die mengsel van kundiges bestaan. Daarom stel hierdie navorsing die ontwikkeling van mengsels van heterogene kundiges voor deur die kombinasie van verskillende masjienleer algoritmes om voordeel te trek uit die sterk punte van die masjienleer algoritmes en om die nadelige effekte van die induktiewe vooroordele van die verskillende algoritmes te verminder.

'n Stel verskillende masjienleer algoritmes word gekies om vier verskillende tipes mengsels van kundiges in die navorsing te ontwikkel. Empiriese ontledings word uitgevoer met behulp van nie-parametriese statistiese toetse om die veralgemenings prestasie van die ensembles te vergelyk. Die vergelyking word uitgevoer om die prestasie van die homogene en heterogene ensembles te ondersoek in 'n aantal modellering studies wat ondersoek is op 'n stel klassifikasie- en regressie probleme deur gebruik te maak van

geselekteerde prestasiemaatstawwe. Die probleme verteenwoordig verskillende vlakke van kompleksiteit en kenmerke om die kenmerke en kompleksiteite te bepaal waarvoor die heterogene ensembles beter as homogene ensembles presteer.

Vir klassifikasie probleme dui die empiriese resultate oor ses modellering studies aan dat heterogene ensembles verbeterde voorspellende prestasie genereer in vergelyking met die ontwikkelde homogene ensembles deur voordeel te trek uit die verskillende induktiewe vooroordele van die verskillende basis kundiges in die ensembles. Spesifiek, die heterogene ensembles wat ontwikkel is deur gebruik te maak van verskillende masjienleer algoritmes, met dieselfde en verskillende konfigurasies, het superioriteit getoon bo ander heterogene ensembles en die homogene ensembles wat in hierdie studie ontwikkel is. Die ensembles het die beste en tweede beste algehele akkuraatheid rangorde oor die klassifikasie datastelle in elke modellering studie behaal.

Vir regressie probleme het die heterogene ensembles beter gevaar as die homogene ensembles oor vyf modellering studies. Alhoewel, 'n ewekansige woud algoritme het mededingende veralgemenings prestasie behaal in vergelyking met die van die heterogene ensembles. Gebaseer op die gemiddelde geledere, het die heterogene ensembles ontwikkel deur gebruik te maak van verskillende masjienleer algoritmes waar die basis lede uit dieselfde bestaan en verskillende konfigurasies steeds beter voorspellende prestasie gelewer het as 'n aantal heterogene ensembles en homogene ensembles oor die modellering studies heen.

Daarom verwyder die implementering van 'n mengsel van heterogene kundiges die behoefte aan die rekenkundig duur proses om die beste presterende homogene ensemble te vind. Die heterogene ensembles van verskillende masjienleer algoritmes is konsekwent die meeste of een van die akkuraatste ensembles oor alle klassifikasie- en regressie probleme. Dit word toegeskryf aan die voordeel om munt te slaan uit die induktiewe vooroordele van die verskillende masjienleer algoritmes en die verskillende konfigurasies van die basis lede in die ensembles.

Acknowledgements

Thanks be to the Almighty God for the successful completion of this research. The author wishes to acknowledge the contributions of the following people and institutions to the success of this work:

- My supervisor, Professor Andries Engelbrecht, for his guidance, support and dedication throughout this research. I appreciate the lifetime opportunity given to me to gain significant research experience and hone technical skills under the supervision of an established researcher like you. Thank you for your inspiration to successfully complete my doctoral research.
- My wife, for your love, patience and peace throughout my studies.
- My mother and siblings, for your advise and unwavering spiritual support.
- The Data Science Research Group, for creating a platform to collaborate and communicate research knowledge which contributed positively to the completion of this research.
- The Computer Science Division at Stellenbosch University, for the space and facilities provided to peacefully work and complete this research and thesis.

Dedications

This research is dedicated to God and my family.

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
ML	Machine Learning
SVM	Support Vector Machine
NN	Neural Network
RF	Random Forest
RFSM	Random Feature Subspace Method
***k*NN**	*k*-Nearest Neighbour
DT	Decision Tree
NB	naïve Bayes
HTE	Heterogeneous Ensemble
CD	Critical Difference
4IR	Fourth Industrial Revolution
ROC	Receiver-Operating-Characteristics
SGD	Stochastic Gradient Descent
CART	Classification and Regression Tree
UCI	University of California Irvine

Chapter 1

Introduction

1.1 Research Background

The fourth industrial revolution (4IR) is characterized by a fusion of technologies that have increased the rate at which data is generated from different sources, including networks, computers, and data-driven devices. The pervasive generation of data from these sources has further resulted in the emergence of large, diverse sets of big data (Lee et al., 2018). Over the years, researchers and professionals of many different fields have relied on conventional statistical methods to analyze the available generated data. However, these methods are only efficient when applied to a considerable amount of data and data complexities. As a result, the methods are limited in the current big data evolution to obtain better predictive models that drive effective decision making (Tan et al., 2017).

Artificial intelligence (AI) is one of the technological deliverables of the 4IR, and the rapid advancement of AI has contributed to the field of machine learning (ML). ML has been introduced to complement the existing statistical methods to process large and complex data in order to derive actionable insights. With the ability to process large amounts of structured and unstructured data, ML algorithms provide the possibility to develop intelligent models that automatically learn to capture trends and patterns, and to extract

valuable insight from big data (Feng et al., 2021).

Ideally, ML involves training of a learning algorithm to capture the relationship between the input features and the corresponding target values of historical data. Then the trained model obtained from the algorithm is an expert that makes predictions for new data derived from similar data distribution as the historical data (Mitchell, 1997; Das and Behera, 2017). The theories, techniques and tools of ML have been applied to solve different problems, including classification, regression, estimation, clustering, and others (Tsai et al., 2011; Elish et al., 2013; Kelleher et al., 2015; Dudek, 2017; Sharma et al., 2020).

Single ML algorithms have been used to develop models for different problems with good prediction performance (Goh and Ubeynarayana, 2017; Poh et al., 2018; Sarkar et al., 2020). However, research has shown that there is no single ML algorithm that performs best on all problems because a ML algorithm generates different views on individual problems (Wolpert, 1996).

These different views arise primarily from the complexities in the structures and mathematical foundation of a ML algorithm, which result in different performance from one problem to another. Due to the complexities and intrinsic nature of the algorithm, the generation of a better predictive outcome by a single model on a problem is uncertain (Feng et al., 2021; Alshdaifat et al., 2021). Therefore, the combination of the decisions of multiple ML algorithms to construct a *mixture of experts*, also referred to as an *ensemble*, offers an efficient solution to obtain improved predictive performance better than a single ML algorithm (Hansen and Salamon, 1990; Wolpert, 1996; Kittler et al., 1998).

Ensemble learning is one of the most promising research directions in ML. The benefits of ensemble learning have been shown in different domains with encouraging predictive outcomes as summarized in Duin (2002) and Polikar (2006). The success of an ensemble is attributed to the possibility of minimizing the influence of sub-optimal learners within an ensemble to generate optimal performance. In addition, the ability to obtain a better approximation for an unknown input-target attribute relationship in a dataset is one of the motivations of ensemble learning (Nguyen et al., 2019b). Thus, the realization of the successes of ensemble learning has transformed ML research into the possibilities of combining the predictions of multiple experts to achieve more accurate predictions and generalization performance than individual experts (Wolpert, 1996; Dietterich, 2000).

A ML ensemble is expected to perform better than the average performance of the individual component or base experts within the ensemble. However, the performance of ensembles varies considerably due to various factors such as the type of base learners, accuracy of the individual base experts, the number of base learners in the ensemble (i.e. ensemble size), combination and decision-making strategy, data sampling technique, and diversity among the individual experts (Bian and Wang, 2006).

Diversity is widely regarded to have a significant impact on the performance of ensembles. Therefore, the crux for developing ensembles is not only to get accurate predictions better than a random guess, but also to create diverse base experts that generate different assumptions and classification errors in their predictions (Verma and Mehta, 2017). Diversity within an ensemble represents the key to improve the generalization performance of the ensemble.

Due to the requirement of diversity in ensembles, combinations of multiple instances of similar or different ML algorithms are possible in order to construct a homogeneous or heterogeneous ensemble. A heterogeneous ensemble is a combination of experts where the individual experts that make up the ensemble are generated from different types of ML algorithms. Each algorithm in a heterogeneous mixture has distinct strategies that induce different assumptions to relate input features to target values in a dataset. These assumptions are referred to as the inductive biases of each algorithm. Therefore, different ML algorithms generate different predictions when trained on the same dataset due to the differences in inductive biases of the algorithms (Mitchell, 1980). This research considers the development of heterogeneous ensembles by capitalizing on the benefits of diversity within the ensembles and the inductive biases of the base algorithms used to construct the ensembles.

1.2 Problem Statement

As is the case with human experts, ML algorithms have a learned bias which results in different ML experts created from the same dataset, resulting in different predictive behaviours. To address the learning bias of ML algorithms, mixtures of experts, such as support vector machine (SVM) ensembles, neural network (NN) ensembles, random forests (RFs), k-nearest neighbour (kNN) ensembles, amongst others, have

been developed. These mixtures of experts generally produce better performance than individual ML algorithms. However, current mixtures of expert models are mostly homogeneous. All of the experts in the mixture model are the same ML algorithm (e.g. a typical NN ensemble consists of only neural networks as members of the ensemble or a RF, which consists of only decision trees). While such an approach is still efficient, the performance of mixtures of experts can be significantly improved if different ML algorithms are included, thus capitalizing on the strengths and inductive biases of a diverse set of experts. In this research, heterogeneous mixtures of experts are developed, where the members of the mixture model are different ML algorithms.

1.3 Rationale of the Research

The rationale behind this approach to mixture modelling is the *No-Free-Launch* theorem (Wolpert, 1996) with different ML algorithms exhibiting different learning (inductive) biases, and therefore performing differently on the same data set. As a result, it is also the case that no one ML algorithm performs best on all problems and that different algorithms show different advantages and disadvantages based on the problem characteristics and data. Moreover, homogeneous ensembles do not capitalize on the inductive biases of different ML algorithms. The heterogeneous mixture model will take advantage of the strengths of the different ML algorithms. In addition, the different inductive biases add an additional behavioural diversity layer to ensembles. Note that behavioural diversity among the members of an ensemble is an essential ingredient to maximize performance. In this research, the heterogeneous mixture models consist of NNs, SVMs, *k*NN, decision trees (DT), and naïve Bayes (NB) algorithms. All of these algorithms are known to exhibit different inductive biases.

1.4 Research Questions

Given the stated research problem and rationale, the following question is identified to guide this research. *"Due to the inductive biases of ML algorithms in an ensemble, can a heterogeneous mixture of experts result in an ensemble that consistently produces more accurate predictions than that of a homogeneous mixture of experts by capitalizing on the advantages of the*

experts that make up the heterogeneous mixture"? The research question is examined under the following dataset configurations: *clean data, skewed class distributions, number of outliers, severity of outliers, bagged subsets, and feature subsets.*

1.5 Goal and Objectives of the Research

The goal of this research is to capitalize on the inductive biases of ML algorithms to develop heterogeneous mixtures of expert models that consistently produce better accuracy and generalization performance on classification and regression problems. The specific objectives of the research are to:

- analyze the bias-variance dilemma and the inductive biases of different ML algorithms;
- investigate different ensemble approaches to create diversity in ensembles;
- investigate different fusion approaches to combine the outcomes of diverse ensemble members in the mixture model;
- perform a critical review of homogeneous and heterogeneous ensembles;
- develop heterogeneous and homogeneous ensembles by capitalizing on the inductive biases of ensemble members with better generalization performance;
- evaluate the performance of the heterogeneous and homogeneous ensembles on simple and complex classification and regression problems;
- conduct empirical analyses to compare the performance of the ensembles on clean data, skewed class distributions, number of outliers, severity of outliers, bagged subsets, and feature subsets in the identified classification problems; and
- conduct empirical analyses to compare the performance of the ensembles on clean data, number of outliers, severity of outliers, bagged subsets, and feature subsets in the identified regression problems.

1.6 Research Methodology

It is necessary to provide a clear research strategy in order to achieve the stated research objectives. The approaches to these objectives follow a ML pipeline that is presented as follows:

- **Problem Identification:** The problem identification is that individual ML algorithms do not perform best on all problems and that homogeneous ensembles do not capitalize on the different inductive biases of the different ML algorithms.
- **Literature Review:** This stage involves performing a critical review of literature on selected ML algorithms, data preprocessing techniques, inductive biases of ML algorithms, the bias-variance dilemma, homogeneous ensembles, heterogeneous ensembles, ensemble approaches, and different performance evaluation metrics. The critical review is essential to capture a detailed body of knowledge in ML to identify the inductive biases of individual ML algorithms, the optimal data pre-processing techniques suitable for each ML algorithm, and the components used to construct of an ensemble. All of these resulted in desired research outcomes.
- **Data Collection and Data Pre-processing:** Different simple and complex problems are collected from the University of California Irvine (UCI) ML Repository for experimentation. The collected datasets are pre-processed under the following criteria: data normalization, feature selection, feature scaling, data encoding, data sampling, handling missing values, removal of outliers, handling skewed class distributions, and handling multi-classes data classification.
- **Development of Ensembles:** Homogeneous and heterogeneous mixtures of experts are developed by combining the selected ML algorithms based on the assumptions made by each ML algorithm in fitting their models on the data. Hence, due to the inductive biases of the selected base algorithms, four types of ensembles are developed. The first ensemble type is the development of homogeneous ensembles using multiple instances of the same ML algorithm, where the instances consist of the same configurations. The second ensemble type is constructed using multiple instances of the same ML algorithm, where the instances were configured differently. The third ensemble type delivers a single ensemble

developed using multiple instances of different ML algorithms, where the instances consist of the same configurations. The last ensemble type provides another single ensemble developed using multiple instances of different ML algorithms, where the instances were configured differently. An optimization approach was employed to randomly search through the hyperparameter space of each algorithm in order to obtain the configurations for the multiple instances of the base algorithms. The different configurations of the base algorithms deliver different experts that actually capitalized on the inductive biases of each algorithm with respect to the identified classification and regression problems. The algorithms are trained, and the obtained ensemble models are evaluated on the pre-processed datasets. The ensembles are developed by capitalizing on different ensemble approaches to enhance the objective of diversity in the research. Model implementation was done in the Python programming language.

- **Empirical Analysis of Results:** The performance of the ensembles are evaluated on a number of classification and regression problems, ranging from simple problems, having small number of features, classes, and samples, to complex problems, having many samples, many features, many classes, skewed class distributions, and outliers. Furthermore, the ensembles are evaluated on the different modelling studies, (i.e. *clean data, skewed class distributions, number of outliers, severity of outliers, bagged subsets, and feature subsets*) for the selected classification and regression problems. The results of the ensembles in each modelling study are analysed using formal statistical tests to determine if differences in the performance of the ensembles are significant or not. Then, a comparative analysis is performed between heterogeneous and homogeneous ensembles using the following performance measures: accuracy, generalization factor, F1-score, and root mean squared error.

1.7 Thesis Organization

This section describes the organization of this thesis. Chapter 1 provided the motivation and goals of the research, while Chapter 2 discusses the concept of ML, the bias-variance dilemma, and the inductive biases of ML algorithms. ML ensemble approaches are discussed in Chapter 3. The reviews of homogeneous and heterogeneous ensembles are

provided in Chapters 4 and 5 respectively.

Chapter 6 describes the ensemble developmental approaches and training methods for the selected ML algorithms to construct diverse mixtures of heterogeneous and homogeneous experts in this research. The empirical process used to evaluate the heterogeneous and homogeneous ensembles is presented in Chapter 7. Chapters 8 and 9 discuss the results for classification and regression problems respectively, while Chapter 10 presents the conclusions and possible future work for the research.

Chapter 2

Machine Learning and Bias-Variance Dilemma

2.1 Background

The need to capture, analyze, interpret, and utilize large, complex and information-rich data has become the next strategic solution to generate useful insight from data. This strategic possibility is achieved by the technological advancement of AI through the introduction of ML algorithms to complement existing statistical methods in order to develop models that learn hidden information, complex relationships, and patterns in data consisting of different characteristics and complexities.

To develop ML models from data with good generalization performance, it therefore becomes imperative to analyze the inductive biases of ML algorithms and the bias-variance tradeoff. Thus, the concept of ML is discussed in Section 2.2, while Section 2.3 discusses the bias-variance dilemma. Section 2.4 describes the inductive biases of selected ML algorithms in this research, while Section 2.5 concludes the chapter with a summary.

2.2 Machine Learning

This section provides general background knowledge of ML. The concept of ML as a subfield of AI is discussed in Section 2.2.1, while Section 2.2.2 presents the categories of learning methods applied to different ML problems.

2.2.1 Concept of Machine Learning

ML is a sub-field of AI that implements a learning algorithm to search through an n-dimensional space of a given dataset to find acceptable generalizations from the data. Generalization in this context indicates that the knowledge and experience learned by a ML model from the samples in the data are used to estimate future predictions on unseen samples (Himika et al., 2008; Muhammad and Yan, 2015; Musumeci et al., 2019; Shailaja et al., 2019).

Formally, " a computer program is said to learn from experience E with respect to a set of tasks T and performance measure P, if the performance of the program at a task in T, as measured by P, improves with experience E " (Mitchell, 1997). Experience in ML refers to historical data available to a ML algorithm to construct a prediction model. The data is usually a benchmark and digitized human-labelled data or real-world dataset collected through interactions with the environment (Sharma et al., 2020).

The Mitchell (1997) formalization shows the difference between a ML system and a classical information system: In a classical information system, a mathematical model of the environmental observations is initially formulated, then model validation with actual data is performed, which is followed by system building based on the model formed. In contrast, a ML system is constructed directly on the actual data with the data allowed to speak for itself (Kantardzic, 2011).

One of the fundamental tasks in ML is *"inductive learning"*, where an unpredictable input-output mapping function of a system is estimated using a limited number of known data samples. The known data samples are referred to as a *"training dataset"* from which a ML algorithm learns and gathers knowledge about the hidden information embedded in the data. In contrast, unknown data samples, referred to as the *"test dataset"* are generated from the same source of data distribution as the training dataset. The test dataset is then

used to evaluate the performance of the trained ML model for prediction.

Typically, inductive learning of a ML algorithm leads to generalizations that are formalized as a set of functions that approximate the behaviour of a system as (Kantardzic, 2011):

$$Y = f(X, w) \tag{2.1}$$

where Y is an output for every input vector X, $w \in W$, such that w is a parameter of the function f, and W is a set of parameters used to index the set of functions. The variable f in equation (2.1), can be any set of approximate functions which may represent different estimations about the system.

2.2.2 Categories of Machine Learning Methods

There are four main categories of ML methods applied to different ML problems. A proper understanding of the characteristics of a problem will provide insight into the type of learning problem to be solved. This section discusses the different learning methods, namely supervised, unsupervised, semi-supervised, and reinforcement learning, applied to different learning problems (Dey, 2016; Das and Behera, 2017; Celik and Altunaydin, 2018; Shailaja et al., 2019; Choi et al., 2020).

Supervised Learning

In supervised learning, datasets with known target labels for each sample in the dataset are split into training and test sets, and a learning algorithm explores the patterns in the training dataset to map input features to the target values. Then a learned model is inferred to make accurate predictions on the test dataset (Choi et al., 2020). The learning tasks in supervised learning include classification and regression (Geurts, 2002).

Unsupervised Learning

Unsupervised learning captures the relationship among input data for theme analysis or grouping purposes when no information about target values is available (Lee and Shin, 2020). Since there are no target values that can relate to input data, the goal is

to identify patterns between the samples in the input dataset and to group the samples to gain meaningful insights. Clustering and association rule mining tasks are usually performed in unsupervised learning (Lee and Shin, 2020).

Semi-supervised Learning

Semi-supervised learning combines the strength of both supervised and unsupervised learning. The learning method is useful for datasets with labelled and unlabelled samples (Choi et al., 2020). A supervised learning method is usually applied to a ML problem when the numbers of labelled samples are significantly less than the number of unlabelled samples. Thus, the inadequate unlabelled samples are used to deduce a pattern about the data (Celik and Altunaydin, 2018).

Reinforcement Learning

In reinforcement learning, a software agent is trained on how to behave using environmental feedback and a reward system. The training provides the software agent to acquire the ability to perceive and interpret its environment, take actions and learn through trial and error. The environmental feedback indicates the degree to which an output, known as *"action"*, fulfils the goals of the agent (Simeone, 2018). Then the objective of the agent is to use the shortest way and correct actions to reach a goal.

When the agent exhibits the desired actions, positive rewards are given to motivate the agent, while negative rewards are assigned for undesired behaviours. In both ways, learning occurs on the way to the goal, and the focus of the agent is to seek long-term and maximum overall reward to achieve an optimal solution (Celik and Altunaydin, 2018).

Having presented the different ML methods for different tasks, the focus of this research is on supervised learning, in which case the goal is to learn a mapping function,

$$f : \mathbb{R}^I \rightarrow \mathbb{R}^K \tag{2.2}$$

from a given dataset such that good generalization performance is achieved. From equation (2.2), I is a set of input features, and K is a target value which could be class labels for classification problems or real-valued quantity for regression problems. However, no

modelling technique is perfect because there is a gap between the best model developed and the true function. The gap is a loss or error obtained due to the bias or variance of the model.

2.3 Bias-Variance Dilemma

The key aspect of every learning system is the ability to produce good generalization performance. However, the generalization performance of a ML model is affected by errors from different sources that subvert the performance of the model. This section discusses the bias-variance dilemma in ML. Section 2.3.1 presents the bias-variance loss, while Section 2.3.2 discusses the bias-variance tradeoff. Sections 2.3.3 and 2.3.4 present the bias-variance loss decomposition and the generalization of ML models, which is followed by the discussion of the factors influencing the bias-variance tradeoff in Section 2.3.5.

2.3.1 Bias-Variance Loss

The realization of an efficient ML model that generalizes well on test data is usually affected by errors categorized into:

- **Irreducible error**: The irreducible error refers to the random intrinsic noise in a dataset that cannot simply be explained by a specific model. The irreducible error is not considered an error that affects the generalization performance of a ML model, because the mean of the noise usually equates to zero (Geman et al., 1992).
- **Reducible error**: The reducible error refers to the expected errors that a ML model can reduce. The reducible error is decomposed into *"bias error"* and *"variance error"*. Therefore, an understanding of the difference between the bias and variance errors helps to develop models that may better estimate the true form underlying an observed data (Geman et al., 1992).

The focus of the research is on supervised learning. Thus, the reducible error of a ML model, characterized by the bias and variance errors, occurs due to the complexity of the model and functional mapping induced from the data representing either classification or regression problems.

The bias error refers to the difference between the expected or average prediction of a

model and the target value (Prachi et al., 2019). The error occurs due to the simplifying assumptions made by a model to ensure that the target function is easily learned. For instance, while most linear ML algorithms are easy to understand and train quickly, the algorithms are less flexible. As a result, lower predictive performance is obtained on complex problems that fail to meet the simplifying assumptions of the bias associated with the linear algorithms.

The variance error refers to the variability of the prediction of a model given a change in the training data (Raschka, 2018). Since the target function is estimated from the training data, the model may generate different levels of variance. The variance error of a ML model is strongly influenced by the sensitivity of the model to the specifics of the training data, which may influence the number and types of parameters used to induce the mapping function between the input features and target values.

Therefore, to obtain a good predictive model, there is a need to find an optimal balance between the bias and variance errors in a modelling process. Striking a balance between these errors often requires understanding the bias-variance dilemma, selecting models with appropriate complexity and flexibility, and suitable training data.

2.3.2 Bias-Variance Tradeoff

As discussed earlier, finding a balance between bias and variance is critical to obtaining a model that generalizes well on data. It is often the case that the techniques employed to reduce variance result in an increase in bias and vice versa. This phenomenon is referred to as the "*bias-variance tradeoff*". Balancing the bias-variance tradeoff in the performance of a model requires an efficient approximation of the mapping function between the input features and target values in a training dataset. Then, the expectation is to ensure that the trained model generalizes well to a test dataset.

The bias-variance tradeoff is interpreted as follows: a low biased model indicates the average model prediction is close to the actual value. A model with a high bias means that the average prediction of the model is far from the actual value because the model is not sufficiently flexible, and individual predictions are not adequately adapted to the data.

On the other hand, a model with low variance means that the model is stable with respect

to a given dataset out of all possible datasets the model encounters. Hence, individual predictions of the model tend to be similar to one another and are close to the average prediction of the model. A model with high variance indicates strong sensitivity to the dataset seen by the model. As a result, individual predictions are different from one another and are far from the average prediction of the model. The bias-variance tradeoff is illustrated in Figure 2.1.

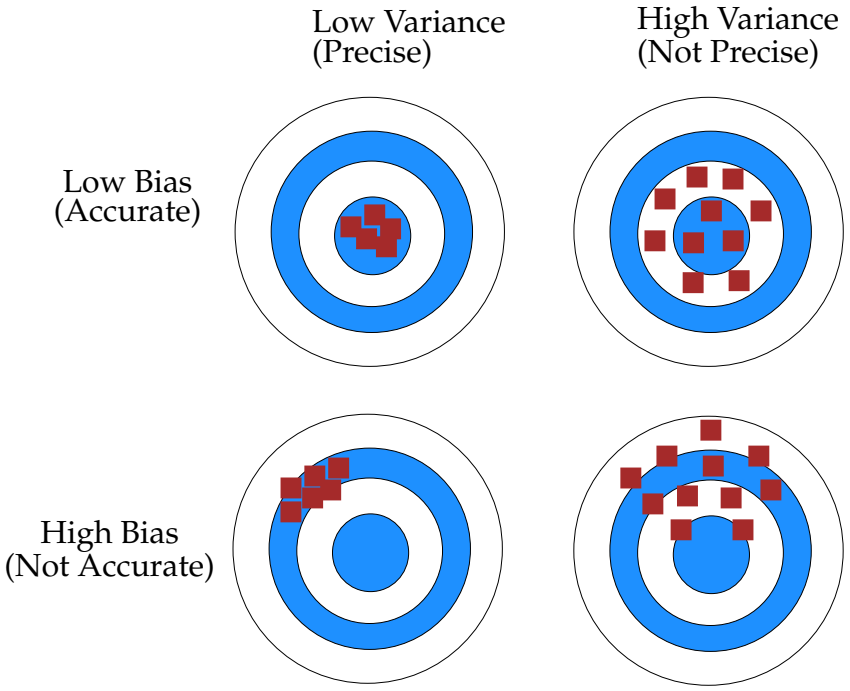

Figure 2.1: Graphical illustration of the bias-variance tradeoff

In Figure 2.1, the center of the target represents the perfect prediction of the actual value for a test point. The portions of the target away from the center represent predictions with errors. Each point represents one manifestation of a model, given the chance variability in the training dataset. The extent to which the center of the point cluster approximates or moves closer to the center of the target represents the bias of the model. The extent to which the different points cluster tightly, i.e. the spread within the dots, represents the variance of the model. Therefore, the bias-variance tradeoff provides a solution to obtain better generalization performance by carefully striking a balance to minimize the bias (being right on average) error and the variance error (being stable with respect to variation in training datasets) (Knox, 2018).

2.3.3 Bias-Variance Loss Decomposition

The bias-variance tradeoff in ML has a long history that is rooted in statistics (Neal, 2019). Early work presented by Geman et al. (1992) on "*neural networks and the bias-variance dilemma*" is the most cited work in the analysis of the bias-variance dilemma. The work of Geman et al. (1992) introduced the bias-variance decomposition to the ML research community.

While the insight into the bias-variance loss decomposition was derived from the field of regression using a squared loss function (Geman et al., 1992; Valentini and Dietterich, 2004), Domingos (2000) defined the bias-variance loss decomposition for classification problems in the context of a 0-1 loss function.

The total error of a learning system is decomposed into three non-negative quantities as

$$Total_{err} = Bias^2[h(\mathbf{x})] + Variance[h(\mathbf{x})] + noise \tag{2.3}$$

where $h(x)$ is the predicted target value learned from the data. Given that noise is an irreducible error with a mean of zero, the noise is not considered when analyzing the loss functions of a ML system. The focus is on the reducible errors that are decomposed into bias and variance, respectively. Formally, the bias-variance tradeoff is defined as follows (Raschka, 2018):

Given a point estimator $\hat{\theta}$ of a true function θ, the bias is the difference between the expected value of the estimator and the true value, given as

$$Bias(\hat{\theta}) = E[\hat{\theta}] - \theta \tag{2.4}$$

where E is the average prediction of the estimator. That is, E is the expectation over training sets and not the expectation over samples in the training set. If the bias is larger than zero, the estimator is positively biased. If the bias is smaller than zero, the estimator is negatively biased, and if the bias is exactly zero, the estimator is unbiased. On the other hand, variance is defined as the difference between the expected value of the squared estimator and the squared expectation of the estimator, given as

$$Var(\hat{\theta}) = E[\hat{\theta}^2] - (E[\hat{\theta}])^2 \tag{2.5}$$

2.3.4 Generalization and Underfitting-Overfitting of Machine Learning Models

The overall goal in ML is to obtain a model that generalizes well from the training data to new test data (Geman et al., 1992). As earlier stated, generalization is the ability of a model to give accurate outputs based on how well the model fits the training and test datasets (Jakubovitz et al., 2019). For the ML model to provide better generalization performance, the model should not memorize the training dataset, but rather learn the underlying rules associated with the data generation process. Then the model derives the capability to extrapolate the learned rules from the training dataset to a new test dataset. However, the construction of a well-generalized ML model depends on the notion that ML algorithms suffer from poor performance due to underfitting or overfitting. A model that generalizes well is a model that neither underfits nor overfits (Jakubovitz et al., 2019).

Underfitting occurs when a ML model fails to correctly capture the underlying trend of a training dataset. Underfitting usually happens when an algorithm tends to build an accurate model with less complexity or training data. On the other hand, overfitting occurs when a model is too complex and is trained on excessive training data until the model starts to learn and memorize the inherent noise and inaccurate entries in the data (Nautiyal, 2018).

Understanding of the dynamics of underfitting and overfitting of ML models is greatly influenced by the bias-variance tradeoff to obtain an optimal predictive model (Brady and Brockmeier, 2018). Therefore, dealing with bias and variance is directed to dealing with the underfitting and overfitting of ML models (Raschka, 2018). The relationship between bias-variance errors and underfitting and overfitting of ML models is presented in figure 2.2.

From figure 2.2, underfitting is derived from erroneous simplifying assumptions in the learning model, which leads to high bias and low variance. The high biased model generates both high training and test errors. In overfitting, the highly complex model interpolates the training data perfectly with the noise or outliers in the training data.

The result is a model with low bias and high variance in prediction. A model with high variance generates a low training error but high test error (Thorhallsson and Singh, 2017).

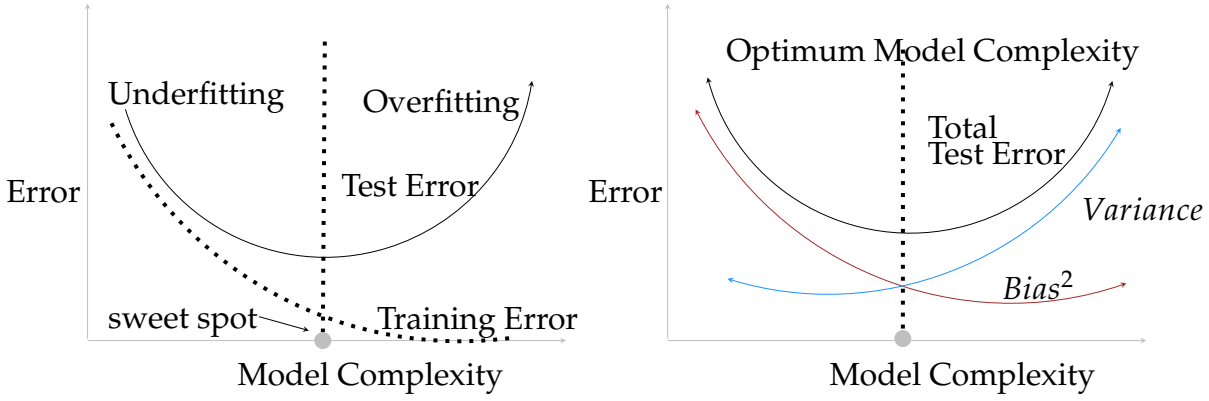

Figure 2.2: Relationship between Underfitting-Overfitting and Bias-Variance Errors

2.3.5 Factors Influencing Bias-Variance Tradeoff in Machine Learning

Several factors influence the bias-variance tradeoff. The effect of model complexity and changes in the size of training data on the bias-variance tradeoff are discussed in this section.

Model Complexity and Bias-Variance Tradeoff

The critical variable modulating bias and variance is model complexity. Model complexity indicates the number of free parameters of a model used to approximate the true functional mapping between input features and the target values (Mehta et al., 2019).

Models with too few free parameters are inaccurate for a given training data size and will underfit, causing high bias. Also, models with too many free parameters are incorrect and will overfit the training data leading to high variance.

Therefore, as more parameters are added to a model, the complexity of the model increases and bias decreases monotonically, while variance becomes the main concern (Merentitis et al., 2014). It is crucial to select appropriate model complexity for the model to generalize well to a test dataset. The influence of model complexity on the bias-variance tradeoff is illustrated in Figure 2.2.

Training Data Size and Bias-Variance Tradeoff

The effect of training data size on the bias and variance tradeoff is observed with increasing and decreasing sample size and feature dimension. For small training sample sizes, low training and high test errors are generated for a highly biased model. As the training sample size increases, the training error increases rapidly while the test error decreases slowly. A point is reached where the training and test errors flatten and remain high as the sample size increases. Therefore, increasing the number of samples may not necessarily reduce the generalization error for a highly biased model (Medicherla, 2018).

For a model with high variance, the training error is usually low, while the test error becomes very high for a small training sample size. With increasing training set, the training error increases slowly, but the test error decreases quickly. A point is reached where the training and test errors flatten and remain low as the sample size increases. Thus, increasing the sample size is significant in the minimization of the generalization error for a model with high variance, provided that the model is not complex and that the increased sample does not increase the noise (Medicherla, 2018).

A graphical representation of how training data size influences the bias-variance tradeoff is given in Figure 2.3:

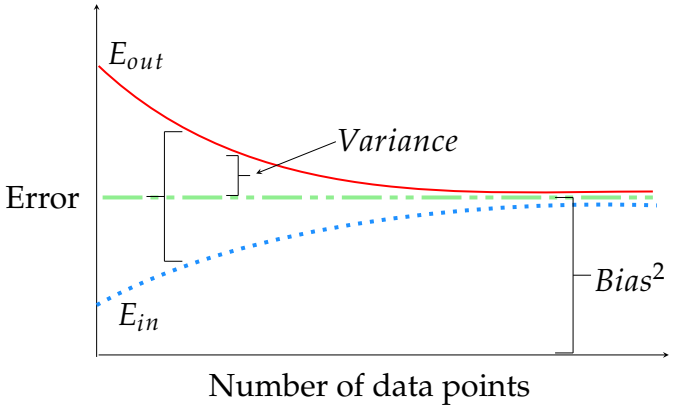

Figure 2.3: Influence of training data size on the bias-variance tradeoff

Furthermore, increasing and decreasing feature size in training data also influence the bias-variance tradeoff. A model trained on fewer features will generate high training and testing errors indicating high bias. Addition of more features to the data gradually decreases the bias. Additionally, a model suffers from high variance when trained

with large and irrelevant features. The reduction of the feature dimension of the data progressively decreases the variance (Medicherla, 2018).

2.4 Inductive Biases of Selected Machine Learning Algorithms

The concept of the bias-variance tradeoff has been established in section 2.3. For a ML model to generalize well on a dataset, it is beneficial to analyze the inductive biases of the ML algorithm that induces the model. This section discusses the inductive biases of selected ML algorithms for this research. Section 2.4.1 introduces the concept of inductive bias in ML, while Section 2.4.2 discusses NN algorithm and the corresponding inductive biases of NNs. Sections 2.4.3 and 2.4.4 present the SVM and DT algorithms including the inductive biases of the algorithms, while Sections 2.4.5 and 2.4.6 discuss the *k*NN and NB algorithms, as well as the inductive biases of the algorithms. Then Section 2.5 concludes the chapter with a summary.

2.4.1 Concept of Inductive Bias

The inductive bias of a ML algorithm refers to the specific assumption made by the algorithm when inferring a relationship between input features and target values in a dataset (Mitchell, 1980, 1997). In the absence of inductive bias, every ML algorithm would make the same predictions on the same dataset (Dietterich and Kong, 1995).

In many cases, ML algorithms operate stochastically and deal with noisy, erroneous and inconsistent datasets. Due to the stochastic nature of the algorithms, the inductive biases of a ML algorithm produce different model generalizations. As a result, there is no one model inductive bias that is best on all problems, because different model inductive biases are better fits for different problems (Wolpert, 1996; Mitchell, 1997).

According to Dietterich and Kong (1995), the inductive biases of ML algorithms also influence the performance of the induced model with respect to the bias-variance tradeoff. Dietterich and Kong (1995) reported that a very strong and inappropriate inductive bias (*very low model complexity*) leads to low variance and high bias in the performance of a ML model. When the inductive bias is too weak but appropriate (*very high model complexity*),

the performance of the model results in high variance and low bias. Therefore, finding the optimal inductive bias may better balance the bias-variance tradeoff (Dietterich and Kong, 1995).

Following the brief introduction of inductive bias in ML and the relationship with the bias-variance tradeoff, the concepts and the inductive biases of NN, SVM, *k*NN, NB and DT algorithms are discussed next.

2.4.2 Neural Networks

A NN also referred to as *"artificial neural network"*, is a ML algorithm constructed to model the decision-making process of the human brain. A NN is a massive parallel distributed processor made up of simple processing units referred to as "*neurons*". These neurons learn experiential knowledge expressed through inter-unit connection strengths and can make such knowledge available for use (Kantardzic, 2011).

The massive parallel distributed structure of a NN provides the algorithm with high computational power and the ability to learn and generalize on a dataset. While it is impossible to model the entire human brain, small NNs having different network structures are constructed to solve various tasks such as classification, regression, estimation, and others (Owens and Tanner, 2017).

The network structure used in this research is the multi-layered feedforward NN (Rosenblatt, 1957), where stochastic gradient descent (SGD) backpropagation algorithm is used as weight optimization algorithm (Rumelhart et al., 1986). Figure 2.4 presents the structure of a NN with five layers. The input layer accepts the input signal $\mathbf{x}_i$ to the network, where I is the number of inputs. The input layer is connected to the first hidden layer using synaptic weights denoted as w_{ij}. It is in the hidden layer that learning occurs in the network. The last hidden layer is connected to the output layer. The output layer provides the output signals, $\mathbf{y}_e$, of the network, where E is the number of outputs.

Learning in a NN is an iterative process in which the free parameters of the network (including weights and biases) are adapted through a process of stimulation by the environment the network is embedded in (Haykin, 1994). Through adaptive learning, the performance of a NN is improved using an interactive process of adjustments applied to the synaptic weights and biases of the network. Then, after every iteration, the network

captures more information about the underlying relationship between the input features and target values (Bonala, 2009).

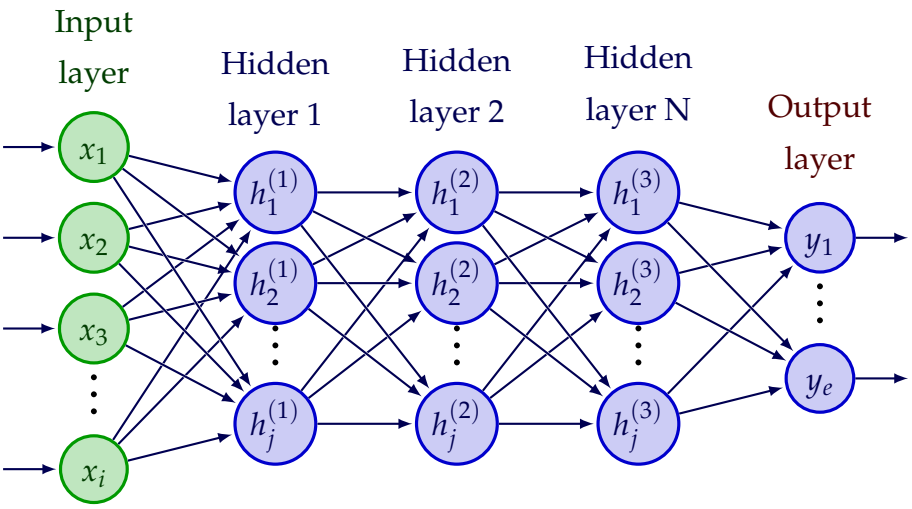

Figure 2.4: Structure of a Feedforward Neural Network

The first phase of training is the feedforward pass. This phase simply calculates the activations of the hidden and output neurons. The next phase is backward propagation. This phase updates the weights using the SGD algorithm to minimize an error function. The error function indicates the difference between the output of the NN and the target value (i.e. error) for each sample in the training set. The error function performs an aggregation of errors over the outputs of the network. Thus, the error function, used as the objective function to be minimized, is the average sum squared error (*also known as L1 loss*) over all available outputs. Other possible error functions include mean absolute error (*also referred to as L1 loss*), cross-entropy (*consisting of binary, categorical and sparse cross entropy*), log loss, exponential loss, hinge loss, Kullback-Leibler divergence loss, and Huber Loss (Bishop, 1995; Feiping et al., 2018).

The different parameters of NNs (including weight optimization algorithm, network size, activation functions, loss functions, learning rate, stopping criterion, epoch, batch size, and others) define the complexity of the network during inductive learning. While there is no standard rule to select the appropriate model complexity for a NN, obtaining an appropriate model complexity plays a significant role to balance the bias-variance tradeoff (Geman et al., 1992). For instance, selection of the appropriate network size defined by the number of hidden layers and hidden nodes in each hidden layer of the network is a factor that influences the generalization performance of NNs (Geman et al.,

1992; Tetko et al., 1995). Several works in literature have provided the possibility of defining the appropriate network complexity for different problems by finding an optimal architecture through trial-and-error, using pruning approaches, growing approaches, regularization, and NN architecture search methods.

Breiman (1996b) categorized NNs as unstable learners making the algorithms susceptible to high variance, which is a result of a too complex network trained on small datasets. The strategies to deal with high variance in NNs include stopping the network early during training to avoid overtraining. Pruning of the network and increasing the size of the training dataset are approaches used to minimize high variance. Other approaches include the introduction of $L1$ and $L2$ weight regularization to modify the loss function and soft weight sharing (Geman et al., 1992; Nowlan and Hinton, 1992). Dropout is another technique developed for deep networks to reduce overfitting in the network. The dropout technique deactivates a certain number of neurons at a layer from firing during training (Srivastava et al., 2014).

Furthermore, NNs exhibit high bias error when a small network is trained on large and complex datasets. An increase in the complexity of the network and the combination of networks may reduce underfitting in the network (Geman et al., 1992; Fitzgerald, 2014). Other techniques are growing approaches which include addition of more features to the data, allowing the network appropriate training time to capture the underlying relationship in the data, and reduction of dropout (Tetko et al., 1995; Srivastava et al., 2014).

Inductive Bias of Neural Networks

An understanding of the inductive bias of NNs is critical to the analysis of the performance of NNs to generalize to new data. The issues in NNs due to the inductive biases of the algorithm are presented as follows:

The inductive bias of NNs is the approximation of continuous target functions from input data to make predictions for corresponding target values. This inductive bias is also informed by the hyperparameters of a NN, such as the weight optimization algorithm, network size, activation functions, loss functions, learning rate, stopping criterion, epoch, batch size, network pruning approaches, network growing approaches, and others.

Each of these hyperparameters provides separate assumptions in NNs that influence the prediction results of the model induced by the network (Archer and Wang, 1993).

The SGD error backpropagation algorithm assumes that NNs are more likely to converge toward a solution with small weights (Snyders and Omlin, 2000). This assumption results in the possibility for an induced predictive model to become stuck in a local minimum due to neuron saturation in the hidden layers of the network. Neuron saturation is a problem that occurs when the hidden neurons in a NN predominantly output values close to the asymptotic ends of an activation function range. The SGD backpropagation algorithm requires the gradient to be a non-zero gradient to update the network weights for learning to occur. The gradient falling to zero usually occurs due to network saturation, which causes the hidden neurons to lose sensitivity to input signals and the inability of the network to propagate error signals backwards. Thus, in most cases, the network may not continue to learn (Bi et al., 2005).

Another inductive bias of NNs is attributed to the complexity of the models obtained from the network during training. The assumption of a small learning rate tends to influence the prediction outcome of NN models. Small learning rates result in small step sizes in the weight space, leading to the possibility that the network may become stuck in a local minimum. When the step sizes are too large, the likelihood that the network converges too quickly to a suboptimal solution increases. As a result, the performance of the induced model may be inconsistent over training epochs (Maier and Dandy, 1998). In addition, large step sizes may overshoot a good local (or even global) minimum.

The performance of a NN is also subjected to the assumption of an optimal network size defined by the hidden layers and hidden nodes and weights. Optimality in this context refers to the smallest network that adequately captures the relationship between the input features and the target values in the training dataset. Smaller networks have been reported to provide better generalizability, require fewer physical resources, and produce higher processing speed during training and testing. However, the error surface of smaller networks is more complicated and has more local minima (Maier and Dandy, 1998). In contrast, larger networks tend to learn quickly in terms of number of training cycles (although slower per training cycle), perform efficiently in complex decision regions, and has a better possibility of avoiding local minima. However, larger

networks have high computational costs and require a large training dataset to achieve a good generalization performance (Maier and Dandy, 1998). Thus, the selection of an appropriate network size is a critical decision toward the generalizability of the network because the decision is usually problem-dependent. Additionally, overfitting and underfitting of the induced NN model is another issue that requires careful consideration during selection of an optimal network size.

Another critical decision in the selection of the optimal network size is the choice of the number of nodes in the hidden layers, and hence the number of connection weights. Given the number of nodes selected, it becomes imperative to find a tradeoff between obtaining sufficient weights for optimal network performance. An optimal number of weights ensures that the function to be learned in the training dataset is adequately approximated. In contrast, an inappropriate number of weights (obtained from too few or too many number of weights) may lead to the network underfitting or overfitting the training dataset. As a result, the network may lack the ability to generalize efficiently to the test dataset.

The stopping criterion decides when to stop the network training process during weight optimization, and therefore provides an assumption that determines whether the model has been optimally or sub-optimally trained. Stopping the training prematurely often subjects the model to not capture sufficient information or patterns in the training dataset, resulting in underfitting. Also, stopping the training too late allows the model to learn the inherent noise in the dataset, which leads to overfitting when there are too many weights. While different approaches have been proposed, research suggests the difficulty in determining the optimal stopping time that would lead to good generalization performance.

The selection of appropriate activation functions in the hidden and output layers of a NN also controls how well the network maps the relationship between the input space and the target space in the dataset, and the type of prediction made by the induced NN model. The assumption of a suitable choice of an activation function in the hidden layer influences the step sizes taken in the weight space, because weight updates are proportional to the derivative of the activation function.

Also, the prediction outcomes obtained when the backpropagation algorithm is used

are sensitive to the initial weight conditions. Typically, weights are initialized to zero-mean random values, and the selection of the upper and lower bounds (-α, α) for the weights is a careful process that affects the decision of the network. If the value of α is too small, the gradient becomes smaller as error signals are backpropagated through the hidden layers, and hence, weight optimization becomes very slow, and in the worst case, network training may stop. This problem results in minor weight updates and slow network convergence. The problem is referred to *vanishing gradient problem*, where the possibility of weights vanishing to zero is high. In contrast, if α is too large, premature saturation of the nodes may occur, which in turn will slow down training and result in the cessation of training at suboptimal levels. In this case, the gradient becomes much larger as error signals are backpropagated through the hidden layers, causing extremely high weight updates that may lead to an overflow in gradient computation and inconsistency in weight optimization. This problem is known as *exploding gradient problem*, where weights are likely to explode to infinity (Maier and Dandy, 2000).

The sum square error, when used as the objective loss function, provides easy computation of the partial derivatives with respect to the weights and is mostly suitable for data that assume a normal distribution. However, to obtain optimal results, the errors are expected to be independently and normally distributed, which is not the case when the training data contain outliers.

Furthermore, research has shown that NNs that do not use a robust estimator are sensitive to outliers in a dataset, causing the network to experience slow training and overfitting. A possible solution to minimize the effect of outliers on the network is to remove outliers in the dataset or use a robust estimator. However, selection of an optimal network configuration is subjective to the type of activation function selected, choice of an error function, hidden layers and hidden nodes, loss function, and other parameters. Hence, NNs without a robust estimator require outliers to be removed from data before model construction (Klein and Rossin, 1999).

The performance of NNs is sensitive to skew class distributions in a dataset. In an imbalanced class scenario for a SGD-based NN model, the gradient vector computed by standard backpropagation for the majority class is much smaller than the gradient vector computed for the minority class. This indicates that the majority class will dominate the

net gradient responsible for updating the weights of the NN model. Thus, the error of the majority class is reduced very quickly during early iterations, while the error of the minority class increases, causing the network to converge slowly (Anand et al., 1993).

Backpropagation NNs require a complete set of input data to create a mapping between input features to target values in a dataset. However, data may contain missing values, and NNs may struggle to handle the missing values, often resulting in biased predictions. Different approaches have been reported to deal with missing values, including deleting all missing values, replacing missing values with random values, and using the mean, median or mode of feature values depending on whether the feature values are categorical or numerical (Rubin, 1976). However, deleting all missing values may lead to the loss of potentially valuable information in the dataset, while it may be challenging to determine the random value to replace missing values. Also, using mean, median, or mode may introduce bias to the dataset (Ennett et al., 2001).

Lastly, NNs assume the requirement of numerical input to create a mapping between input space and target space in a dataset. The algorithm cannot work on data consisting of categorical or multivariate features intrinsically, except the values of the input feature are encoded into continuous values.

2.4.3 Support Vector Machines

The SVM algorithm was developed by Vapnik and Chervonenkis (Vapnik and Cortes, 1995; Vapnik, 2000) based on statistical learning theory to solve binary classification problems. The SVM works on the principle of structural risk minimization to find a function that minimizes the expectation of model error on new data for better generalization. The principle allows the SVM algorithm to exploit theorems bounding the actual risk in terms of the empirical risk rather than estimating error using asymptotic convergence to normality. Hence, even with small sample sizes, SVMs can produce accurate estimates of the prediction error, while making no distributional assumptions about the data (Wilson, 2008). Also, the SVM algorithm has been reported to provide efficient computational performance on high dimensional data (Nah and Lee, 2016). The SVM decision function can be a classification function (*support vector classifier with binary class labels*) or a regression function (*support vector regression with real-valued outputs*)

(Kantardzic, 2011).

The SVM represents data in an n-dimensional space, and the goal of the SVM is to find the hyperplane (also known as *decision boundary*) out of several hyperplanes that has the maximum margin in separating samples of two classes in the n-dimensional space. The process is achieved by searching for the set of samples of opposing classes (*referred to as support vectors*) closest to the lines from both the classes such that the margin around the calculated hyperplane is maximized (Vapnik and Cortes, 1995). The hyperplane for which the margin is maximum is the optimal hyperplane, illustrated in figure 2.5.

From Figure 2.5, the optimal separating hyperplane represented by the thick centre line classifies the training samples into two classes with the maximum margin; the positive class "1" above the hyperplane and the negative class "-1" below the hyperplane. The filled and unfilled red dots denote the support vectors that influence the optimum location of the separating hyperplane. The width of the margin is $\frac{2}{\|\mathbf{w}\|}$, $\mathbf{w}$ is an n-dimensional vector, $\mathbf{x}$ is the set of training data, and b represents the bias term.

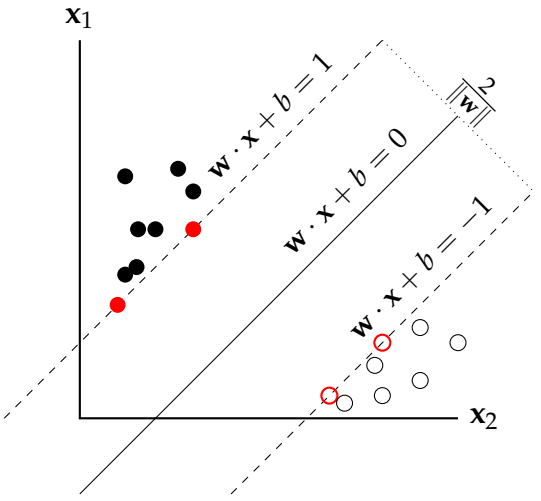

Figure 2.5: SVM in a 2-D space

The SVM performs different classification tasks that fall under the following problems: linearly separable, linearly non-separable, non-linearly separable, and multi-class problems.

SVM learning for a linearly separable problem is illustrated in Figure 2.5, where there are no training errors, and the optimal hyperplane is the hyperplane that maximizes the margin. In this case, the optimal hyperplane for a set of training data, $\mathbf{x}_i$ $(i = 1, 2, 3, ..., n)$,

where n is the number of training samples, is defined as:

$$\mathbf{w}.\mathbf{x} + b = 0 \tag{2.6}$$

The important properties of this hyperplane include having the least possible error in the separation of data and that the distance from the closest data of each class must be maximal. Under these scenarios, data of each class can only be above $(y_i = 1)$ or below $(y_i = -1)$ the hyperplane. Therefore, two margins are defined as shown in Figure 2.5 to separate the data, i.e.

$$\mathbf{w}.\mathbf{x} + b \begin{cases} \geq 1 \quad for \;\; y_i = 1, \\ \leq -1 \quad for \;\; y_i = -1 \end{cases} \tag{2.7}$$

However, the generalization region for the hyperplane can be anywhere between 1 and -1, and there are many margins that can be considered the boundary of each class. Hence, to find the optimal hyperplane, the distance (d) between the margins should be measured and maximized using (Nah and Lee, 2016)

$$d(\mathbf{w}, b, \mathbf{x}) = \frac{|(\mathbf{w}.\mathbf{x} + b - 1) - (\mathbf{w}.\mathbf{x} + b + 1)|}{\parallel \mathbf{w} \parallel} = \frac{2}{\parallel \mathbf{w} \parallel} \tag{2.8}$$

Therefore, maximizing the margin is equal to minimizing the dimensional vector $\mathbf{w}$, and the SVM learning problem is given as

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2} \parallel \mathbf{w} \parallel^2 \\ \text{s.t} \quad & y_i((\mathbf{w}.\mathbf{x}_i) + b) \geq 1, \; i = 1, 2, 3, \ldots, n. \end{aligned} \tag{2.9}$$

The Lagrangian method is used to transform equation (2.9) to a quadratic programming problem to derive the optimal hyperplane as

$$L_p = \frac{1}{2} \parallel w \parallel^2 - \sum_{i=1}^{n} \alpha_i \{y_i(\mathbf{w}.\mathbf{x}_i + b) - 1\} \tag{2.10}$$

where L_p denotes the primal problem and α_i are the lagrange multipliers, one for each

training sample. Following equation (2.10), L_p is minimized with respect to $\mathbf{w}$ and b, and requires that the derivatives of L_p with respect to all α_i vanish as

$$\frac{\partial L}{\partial w} = \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \tag{2.11}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.12}$$

Finally, substitution of equations (2.11) and (2.12) into equation (2.10) gives the equation of the SVM for a linear separable problem as (Gholami and Fakhari, 2017)

$$\begin{aligned} L_D &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i . x_j) \\ \text{s.t} &\begin{cases} \alpha_i \geq 0 \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.13}$$

L_D is the referred to as a dual problem. Thus, the solution to the learning problem is the minimization of L_p or the maximization of L_D. Since there is a Lagrange multiplier α_i for every training sample, the samples for which $\alpha_i > 0$, are the support vectors (SVs) . The classifier is constructed as

$$f(x) = sign(\mathbf{w}.\mathbf{x} + b) = sign\left(\sum_{i \in SVs} y_i \alpha_i (\mathbf{x}_i . \mathbf{x}) + b \right) \tag{2.14}$$

Linearly non-separable problems usually occur due to the similarity of a number of features or noise in the training data. In this case, the SVM adopts a soft margin approach by introducing a penalty cost, C, and slack variables, ξ_i. The penalty cost softly penalizes misclassified samples, while slack variables denote the distance that is measured and minimized between the misclassified samples of a class from the margin of the class. The penalty cost function is given as

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{n} \xi_i \\ \text{s.t} \quad & y_i((\mathbf{w}.\mathbf{x}_i) + b) \geq 1 - \xi_i, \; i = 1, 2, 3, \ldots, n. \end{aligned} \tag{2.15}$$

The penalty cost C in equation (2.15) is referred to as the *"trade-off"* parameter added to maximize the margin and to minimize the classification error (Gholami and Fakhari, 2017). Using Lagrangian multipliers, the optimization problem in equation (2.15) is converted into a dual problem for a soft margin support vector classifier as

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i . \mathbf{x}_j)$$
$$\text{s.t} \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{2.16}$$

The difference between equations (2.16) and (2.13) is the constraint imposed on the Langrange multiplier α to be either equal to or less than the penalty cost C.

For nonlinear separable problems, a kernel function, $\mathbf{K}$, is employed to map or transform the input data $\mathbf{x}$ onto a higher dimensional feature space, also referred to as *feature or Hilbert space* (Mercer, 1909). The mapping process using the kernel function still allows the nonlinearity of data in the input space, while ensuring the creation of a linear support vector classifier in the feature space to separate the data into classes. As a result, the input data $\mathbf{x}$ is represented in the feature space to allow dot product computation (*also referred to as inner product computation*) $\phi(\mathbf{x})$ of the input data in the feature space using a given kernel function as

$$\mathbf{K}(x_i, x_j) = \phi(x_i)\phi(x_j) \tag{2.17}$$

Then, the general dual equation derived for a linearly non-separable problem in equation (2.16) is reformulated for a nonlinear classification problem as

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$$
$$\text{s.t} \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{2.18}$$

Finding the optimal hyperplane in this equation (2.18) is not a straightforward task due

to the unknown value of ϕ, which makes it challenging to calculate the weighting vector $\mathbf{w}$. The weighting vector $\mathbf{w}$ is represented in the feature space as

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \phi(x_i) \tag{2.19}$$

However, knowing that the hyperplane is defined as

$$d(\mathbf{x}) = \mathbf{w}.\phi(\mathbf{x}) + b, \tag{2.20}$$

the kernel trick provides the realization to ignore the computation of the weight vector in the feature space by substituting equation (2.19) into equation (2.20) to obtain the optimal hyperplane as

$$d(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i \mathbf{K}(x, x_i) + b \tag{2.21}$$

where $f(x) = sign(d(\mathbf{x}))$ is the classification output. Hence, SVMs can efficiently solve nonlinear classification problems by selecting an appropriate kernel function that is usually problem dependent. Commonly used kernel functions include linear, polynomial, Gaussian radial basis function and the sigmoid function (Liu et al., 2007; Nan and Xiang, 2014).

The SVM, as discussed thus far, has been developed for binary classification problems. However, the SVM has to deal with multi-class classification problems. Two common approaches to scale SVM to multi-class classification problems are (Prakash et al., 2012):

- one-versus-one, where $k(k-1)/2$ models are constructed, and k indicates the number of classes; and
- one-versus-many, where a pairwise classification is performed such that there is one binary SVM for each pair of classes to separate members of one class from members of the other.

The working principle of the support vector regression (SVR) involves finding the optimal hyperplane that can satisfactorily explain the relationship of the real-valued target output from the input features as

$$f(\mathbf{x}) = \mathbf{w}.\mathbf{x} + b \tag{2.22}$$

where $\mathbf{w}$ is the weighting vector and b is the the intercept parameter (*referred to as bias in SVM classification*) of the regression model. For regression problems, the optimal hyperplane is expected to have the minimum possible prediction error when selected. Therefore, achieving the minimization of the empirical risk (i.e. error) requires defining an insensitive parameter ε to measure the variation between the real and predicted values (Vapnik, 2000). The sum of ε_i can then be minimized using a loss function such as the mean square error, mean absolute error, mean absolute percentage errors, and others to obtain the optimal hyperplane.

For a linear SVR problem, the optimal hyperplane is selected such that minimum deviation from the insensitive ε parameter is derived. As a result, the SVR ignores the error posed by the input data confined in the ε margins, and considers the remaining error to find the optimal hyperplane using the slack variables, ξ_i. The formulation of the regression learning problem using the SVR is defined by an objective function L_p whose goal is to find the optimal value of the weighting vector $\mathbf{w}$ such that the empirical risk is minimized as (Gholami and Fakhari, 2017)

$$\begin{aligned} L_p &= \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i') \\ \text{s.t} &\begin{cases} y_i - \mathbf{w}.\mathbf{x} - b \leq \xi_i + \varepsilon \\ y_i + \mathbf{w}.\mathbf{x} + b \leq \xi_i + \varepsilon \\ \xi_i, \mathbf{x}_i \geq 0 \end{cases} \end{aligned} \tag{2.23}$$

where ξ_i, ξ_i' are slack variables for the mutually exclusive situations in the constraints. The constraints introduced in equation (2.23) guarantees that any error less than ε would not be computed in the objective function, illustrating the insensitivity of the ε loss function proposed by Vapnik et al. (1996) and Vapnik (2000). The insensitivity ε measures the cost of the errors on the training points such that the loss is zero if the difference between the predicted and real value is less than ε; otherwise, the loss is measured as the absolute difference between the predicted and real values in the input data.

Like the SVM classification, to resolve the optimization problem in equation (2.23), the Lagrange multipliers (α, α') associated with the constraints of the primal problem are used by following the Karush–Kuhn–Tucker (KKT) conditions as (Karush, 1939; Kuhn and Tucker, 1951)

$$\frac{\partial L}{\partial w} = 0 \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} (\alpha_i - \alpha'_i)\mathbf{x}_i \tag{2.24}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} (\alpha_i - \alpha'_i) = 0 \tag{2.25}$$

Substitution of equations (2.24) and (2.25) into equation (2.23) gives the general equation of the linear SVR formulated as (Steinwart, 2008)

$$\begin{aligned} L_d = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i - \alpha'_i)\mathbf{x}_i^T \mathbf{x}_j (\alpha_i - \alpha'_j) \quad + \sum_{j=1}^{n} ((\alpha_i - \alpha'_i) y_i - (\alpha_i + \alpha'_i)\varepsilon) \\ \text{s.t } 0 \leq (\alpha_i - \alpha'_i) \leq C \end{aligned} \tag{2.26}$$

Having obtained nonzero Lagrange multipliers, the weighting vector $\mathbf{w}$ of the optimal hyperplane is derived from equation (2.24). Then the intercept b of the regression equation is determined in one of the equations below

$$\begin{aligned} y_i - \mathbf{w}.\mathbf{x}_i - b + \varepsilon = 0 \\ -y_i + \mathbf{w}.\mathbf{x}_i + b + \varepsilon = 0 \\ \alpha_i, \alpha_j \prec C \end{aligned} \tag{2.27}$$

where $\prec$ denotes an order defined on C such that α_i and α_j are binary relations of C. The implementation of the SVR for nonlinear regression problems follows the same analysis as the linear regression problem presented above, where an insensitive loss margin is defined to minimize the prediction error. However, the problem of transforming input data into the high-dimensional feature space has to be determined. The nonlinear problem is represented as

$$f(\mathbf{x}) = \mathbf{w}.\phi\mathbf{x} + b \tag{2.28}$$

The optimal value of the weighting vector $\mathbf{w}$ obtained for a linear regression problem in equation (2.24) is transformed in the feature space as

$$\mathbf{w} = \sum_{i=1}^{n}(\alpha_i - \alpha'_i)\phi\mathbf{x}_i \tag{2.29}$$

As stated earlier, in the case of SVM classification, the value of ϕ is unknown in the feature space and therefore it becomes challenging to calculate the weighting vector $\mathbf{w}$. However, this problem is resolved using the kernel function $\mathbf{K}$, in which case equation (2.29) is substituted into the nonlinear regression problem in equation (2.28) to formulate the general equation of the SVR for nonlinear regression problems as

$$y_i = \sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha'_i)\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) + b \quad = \sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha'_i)\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) + b \tag{2.30}$$

By capitalizing on the benefit of the kernel function in which the computation of the weighting vector is ignored, the intercept b of the nonlinear regression equation is calculated as

$$b = y_i - \sum_{i,j=1}^{n_{sv}} \alpha_i y_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \tag{2.31}$$

To balance the bias-variance tradeoff, a SVM uses a set of hyperparameters such as the penalty cost C (*also known as a regulation factor*), kernel functions, degree of polynomials, and kernel width, σ, to control the tradeoff between the generalization error and complexity of a model (Ma et al., 2011). It is important to note that the hyperparameters are specific to the type of kernel function selected. For all kernels, a small value of C may result in underfitting the model and leads to high bias for a fixed size training dataset. A large value of C increases the cost of misclassifying samples, which results in a more accurate model but generalizes poorly to test data (Elaidi et al., 2018).

For Gaussian radial basis, polynomial and sigmoid kernels, small values of σ cause high bias in a model, while very high values of σ lead to high variance. The degree

of polynomial is only applied to the polynomial kernels. For polynomial kernels, the tradeoff is influenced by the degree of the polynomial and the value of C. While the polynomial kernel behaves similarly to a linear kernel in terms of the penalty cost C, an increase in the polynomial degree reduces the high variance caused by an increase in the value of C (Valentini and Dietterich, 2004).

For regression problems, the ε-insensitive loss function influences the behaviour of the SVR by allowing a tolerable error during training. Hence, setting of the optimal value of ε that may better balance the bias-variance is essential, but problem-dependent. A high value of ε indicates a high tolerable error, which may result in overfitting the model. In contrast, a small value of ε leads to a low tolerable error, but the model predictions may be far from the real target value due to the underfitting of the model.

Inductive Bias of Support Vector Machines

The inductive bias of a SVM is the assumption of being designed for binary classification problems. For multi-class classification problems, one-versus-one or one-versus-all approaches are required.

Another assumption made by SVMs is that the binary classes to be classified are linearly separable. The capability of a SVM to solve linearly non-separable problems using its intrinsic properties and capitalizing on the penalty cost C to softly penalize misclassified samples is illustrated in equations (2.9) and (2.15). For non-linear problems, this inductive bias requires the use of a kernel function to transform the input sample space to a high dimensional feature space. In addition, the optimal kernel function is problem-specific.

Also, an inductive bias of the SVM is the maximum margin for linear separable problems, which is obtained through the selection of the optimal hyperplane out of several hyperplanes. The selection of the optimal hyperplane is influenced by the feature dimensions in the input data and the support vectors that determine the margin of each class through which the optimal hyperplane is selected.

For the assumption of maximum margin made by SVMs for linear and nonlinear separable problems, only the data samples on the margin (i.e. the support vectors) contain non-zero weights in the prediction function, and the data samples beyond the margin are ignored. As a consequence, the optimal hyperplane is likely to be too sensitive to outliers

around the margin (if any), and not sufficiently sensitive to the density of data samples beyond the margin (Liu et al., 2002).

SVM requires that the optimization problem in equation (2.15) is minimized to determine the optimal hyperplane. Minimization of this problem is equivalent to solving a linearly constrained quadratic programming problem which becomes challenging as the number of samples increases. As a result, SVM does not scale well on a large number of samples in the training dataset (Osuna et al., 1997).

Furthermore, SVMs have problem-dependent parameters that are influenced by the nature of the input data (i.e. the degree of nonlinearity in the data). Selection of the kernel functions, the trade-off parameter C, polynomial degree, kernel width, and ε-insensitive parameter requires computationally expensive control parameter tuning to find the best SVM for a specific classification or regression problem (Wang et al., 2009; Gholami and Fakhari, 2017). Selection of the optimal values for these parameters is problem-specific, and may influence the optimization to be solved to obtain the optimal hyperplane.

Another issue is that SVMs are not very robust to outliers. The presence of outliers in a dataset may lead to a high rate of misclassifications, especially when the hard margin approach is used to construct the separating hyperplane. Also, the soft margin approach allows more misclassifications to accommodate outliers in the dataset. However, for extreme outlier cases, the naive maximum margin principle of SVMs influences the soft margin approach to yield poor predictive results, because the separating hyperplane becomes determined by the outliers (Scholkopf et al., 2000; Kanamori et al., 2014).

While SVMs work well with balanced datasets, SVMs are sensitive to imbalanced datasets when the soft margin approach is used. The effect of imbalanced datasets on SVM classification often results in suboptimal predictive results. When training SVMs on a skewed class dataset, the density of the negative samples in the majority class is higher than the density of the positive samples in the minority class, even around the class boundary region, where the ideal hyperplane would pass through. Thus, to reduce the total number of misclassifications, the separating hyperplane is skewed towards the majority class. The skewness of the separating hyperplane leads to higher model predictions towards the majority class, while low model prediction on the minority class is generated (Batuwita, 2013).

SVMs cannot work with missing values and requires all feature values to be available in the dataset. Also, due to samples in a training dataset having more local influence on the margin with non-linear kernels, missing values are less problematic for linear SVMs than for non-linear SVMs (Stewart et al., 2018).

Lastly, SVMs can only work with continuous-valued descriptive features. As a result, input data consisting of categorical or multivariate features are encoded into continuous values for SVMs to be implemented.

2.4.4 Decision Trees

DTs are hierarchical models that predict target values for samples through the construction of a tree-like decision structure from a set of input-output samples. The tree construction follows the application of a top-down strategy that implements sequences of recursive splitting of the features in a dataset to construct the tree (Quinlan, 1986; Kantardzic, 2011).

The invention of a DT as a simple and consistent prediction model is credited to the works presented by Morgan and Sonquist (1963), Quinlan (1986), and Breiman et al. (1993). The authors proposed different inductive algorithms to induce DTs for classification and regression problems. Figure 2.6 illustrates the hierarchical structure of a DT, which consists of a root node, branches, internal nodes and leaf nodes.

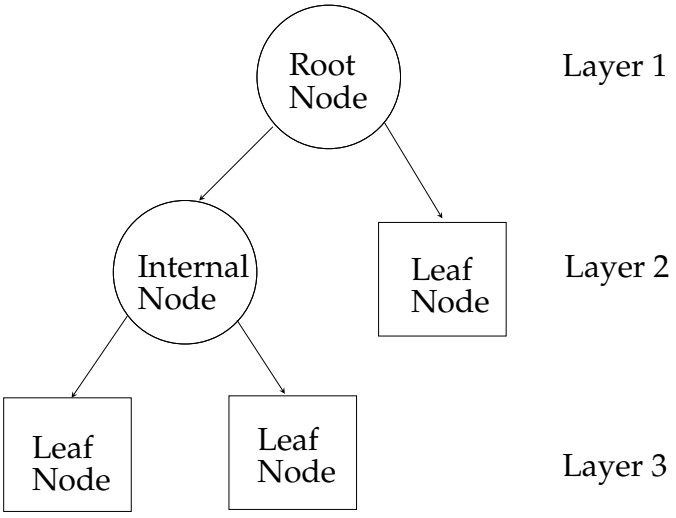

Figure 2.6: Structure of a Decision Tree

As shown in Figure 2.6, the top-down knowledge representation of a DT is induced using

a single root decision node as the start of the tree, internal decision nodes specifying a test on one or more features in the training data, and leaf nodes (i.e. terminal node) representing the final prediction outcome of the tree. The test in the internal decision nodes produces a sub-tree for each possible outcome of the test through the application of a divide and conquer strategy.

The strategy performs a greedy search to identify the best split points within a tree. The best split point in the tree is obtained through the selection of the most important feature offering the maximum accuracy for classification trees or minimum error for regression trees at each step or node of the tree-building process. The dataset is then split along the values of the features such that the target feature values at the resulting nodes are as pure as possible, i.e. *homogeneous*. The splitting process is repeated recursively in a top-down fashion until all or the majority of samples in the dataset have been predicted in the leaf node, or when a suitable stopping criterion is satisfied (Mitchell, 1997; Patel, 2012).

In the leaf node of the tree, the predicted class value for the samples in a data subset is the majority label of the samples for classification problems. In the case of regression problems, the average over the target values of the training samples is the final tree prediction obtained as the real-valued output.

The steps required for the construction of a DT include splitting, stopping, and pruning. For the splitting step, the critical decision to induce the DT is how to select a suitable splitting criterion and the best feature at a node to split the data samples into subsets. Typical splitting criteria used to select the best feature at a node during the construction of classification trees include the information gain, entropy, gain ratio, Gini index, classification error, and twoing criterion (Patel, 2012).

The information gain works on the concept of information entropy obtained from information theory (Shannon, 1948). Entropy quantifies the amount of impurity present in the sample values of a training dataset at a node. The range of entropy values is between 0 and 1. Entropy is zero for a homogeneous node where all samples of the data subset are of the same class. Thus, the node is considered a pure node. On the other hand, when the classes of samples in the data subset are equally distributed, entropy is 1.

Information gain represents the difference in entropy before and after a split on a given feature. Maximization of the information gain results in minimization of the information

entropy. Therefore, in order to select the best feature to split on at a node and to obtain an optimal DT, the feature with the smallest amount of entropy (or the highest information gain) is selected at the node. A node with the smallest entropy indicates maximum purity with reference to the samples of that node, and vice versa (Quinlan, 1986). The information gain of a dataset T is given as (Ruggieri, 2002)

$$Gain(D,T) = Entropy(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \cdot Entropy(D_i) \tag{2.32}$$

where D_i is the subset of the samples with outcome i of the test on a feature, k is the number of outcomes of a test on the feature, and $Entropy(D)$ is the entropy of the dataset with respect to class labels, given as

$$Entropy(D) = -\sum_{j=1}^{J} p(D,j).log_2(p(D,j)) \tag{2.33}$$

where $p(D,j)$ is the proportion of outcomes in D that are associated with the j^{th} class label and J is the number of class labels. The feature with the highest information gain is selected as the feature used to form a test.

The Gini index, also referred to as Gini impurity, measures the probability of a given feature being wrongly classified when the feature is randomly selected. The Gini index also represents a measure of node purity in a classification tree, and the values of the Gini index vary between 0 and 1. A Gini index of 0 indicates that all samples are classified in a certain class or if there exists only one class, while a Gini index of 1 illustrates that the samples are randomly distributed across various classes (Tangirala, 202). The Gini impurity index is calculated as

$$Info\_Gini(D) = 1 - \sum_{j=1}^{J} p(D,j)^2 \tag{2.34}$$

Hence, for the construction of a DT using the Gini index, the feature with the lowest Gini index is selected for a split at a node.

Regression trees were introduced in the classification and regression trees (CART) system by Breiman et al. (1993). CART incorporates a decision tree inducer for discrete classes

and a strategy to induce regression trees. The construction of regression trees follows a similar process as that of the classification trees, except for predicting a real-valued output. The other difference between regression and classification trees is the selection of a loss function as the splitting criterion instead of information gain, entropy, or the Gini index employed for classification trees. Typical splitting criteria used in regression trees include mean squared error, mean absolute error, variance, and standard deviation.

For the construction of regression trees, the objective at each node is also to select the most important feature to split the data into subsets, such that the overall error between the predicted value and the actual value in the data subset is minimized. Using the mean squared error, the objective function for regression trees is calculated as

$$MSE(D) = \sum_{j=1}^{D} \sum_{i \in d_j} (y_i - \hat{y}_{d_j})^2 \tag{2.35}$$

where d_j denotes the predictions for samples in d^{th} training subset of D training dataset, y_i is the actual value in the dataset, and $\hat{y}_{d_j}$ is the predicted value representing the mean response for d_j.

Having discussed splitting criteria used for classification and regression trees, it is important to note that the splitting criteria are implemented by different induction algorithms used to generate DTs. The ID3, C4.5 and C5.0 algorithms (Quinlan, 1986) consider the information gain to split features at a node, while the classification and regression tree (CART) algorithm (Breiman et al., 1993) implements the Gini index for classification trees and various loss functions for regression trees.

While the ID3 algorithm considers only categorical features, the C4.5 algorithm provides an improvement over the ID3 algorithm by removing the restriction on categorical features. Hence, C4.5 deals with both discrete and continuous features. Further, research has shown that the information gain criterion is limited by preferring tests with many outcomes higher up in the tree (Mitchell, 1997). This preference often subjects DTs to overfitting, and as a result, the C4.5 algorithm introduces the information gain ratio obtained from a split information value used to normalize the information gain (Quinlan, 1986). The split information and information gain ratio are calculated as follows (Ruggieri, 2002):

$$Split\_Info(D,T) = -\sum_{i=1}^{k} \frac{|D_i|}{|D|} \cdot log_2 \frac{|D_i|}{|D|} \tag{2.36}$$

$$Info\_Gain\_Ratio(D,T) = \frac{Info\_Gain(D,T)}{Split\_Info(D,T)} \tag{2.37}$$

If the outcomes associated with the test are discrete, branches are constructed for each possible outcome. However, if the values of a feature are numeric, the C4.5 algorithm determines an appropriate threshold value that splits the feature values as follows (Podgorelec et al., 2002): The training samples are sorted on the feature values as $v_1, v_2, v_3, ..., v_m$, where m is the sample with the smallest feature values. Then information gain is calculated on each midpoint $(v_i + v_{i+1})/2$, and the midpoint that maximizes information gain is selected to split the dataset. The C4.5 algorithm ensures that the selected threshold value appears in the final DT.

Another important step in the construction of DTs is the selection of a suitable stopping criterion to determine the exact point to stop growing the tree. The selection of an appropriate stopping criterion also determines the complexity of the trees. Smaller trees are likely to result in non-pure leaf nodes because the smaller the tree, the more likely the tree will underfit. In contrast, larger trees will have a higher probability of having pure nodes, but also a higher chance of overfitting. Therefore, induction to ensure pure nodes results in larger trees that overfit.

Different stopping criteria are used to control the bias-variance tradeoff in DTs. Possible stopping criteria include the number of nodes in a tree, number of leaves, number of features, number of samples in a node before splitting, and the depth of the tree (Song and Lu, 2015). To illustrate the importance of a stopping criterion to control the bias-variance tradeoff, when the tree grows deeper, more complex models are induced due to more split conditions. Thus, the model learns more information including the inherent noise in the training data. The outcome is a model with low bias and high variance. For shallow DTs, the induced models generate performance susceptible to high bias and relatively reduced variance. Therefore, it becomes necessary to select appropriate tree complexity for optimal performance.

Pruning is another method used to reduce the complexity of DTs and to ensure induction

to maximize purity in leaf nodes does not result in overfitting and poor generalization. Pruning involves the removal of nodes that contain features with low importance from a tree to obtain an optimally sized tree. Optimality in this context is with reference to obtaining optimal generalization performance. The pruning process is carried out using one of two approaches, namely, *pre-pruning* or *post-pruning*. Pre-pruning, also referred to as *"early stopping"*, is performed to prevent the generation of irrelevant nodes in the tree before a full tree is induced. In contrast, post-pruning is employed to remove nodes after the full tree has been generated such that the overall accuracy is maximized or the total error is minimized. Thus, finding an appropriate pruning strategy is necessary to obtain better generalization performance (Hastie et al., 2001).

From the perspective of the bias-variance tradeoff, Breiman (1996a) provides an empirical report that categorizes DTs as unstable learners, because DTs are sensitive to small changes in the sample space of a training dataset. The sensitivity of DTs to the small changes results in different tree structures that may produce poor generalization performance on the same dataset. Therefore, DTs are known to be susceptible to high variance but low bias in prediction (Breiman, 1996a).

Inductive Bias of Decision Trees

For higher branching factors (branching factor depends on the number of values that can be assigned to a feature) in DTs, the inductive bias of DTs selects shorter trees over longer trees when the ID3 search strategy is used to induce the trees (Mitchell, 1997).

Another inductive bias of DTs is the preference given to trees that place high information gain on features close to the root over trees that do not (Mitchell, 1997).

DTs that use information gain have an inductive bias that tests with many outcomes are favoured higher up in the tree. The result of this inductive bias is very large and bushy trees (Thakur et al., 2010; Deng et al., 2011). However, this inductive bias is resolved using the gain ratio.

Furthermore, DTs have an inductive bias due to the requirement that the entropy of leaf nodes is equal to zero. While this outcome depends on the induction stopping condition used, the requirement means that DTs are induced to overfit. This inductive bias results in the possibility to obtain poor generalization performance if test instances are not similar

to training instances. The inductive bias of trees being induced to overfit is addressed either by early stopping or post-pruning (Quinlan, 1986; Fitzgerald, 2014).

DTs form axis-parallel decision boundaries, because the test condition involves a single feature at-a-time. For problems with non-axis parallel boundaries, the induction of DTs results in more complex trees. This inductive bias has resulted in methods proposed to find non-axis parallel boundaries, such as oblique and multivariate methods (Heath et al., 1993).

Regression trees are strongly influenced by outliers in the target features, due to the fact that the predicted value is the average over all target values. Thus, average is biased by outliers. In addition, outliers may influence the splitting point and the potential feature to be selected during tree induction, which could lead to poor predictive performance (Ch'ng and Mahat, 2020).

Another issue is the sensitivity of DTs to class imbalanced datasets. While sampling approaches are understood to improve the induction of DTs, the interaction between sampling and appropriate tree structures is not easily determined (Cieslak and Chawla, 2008). Also, the information gain and Gini measure used by C4.5 and CART algorithms have been reported to be skew-sensitive to class-imbalanced dataset (Flach, 2003). In addition, pruning is considered detrimental to learning from class-imbalanced datasets, because pruning can potentially collapse (small) leaves belonging to the minority class, thus increasing bias towards the majority class (Chawla, 2003).

Lastly, one of the key strengths of DTs is the ability to handle missing values. However, there is no unique approach to properly deal with missing values in DTs regarding tree induction from data. One approach passes all missing values to the node with the highest number of samples. Another approach considers the distribution of the missing values to all child nodes, but with diminished weights, proportional to the number of samples from each child node. A random distribution of the missing values to only one child node based on a categorical distribution is also used to handle missing values. The last approach is the surrogate split method. The surrogate split method means that when a value for a feature is missing, and the feature is required to determine a split, an alternative feature that is highly correlated with the missing feature is considered to determine the direction of the split (Breiman et al., 1993; Tierney et al., 2015; Khosravi

et al., 2020).

2.4.5 *k*-Nearest Neighbour Algorithm

The *k*NN algorithm was introduced by Fix and Hodges (Fix and Hodges, 1951) as a non-parametric method (i.e. *a method that makes no assumption about the underlying data distribution*) for classification and regression problems. The algorithm predicts the class label for a test sample based on the class of the closest k samples in the training data to the test sample for classification problems. The predicted class label is determined using majority voting. For regression problems, *k*NN approximates the relationship between input features and the real-valued target output by averaging the samples in the same neighbourhood (Fix and Hodges, 1951; Pandey, 2017).

The value of k in the algorithm represents the number of nearest training samples, also referred to as *nearest neighbours*. The closest training sample is obtained by computing the similarity or distance between the test sample and each sample of the training dataset. Commonly used distance metrics include the Euclidean distance metric, Minkowski distance, Manhattan measure, cosine similarity, and Hamming distance. While Hamming distance is only used for Boolean features, the selection of a specific metric from other distance metrics depends on the data types of the descriptive features in a dataset (Tahir and Smith, 2010; Hussain et al., 2012; Devi and Sumanjani, 2015).

*k*NN algorithms are known as instance-based learners because the training samples are memorized. The stored samples are then used to predict or estimate a test sample (Pandey, 2017). Figure 2.7 illustrates *k*NN classification using three and seven nearest neighbours for the values of k. For $k = 3$, the test sample, represented as the "green square", is predicted class label 2, i.e. the blue triangles. For $k = 7$, the test sample is assigned to class 1.

The choice of k is critical to the performance of *k*NN to obtain a model that offers the maximum accuracy during prediction. While an odd value of k is usually preferred to avoid ties for classification problems with an even number of classes, selection of the optimal value of k and suitable distance or similarity metric is essential. Appropriate values for these parameters also determine the realization of a generalized *k*NN model that may better balance the bias-variance tradeoff.

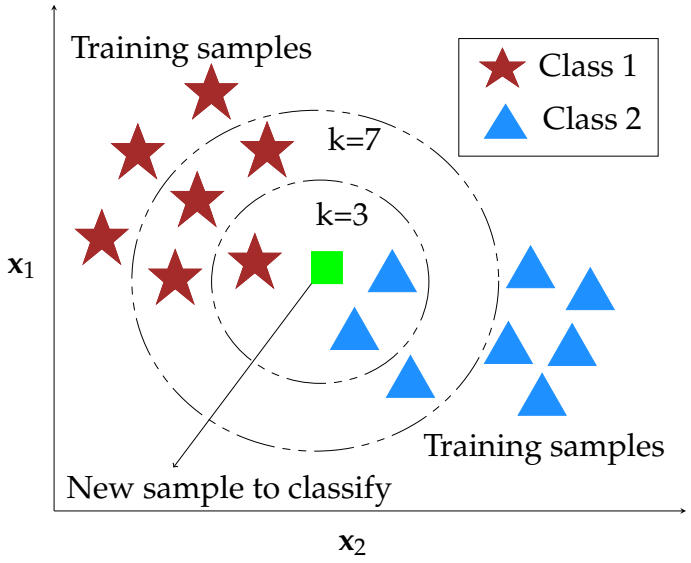

Figure 2.7: kNN Classification

Given an appropriate selection of the value of k and distance metric, research has shown that the generalization performance of a kNN model also depends on the quality of the training dataset, which can be influenced by a number of factors, such as noise, outliers, and skewed class distribution. For a training dataset with noise, a small value of k allows the model predictions to be influenced by the presence of noise in the training dataset and may lead to overfitting. In contrast, a large value of k is required to lower the impact of noisy data, because noisy samples become a minority the larger the value of k, and hence exert little influence on the final vote for classification problems. For regression problems, the effect of noisy data is minimized due to averaging as the value of k increases. However, an excessive increase in the value of k leads to expensive computation during prediction and may result in underfitting (Ougiaroglou and Evangelidis, 2015).

The sensitivity of kNN to outliers is also subject to the value of k, and depends on whether the problem is a classification or regression problem. For classification problems, outliers will likely not be selected as a neighbour for a small value of k. On the other hand, for large values of k, if an outlier is selected as a neighbour, the outlier will be in the minority and will therefore not have a strong impact on the voting process.

For regression problems, outliers are considered with respect to target features and descriptive features in a dataset. For target features, outliers strongly influence model prediction irrespective of the value of k, because the predicted value is the average over

target values. The effect is that the *k*NN regression model will show unstable prediction capability that illustrates more focus on the outliers in the target feature. For descriptive features, outliers are not likely to be selected for small values of k. Very large values of k may include outlier descriptive features and may lead to underfitting and poor prediction performance.

The performance of *k*NNs is also sensitive to skewed class distributions, where one of the classes contains more training samples than the other classes. A very large value of k results in the predictions of a *k*NN model to be biased towards the majority class. This is attributed to the likelihood that the closest neighbours to the test sample are in the majority class, which results in an underfitted prediction. On the other hand, a small value of k reduces the dominance of the majority samples over the minority samples, and may lead to overfitted prediction and low generalization performance (Shi, 2020). Thus, in order to obtain better generalization performance, the requirement is to optimize the value of k.

Upon the analysis of the performance of *k*NN algorithms on the bias-variance tradeoff, *k*NN algorithms have also been reported to be stable algorithms, because *k*NN algorithms are less sensitive to small changes in the training data (Breiman, 1996a; El-Hindi et al., 2018). While this assertion is problem dependent, the stable nature of the algorithm often results in little to no significant improvement in prediction performance (Beliakov and Li, 2012).

Inductive Bias of *k*-Nearest Neighbour Algorithm

For classification problems, the inductive bias of a *k*NN algorithm is the assumption that the prediction of a test sample will be similar to the class of the closest training samples. However, this inductive bias is subject to an appropriate selection of the value of k and efficient computation of a given distance metric based on the data type (Archana and Elangovan, 2014).

For regression problems, the assumption is that the predicted value is similar to the target value of similar samples. However, the average computation is influenced by noise, outliers in descriptive features, and outliers in target features.

The *k*NN algorithm assumes that all features are considered with equal importance to

calculate the distance metric between samples. As a result, the distance computation requires that input attribute values be normalized to ensure that all input feature values are of the same scale. If input feature values are not of the same scale, features with large values will make the most contribution to distance calculation than that of small feature values.

Furthermore, average prediction is influenced by noise if the noise does not have a zero mean. Also, outliers in the input features will influence the distance calculation (although depending on the type of distance metrics). For instance, Euclidean distance is more sensitive to outliers due to the squaring considered when calculating the distance between two samples for all features. As a result, outliers in the input feature will dominate other feature values in the dataset, causing bias during the distance computation (Soundarya and Balakrishnan, 2014; Boehmke and Greenwell, 2020).

The kNN algorithm provides different predictions that are informed by the choice of the value of k. As a result, it is necessary to select an optimal value of k to achieve the best predictive performance.

As discussed earlier, the kNN algorithm is sensitive to noise and outliers for both classification and regression problems as well as skewed class distributions in classification problems. The sensitivity of the kNN algorithm to noise, outliers, and skewed class distributions is subject to the value of k.

kNN implicitly assumes that missing values are uniformly distributed at random in a dataset. The missing values in the input features are simply ignored in the distance or similarity computation, provided that samples do not have too many missing values.

2.4.6 Naïve Bayes Algorithm

The NB algorithm predicts the membership probabilities for a given test sample belonging to a particular class. The NB algorithm is also considered a stable algorithm, because the algorithm is insensitive to noisy data (Breiman, 1996a). The algorithm has shown to be competitive to other ML algorithms. The NB algorithm is useful when the available dataset is characterized by high dimensionality, and has been developed for classification problems (El-Hindi et al., 2018).

To classify a test sample $\mathbf{x}$ with unknown class label, the class j for $\mathbf{x}$, with posterior probability, $P(j|\mathbf{x})$, is predicted using

$$P(j|\mathbf{x}) = \underset{j \in \mathbf{j}}{argmax} \frac{P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n|j) \cdot P(j)}{P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n)} \tag{2.38}$$

where $P(j|\mathbf{x})$ is the posterior probability of class j to be predicted, $\mathbf{j}$ is a vector of all class values and $P(j)$ is the prior probability of class j. $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n|j)$ is the likelihood for the conditional probability that features $1, 2, 3, ..., n$ will take the values $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n$ given that the sample is of class j, while $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n)$ is the feature prior probability that feature $1, 2, 3, ..., n$ will take the values $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n$ respectively (El-Hindi et al., 2018).

Because $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n)$ is the same for all classes, only the product $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n|j) \cdot P(j)$ is maximized (Kantardzic, 2011). The class prior probability is computed as

$$P(j) = \frac{|D_j|}{n} \tag{2.39}$$

where D_j contains all samples of each class j in the class vector $\mathbf{j}$, and n is the total number of training samples. Due to the complex computation of the conditional probability $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n|j)$ for large datasets, the naïve assumption of feature independence is made given the class label. As a result, the conditional probability of all features given the class labels is transformed into independent conditional probabilities of each feature value given the class label. Then the product of the independent probabilities is calculated as

$$P(\mathbf{x}|j) = \prod_{i=1}^{n} P(\mathbf{x}_i|j) \tag{2.40}$$

where x_i are values for features in sample $\mathbf{x}$ and the probability of $P(\mathbf{x}_i|j)$ is estimated from the training dataset. Thus, the new sample $\mathbf{x}$ is classified to a particular class by selecting the class j that yields the largest value in equation (2.40). This decision rule is referred to as the *"maximum a posteriori"* rule given as

$$P(j|\mathbf{x}) = \underset{j \in \mathbf{j}}{argmax} \prod_{i=1}^{n} P(\mathbf{x}_i|j).P(j) \tag{2.41}$$

To balance the bias-variance tradeoff, Breiman (1996b) reported that the NB algorithm often generates low variance, but high bias subject to a defined parameter configuration. The low variance produced by the NB classifier is attributed to the stable nature of the algorithm, because stable classifiers are insensitive to small changes in the sample space of a training dataset. However, Breiman (1996b) stated that the use of an effective feature selection technique may reduce the bias error of an induced NB model.

Inductive Bias of Naïve Bayes Algorithm

The inductive bias of a NB algorithm is the naïve assumption that features in a dataset are conditionally independent given the target (Shi and Lv, 2010). Despite the non-applicability of this assumption to most real-world problems, the NB classifier still produces good generalization performance (Lewis, 1998, 2015).

Another inductive bias of the NB algorithm is the assumption of equal weight for each training sample when computing the conditional probabilities (Liu et al., 2002).

The naïve assumption of the NB algorithm requires features to be uncorrelated. If there are correlations between any features, the calculation of the likelihood may result in over-weighting the correlated features, leading to poor generalization (Gastón and Bagnasco, 2007).

The possibility of a zero probability problem in the dataset is another inductive bias of the NB algorithm. When a sample in the test dataset has a class label that was absent during training, the estimation of the frequency probabilities in equation (2.41) will be zero. Thus, it becomes difficult for the NB algorithm to make predictions (Lowd and Domingos, 2005).

An inductive bias of the NB algorithm is the preference for small probabilities. Based on this bias, multiplication of large feature values may lead to an overflow of the probability estimation due to limited floating-point precision. The problem tends to result in high classification errors during prediction. This bias is addressed by mapping the posterior probability to a log space, which does not always lead to an efficient result (Gastón and

Bagnasco, 2007).

The NB algorithm determines the posterior probabilities of a test sample by computing the frequency of class label for categorical features as given in equation (2.41). For continuous features, the algorithm models the feature distribution as a mixture of Gaussian distributions. With the assumption of a Gaussian distribution, the algorithm models each numeric feature to associate probabilities with the values of the feature. While the Gaussian distribution is a reasonable data distribution assumption for many applications, the distribution is not always accurate in practical applications (John and Langley, 1995).

Further, the NB algorithm can make different assumptions due to the sensitivity of the algorithm to outliers in the training dataset. This problem is attributed to the fact that the prediction of a Gaussian NB algorithm relies on the method of maximum likelihood for parameter estimation, which is defined by the mean vectors and diagonal covariance matrix based on the training dataset. The presence of outliers in the dataset may bias the values of the computed mean vectors and diagonal covariance matrix, leading to an unreliable prediction outcome (Ahmed et al., 2017).

For skewed class distributions, the computation of the posterior probability may generate a misleading result due to the computation of the prior probabilities that will most likely be biased towards the majority classes. Thus, the NB algorithm may achieve poor prediction performance because the chance of the prior probability of minority class resulting in zero is increased.

NB is insensitive to missing values because missing values are inherently handled differently based on whether the missing values exist in the model training or prediction phase. During training, input features with missing values are not included in the frequency count for feature value-class combinations. Also, the input features with missing values are ignored in probability calculation during prediction (Kohavi et al., 1997; Kalousis and Hilario, 2000). However, it is important to consider the problem that may arise from a dataset containing many missing values, which will result in the NB classifier ignoring many input features that may be relevant to the target feature in the dataset.

2.5 Chapter Summary

This chapter discussed the concepts of ML and the bias-variance dilemma. The background information of ML with respect to the conceptual definition and the categories of ML methods was discussed. The chapter further discussed the bias-variance dilemma with a focus on the bias-variance tradeoff, bias-variance loss decomposition, generalization of ML algorithms, and the analysis of underfitting and overfitting of ML models. Then the inductive biases of selected ML algorithms were described in the chapter.

Chapter 3

Machine Learning Ensemble Approaches

3.1 Background

Single ML algorithms have been successfully applied to a wide range of problems providing good predictive performance (Goh and Ubeynarayana, 2017; Poh et al., 2018; Sarkar et al., 2020). However, no single ML algorithm performs best on all problems because different ML algorithms have different inductive biases and exhibit different predictive performances for different parts of data space. This realization has transformed ML research into the possibility of combining the predictions of multiple experts into an ensemble to achieve better generalization performance than the individual experts.

This chapter provides background on ensemble learning. Section 3.2 provides an introduction to ML ensemble learning, while Section 3.3 discusses different ML ensemble approaches. Section 3.4 discusses fusion approaches used to aggregate the predictions of individual experts to obtain an overall ensemble prediction, while Section 3.6 concludes the chapter with a summary.

3.2 Machine Learning Ensembles

Ensemble learning is a learning paradigm in which multiple ML experts are trained to generate different predictions, which are combined to obtain a final ensemble prediction (Akila, 2017). The ensemble paradigm originated from the work of Hansen and Salamon (Hansen and Salamon, 1990) to improve the generalization ability of a neural network through the integration of a number of component neural networks (Chen and Yao, 2009; Zhao et al., 2015; Larasati, 2017; Zhang and Zimba, 2017).

Ensemble learning has been applied to both classification and regression tasks and has shown outstanding prediction performance (Tang et al., 2019; Almeida et al., 2019). For classification problems, the experts learn different decision boundaries on the training data. The predictions obtained due to the different decision boundaries formed by the different members of the ensemble are combined by voting in order to reach a final classification prediction. The combined prediction is expected to outperform the individual predictions of component experts (Rahman and Verma, 2011). In the case of regression problems, the members of an ensemble of regressors learn different relationships between the input features and the target value in the training data. The resulting experts make different predictions that are averaged, such that the final ensemble prediction is usually better than any of the single base experts because the error of the ensemble is reduced. Thus, a ML ensemble is a multiple learner system where each component expert tries to solve the same task (Zhang and Street, 2008).

There are several reasons for creating a ML ensemble. One of the reasons is for *statistical reasons*, where the opinions of multiple experts are consulted to obtain improved performance and to reduce the overall risk of making poor decisions.

Another reason is to deal with *too large and too small datasets*. For specific applications, the amount of data to be analyzed can be too much to be handled effectively by a single ML algorithm. Training a single ML algorithm with an extremely large dataset is challenging and usually impracticable. As a result, splitting the dataset into smaller subsets to train different base learners and combining the predictions of the resulting experts into an ensemble is often a more efficient solution. For small datasets, where there is an inadequacy in the representative set of training data, resampling techniques

can be used to draw overlapping random subsets of the available data, each of which can be used to train different base learners to induce experts that make different predictions combined into an ensemble.

Ensemble learning also performs a *divide and conquer* function when certain predictive problems are too difficult for a given ML algorithm to solve. For instance, for a difficult classification problem, the decision boundary induced by a single classification model may be too complex to separate samples of different classes. In this case, an ensemble of classifiers will outperform the single model by efficiently learning the underlying complex decision boundary in the classification problem. For a difficult regression problem, a single regressor may induce a regression line (*for two-dimensional space*) or a hyperplane (*for higher dimensional space*) that lacks sufficient complexity to capture the underlying patterns in a dataset. Also, the single regression model may perform poorly when the dataset contains outliers, noise, and other issues. On the contrary, an ensemble of regressors can deliver different regression hyperplanes to efficiently learn the relationship between input features and target values in the dataset. Moreso, the ensemble of regressors can neutralize the effect of noise and outliers in the dataset to achieve better predictive performance than a single regressor model. Lastly, ensemble learning has successfully been used for applications in which data from different sources are combined, while single ML algorithms do not perform efficiently for such applications (Polikar, 2006; Vaghela et al., 2009; Iglesias et al., 2014).

According to Dietterich (2000), a single ML approach produces experts with different predictions when trained on different data subsets for the same problem. However, the predictive performance of these experts can be improved by an ensemble of experts from different ML algorithms or multiple instances of the same ML algorithms trained on different data subsets to minimize the errors due to bias and variance or both (Dietterich, 2000). An ensemble can reduce bias-variance errors because the models obtained from a single ML algorithm may be underfitting or overfitting. For datasets characterized by small sample and feature sizes, a single ML algorithm easily makes simplifying assumptions that may be insufficient to capture the underlying trend in the dataset, leading to underfitting. Even when the algorithm is tuned, the algorithm is still constrained to learn only an aspect of the structure of the dataset and may not generalize well on the test dataset.

On the other hand, when the single ML algorithm is too complex, overtrained or trained on an extremely large dataset, the algorithm is likely to capture any inherent noise in the training dataset, which causes overfitting of the resulting model. Therefore, combining multiple experts can significantly contribute to the required prediction accuracy. This is because an ensemble of different ML algorithms or an ensemble of multiple instances of the same ML algorithm trained on different data subsets will provide the capability to learn different aspects of the training dataset. It is expected that the multiple different base experts within an ensemble will complement one another, and more patterns from the underlying structure of the training data can be adequately represented (Brown et al., 2005).

Furthermore, the rationale behind the introduction of ML ensembles is not only to obtain improved predictive accuracy, but also to generate diverse base experts that will result in a reduced generalization error produced by an ensemble in comparison to generalization errors of individual base experts (Kantardzic, 2011). For an ensemble to achieve the stated rationale, each base expert is expected to perform better than a random guess. In addition, the base experts are expected to make different prediction errors on the samples of the training data, such that the resulting error is adequately reduced by the ensemble (Boström, 2007; Rahman and Verma, 2011).

In a ML ensemble, different prediction errors are made because the base experts learn different mapping functions for the data subsets of the same problem, thereby promoting behavioural diversity among the base experts that form the ensemble (Kantardzic, 2011). Diversity in this context refers to the differences in the predictions made by base experts in an ensemble. The role of diversity is crucial to the generalization performance of an ensemble. Diverse experts result in experts with different decision boundaries (*in the context of classification of problems*) or different regression hyperplanes (*in the context of regression of problems*). Different decision boundaries or regression hyperplanes result in experts that make different errors on different samples. Therefore, a combination of the decision boundaries or regression hyperplanes of different experts results in decisions more accurate than individual base experts (Polikar, 2006).

Moreso, for a training data subset, a key focus of ensemble learning is to ensure that the base experts do not agree to similar predictions combined under a given aggregation

scheme to guarantee diversity. Otherwise, the final prediction of the ensemble will be identical to that of any individual base expert, thereby invalidating the significance of ensemble learning (Webb and Zheng, 2004; Domeniconi and Yan, 2004). The diverse base experts are developed using different ML ensemble approaches, including bagging (Breiman, 1996a), boosting (Schapire, 1990; Freund and Schapire, 1996), random feature subspace method (Ho, 1998), stacking (Wolpert, 1992), and others. Furthermore, diversity among the individual base experts is enhanced by the different inductive biases of the base experts to generalize in distinctive ways (Zhang and Street, 2008).

Typically, ensemble learning consists of two phases, as illustrated in Figure 3.1. The first phase involves the generation of multiple base experts, and the second phase considers the combination of the predictions made by the base experts (Webb and Zheng, 2004). In the first phase, the central focus is not only to generate multiple base experts, but also to efficiently select an optimal number of multiple base experts, bag sizes, data sampling technique, and combination strategy that will guarantee a reduced generalization error by an ensemble compared to that of the component experts (Bian and Wang, 2006; Cai and Wu, 2010).

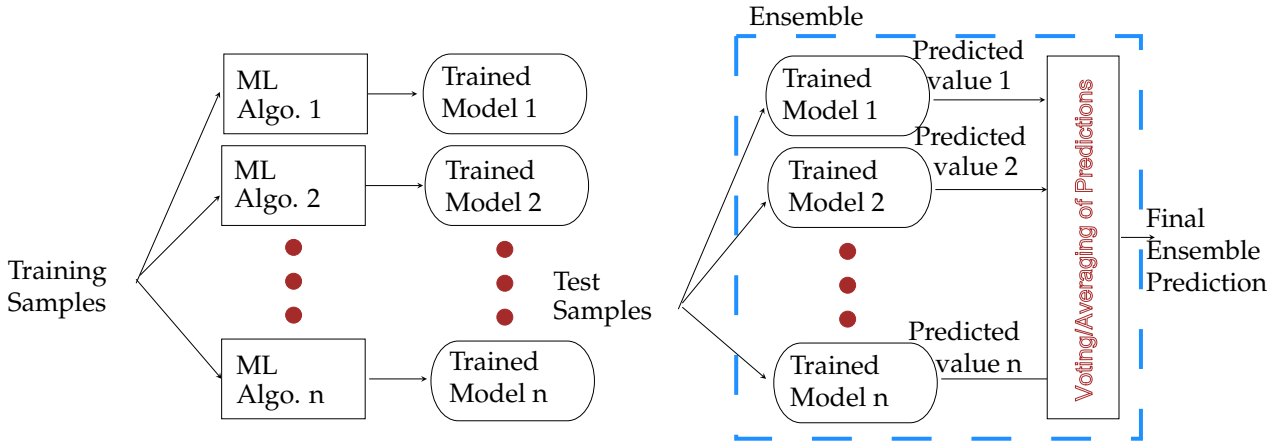

Figure 3.1: ML Ensemble

Further, the generalization error of an ensemble is reduced from the perspective of bias-variance loss decomposition. The plausibility is that an ensemble addresses the bias-variance tradeoff by capitalizing on the benefits of the different ensemble approaches (Thewsuwan et al., 2018). Each ensemble approach contributes to the reduction of the generalization error of the ensemble using different strategies. For this reason, it is much more likely that the ensemble will misclassify less than the individual members of the

ensemble (Kantardzic, 2011; Alkubaisi, 2019).

3.3 Machine Learning Ensembles Approaches

As stated in Section 3.2, the goals of any ML ensemble are to generate accurate and diverse base experts. Then, the predictions made by the base experts are combined to obtain improved generalization performance better than individual component experts of the ensemble (Yang et al., 2013). This section discusses different ensemble approaches used to generate diverse base experts within an ensemble.

3.3.1 Bagging

Bagging, coined from *boostrap aggregation*, is an ensemble approach proposed by Breiman (Breiman, 1996a) to create diverse experts within an ensemble (Fazelpour et al., 2016). Breiman (1996a) observed that multiple versions of an expert could be generated using bootstrapped replicates of a training dataset. The bootstrapped replicates are referred to as *bagged subsets*, drawn uniformly using random sampling with replacement (Li et al., 2011a; Rahman and Verma, 2011).

Sampling with replacement creates the possibility that a training sample may appear more than once or not at all in a bagged subset (Kantardzic, 2011; Liang and Zhou, 2012). Therefore, training of individual base experts on the bagged subsets guarantees that the experts behave differently on each bagged subset. The base experts are homogeneous in nature, and training on the bagged subsets is performed in parallel, which results in significant differences in the decisions among experts (Polikar, 2006; Rahman and Verma, 2011).

A fusion method is required to combine the individual predictions of the experts to obtain an overall prediction of an ensemble. Fusion of expert predictions can be realized using a voting method for classification problems or an averaging method for regression problems (Li et al., 2011a). Random forest (Breiman, 2001), pasting small votes method (Breiman, 1999a), and wagging (Bauer and Kohavi, 1999) are variants of the bagging approach that have generated reliable results in different ML applications.

3.3.2 Boosting

Boosting is another ensemble approach, proposed by Freund and Shapire (Schapire, 1990; Freund and Schapire, 1996), to create diverse base experts within an ensemble. The boosting approach involves the combination of weak base experts to generate a strong expert that can correctly classify all but a small fraction of test data using a sequential learning approach (Schapire, 1990; Polikar, 2006; Ahmadian et al., 2007; Verma and Mehta, 2017; Lawi et al., 2018).

Boosting approaches assign equal initial weights to all samples in a training dataset, and weak base learners are trained to induce an expert. After the base experts make predictions, samples in the training data are adjusted to concentrate only on the samples misclassified by the weak base experts. Then weight adjustment is performed such that the weights of misclassified samples are increased while the weights of correctly classified samples are decreased. The misclassified samples are later included in the training set for the next training. This way, resampling is strategically performed to provide the most informative training data for consecutive base experts (Wan and Yang, 2013). Then subsequent base experts are constructed by fitting the misclassified samples of the initial experts from the training data (Baneriee et al., 2018). The final strong expert is obtained through the combination of the weighted votes of all weak base experts (Vaghela et al., 2009).

Boosting approaches include the Adaboost algorithm (Freund and Schapire, 1996), gradient boosting algorithm (Freund and Schapire, 1997; Breiman, 1999b; Friedman, 2001) and extreme gradient boosting (XGBoost) algorithm (Chen and Guestrin, 2016).

3.3.3 Stacked Generalization

Another approach to creating diverse experts to generate varying predictions within an ensemble is to combine different types of ML algorithms. Stacked generalization, also known as *stacking*, was proposed by Wolpert (Wolpert, 1992) to improve the predictive performance of a ML ensemble using different ML algorithms.

The different base learners are trained on random subsets of a training dataset using a cross-validation method (Lee, 2017). Then by capitalizing on the inductive biases of the

different underlying base learners, multiple different experts are induced and generate different predictions. The predictions made by the base experts are combined with the target class of the original training data to construct meta-data. The meta-data is used as a new training dataset to train another ML algorithm which performs the function of a meta-classifier that generates the final prediction of the ensemble (Jurek et al., 2011; Gupta and Thakkar, 2014; Czarnowski and Jedrzejowicz, 2017; Hnoohom and Jitpattanakul, 2018). Therefore, stacking guarantees diversity within an ensemble by capitalizing on the intrinsic properties of different ML algorithms used to construct the ensemble (Tahir and Smith, 2010).

3.3.4 Random Feature Subspace Method

The random feature subspace method (RFSM) is an ensemble approach proposed by Ho (Ho, 1998). The RFSM approach has been used to solve the problem arising from the curse of dimensionality in a dataset. The curse of dimensionality in a dataset occurs when the number of features largely outnumbers the number of samples in the dataset (Biggio and Corona, 2015). For effective model performance, the dimension of the data is reduced through feature selection and resampling.

Input features are randomly selected with replacement into different subsets to train the base learners in order to construct an ensemble. Each feature subset is referred to as a *"feature subspace"*. Hence, the RFSM ensures that the base experts within the ensemble behave differently in prediction by generating diverse predictions combined to obtain the final prediction of the ensemble (Ho, 1998; Wang et al., 2015). The approach also guarantees augmentation with additional changes, which include using bootstrap or a random sample of the rows in a training dataset (Ho, 1998).

3.3.5 Parameter and Hyperparameter Tuning

ML algorithms have internal parameters that are not tuned but learned from data during model training. The parameters are referred to as *decision variables*. Furthermore, the algorithms consist of hyperparameters that are tuned during model building and are referred to as *control parameters* (Engelbrecht, 2007; Lin et al., 2017).

The tuning of the hyperparameters of ML algorithms has been formalized as an

optimization problem to address diversity within an ensemble (Cachada et al., 2017; Preuveneers et al., 2020). The formalization involves selection of different configurations for the hyperparameters of base learners in the ensemble. As discussed in Section 2.3, the different configurations of the base learners define different levels of model complexity obtainable from the learners. Then, the inductive biases and the model complexity of individual base learners lead to different predictions by the resulting diverse experts.

Due to the large numbers of hyperparameters in different base learners within an ensemble, the possibility to obtain optimal predictive performance in the ensemble depends on the methods used to optimize the hyperparameters of the base learners. Generally, two methods are used to perform hyperparameter optimization of ML algorithms, i.e. *manual search* and *automatic search* methods (Claesen and De Moor, 2015; Dodge et al., 2017; Gokalp and Tasci, 2019; Wu et al., 2019).

The manual search method is used to tune the values of the hyperparameters of base learners by hand. The method requires technical experience, which is usually subjected to trial and error. The key to obtaining optimal ensemble results using manual search is to select hyperparameters that are more significant for individual base learners in the ensemble (Wu et al., 2019).

As a result of the subjective requirements of the manual search method, the method is usually time-consuming (Feurer et al., 2014). Another problem of the manual search method is managing a large number of hyperparameters and the range of values for the hyperparameters. In addition, it becomes challenging to handle data with high dimensions. These challenges often result in possible misinterpretation of the relationships and trends among hyperparameters (Wu et al., 2019).

Automatic search methods have been proposed to solve the problems of manual search methods (Feurer et al., 2014; Claesen and De Moor, 2015; Luo, 2016; Dodge et al., 2017; Mantovani et al., 2019; Kadam and Jadhav, 2020). Grid search (Bergstra et al., 2011) is a simple automatic method that implements exhaustive search through specified hyperparameter values (Cachada et al., 2017). Grid search methods train base learners with different combinations of hyperparameter values to obtain optimal ensemble performance. However, the performance of base experts quickly drops when the hyperparameters and the range of values of the hyperparameters increase excessively

(Wu et al., 2019).

Bergstra and Bengio (2012) proposed a random search method to solve the expensive computational cost of grid search methods. Random search method provides optimal hyperparameter values through the selection of important hyperparameters for individual base learners to achieve optimal performance. The random search method implements random combinations of hyperparameter values. This way, the hyperparameter search space is reduced to hyperparameters that only contribute to the final ensemble results (Wu et al., 2019). However, Bergstra and Bengio (2012) reported that the random search method might not be effective to optimize complex ML experts and suggested the use of advanced hyperparameter tuning strategies. Such advanced optimization methods include the Bayesian optimization and evolutionary algorithms (Hutter et al., 2011; Bergstra et al., 2011; Bachoc, 2013; Eggensperger et al., 2013).

Bayesian optimization repeatedly fits a probabilistic model as each hyperparameter combination is tested. The method adopts the Bayesian theorem, and the outputs of any hyperparameter combination provide good suggestions for the next value combination (Cachada et al., 2017). Evolutionary optimization strategies include particle swarm optimization (Kennedy and Eberhart, 1995; Engelbrecht, 2007; Lin et al., 2008; de Miranda et al., 2012), genetic algorithms (Tsai et al., 2006; Reif et al., 2012), coupled simulated annealing (Souza et al., 2010), tabu search (Gomes et al., 2012), and racing algorithms (Birattari et al., 2010).

3.3.6 Class Manipulation

Class manipulation is an approach used to solve multi-class classification problems in ensemble learning (Joshi and Kulkarni, 2014).

A multi-class problem is transformed into multiple smaller binary classification problems such that the base learners within an ensemble are trained to solve a two-class problem. The base learners are constructed with different and simpler representations of the target classes in the training data. As a result, the scope of the new classes is made smaller than the original class in the training data.

The different predictions made by the base experts are representative solutions of the original multi-class problem. Thus, the base learners are considered to solve different

target concepts, which guarantees diversity in the ensemble. The different predictions are combined to obtain the final prediction of the ensemble (García-Pedrajas and Ortiz-Boyer, 2011; Rocha and Goldenstein, 2014; Gopika and Azhagusundari, 2014).

Dietterich (2000) discussed a formal analysis of a multi-class problem into learnable two-class problems by the base learners in an ensemble. The study discussed three methods to transform a multi-class problem into a multiple binary classification problem. The methods include one-vs-one (Knerr et al., 1990), one-vs-all (Anand et al., 1995), and error-correcting output codes (Dietterich and Bakiri, 1995). Each of the methods has been shown to enhance diversity within an ensemble (Dietterich, 2000; Platt et al., 2000; Raschka, 2018; Bagheri et al., 2014).

3.4 Fusion Approaches for the Predictions of Experts

A fusion approach is necessary to combine the predictions made by the diverse base experts in an ensemble. A key factor is to select an appropriate fusion approach to optimally realize complementarity among the diverse base experts (Zhang et al., 2018). By complementariness, the individual experts could make up for the deficiencies of one another in the mixture model to enable the ensemble to generate a correct and improved decision (Martinez-Muñoz et al., 2009; Yang et al., 2013). This section discusses different voting and averaging methods used to combine the predictions of experts in an ensemble.

3.4.1 Voting Methods for Classification Problems

Voting is a method used to fuse the decisions of base experts in an ensemble for classification problems (Zhou, 2012; Xie et al., 2017). The method is mostly used to combine categorical predictions generated by classification algorithms (Opitz and Shavlik, 1996; Freund and Schapire, 1997). The different voting methods, i.e., unweighted and weighted majority voting, are discussed next.

3.4.1.1 Unweighted Voting

Unweighted voting is used to combine the predictions of base experts within an ensemble when the prediction of each expert is observed as a single vote. Then, the individual

predictions of the experts are known as votes. The different approaches of the unweighted voting are presented next.

Majority Voting

Majority voting is considered the most popular voting method. Austen-Smith and Banks (1996) explained the concept of majority voting in relation to the condercent jury theory. Austen-Smith and Banks (1996) stated that in an uncertain situation where a decision is to be made, the probability that the majority will make correct decisions is higher than the probability of any individual decisions (Austen-Smith and Banks, 1996).

Each base expert votes a single class label, and the final class prediction for an ensemble is the class that has more than half of the votes. A test sample is rejected when there is no class label out of the predictions that takes the majority of the votes. While this situation may not likely occur in an ensemble with many base experts, it is still a potential issue in majority voting. The outcome will be the inability of the ensemble to classify samples correctly (Van Erp et al., 2002). Formally, the class label that receives half of the votes is calculated as

$$H(\mathbf{x}) = \begin{cases} C_j & if \ \sum_{i=1}^{T} h_i^j(\mathbf{x}) > \frac{1}{2} \sum_{k=1}^{l} \sum_{i=1}^{T} h_i^k(\mathbf{x}) \\ reject & otherwise \end{cases} \tag{3.1}$$

where $H(\mathbf{x})$ is the final ensemble prediction, C_j is returned as the class label with majority votes, $h_i^j(\mathbf{x})$ represent the class prediction for a sample $\mathbf{x}$ by a base expert, and T indicates the total number of base experts in the ensemble. The ensemble will then make a correct prediction if at least $(T/2+1)$ experts select the correct class label (Zhou, 2012). Thus, majority voting gives the accuracy of the ensemble, P_{maj}, as

$$P_{maj} = \sum_{f=(n/2)+1}^{n} \binom{n}{f} p^f (1-p)^{n-f} \tag{3.2}$$

where f indicates the number of majority votes, n is the number of base experts in the ensemble, and p represents the probability of a base expert to generate an accurate prediction. The majority voting calculated in equation (3.2) indicates that the accuracy of

the individual experts within an ensemble is directly proportional to the overall accuracy of the ensemble. As the value of n approaches infinity, the accuracy of the ensemble, P_{maj}, monotonically increases when $p \geq 0.5$ as

$$P_{maj} \rightarrow 1 \ \ if \ \ p > 0.5, \tag{3.3}$$

while the accuracy monotonically decreases when $p \leq 0.5$ as

$$P_{maj} \rightarrow 0 \ \ if \ \ p < 0.5. \tag{3.4}$$

Also, the accuracy of the ensemble equals 0.5 for any value of n as

$$P_{maj} \rightarrow 0.5 \ \ if \ \ p = 0.5 \tag{3.5}$$

Plurality Voting

Contrary to majority voting, plurality voting computes the final classification result of an ensemble as the most frequent class label. This means that the class label that generates the highest number of votes becomes the ensemble prediction (Zhou, 2012). The final class label is determined using

$$H(\mathbf{x}) = C_{\underset{j}{arg\ max}} \sum_{i=1}^{T} h_i^j(\mathbf{x}) \tag{3.6}$$

where C is the class label with majority votes out of j votes. The plurality voting computed in equation (3.6) illustrates that there is no reject option, unlike the simple majority voting in equation (3.1). This is because plurality voting always finds a label that receives the largest number of votes. Moreover, for binary classification problems with only two class labels, the plurality voting strategy is equivalent to majority voting, because the class label that has the highest number of votes becomes the class with the majority of the votes. However, plurality voting is not effective for multi-class classification problems. The ineffectiveness of plurality voting is attributed to designating the final prediction of the ensemble as the class label that has not received many votes, but has received more than any other. Then, as the number of class labels to be predicted increases, so too does

the possibility of this drawback occurring (Van Erp et al., 2002; Zhou, 2012).

Amendment Voting

The drawback of plurality voting for multi-class classification problems is solved using the amendment voting rule. First, the amendment voting rule determines the majority voting result between two class labels. The label that receives the highest votes is then matched to the next label with another majority vote. The process continues until a final class label with the majority votes between the last two labels is obtained. The final class label becomes the prediction of the ensemble prediction. However, the amendment voting strategy has been shown to be biased towards the class labels that are matched and analyzed in the final voting process (Van Erp et al., 2002).

3.4.1.2 Weighted Majority Voting

Weighted majority voting is a strategy used to measure the confidence of the base experts within an ensemble when the experts generate unequal accuracy. For the weighted majority strategy, more relevance is given to base experts with more accurate predictions (Zhou, 2012). This means that weights are assigned to experts based on the classification performance of individual experts in the ensemble.

Before training begins, the weights of the base learners are initialized to 1. As each learner processes the training samples, the weights of the experts that correctly predict the correct class label of a sample are incremented by the ratio of the number of experts with wrong predictions to the total number of base experts (Dogan and Birant, 2019). Given the weight w_{ij} of a number of base experts ($j = 1, 2, 3, ..., n$) on training samples ($i = 1, 2, 3, ..., m$), the weight for the vote of each base experts is recalculated as

$$w_{ij} = \begin{cases} w_{1-1,j} + \alpha_i & if\ j^{th}\ base\ expert\ correctly\ predict\ i^{th}\ sample \\ w_{1-1,j} & if\ j^{th}\ base\ expert\ incorrectly\ predict\ i^{th}\ sample \end{cases} \tag{3.7}$$

where α_i is the change in weight and calculated as $\alpha_i = Y_i/n$, where Y_i represents the number of incorrect predictions for the i^{th} sample, and n is the number of base experts. When all the base experts have traversed all training samples, the obtained weights are used to achieve the vote for each base expert to predict the class labels of the samples.

The final decision is computed as the summation of all weighted votes for each class as

$$g_k(\mathbf{x}) = \sum_{j=1}^{n} w_j \, d_{jk} \tag{3.8}$$

where w_j is the weight assigned to the prediction of j^{th} expert, n represents the number of base experts, and k is the index of a specific prediction for j^{th} expert. The value d_{jk} is calculated for all class labels as

$$d_{jk} = \begin{cases} 1 & if \ R_j \ selects \ C_j \\ 0 & if \ otherwise \end{cases} \tag{3.9}$$

where R_j is a base expert j, and C_j is the predicted class label of a base expert out of all possible class labels in the training data. The class label that receives the highest weighted vote is designated as the final ensemble prediction.

The range of the values for the weight assigned to the prediction of an expert after traversing each training sample is between zero and one. Hence, a normalization equation is computed to ensure that the weight gain value cannot exceed one as

$$w_j \geq 0 \text{ and } \sum_{j=1}^{n} w_j = 1 \tag{3.10}$$

The process of weight assignment to base experts in an ensemble is only efficient once the performance of each expert has been determined. However, the process is considered a weakness to the weighted majority voting strategy because determining the weight values is complex and subjective. When incorrect weight values are assigned to each expert, the ensemble tends to generate an overall performance worse than the performance obtained using the unweighted voting strategies (Dogan and Birant, 2019).

3.4.2 Averaging Methods for Regression Problems

Averaging is applied to regression problems to combine real-valued outputs generated by base experts to obtain the final prediction of an ensemble (Khan et al., 2018). Averaging methods, which consist of simple and weighted averaging methods, are discussed in this

section.

3.4.2.1 Simple Averaging

Simple averaging is the most popular averaging method to generate an overall ensemble prediction (Zhou, 2012). Each base expert within an ensemble has the same influence due to the assignment of equal weights to the predictions of all experts (Fu and Browne, 2007). The overall prediction of an ensemble is calculated using the simple averaging method as

$$H(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} h_i(\mathbf{x}) \tag{3.11}$$

where $h_i(\mathbf{x})$ is the i^{th} prediction of a base expert for sample $\mathbf{x}$. The simple averaging method provides the final prediction with the assumption that the errors of the individual base learners are uncorrelated. However, despite the simplicity of the method, there is an inefficiency to reduce the prediction errors in an ensemble. The inefficiency is attributed to the high correlation of the errors generated by the base experts in the ensemble because the experts are induced on the same problem (Zhou, 2012). Furthermore, the simple averaging method is adversely influenced by outliers in a dataset, which results in the possibility of the base experts within an ensemble providing misleading predictions. These misleading predictions lead to underestimation or overestimation in the final ensemble prediction (Kwak and Kim, 2017).

3.4.2.2 Weighted Averaging

Weighted averaging measures the importance of the predictions of individual experts in an ensemble. The weighted sum of the predictions provides an overall prediction of the ensemble (Fu and Browne, 2007). Weight assignment is performed in terms of the accuracy of prediction generated by each expert or other selected performance metrics. An overall ensemble prediction is calculated using the weighted averaging method as

$$H(\mathbf{x}) = \sum_{i=1}^{n} w_i \, h_i(\mathbf{x}) \tag{3.12}$$

where w_i is the weight assigned to the prediction of each expert, which follows the approach used to compute weights in equation (3.9). The weights are also subject to the

constraints given in equation (3.10).

While weighted averaging is considered an advanced case of simple averaging, there has been no reported superiority between the two methods. Weighted averaging is usually not preferred when the available data is noisy and insufficient. In addition, learning different weights when the ensemble size is too large tends to lead to overfitting. Also, weight determination is regarded as a computationally hard problem (Zhou, 2012). On the other hand, weighted averaging is preferred when the base experts exhibit different performances. Therefore, there is no averaging method that is consistently the best because the selection of each method depends on different factors such as data types, complexities of data, and base algorithms.

3.4.3 Advanced Fusion Methods

Advanced fusion methods are reported in literature to combine the predictions of base experts in an ensemble. *Bayesian combination* computes the weight assigned to each expert in an ensemble as the posterior probability of the expert given a training dataset (Buntine, 1990). *Performance weighting* relies on the evaluation of the weight of an expert as a direct proportion to the accuracy of the expert on a validation set (Opitz and Shavlik, 1995). *Distribution summation* computes the sum of the conditional probability vectors obtained from each expert (Clark and Boswell, 1991).

Another method is the *Dempster–Shafer* strategy (Dempster, 1967; Shafer, 1976) which applies the principle of basic probability assignment to combine the predictions of experts within an ensemble (Shilen, 1990). The *variance optimized bagging* strategy, also known as *Vogging*, has been used to perform optimization of a linear combination of experts to improve accuracy and to reduce variance (Derbeko et al., 2002).

3.5 Impact of Ensemble Approaches on the Bias-Variance Dilemma

This section discusses the impact of the ensemble approaches on the bias-variance tradeoff, to obtain better generalization performance.

For the bagging approach, the diversity among the base experts of an ensemble has its

origin in the statistics of the random bootstrap sampling process. As discussed in Section 3.2, the number of experts combined in an ensemble is one of the factors that influence the performance of the ensemble. The number of experts indicates the ensemble size, which is usually determined by the required accuracy, computational cost, the nature of the data, number of iterations in training, and the number of processors available for parallel learning (Rokach, 2009). The ensemble size is significant to the construction of bagging-based ensembles.

The ensemble size has an influence on balancing the bias-variance tradeoff because the generalization error of a bagged ensemble becomes smaller as the number of experts aggregated in the ensemble increases. A point is reached whereby the reduction of the generalization error reaches a plateau as more experts are added. This point is considered the best result achieved by the bagging approach (Martinez-Muñoz et al., 2009). For instance, Breiman (1996a) showed empirically that an ensemble size of 10 experts generated the most significant decrease in misclassification rate and that more than 25 experts are unnecessary. Hence, the amount of overfitting does not generally increase with the number of experts combined in the ensemble due to the statistical origin of error reduction in the bagging approach (Breiman, 2001).

Furthermore, the bagging approach allows the base learners within an ensemble to be trained in parallel, where the induced base experts make independent and uncorrelated prediction errors (i.e. variance). It is expected that the combination of these errors made by the base experts will result in an ensemble that decreases the overall prediction error.

Freund and Schapire (1996) explained the application of the boosting approach to address the bias-variance tradeoff: boosting reduces the bias error by focusing on misclassified samples of previous base experts to obtain the next base experts. The final strong expert is largely different from individual experts in terms of generalization error. In addition, the boosting approach is credited with the capability to construct a strong expert that is not producible by the base experts with simple inductive biases. An illustration of this is to change linear predictions of a classifier into a classifier that contains non-linear predictions (Opitz and Maclin, 1999). Therefore, due to the different inductive biases of the base experts, the adaptive weight adjustment of the boosting approach influences the reduction of the bias error.

Breiman (1996b) showed empirically that the boosting algorithm is capable of reducing the variance error using a stopping iteration criterion during training. Very late stopping of the algorithm may cause the final ensemble model to overfit due to an increase in the complexity of the final expert. On the other hand, stopping the algorithm too early would not only lead to higher error on training data, but could as well result in poor predictions on new data (Mayr et al., 2014).

For stacked generalization, the inductive biases of the different algorithms within an ensemble contribute to the generation of different prediction errors made by individual algorithms. This is significant to balancing the bias-variance tradeoff within the ensemble. The advantage of combining the different algorithms, therefore, influences the minimization of the generalization error of the ensemble (Wan and Yang, 2013). A meta-learner is used to combine the predictions of the base algorithms. The meta-learner fits the prediction errors generated by the different base experts induced by the algorithms. The combination of these prediction errors results in an overall reduction in the generalization error of the ensemble (Large et al., 2019). However, the decision to construct a stacked-based ensemble that will produce effective generalization performance is determined by different factors. These factors include the types of base algorithms and meta-learner selected, ensemble size, appropriate configurations of the control parameters for base learners and meta-learner, and the number of folds required for the cross-validation process (Gupta and Thakkar, 2014; Moudrik and Neruda, 2015; Kadkhodaei and Moghadam, 2016).

Ho (1998) stated that the presence of redundant and irrelevant features in a training dataset leads to high dimensionality in the feature space for RFSM. The redundant and irrelevant features significantly influence the bias-variance tradeoff in an ensemble. Removal of irrelevant features has less importance in the reduction of the bias error for the base learners in the ensemble, because irrelevant features are treated as noise (Van Der Putten and Van Someren, 2004).

Furthermore, selection of relevant features is directed at reducing the variance error. This way, individual learners estimate fewer parameters to generate low independent errors while the amount of relevant information that is removed is minimized. However, the removal of relevant features may lead to an increase in intrinsic bias (Van Der Putten and

Van Someren, 2004; Merentitis et al., 2014). Thus, by taking advantage of the principle of RFSM, minimal bias and variance errors are generated. In addition, the inductive biases of the diverse base experts in the ensemble contribute to the reduction of the final generalization error.

Hyperparameter optimization performs an important role in the reduction of the generalization error of an ensemble. As discussed in Section 2.3.5, a too complex model will memorize training data and learn the inherent noise present in the training data. The outcome is a model with poor generalization performance due to *model overfitting*, i.e. high variance.

On the other hand, a model with too low complexity will fail to capture all the information and patterns present in the training data, leading to high bias. The model will also perform poorly on test data. This refers to *model underfitting*. Therefore, the selection of a suitable level of model complexity defined by the hyperparameters of the base learners in an ensemble is significant in balancing the bias-variance tradeoff.

Class manipulation has been shown to address the bias-variance tradeoff in an ensemble (Kong and Dietterich, 1995; Windeatt and Ghaderi, 2003; Brown et al., 2005). However, the approach has received little attention because the application of the method is mainly found in multi-class classification problems (Furnkranz, 2002). Furthermore, there is no significant improvement in generalization performance reported over other ensemble approaches (Wang and Zhang, 2010).

3.6 Chapter Summary

This chapter discussed all relevant information regarding ensemble learning. Section 3.1 presented the background to the chapter. ML ensembles and ensemble approaches to create diverse experts in an ensemble were discussed in Sections 3.2 and 3.3.

Section 3.4 discussed different fusion approaches to combine the predictions of experts. The discussion showed that there is no fusion approach that is consistently the best on every problem. The selection of a suitable fusion approach depends on the formulated research problem, available data, ensemble approach, selected base ML algorithms, data types, and complexities of data. The impact of the ensemble approaches on the bias-

variance dilemma was presented in Section 3.5.

Chapter 4

Homogeneous Ensembles

4.1 Introduction

This chapter presents a critical review of mixtures of homogeneous experts, i.e. homogeneous ensembles. The review investigates the different components of a homogeneous ensemble including diversity approaches, fusion approaches used to combine the predictions of base experts within the ensemble, ensemble size, and tuning strategies used to address the inductive biases of the base learners and the bias-variance dilemma to obtain efficient generalization performance. Section 4.2 presents background information of the chapter, while Section 4.3 discusses the review strategies followed in the chapter. Sections 4.4 to 4.8 reviews past studies of SVM ensembles, NN ensembles, RF, nearest neighhbour ensembles, and NB ensembles algorithm. Section 4.9 presents the limitations of the different implementations of homogeneous ensembles, while Section 4.10 concludes the chapter with a summary.

4.2 Background

In ML, the *"No-Free-Launch"* theorem (Wolpert, 1996) presents the concept that there is no single ML algorithm that performs best on all problems, because different ML algorithms

exhibit different learning biases and different assumptions on data. Hence, integration of ML algorithms into an ensemble can result in better generalization performance and may better balance the bias-variance dilemma (Breiman, 1996b) of the base learners (Tang et al., 2019). This objective is achieved by combining multiple instances of the same learning algorithm or different learning algorithms to build an ensemble. Combinations of the same ML algorithms as the base learners in an ensemble are known as *"Homogeneous Ensembles"*, while combinations of different base learning algorithms are referred to as *"Heterogeneous Ensembles"* (Kilimci et al., 2016).

The crux for developing homogeneous ensembles is not only to get accurate predictions, but also to create diverse learners that generate different assumptions and prediction errors. Diversity with respect to the inductive biases of learning algorithms in an ensemble represents the key to improving the prediction performance of the ensemble (Polikar, 2006). In a bid to address diversity within an ensemble, several authors have investigated the significance of introducing diversity in homogeneous ensembles using various ensemble approaches such as sampling training data, altering feature sets, and using different parameters and hyperparameters for each ensemble member (Wang et al., 2018). Also, using a RFSM has been shown to be an effective strategy in enhancing diversity in homogeneous ensembles (Kilimci et al., 2016). All of these approaches have been discussed in Section 3.3 of this thesis.

4.3 Review Strategy

The goal of the review conducted in this chapter is to identify and analyze different implementations of homogeneous ensembles. Therefore, the strategies followed to perform a detailed systematic review of the construction of different homogeneous ensembles are discussed next:

- **Search for relevant terms and form search strings:** The search terminologies and search strings used include "Machine Learning Ensemble", "Homogeneous Ensembles", "Ensemble Approaches", "Diversity in Ensembles", "Fusion Approaches in Ensembles", "Hyperparameter Optimization in Ensembles", "Ensemble Machine Learning", "Support Vector Machine Ensemble", "Neural Network Ensembles", "Random Forest", "Nearest Neighbour Ensembles", and

"Naïve Bayes Ensembles".

- **Select Bibliographical Database:** The bibliographical databases IEEE Xplore, Scopus, ACM Digital Library, and PubMed were used to select papers reporting the implementations of different homogeneous ensembles constructed for the identified ML algorithms, i.e. SVM, NN, RF, *k*NN, and NB. The literature studies published between 2008 and 2022 were considered. A total of 83 articles were finally selected for the review of homogeneous ensembles developed for each of the ML algorithms.
- **Inclusion Criteria:** A number of factors were considered as the inclusion criteria for the selection of relevant studies out of all available studies for each homogeneous ensemble. One of the factors includes articles that have used SVM, NN, DT, *k*NN, and NB as base learners for the construction of individual ensembles. Another factor considers the articles where the components of an ensemble (*i.e. diversity, fusion approach, tuning approach, and ensemble size*) were implemented. Other factors include articles available in full text, and articles written in English.
- **Exclusion Criteria:** Also, a number of factors were identified as exclusion criteria to ignore selection of irrelevant studies out of all available studies for each homogeneous ensemble. One factor is the exclusion of articles written in other languages. Duplicated and repeated articles are also excluded. Articles that have used more than one of the identified ML algorithms as base learners were excluded, because the focus of the review is mainly pure homogeneous ensembles constructed with base learners of the same ML algorithms.
- **Study, analyze and summarize findings:** For each homogeneous ensemble, the selected articles were studied and analyzed comparatively to examine the approaches used to implement the components of the homogeneous ensembles. The findings are summarized and presented in this chapter.

4.4 Support Vector Machine Ensembles

The application of SVMs to various problems has recorded huge success due to the capability of SVMs to solve both linear and non-linear problems. These capabilities have motivated different authors to use SVMs to develop homogeneous ensembles. In

a review of SVM ensembles where the base learners are SVM algorithms, the different implementations of the components of SVM ensembles are presented in Table 4.1.

Table 4.1: Comparison of the implementations of SVM ensembles

Study	Diversity Approach	Fusion Method	Tuning Approach	Ensemble size
Guo et al. (2010)	Non-random data sampling	Unspecified	Grid search	Unspecified
Kundu and Ari (2019)	Non-random data sampling	Simple averaging	Manual	5
Bhatnagar et al. (2016)	Random data sampling	Simple averaging	Manual	17
Sun et al. (2013)	Cross validation	Weighted majority voting	Manual	15
Arefi and Chowdhury (2017)	Cross validation	ANFIS	Grid search	6
Pang et al. (2017)	Cross validation	Majority voting	Simulated annealing	Unspecified
Ma et al. (2011)	Bagging	Majority voting	Manual	20
Eeti and Buddhiraju (2018)	Bagging	Majority voting	Manual	25
Bhavan et al. (2019)	Bagging	Simple averaging	Manual	20
Liu and Huang (2019)	Adaboost	Weighted majority voting	Particle swarm optimization	Unspecified
Nakayama et al. (2009)	Bagging, Boosting	Majority voting	Manual	3
Lu and Wang (2011)	Bagging, Boosting	(v-SVR) technique	Manual	Unspecified
Abdullah et al. (2009)	RFSM	Product rule	Grid Search	Unspecified
Li et al. (2017a)	RFSM	SVM metalearner	Grid Search	5
Lee and Lee (2014)	k-means clustering	Simple averaging	Manual	20
Ahmed et al. (2010)	RFSM, k-means clustering	Majority voting, SVM metalearner	Grid Search	Unspecified
Liu et al. (2020)	fuzzy c-means clustering	Simple averaging	Manual, Genetic algorithm	4

The comparison of the implementations in Table 4.1 highlights the different approaches used to address diversity in SVM ensembles. A non-random data sampling approach was implemented in the studies of Guo et al. (2010) and Kundu and Ari (2019), while Bhatnagar et al. (2016) randomly sampled the training dataset into different subsets to train the base learners. Kundu and Ari (2019) improved the work of Bhatnagar et al. (2016) by scaling the output scores of the base SVM learners before applying a combination rule. Kundu and Ari (2019) observed that the output scores of base learners were at different

signal levels and applied a min-max normalization to scale the output scores to a range of zero and one. The normalization was performed to minimize the effect of the outputs of the base SVM experts dominating one another in the ensemble. However, the non-random data sampling is not an efficient approach to guarantee diversity among the base learners of an SVM ensemble.

Cross validation is another approach used in the studies to ensure that the base learners generate diverse predictions. This is shown in the work of Sun et al. (2013), Arefi and Chowdhury (2017), and Pang et al. (2017). Sun et al. (2013) reported that the proposed SVM ensemble does not perform well on imbalanced datasets. A number of studies have implemented the bagging approach to ensure diversity in SVM ensembles (Ma et al., 2011; Eeti and Buddhiraju, 2018; Bhavan et al., 2019). On the other hand, Liu and Huang (2019) considered training the base learners in a SVM ensemble using the Adaboost algorithm (Freund and Schapire, 1996). Liu and Huang (2019) employed the particle swarm optimization algorithm (Kennedy and Eberhart, 1995; Engelbrecht, 2007) to tune the control parameters of the base learners for efficient generalization performance.

Furthermore, bagging and boosting approaches have been combined in the studies of Lu and Wang (2011) and Nakayama et al. (2009) to guarantee diversity in an SVM ensemble. While both studies are implemented in the same domain, Lu and Wang (2011) extended the work of Nakayama et al. (2009) by employing a variant of SVR (v-SVR) technique proposed by Scholkopf et al. (2000) to combine the predictions of the base learners in the ensemble.

Abdullah et al. (2009) and Li et al. (2017a) demonstrated the benefit of the RFSM to ensure the base learners in a SVM ensemble behave differently. While both studies considered tuning the base learners using a grid search method, Abdullah et al. (2009) obtained the final ensemble prediction using a product rule (Tax et al., 1997) and Li et al. (2017a) trained a SVM algorithm as a meta-learner.

The k-means algorithm (Tou and Gonzalex, 1974) and fuzzy c-means clustering algorithm (Bezdek et al., 1984) were implemented in the works of Lee and Lee (2014), Ahmed et al. (2010), and Liu et al. (2020), where base learners are trained on different clusters consisting of random data and feature subsets. However, the implementations in these studies are influenced by the optimal value of k and c in the clustering algorithms to induce base

experts to produce diverse predictions for efficient generalization performance.

Additionally, Table 4.1 further compares the implementations of other components in the SVM ensemble, including fusion methods, tuning approaches and ensemble size). The studies employed different fusion approaches to combine the predictions of the base learners in the ensembles. The majority voting and simple averaging approaches were the dominant fusion approaches used in the past studies. Also, different tuning approaches were implemented in the studies, but the manual approach is the main tuning approach employed to tune the control parameters of the base learners to ensure the ensembles achieved optimal predictive performance compared to the component members.

4.5 Neural Network Ensembles

The application of NN ensembles (NNEs) to different real-world problems has been reported in several studies. The successful implementation of NNEs in these studies has been attributed to the ability of a NN algorithm to model complex non-linear relationships in large data. Thus, the comparison of the different implementations of NNEs with respect to the components of NNEs is provided in Table 4.2.

From Table 4.2, a random perturbation of the sample and feature space of a training dataset to obtain different subsets was implemented by Zhang et al. (2018), Zaamout and Zhang (2012), Kaur et al. (2018), and Wang et al. (2018), while Khan et al. (2018) utilized the cross-validation approach due to small size of the training dataset in the study to ensure the base learners behaved differently. Zhang et al. (2018), Zaamout and Zhang (2012), Kaur et al. (2018), and Wang et al. (2018) considered backpropagation neural networks (BPNNs) as the base learners, while Khan et al. (2018) proposed evolutionary wavelet neural networks (EWNNs) (Khan et al., 2017) as the component members. In these implementations, Zhang et al. (2018) and Khan et al. (2018) tuned the control parameters of the base learners using ant lion optimization (Mirjalili, 2015) and genetic algorithm (Holland, 1975), while other studies employed a manual approach.

Bagging is another approach implemented in the studies to introduce diversity in a NNE. This is shown in the work of Peng and Zhu (2009), Li et al. (2010), Yan et al. (2016), Li et al. (2017b), Almeida et al. (2019), and Nguyen et al. (2019a). A notable difference in

these studies is the introduction of BPNN with random weights (BPNN-RW) (Pao and Takefuji, 1992) as base learners in the work of Almeida et al. (2019).

Table 4.2: Comparison of the implementations of Neural Network Ensembles

Study	Diversity Approach	Fusion Method	Tuning Approach	Ensemble size	Base Learners
Zhang et al. (2018)	Random data selection	Weighted averaging	Ant Lion optimization	4	BPNN
Zaamout and Zhang (2012)	Random feature subsets	BPNN metaclassifier	Manual	4	BPNN
Kaur et al. (2018)	Random feature subsets	Simple averaging	Manual	10	BPNN
Wang et al. (2018)	Random data, feature selection	Simple averaging, Weighted averaging	Manual	30	BPNN
Khan et al. (2018)	Cross validation	Majority voting	Genetic algorithm	14-17	EWNN
Peng and Zhu (2009)	Bagging	Simple averaging	Manual	Unspecified	BPNN
Li et al. (2010)	Bagging	Simple averaging, Weighted averaging	Bayesian regularization	6	BPNN
Yan et al. (2016)	Bagging	Majority voting	Manual	10	BPNN
Li et al. (2017b)	Bagging	Majority voting	Manual	5	BPNN
Almeida et al. (2019)	Bagging	Simple averaging	Full factorial design	10	BPNN-RW
Nguyen et al. (2019a)	Bagging	Simple averaging	Manual	200,400,500, 600,1000	FFNN-LM
Hui et al. (2015)	Adaboost	Weighted averaging	Manual	Unspecified	BPNN
Peerlinck et al. (2019)	Adaboost	Weighted averaging	Grid search	5,10,20	FFNN
Osman and Aljahdali (2020)	Adaboost	Weighted majority voting	Manual	Unspecified	RBFNN
Li et al. (2011b)	Adaboost, *k*-means clustering	Majority voting	Manual	5,15,25,50	BPNN
Chen and Wang (2016)	*k*-means	Weighted averaging	Manual	Unspecified	BPNN
Staroverov and Gnatyuk (2016)	Stacking	ElmanBPNN metaclassifier	Manual	50	different NN architectures
Zhao et al. (2015)	Hyperparameter Optimization	Weighted averaging	Multiple PSO techniques	6	FFNN
Rohman and Kurniawan (2017)	Hyperparameter Optimization	Majority voting	Manual	3	BPNN

Also, Li et al. (2010) tuned the control parameters of the base learners using the Bayesian regularization approach, while Almeida et al. (2019) used a full factorial design of experiment (FFDOE) strategy (Montgomery, 2012) to tune the base learners. However, as noted by Almeida et al. (2019), the assumptions made by the FFDOE strategy that

populations are normally distributed, populations have equal variances, and samples are randomly and independently drawn are not always true, which may influence the ensemble performance. On the other hand, Nguyen et al. (2019a) adopted the Levenberg Marquardt (LM) algorithm (Levenberg, 1944; Marquardt, 1963) to train the component BPNNs, while other studies utilized BPNNs.

Furthermore, a boosting approach was implemented by Hui et al. (2015), Peerlinck et al. (2019), and Osman and Aljahdali (2020) to ensure that the base learners generated diverse predictions that were combined using weighted averaging and weighted majority voting, respectively. Hui et al. (2015) developed four NNEs and employed four weight optimization algorithms, including gradient descent, gradient descent with momentum with adaptive learning rate backpropagation, conjugate gradient backpropagation with Fletcher-Reeves updates, and the Broyden–Fletcher–Goldfarb–Shanno algorithm to train the base learners in each NNE. On the other hand, Peerlinck et al. (2019) developed three NNEs using two variants of Adaboost algorithm proposed by Solomatine and Shrestha (2004) and Bertoni et al. (1997), while the third NNE was the combination of the two variants. Osman and Aljahdali (2020) utilized radial-basis function NNs (RBFNNs) (Broomhead and Lowe, 1988) as the base learners and pointed out that an efficient optimization technique is required to improve the performance of the NNE developed in their work.

The *k*-means clustering algorithm has also been used in the studies of Li et al. (2011b) and Chen and Wang (2016) to split the training data into clusters consisting of random data. The component members are then trained on each cluster to generate diverse predictions. However, the authors suggested that the performance of the proposed NNEs may be limited by an inefficient setting of the number of clusters in the algorithm.

Staroverov and Gnatyuk (2016) employed a stacking based approach that is focused on the base NNs consisting of different architectures to develop a NNE. Staroverov and Gnatyuk (2016) developed the NNE by considering different optimization algorithms in a three-level network architecture to forecast energy consumption. The first level consists of four NNs, the second level consists of two NNs, and the third level is made up of one NN used as the meta-learner. In the first level, the outputs of two NNs (consisting of Elman backpropagation and feedforward backpropagation algorithms) and the other two NNs

(consisting of Elman backpropagation and cascadeforward backpropagation algorithms) served as input to train the two NNs (both consisting of an Elman backpropagation network) in the second level. Then the outputs of the two NNs in the second level were fed as input to the last NN (trained with an Elman backpropagation algorithm) in the third level to generate the final ensemble prediction. Staroverov and Gnatyuk (2016) showed that the NNE generated low forecast errors for energy consumption schedules than a single NN.

Also, Zhao et al. (2015) and Rohman and Kurniawan (2017) directly tuned the control parameters of the base learners consisting of FFNNs and BPNNs to guarantee diversity in the developed NNEs. While Zhao et al. (2015) capitalized on the benefits of multiple particle swarm optimization (PSO) techniques to obtain optimal control parameter values, Rohman and Kurniawan (2017) employed a manual approach which is subjective to trial and error.

4.6 Random Forest

A RF algorithm is an ensemble of DTs developed through the combination of bagging and RFSM approaches. In a RF algorithm, bootstrapped subsets are first generated from a training dataset to train the base trees. Then in order to select the most relevant feature to determine an optimal split in a decision node, the base trees in a RF consider only a random subset of possible features when creating each decision split. In contrast, a single DT considers each feature from the set of features individually to determine the most relevant features. This is a major difference between DTs and RFs. Thus, bagging and RFSM approaches introduce different instances of randomness that increase the diversity in the training dataset and reduces the error correlation among base trees in the construction of a RF (Breiman, 2001; Bernard et al., 2009).

Based on the problem type, determination of the final RF prediction will vary. For a classification problem, the final prediction is obtained by taking a majority vote of the class labels predicted by the base tree models. For a regression problem, the average of the outputs of all base tree models is computed as the final prediction (Xu et al., 2009).

Several studies have shown the successful application of RF to different classification and

regression problems due to the ability of a RF algorithm to handle large non-linear data efficiently, perform well on large dimensional and imbalanced data, and to produce faster tree induction due to the consideration of feature subsets during tree construction (Paul et al., 2018). A comparison of the implementations of the different components in a RF is provided in Table 4.3.

Table 4.3: Comparison of the implementations of Random Forest

Study	Diversity Approach	Fusion Method	Tuning Approach	Ensemble size
Bernard et al. (2009)	Random data subsets	Simple averaging	Manual	300
Nah and Lee (2016)	Random data subsets	Majority voting	Unspecified	Unspecified
Cheng et al. (2012)	Random feature subsets	Simple averaging	Manual	Unspecified
Sun et al. (2010)	Random data and feature election	Majority voting	Unspecified	Unspecified
Vincenzi et al. (2011)	Bagging and RFSM	Simple averaging	Manual	700
Dong et al. (2013)	Bagging and RFSM	Majority voting	Manual	350
Zhang et al. (2013)	Bagging and RFSM	Majority voting	Manual	200
Fawagreh et al. (2014)	Bagging and RFSM	Weighted majority voting	Manual	500
El-Habib Daho et al. (2014)	Bagging and RFSM	Weighted majority voting	Manual	100
Feng et al. (2015)	Bagging and RFSM	Weighted majority voting	Manual	10
Xue and Li (2015)	Bagging and RFSM	Majority voting	Manual	100-400
Pachange et al. (2016)	Bagging and RFSM	Majority voting	Manual	2-3
Anbarasi and Janani (2017)	Bagging and RFSM	Majority voting	Unspecified	Unspecified
Cano et al. (2017)	Bagging and RFSM	Majority voting	Manual	300
Lin et al. (2017)	Bagging and RFSM	Majority voting	Manual	5,10,25,50,100
Man et al. (2018)	Bagging and RFSM	Majority voting	Unspecified	Unspecified
Liu and Wu (2018)	Bagging and RFSM	Majority voting	Manual	100
Tang et al. (2019)	Bagging and RFSM	Simple averaging	Manual	25
Yao et al. (2019)	Bagging and RFSM	Simple averaging	Manual	500
Xing et al. (2019)	Bagging and RFSM	Simple averaging	Manual	51,66,68
Victoriano et al. (2020)	Bagging and RFSM	Majority voting	Unspecified	Unspecified

The studies of Bernard et al. (2009) and Nah and Lee (2016) considered implementation of diversity through random splitting of data into subsets, while Cheng et al. (2012) opted to

randomly split the features in a training dataset into different feature subsets. Sun et al. (2010) trained the base trees on random data subsets, but utilized random selection of a single feature during node splitting.

The classical bagging and RFSM approaches in a RF were implemented by the majority of the studies as shown in Table 4.3. Also, a number of studies provided a significant contribution to the implementation of a RF. Man et al. (2018) combined ID3 and CART algorithms during tree induction to achieve better prediction performance.

El-Habib Daho et al. (2014) experimented with the replacement of the Gini index with twoing and deviance metrics to split decision nodes during tree construction. To improve the prediction performance of the RF model developed by Yao et al. (2019), the authors first capitalized on the feature importance property of a RF algorithm to rank features in a training dataset in descending order. Then the top 20 features formed a subset, followed by the top 40, top 60, top 80, and top 100 features, which were used to train the RF algorithm.

Zhang et al. (2013) proposed an instance-based RF with rotated feature space to improve the diversity of component trees in a RF. The authors achieved diversity by partitioning the feature space into several subspaces, and each subspace was rotated using a rotation matrix algorithm and principal component analysis (PCA). Then the rotated subspaces were concatenated with the original feature in the training dataset to train the base trees. Zhang et al. (2013) coined an instance-based approach to select the tree models that provided accurate predictions, while models with wrong predictions were deleted.

Fawagreh et al. (2014) enhanced the diversity of a RF using an absolute predictive power (APP) approach proposed by Cuzzocrea et al. (2013) to assign a weight to each training subset based on the predictive power of the base trees. The APP approach was also employed in the prediction stage of the base learners to implement a weighted voting technique to combine the diverse predictions of the base learners.

For the fusion approaches, Fawagreh et al. (2014), El-Habib Daho et al. (2014), and Feng et al. (2015) employed the weighted majority voting technique to combine the predictions of the base trees, while other studies utilized the majority voting and simple averaging techniques.

Furthermore, it can be observed that a number of studies did not provide information or considered tuning the control parameters of base trees. This is shown in the works of Nah and Lee (2016), Sun et al. (2010), Anbarasi and Janani (2017), Man et al. (2018), and Victoriano et al. (2020). Other studies manually configured the control parameters of the base trees such as the number of trees, and the number of features for each base tree.

4.7 Nearest Neighbour Ensembles

Research has shown the successful application of *k*NN ensembles to different problems. Although, the stable nature of a *k*NN algorithm has also been reported to limit the wide applicability of the algorithm to develop *k*NN ensembles that will provide efficient generalization performance compared to the base learners in the ensemble (Breiman, 1996a; El-Hindi et al., 2018). Breiman (1996a) pointed out that stable algorithms, such as *k*NN and naïve Bayes, do not often yield improved predictive performance when the training dataset is randomly perturbed to develop ensembles. However, while the suggestion made by Breiman (1996a) may be considered to be problem-dependent, several studies have since applied *k*NN ensembles to different problems due to the simple implementation and interpretation of a *k*NN algorithm. Thus, the review of nearest neighbour ensembles (NNGBEs) is provided in Table 4.4 with an emphasis on the implementation of the components of the ensemble.

For diversity approaches, the RFSM has been implemented to ensure that the base learners in a NNGBE behave differently (Okun and Priisalu, 2009; Hamzeloo et al., 2012; Yu et al., 2016). Due to the small size of the dataset used to train the *k*NN ensemble developed by Okun and Priisalu (2009), a bolstered re-substitution error (BRE) technique (Braga-Neto and Dougherty, 2004) was adopted to generate artificial data for better ensemble performance. However, Okun and Priisalu (2009) suggested that the BRE technique may subject the ensemble to overfitting. On the other hand, Hamzeloo et al. (2012) introduced two probability functions to adaptively split the training features into different subsets. However, the authors suggested that the computation of a negative constant value by the probability functions during feature selection could impair the predictive performance of the ensemble.

Zhang et al. (2022) trained the base learners in a NNGBE using a bagging approach, while

Wang et al. (2019b) randomly split the data into subsets to introduce diversity among the base learners. However, Wang et al. (2019b) opined that the performance of the developed ensemble would negatively be affected by the manually configured threshold and the size of the sliding window introduced in the study. A related work to Wang et al. (2019b) was conducted by Bandaragoda et al. (2015) where the base learners were trained on random data and feature subsets.

Table 4.4: Comparison of the implementations of Nearest Neighbour Ensembles

Study	Diversity Approach	Fusion Method	Tuning Approach	Ensemble size
Okun and Priisalu (2009)	RFSM	Majority voting	Manual	3,5,7,9
Hamzeloo et al. (2012)	RFSM	Weighted majority voting	Manual	21
Yu et al. (2016)	RFSM	Majority voting	Manual	20
Zhang et al. (2022)	Bagging	Simple averaging	Manual	1-12
Wang et al. (2019b)	Random data subsets	Weighted majority voting	Manual	50-200
Bandaragoda et al. (2015)	Random data and feature subsets	Simple averaging	Manual	25
Tahir and Smith (2010)	Different feature subsets, different distance metrics	Majority voting	Manual	Unspecified
Fuchs et al. (2015)	Cross validation	Weighted averaging	Manual	Unspecified
Abed et al. (2018)	Cross validation	Simple averaging	Manual	3
Khan et al. (2020)	Cross validation and RFSM	Majority voting	Manual	Unspecified
Sun et al. (2020)	LOOCV and Different distance functions	Simple averaging	Grid search	2-8
Haixiang et al. (2016)	Adaboost and RFSM	Majority voting	Manual	Unspecified
Iswarya and Radha (2015)	Single pass clustering	Majority voting	Manual	Unspecified

Tahir and Smith (2010) and Sun et al. (2020) considered the use of different distance metrics for the base learners to develop a NNGBE. To train the base learners in the ensemble, Tahir and Smith (2010) used different feature subsets that were selected by a Tabu search technique (Glover, 1989). In contrast, Sun et al. (2020) trained the base learners using a leave-one-out cross-validation approach (LOOCV). The cross-validation approach was also considered in the studies of Fuchs et al. (2015), Abed et al. (2018), and Khan et al. (2018).

Furthermore, the comparison of the implementation of the other components is provided in Table 4.4. Majority voting and simple averaging are the mostly used fusion approaches

in the reported studies. Hamzeloo et al. (2012) and Wang et al. (2019b) employed a weighted majority voting approach to combine the decisions of the base learners, while Fuchs et al. (2015) utilized a weighted average to obtain the final prediction of the ensemble developed in the study. Also, while Sun et al. (2020) tuned the control parameter values of the base learners using a grid search, other studies implemented the manual approach, i.e. trial and error.

4.8 Naïve Bayes Ensembles

The assumption of feature independence made by a NB algorithm has been reported to limit the predictive performance of a NB ensemble (NBE). Also, the stable nature of a NB has further influenced the wide applicability of a NB algorithm to develop efficient NBEs, compared to ensembles developed using other algorithms. Though stability makes the NB algorithm robust to noise, it also makes it challenging to construct a NB ensemble using bagging and boosting approaches. This is actually the case with any stable classifier. The reason behind it is the fact that slightly different data samples do not cause a base learner to generate sufficiently diverse classifiers (Breiman, 1996a; El-Hindi et al., 2018). However, despite the assumption of feature independence and the stable nature of a NB algorithm, NBEs have shown remarkable performance in a number of studies (El-Hindi et al., 2018; Li and Hao, 2009), where different approaches were proposed to construct a NBE with better generalization performance. Thus, the review of NBEs is provided in Table 4.5 to investigate the implementation of the components of a NBE.

For the implementation of diversity in a NBE, Klement et al. (2012) and Lutu (2015) trained the base learners on random subsets of a training dataset, resulting in diverse predictions combined using a majority voting approach. While the ensembles in the studies by Klement et al. (2012) and Lutu (2015) outperformed single base learners, Klement et al. (2012) reported that the developed ensemble recorded a high false negative error, illustrating inconsistency in the prediction performance of the ensemble.

The bagging approach has also been utilized to guarantee diversity in NBE, as shown in the studies of Li and Hao (2009) and El-Hindi et al. (2018). A notable difference between the studies is the introduction of a random oracle selection technique (Kuncheva and Rodriguez, 2007) by Li and Hao (2009) to complement the bagging approach for

improved diversity in a NBE. Li and Hao (2009) used the oracle selection technique to create hyperplanes of data subsets on which base NB learners were trained. However, while Li and Hao (2009) did not apply any tuning approach to improve the generalization performance of the NBE, Klement et al. (2012) proposed a fine-tune NB algorithm (FTNB) (El-Hindi, 2014) to optimize the parameter computation of the likelihood and class prior probabilities. The FTNB was introduced to solve the stability problem of the base NB learners.

Table 4.5: Comparison of the implementations of Naïve Bayes Ensembles

Study	Diversity Approach	Fusion Method	Tuning Approach	Ensemble size
Klement et al. (2012)	Random data subsets	Majority voting	Unspecified	10
Lutu (2015)	Random data subsets	Majority voting	Manual	3
Li and Hao (2009)	Bagging, Oracle Selection	Majority voting	Unspecified	10-50
El-Hindi et al. (2018)	Bagging	Majority voting	FTNB	10
Shi and Lv (2010)	Adaboost	Weighted majority voting	Parameter expectation	Unspecified
Nikolić et al. (2014)	Adaboost	Majority voting	Unspecified	3
Srisuan and Hanskunatai (2014)	Different feature subsets	Majority voting	Unspecified	Unspecified
Maia et al. (2021)	RBS	Majority voting	Unspecified	500
Bang and Wu (2016)	*k*-means clustering	Majority voting	Unspecified	7
Sumathi and Poorna (2017)	Fuzzy *k*-medoids clustering	Majority voting, Weighted averaging	Unspecified	Unspecified
Kilimci et al. (2016)	Bagging, RFSM, Adaboost, RF	Majority voting, Weighted Majority voting	Unspecified	100
Alkubaisi (2019)	MNB and MVNB	Majority voting	Manual	2

The boosting approach was also used to implement diversity in a NBE. Shi and Lv (2010) and Nikolić et al. (2014) trained the base learners in a NBE using the Adaboost algorithm to generate diverse predictions combined using weighted and simple majority voting approaches, respectively. To improve the generalizability of a NBE, Shi and Lv (2010) proposed an approach referred to as *"parameter expectation"* to sum the weighting parameters of the base learners when training the base learners based on Adaboost architecture. Parameter expectation was included in the computation of the conditional probability.

The construction of a NBE based on training the base learners on different feature subsets was implemented in the studies of Srisuan and Hanskunatai (2014) and Maia et al. (2021). Srisuan and Hanskunatai (2014) used two feature selection techniques, i.e. ReliefF (Kononenko, 1994) and Chi-square to generate different feature subsets on which the base learners were trained. In contrast, Maia et al. (2021) introduced a probability-based feature bias to the classical RFSM approach, termed as *"RBS"* to measure and select features with low noise. The selected features were split into subsets to train the base learners. Srisuan and Hanskunatai (2014) and Maia et al. (2021) obtained the final ensemble prediction using a majority approach.

Bang and Wu (2016) and Sumathi and Poorna (2017) explored the potential of k-means and Fuzzy k-medoids (Krishnapuram et al., 1999) clustering algorithms to cluster the training dataset into different subsets used to train the base learners in a NBE. Kilimci et al. (2016) developed four NBEs using bagging, Adaboost, RFSM, and RF approaches. Kilimci et al. (2016) adapted the base learners of the RF algorithm to include changing the classical base tree models to NB classifiers. Additionally, Alkubaisi (2019) capitalized on the advantage of two variants of a NB algorithm, i.e. multinomial NB (MNB) (McCallum and Nigam, 1998) and multivariate Bernouli NB (MVNB) (Kalt and Croft, 1996), to achieve diversity in a NBE. However, both base learners belong to the same NB algorithm.

4.9 Limitations of the Different Implementations of Homogeneous Ensembles

Generally, by examining the reported studies for each ensemble type from Sections 4.4 to 4.8, it is noteworthy to conclude that the aforementioned approaches did not efficiently explore the inductive biases of the base learners in each ensemble type. Also, while tuning approaches were implemented to ensure that the ensembles achieved better generalization performance compared to a single learner, the base learners within each ensemble were mostly configured only with fixed parameter values which do not consider the inductive biases of the base learner that would result in different base experts. It is important that the base learners consist of different control parameter values to induce different experts within each ensemble.

Further, despite tuning the control parameter values of the base learners, there were studies where the ensembles showed inconsistent generalization performance. Additionally, the reported implementations are limited with respect to the number of classification and regression problems for the evaluation of the ensembles. Most of the datasets used to evaluate the ensembles consist of small sizes, which do not adequately reflect efficient generalization performance when trained with the ensembles. Also, the review showed that the ensemble size is problem-dependent.

4.10 Chapter Summary

The chapter provided an extensive review of the construction of homogeneous ensembles considering the implementation of the components of an ensemble, i.e. approaches used to address diversity in an ensemble, fusion approaches to combine the predictions of base experts, ensemble size, and strategies used to control overfitting for better generalization performance. It was observed that authors agreed that homogeneous ensembles significantly generate better predictive performance than a single ML algorithm due to the combination of diverse and multiple experts.

However, despite achieving good results through the application of homogeneous ensembles in different domains, limitations of the ensembles to effectively address diversity, inductive biases, and the bias-variance tradeoff are still observed.

Chapter 5

Heterogeneous Ensembles

5.1 Background

A heterogeneous ensemble (HTE) is developed using different ML algorithms. The rationale to develop a HTE is to obtain improved generalization performance, better than the individual experts of the ensemble. For a HTE, the different advantages and characteristics of the base ML algorithms serve as a source of diversity in the ensemble (Dudek, 2016). Due to different inductive biases, diversity is obtained when individual ML algorithms generate different predictions when trained on the same dataset. This chapter provides a review of HTEs. The review focuses on the implementation of the different components of a HTE, which include the approaches used to address diversity within the ensemble, fusion approaches to combine the predictions of base experts, tuning strategies used to balance the bias-variance tradeoff, ensemble size, and the base algorithms selected. Section 5.2 provides the review strategy used in this chapter, while Section 5.3 discusses the past studies on the construction of HTEs, i.e HTEs. The limitations of the different implementations of HTEs are discussed in Section 5.4, and Section 5.5 presents the summary of this chapter.

5.2 Review Strategy

The review conducted in this chapter aims to identify and analyze different implementations of HTEs. The strategy to perform a detailed systematic review of the construction of different HTEs follow the same strategy used for the review of homogeneous ensembles. However, there are a number of differences in the strategy. The first difference is the consideration of relevant terms in the search strings to identify articles where HTEs were implemented. The relevant terms include "Heterogeneous Ensembles", "Heterogeneous Mixtures of Experts", "Mixtures of Heterogeneous Experts", "Combination of Machine Learning Algorithms", and "Mixtures of Machine Learning Algorithms".

Another difference is the selection of articles in bibliographical databases where two or more different ML algorithms were identified as the base learners for the construction of HTEs. A total of 25 relevant articles were selected for the review of HTEs.

For the inclusion criteria, the review considers articles that have used different ML algorithms such as SVM, NN, DT, *k*NN, NB, and others as base learners for the construction of a HTE.

For the exclusion criteria, articles where an ensemble was developed using multiple instances of the same ML algorithm, were excluded, because the focus of the review is mainly on HTEs constructed using different ML algorithms.

5.3 Review of Heterogeneous Ensembles

Several experimental studies have reported the superiority of HTEs over homogeneous ensembles and a single learner, based on improved predictive performance generated for a given task (Wichard et al., 2002). The superior performance of HTEs has been attributed to low correlation error terms of the different base models induced by the different ML algorithms within the ensemble (Dudek, 2016). While these studies have investigated the development and applications of HTEs across various domains, not much attention has been given to effectively discuss the inductive biases of the base algorithms, to address diversity and the bias-variance dilemma. Moreover, with respect

to diversity, very little work has been done in the selection of the best algorithmic-specific data preprocessing techniques suitable for individual base experts that may better result in optimal performance of a HTE. Thus, the implementation of the different components of a HTE is provided in Table 5.1, which is followed by the discussion of the studies.

As provided in Table 5.1, Balogun et al. (2017) trained NB, RBFNN, and repeated incremental pruning to produce error reduction (RIPPER) (Cohen, 1995) on random data subsets to implement diversity in the proposed HTE. Luong et al. (2020) employed random projections (John, 1995) to generate different training subsets on which the base learners were trained. While both studies considered majority voting as one of the fusion approaches, Balogun et al. (2017) also used a meta-classifier, multi-scheme, and minimum probability (Min Prob) approaches, while Luong et al. (2020) introduced a sum rule (Kittler et al., 1998) to combine the predictions of base learners. However, in the work of Balogun et al. (2017), different unstable results were generated by the HTEs and individual base classifiers, in which most single learners outperformed the HTEs. In contrast, Luong et al. (2020) suggested using better ensemble approaches and unstable base learners to construct a HTE to realize improved prediction performance.

Tuarob et al. (2014) and Sagayaraj and Santhoshkumar (2020) achieved diversity by training the base learners on different feature subsets. Tuarob et al. (2014) selected RF, SVM, RIPPER, and MVNB as the base algorithms, while Sagayaraj and Santhoshkumar (2020) used a DT and logistic regression (LogR) (Cox, 1958). Feng et al. (2021) opted to randomly split features in the training datasets into subsets to train the base learners consisting of DT, RF, SVM, extreme gradient boosting (XGboost) (Chen and Guestrin, 2016), and light gradient boosting machine (LGBM) (Ke et al., 2017). Aside from the implementation of majority voting and weighted averaging approaches to combine the predictions of the base learners, Tuarob et al. (2014) also used multi-staging (MS) and reversed multi-staging (RevMS) approaches (Ted, 2005). On the other hand, Feng et al. (2021) proposed a weighted area under curve-based integration mechanism (WAUCE) as the fusion approach. A critical analysis of the work of Tuarob et al. (2014) showed that there is no significant improvement in the performance of the HTE over the single base classifiers in terms of accuracy, and Sagayaraj and Santhoshkumar (2020) did not specify the approach used to tune the base learners. Additionally, Feng et al. (2021) reported that the proposed HTE was developed using a relatively small dataset and may perform

Table 5.1: Comparison of the implementations of Heterogeneous Ensembles

Study	Diversity Approach	Fusion Method	Tuning Approach	Ensemble size	Base Learners
Vinay et al. (2011)	Adaboost, Stacking	Majority voting, DECORATE	Manual	7	DT, RF, NB, *k*NN, SVM, RBFNN, BPNN
Tsai et al. (2011)	Bagging	Majority voting	Manual	3	BPNN, DT, LogR
Marqués et al. (2012)	Bagging, Adaboost, DECORATE, RFSM, RFST	Majority voting	Manual	10	BPNN, DT, SVM, 1-NN, NB, LogR
Son et al. (2013)	Cross validation	Majority voting	Manual	6	SVM, BPNN, DT, NB, LogR, *k*NN
Elish et al. (2013)	Bagging, Boosting	Linear, Non-Linear combiners	Genetic algorithm	3	BPNN, SVR, ANFIS
Tuarob et al. (2014)	Different feature subsets	Majority voting, Weighted averaging, MS, RevMS	Manual	5	RF, SVM, RIPPER, MVNB, MNB
Chali et al. (2014)	Cross validation	Weighted voting	Manual	4	SVM, HMM, CRF, Max Ent
Ala'Raj and Abbod (2015)	Bagging	Majority voting	Manual	3	LogR, BPNN, SVM
Mendes-Moreira et al. (2015)	Cross validation	DWS	Manual	5,10,15,20,25	RF, SVM PPR
Chaudhary et al. (2016)	Bagging	Majority voting	Unspecified	2	LogR, NB
Balogun et al. (2017)	Random data subsets	Majority voting, Meta-classifier, multi-scheme, Min Prob	Manual	3	NB, RBFNN, RIPPER
Palaninathan et al. (2017)	CEEMDAN	Median aggregate	Manual	3	SVR, BPNN, RF
Kilimci et al. (2017)	Cross validation	Majority voting, Meta-learner	Unspecified	4	MVNB, MNB, SVM, RF
Li et al. (2018)	Cross validation	Weighted averaging	TPE	3	XGBoost, DNN, LogR
Nguyen et al. (2018)	Cross validation	Fuzzy meta-classifier	Manual	3	LDA, NB, *k*NN
Xu and Zhang (2019)	Unspecified	BPNN meta-classifier	Manual	2	1D-CNN, LSTM
Nguyen et al. (2019b)	Cross validation	Sum Rule, MLR	Manual	3	*k*NN, LogR, NB
Nguyen et al. (2020)	Cross validation	Sum Rule	Manual	5	*k*NN, LogR, MVNB, DT, RF
Luong et al. (2020)	Random projections	Sum rule, Majority voting	Manual	3	LDA, NB, *k*NN
Zhao et al. (2020)	Cross validation	DNN meta-classifier	Grid search	6	SVM, DT, NB, *k*NN, RF, LogR
Sagayaraj and Santhoshkumar (2020)	Different feature subsets	Majority voting	Unspecified	2	DT, LogR
Zain et al. (2020)	Stacking	Majority voting, Decision template, Dempster-shafer	Manual	3	ELM, SVM, RF
Tewari and Dwivedi (2020)	Stacking	Majority voting, LGBM meta-classifier	Grid search	4	BPNN, SVM, RF, LGBM
Alshdaifat et al. (2021)	Cross validation	Majority voting, Average probability	Unspecified	6	NB, DT, RB, SVM, *k*NN, BPNN
Feng et al. (2021)	Different feature subsets	WAUCE	Unspecified	5	DT, RF, SVM, XGBoost, LGBM

poorly on a larger dataset consisting of skewed class distributions and noise characteristics.

Furthermore, it can be observed that the cross-validation approach was implemented in a number of studies to guarantee diversity in a HTE. The cross validation approach is implemented in the work of Son et al. (2013), Chali et al. (2014), Mendes-Moreira et al. (2015), Kilimci et al. (2017), Li et al. (2018), Nguyen et al. (2018), Nguyen et al. (2019b), Nguyen et al. (2020), Zhao et al. (2020), Alshdaifat et al. (2021), and Tewari and Dwivedi (2020). While voting and averaging approaches were used in these studies, Mendes-Moreira et al. (2015) applied a dynamic weighting with selection (DWS) method (Rooney et al., 2004) to combine the outcomes of the base learners consisting of RF, SVM, and projection pursuit regression (PPR) (Friedman and Stuetzle, 1981).

A notable difference in the tuning component of the studies is in the work of Li et al. (2018), where a variant of Bayesian optimization, called tree parzen estimator (TPE) (Bergstra et al., 2011) was implemented to tune the control parameters of the base learners for a better prediction performance. Li et al. (2018) used XGBoost, deep neural network (DNN) (Bengio, 2009) and LogR as base learners. Another notable difference is in the studies of Nguyen et al. (2018, 2019b, 2020) which employed a fuzzy rules selection algorithm (Ishibuchi et al., 1999), BPNN, sum rule, and multi-response linear regression (MLR) (Ting and Witten, 1999) as meta-classifiers to obtain the final prediction of a HTE. Nguyen et al. (2018, 2019b, 2020) considered NB, *k*NN, LogR, MVNB, DT, RF and linear discriminant analysis (LDA) (Fisher, 1936) as base algorithms in the proposed HTEs. However, despite the implementation of a cross-validation approach and other components to develop HTEs that outperformed a single learner in these studies, the following limitations are still observed. In the work of Son et al. (2013), a weighted count of errors and correct results (WCEC) method (Aksela and Laaksonen, 2006) was adopted to measure the diversity of errors in the base classifiers, and the classifiers having the best WCEC value and validation accuracy were selected as ensemble members. However, Son et al. (2013) reported that WCEC may not effectively manage diversity when the base classifiers generate high diversity errors in the ensemble. Also, the HTE developed by Chali et al. (2014) achieved very low F1-scores illustrating poor prediction of positive classes. Mendes-Moreira et al. (2015) stated the HTE developed in the study recorded high computational complexity leading to inconsistent prediction performance in certain

cases, while the meta-learner for the proposed ensemble in Kilimci et al. (2017) could not be ascertained.

Furthermore, the HTE developed by Li et al. (2018) did not achieve much improvement over the base classifiers, while the use of stable base algorithms in Nguyen et al. (2018) and Nguyen et al. (2019b) does not guarantee efficient predictive performance, as pointed out by Breiman (1996a). Also, Zhao et al. (2020) suggested the potential of training base learners in the study on different dataset features to achieve better predictive performance. Alshdaifat et al. (2021) defined a threshold value to determine the best classifiers in an ensemble. However, the threshold value is subjective to the authors and may not effectively define the boundary to exclude poor performing classifiers.

Another diversity approach used to ensure the base learners in a HTE produce different predictions is bagging, which is considered in Tsai et al. (2006), Marqués et al. (2012), Elish et al. (2013), Ala'Raj and Abbod (2015), and Chaudhary et al. (2016). A boosting approach was also implemented in Marqués et al. (2012) and Elish et al. (2013). Marqués et al. (2012) considered diverse ensemble creation by oppositional relabelling of artificial training examples (DECORATE) (Melville and Mooney, 2005) and rotation forest (RFST) (Rodriguez et al., 2006) approaches to develop HTEs. All of these studies, except Elish et al. (2013), employed a majority voting to combine the predictions of the base learners. Elish et al. (2013) experimented with different fusion approaches using simple and weighted averaging as linear combiners, and a number of meta-learners, including BPNN, SVR, fuzzy c-means clustering (FCM), subtractive clustering (SC) (Chiu, 1994), adaptive neuro-fuzzy inference system (Jang, 1993) (ANFIS)-FCM, and ANFIS-SC. A critical analysis of the ensembles developed in Tsai et al. (2006) showed that there is little difference in performance between the developed heterogeneous and homogeneous ensembles based on the predictions for return on investment. Marqués et al. (2012) stated that the best individual classifier did not contribute significantly to the ensemble performance, which was attributed to the manual approach used to tune the base learners. Ala'Raj and Abbod (2015) reported that the proposed HTE generated high false negatives with respect to the high risk of classifying bad loan applicants as good loan applicants. Thus, efficient risk prediction by the HTE could not be achieved in certain cases.

Stacking is another approach used to guarantee diversity in a HTE, which is implemented

in the works of Zain et al. (2020) and Tewari and Dwivedi (2020). To obtain the final prediction of a stacked-based HTE, Zain et al. (2020) employed majority voting, decision template (Kuncheva et al., 2001) and Dempster-Shafer approaches. However, Zain et al. (2020) considered an extreme learning machine (ELM) (Huang et al., 2004) implemented with a single hidden layer as one of the base learners, which produced low computational complexity on the high dimensional Mallay dataset used in the study. Thus, the ELM may perform poorly on large datasets with respect to samples and features, which could degrade the predictive performance of the ensemble. In contrast, Tewari and Dwivedi (2020) used majority voting and a LGBM meta-classifier to combine the predictions of the base learners. Tewari and Dwivedi (2020) reported that the developed HTE outperformed other classifiers for small classes in the oil-field dataset used in the study. However, the authors stated that the HTE may perform poorly on a large dataset with a large number of classes. Moreover, all HTEs developed in the study were evaluated with only one dataset.

A different diversity approach was used in Palaninathan et al. (2017). The authors employed a complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) method (Torres et al., 2011) to split a training dataset into different data subsets. The base learners were trained on the data subsets to develop a HTE that forecasted four output horizons. In the study, two base models were selected at a time to forecast each horizon. However, Palaninathan et al. (2017) suggested that using more than two base models may improve the load demand forecast of the HTE.

Fernandez-Aleman et al. (2019) presented a systematic review of 107 published articles to investigate the methods used to construct ML ensembles for the diagnosis and prediction of potential diabetes disease. The authors reported that the first application of a ML ensemble for diabetes classification appeared in a paper published in 2003 and stated that 12 different single ML algorithms with each variant of the algorithms have been applied to construct both homogeneous and heterogeneous ensembles. Fernandez-Aleman et al. (2019) stated that NNs, SVMs, and DTs have been the most frequently used base classifiers.

Following this, a high percentage (85%) of homogeneous ensembles were reported to have been extensively applied to diabetes classification in contrast to a small percentage (15%) of applying HTEs for the same task. Different diversity measures were

reported in the study, where bagging, boosting, and RSM methods were the frequently used approaches to achieve diversity, mostly in homogeneous ensembles. Then, 17 combination rules have been used to obtain a final ensemble prediction, stating that the majority voting and weighted majority voting rules still served as the most widely used fusion approaches. Given the small percentage of HTE applications in diabetes classification literature, Fernandez-Aleman et al. (2019) concluded the review by stating that the continuous application of HTEs for different classification problems is still an interesting area to explore. It is noteworthy to mention that this research also extended the conclusion of Fernandez-Aleman et al. (2019) to the domain of regression problems.

5.4 Limitations of the Different Implementations of Heterogeneous Ensembles

Generally, apart from the limitations reported for each study, the limitations of the constructed HTEs in the studies are also identified, and a summary of the limitations is provided below:

- Inefficient implementation of the strategies used to address the bias-variance tradeoff with respect to underfitting and overfitting of the ensembles.
- While a number of studies did not tune the control parameters of the base learners for optimal prediction performance, most studies employed a manual approach which is also subject to trial and error.
- Elish et al. (2013) considered a genetic algorithm to tune the base learners, while a grid search method was implemented in the works of Zhao et al. (2020) and Tewari and Dwivedi (2020). However, a grid search method is adversely influenced by an expensive computational cost due to the exhaustive search of the optimal control parameter value for each base learner. Hence, the performance of base learners quickly drops when the control parameters and the range of values of the control parameters increase excessively (Wu et al., 2019; Cachada et al., 2017).
- Additionally, the tuning process performed across the studies was only to obtain the best single parameter value for each base algorithm. In contrast, this research focuses on considering different parameter values for multiple instances of the base

algorithm to induce diverse base experts in the HTEs proposed in the study.

- Furthermore, no efficient strategies are used to address the inductive biases of the base algorithm to develop efficient HTEs.
- More importantly, the reported studies did not consider different cases of exploring and analyzing the inductive biases of the base algorithms towards the construction of the HTEs.
- The inductive biases of the base algorithms are considered as additional behavioural diversity layers to the classical diversity approaches, i.e. bagging, boosting, stacking, and others for the construction of the HTEs proposed in this research. This aspect is not considered in these past studies.

5.5 Chapter Summary

The chapter provided a review of HTEs. Sections 5.1 and 5.2 provided a brief introduction to the chapter and the strategies followed in conducting the review. The review of the past studies of HTEs is discussed in Section 5.3. The review focused on the different components of a HTE, including approaches used to address diversity, fusion approaches, and the tuning approaches used to balance the bias-variance tradeoff for improved predictive performance. In addition, the ensemble size and ML algorithms selected to construct HTEs in the studies were discussed.

Chapter 6

Ensembling Diverse Heterogeneous and Homogeneous Experts

6.1 Introduction

This chapter presents the approaches used to develop the heterogeneous and homogeneous ensembles of this research. Section 6.2 presents the different types of ensembles developed in this research, while Section 6.3 discusses the sampling approach used to create bagged and feature subsets from training datasets. Section 6.4 describes the training process of the individual ML algorithms that make up an ensemble. The approaches used to determine the predictions of an ensemble are described in Section 6.5, and Section 6.6 provides a summary of the chapter.

6.2 Ensemble Model Development

This section describes the four types of ensembles developed in this research. The ensemble types were developed to explore the inductive biases of the selected ML algorithms, and to maximize the benefits inherent in the control parameter configurations of the base learners that would result in efficient base experts within the ensembles. The

four ensemble types are as follows:

- Ensembles where multiple instances of the same ML algorithm are used, where the instances consist of the same control parameter configuration. The ensembles are referred to as *"pure homogeneous ensembles"* in this research and are denoted as NBE, *k*NNE, DTE, SVME, and NNE. The RF algorithm is also used.
- Ensembles where multiple instances of the same ML algorithm are used, and the instances consist of different control parameter configurations that actually result in different experts. The ensembles in this category are denoted as NBhte, *k*NNhte, DThte, SVMhte, and NNhte.
- An ensemble where multiple instances of different ML algorithms are considered as members of the ensemble, and the instances of each ML algorithm of the ensemble have the same control parameter configuration. The ensemble is denoted as HTEsm.
- An ensemble where multiple instances of different ML algorithms are considered, where the instances of each ML algorithm have different control parameter configurations. The ensemble is denoted as HTEdf.

Generally, the HTEs and homogeneous ensembles were constructed with NB, *k*NN, DT, RF, SVM, and NN algorithms for classification problems. The regressors of these algorithms were considered in the case of regression problems, except for the exclusion of the NB algorithm, because the NB algorithm is basically developed for classification problems. Also, SVR was used for regression problems instead of SVM.

Thus, the different types of ensembles developed in this research resulted in the construction of 13 ensembles for classification problems and 11 ensembles for regression problems, in which NBE and NBhte were excluded. Each ensemble consists of 10 component members that define the size of the individual ensembles. The selected ensemble size has been shown to produce good generalization performance across different problems (Hansen and Salamon, 1990; Breiman, 1996a; El-Hindi et al., 2018).

6.3 Data Sampling

Due to the recursive nature and complex implementation of boosting approaches, which is outside the scope of creating ensembles for this research, the bagging approach is

used to create bagged subsets for classification and regression problems. As discussed in Section 3.3.1, the bagging approach randomly samples training data with replacement to obtain bootstrapped replicates. The first phase of the sampling process in the research is to randomly split the entire dataset into 70% training and 30% testing datasets. The second phase is to create bagged subsets from the training set, on which multiple instances of base algorithms are trained to induce experts that generate different predictions on the test dataset. It is essential to sample bagged subsets only from the training datasets. If not, the algorithms would have already seen the samples in the testing dataset during model evaluation.

Another justification for the selection of the bagging approach to develop ensembles in this research is discussed in Hansen and Salamon (1990) and Breiman (1996a). The bagging approach basically produces an overall reduced generalization error using an ensemble when compared to the generalization errors produced by the individual learners within the ensemble. This is possible because the base learners are trained in parallel, and each induced base expert makes independent generalization errors on different subsets of the input space. It is then expected that the combination of the independent errors of these experts, given the selection of an appropriate ensemble size, will result in minimization of the overall generalization error produced by the ensemble compared to that of the base experts.

Additionally, for the experiment on feature subsets, the RFSM ensemble approach was used to construct ensembles to further investigate the significance of diversity in the research. The feature subsets were obtained by sampling the input features in the training dataset. The performance of the ensembles were evaluated on these feature subsets across all classification and regression problems.

6.4 Training of Machine Learning Algorithms

The ML algorithms to be combined to develop the ensembles in this research contain different characteristics and training processes. Based on the goal of this research, the training behaviour, high-performance, and inductive biases of the selected ML algorithms discussed in Section 2.4, motivated the choice of the NB, *k*NN, NN, SVM, DT, and RF algorithms for the research. The selection of these algorithms is further

justified by the different intrinsic properties of the algorithms with respect to the mathematical foundation, model architecture, model structure, and model complexity of the algorithms. These intrinsic properties inform the differences in training and generalization performance produced by the algorithms when trained on the same dataset. The algorithms also provide specific diversity benefits to the development of the ensembles with respect to the combination of the intrinsic inductive biases of the algorithms. Additionally, the algorithms provide specific assumptions based on the characteristics and complexities of the selected datasets identified for this research. Therefore, this section discusses the training process for individual algorithms, the algorithm-specific parameters, and the default methods of the algorithms in the Python programming language.

6.4.1 Naïve Bayes

As discussed in Section 2.4.6, the NB classifier predicts the target label based on the calculations of conditional posterior probabilities. Laplace smoothing is used to avoid the problem of frequency-based zero probability. The problem occurs when the NB classifier encounters an input feature value in the test dataset that has not appeared in the training dataset and vice versa. Without Laplace smoothing, probabilities calculated by the classifier are equal to zero if there exists a frequency-based zero probability. Thus, the NB classifier will be unable to make a prediction.

The NB algorithm used in implementations assumes different data distributions such as Gaussian, Multinomial, Bernoulli and categorical for integer-valued and continuous-valued variables. These assumed distributions remove the need for the discretization of numeric variables. The NB classifier is accessed through the *naive_bayes* package of the *sklearn* library in Python.

6.4.2 *k*-Nearest Neighbour

The *k*NN algorithm makes predictions for a test data sample based on the closest samples in the training dataset. The number of surrounding samples to be considered depends on the value of k, denoting the number of nearest neighbours. The different values of k used by the *k*NN base learners influence the generation of diverse base experts within each

ensemble. The *k*NN algorithm is accessed in the *neigbors* package of the *sklearn* library in Python.

6.4.3 Decision Tree

The DT algorithm induces a tree-based model based on the features and samples in the training set. The CART induction algorithm is used to construct trees, as discussed in Section 2.4.4. CART is very similar to C4.5, but differs in that it supports numerical target variables for regression problems.

CART constructs binary trees using the feature and threshold that yield the largest information gain at each node. The default CART functions perform post-pruning, and use the Gini index and information gain for splitting. The *sklearn* library implements DT algorithms using an optimized version of the CART algorithm that is accessed via the *tree* package.

6.4.4 Support Vector Machines

By default, the SVM algorithm creates a separating hyperplane that allows the classification of samples in the training and test sets for binary class datasets. The one-vs-all approach to SVM was used for multi-class classification problems. The SVM method requires pre-determined parameters, including the kernel function, cost of misclassifying points, C, and kernel width, γ, all of which control the bias-variance tradeoff of the algorithm.

As discussed in Section 2.4.3, the kernel function selection is problem-dependent. The C parameter performs a regularization function for the penalty cost of misclassifying samples, while γ defines the influence of a single sample during training. A low γ value means little influence, and a high γ value indicates a significant influence. The SVM algorithm is accessed from the *svm* package of the *sklearn* library in Python.

6.4.5 Neural Networks

NNs map a set of inputs to produce output values through an interactive process of adjustments applied to the synaptic input weights and bias levels of the network. The weight adjustment of a NN ensures that the free parameters of the network are adapted

through forward and backward learning processes. As discussed in Section 2.4.2, the multi-layer perceptron (MLP) is selected to develop ensembles and the standard gradient-descent backpropagation algorithm is used for weight optimization.

The MLP python method requires setting a number of parameters including the number of hidden layers, number of hidden nodes, weight optimizer (solver), the maximum number of iterations, learning rate, tolerance value, momentum, and activation function. The maximum number of iterations indicates the number of iterations before the training process of a NN learner is stopped. The tolerance value is used to set a condition on which a stopping criterion for weight optimization is defined. When the change in the error of the backpropagation process is less than the tolerance value, the algorithm performs no further optimization. It is important to use a set of parameters to implement NNs in order to achieve an optimal balance between generalization accuracy and computational complexity of the network. The *neural_network* package in the *sklearn* library is used to access neural networks in Python.

6.4.6 Random Forest

As described in Section 4.5, a RF algorithm combines multiple classification or regression trees to obtain better generalization performance. A RF algorithm gives the final prediction as the majority vote of the predictions of base trees for classification problems. For regression problems, the average of the predictions of base trees in the forest is obtained. In Python, the RF algorithm is accessed from the *"ensemble"* package in the *"sklearn"* library.

6.5 Fusion of the Predictions of Experts

For classification problems, majority voting is used to combine the predictions of base experts within an ensemble, and ties are determined arbitrarily. The majority voting determines the final prediction as the label that is most frequently predicted for each sample of the test set. For regression problems, the average of the predictions of the base experts is computed. Majority voting and averaging were discussed in Section 3.4.

6.6 Chapter Summary

The methods used to construct the ensembles of this research were discussed in this chapter. The ensembles consider multiple instances of the base algorithms, the inductive biases of base algorithms, and diversity within the ensemble. In addition, the ensembles capitalized on the advantage of combining the strength of multiple instances of the selected base algorithms.

Chapter 7

Empirical Process

7.1 Introduction

The empirical process provides all the information necessary for the comparison of HTEs and homogeneous ensembles developed in this research. The research question to compare the HTEs and homogeneous ensembles states that *"due to the inductive biases of the base algorithms in an ensemble, can a heterogeneous mixture of experts result in an ensemble that consistently produces more accurate predictions than that of a homogeneous mixture of experts by capitalizing on the advantages of the experts that make up the heterogeneous mixture"*? This research question is examined under the following dataset configurations: *clean data, skewed class distributions, outliers, bagged subsets, and feature subsets.*

The empirical process to follow to answer the research question under each dataset configuration, also referred to as the *modelling study*, is discussed in the following sections. Section 7.2 presents the modelling studies on which the developed ensembles were evaluated, while Section 7.3 presents the selected benchmark problems on which individual ensembles were developed. The pre-processing of the datasets is discussed in Section 7.4, and Section 7.5 describes the performance measures used to compare the HTEs and homogeneous ensembles. Section 7.7 provides the algorithm control parameter values for the selected ML algorithms. Section 7.8 describes the statistical tests used

to investigate statistically significant differences between the HTEs and homogeneous ensembles. Lastly, Section 7.9 provides a summary of the chapter.

7.2 Modelling Studies

The research question identified in this research resulted in six modelling studies on classification problems and five modelling studies on regression problems. The modelling studies conducted on classification problems are provided as follows:

- Clean Data: This study involves the development of the ensembles on clean training dataset consisting of full sample and features with balanced classes as well as removal of outliers and missing values.
- Skewed classes: The study involves the introduction of skewed class distributions in various percentages to the clean dataset. For binary classification problems, the skewed classes were considered in the range of 10-90%, 15-85%, 20-80%, 25-75%, 30-70%, 35-65%, 40-60%, 45-55%, and 50-50%. For multi-class classification problems, one of the classes was undersampled while the other classes were balanced.
- Number of outliers: In this study, outliers were introduced to the clean data in various percentages. The number of outliers was included within a range of 1% to 5%. This range was considered due to the significant impact of outliers in the predictive performance of ML models.
- Severity of outliers. The introduction of outliers to the clean data is presented in this modelling study and outliers were included from 2.0, 2.5, 3.0, 3.5, to 4.0 standard deviations from the mean.
- Different bagged subsets: This modelling study considers the investigation of the performance of the ensembles on different bag sizes of the clean training data sampled in various percentages. The different bagged sizes range from 10%, 20%, 30%, ... to 80%, 90%, and 100% as provided in Table 7.1.
- Different feature subsets: The performance of the ensembles was also investigated on different feature sizes of the clean training data sampled in various percentages. The different feature sizes range from 10%, 20%, 30%, ... to 80%, 90%, and 100% as provided in Table 7.1

For regression problems, all modelling studies were performed except for the skewed class distribution study, because the prediction outcome of a regression problem is a numeric value and not a class label. The choice of sampling in all modelling studies except for the clean data study provides sufficient resolution to investigate the relationship of the ensemble performance with the different sampling sizes in each study. All of the ensemble models identified in Section 6.2, i.e. 13 ensemble models for classification problems and 11 ensemble models for regression problems, were implemented for each of the modelling studies.

Table 7.1: Data Configurations

Skewed Classes (%)	Number (%)	Severity (σ)	Bag Size (%)	Feature size (%)	k-fold	runs
10-90% - 50-50%	1-5	2-4	10-100	10-100	10	10

7.3 Selection of Benchmark Problems

In order to conduct a critical empirical analysis of the HTEs and homogeneous ensembles, different benchmark problems were selected from the University of California Irvine (UCI) ML repository. The datasets found in the UCI ML repository are publicly available datasets that provide information about different problems across several applications[1].

The selected datasets cover a range of problem characteristics and complexities. Datasets of different complexities ensure that a general conclusion is reached about the performance of the different ensembles. The scope of this research is the analysis of the performance of different ensembles on classification and regression problems, and therefore both classification and regression problems are included. Ten classification and ten regression problems were selected. The characteristics and complexities of the selected problems are presented as follows:

- **The number of samples in the dataset:** Datasets with small and large numbers of samples represent different levels of complexity. While datasets having small samples were collected, more focus was on the collection of larger datasets because it is for large problems that ensembles have been developed.
- **The number of input features**: The more input features, the higher the complexity

[1]https://archive.ics.uci.edu/ml/datasets.php

of a problem. Datasets that consist of a large number of input features lead to the curse of dimensionality problem.

- **The distribution of class features:** If the class distribution of the target feature is imbalanced, a problem is deemed more difficult. Datasets with class imbalance often result in biased predictions towards the majority class in the datasets.
- **The number of class labels:** Binary and multi-class classification problems represent different levels of complexity due to varying methods in training ensembles. For instance, a number of ML algorithms, including *k*NN, DT, NB, inherently support multi-class classification while the SVM algorithm does not. The SVM algorithm naively supports binary classification, but requires extensions such as one-vs-one or one-vs-all to handle multi-class classification. For each extension, the SVM algorithm needs to handle additional parameters and constraints in solving the optimization problem to perform multi-class classification efficiently.
- **The type of input feature data:** Datasets with a variety of input feature types, such as categorical or numeric, are more challenging. For instance, NN, SVM and *k*NN algorithms intrinsically perform well with numeric features while the DT algorithms are designed to work with categorical features. However, research has shown that DTs and NNs do not perform well on data with a mix of categorical and numeric types. This is not the case for *k*NN, NB and SVM algorithms because these algorithms work well with data having a mix of input feature types. However, pre-processing is required for the algorithms.
- **Problems with noise and outliers**: Datasets with different noise characteristics and outliers in input and target features present different levels of complexity.

7.3.1 Classification Problems

The selected classification problems for this research are discussed in this section. The characteristics of these classification problems are summarized in Table 7.2.

Sonar Dataset

The sonar dataset specifies sonar returns collected from a metal cylinder and a cylindrically shaped rock positioned on a sandy ocean floor. The classification problem

specified in the dataset is to classify if a sonar sample is labelled as a mine metal cylinder (M) or a rock (R) sonar signal (Gorman and Sejnowski, 1988). There are 60 continuous input features that describe sonar samples. The input features represent energy within a particular frequency band achieved from the received sonar signals.

The complexity of this dataset refers to the characteristics of the dataset having a binary classification problem with an almost balanced distribution of target feature classes, a small number of samples, a large number of features, and features of the continuous type.

Table 7.2: Characteristics of Selected Classification Problems

Dataset	Total Samples	Features	Majority Samples	Minority Samples	Imbal. Ratio (Maj./Min.)	Number of Classes	Feature Type
Sonar	208	60	111	97	1.14	2	Continuous
Breast Cancer	286	9	201	85	2.36	2	Categorical
Indian Liver	583	10	416	167	2.49	2	Multivariate
Credit Approval	690	15	383	307	1.25	2	Multivariate
Red Wine	1600	12	1519	81	18.75	6	Continuous
Car Evaluation	1728	6	1594	134	11.89	4	Categorical
White Wine	4898	12	4873	25	194.92	7	Continuous
Nursery	12960	9	12630	330	38.27	4	Categorical
Bank Marketing	45211	17	36535	8676	4.21	2	Multivariate
Censor Income	48842	14	37155	11687	3.18	2	Multivariate

Breast Cancer Dataset

The breast cancer dataset consists of nine input features predicting whether the breast cancer of a patient is recurring or not (Tan and Eshelman, 1988). The complexity of this dataset refers to the characteristics of the dataset consisting of a binary classification problem with an unbalanced class distribution, a small number of samples, a small number of features, and input features of the categorical type.

Indian Liver Patient Dataset

The Indian liver patient dataset consists of 10 input features that specify liver disease in patients. Such input features include age, gender, total Bilirubin, direct Bilirubin, total proteins, among other features (Ramana et al., 2011). The target feature is a selector representing a class label associated with the presence or absence of liver disease in a

patient. The complexity of this dataset refers to the characteristics of the dataset including a binary classification problem with an unbalanced class distribution, a small-moderate number of samples, a small number of features, and multiple input features types.

Credit Approval Dataset

The credit approval dataset is made up of samples representing approved or declined requests for credit cards in a financial company (Quinlan, 1987). The complexity of the dataset refers to the characteristics of the dataset consisting of a binary classification problem with an almost balanced target class distribution, a small number of samples, a moderate number of features, and multiple input feature types.

Red Wine Dataset

The red wine dataset describes red wine quality using 11 input features. The target feature is a class label associated with the quality of wine samples. The input features consist of features such as acidity, residual sugar and alcohol, among other features (Cortez et al., 2009). The complexity of this dataset refers to the characteristics of the dataset having multiple classes to predict, an unbalanced class distribution, a small-moderate number of samples, a small number of features, and input features of the continuous type.

Car Evaluation Dataset

The car dataset describes the specifications of cars based on six categorical input features. These features include technical characteristics such as comfort, safety, and price data. The target feature consists of four classes such as unacceptable, acceptable, good and very good (Bohanec and Rajkovic, 1988; Dua and Graff, 2019). The complexity of this dataset refers to the characteristics of the dataset consisting of multiple classes with an unbalanced distribution, a small-moderate number of samples, a small number of features, and input features of the categorical type.

White Wine Dataset

The white wine dataset has the same input features as the red wine dataset. The white wine dataset, however, has a different class distribution than the red wine dataset, more samples and an additional quality class label (Cortez et al., 2009; Dua and Graff, 2019).

The complexity of the white wine dataset refers to the characteristics of the dataset having multiple classes to predict, an unbalanced class distribution, a moderate number of samples, and a low number of features with input features of the continuous type. Additionally, the dataset is characterized by outliers in the values of the features.

Nursery Dataset

The nursery dataset contains eight categorical input features that describe the ranking of a nursery school admission application. These features include the employment of parents, family structure of a child, financial status of the parents, social and health profiles of the family, among others. The target feature represents one of four levels of recommendation for each nursery application (Olave et al., 1989; Zupan et al., 1997). The target feature is almost balanced, with three significant classes having an almost even distribution.

The complexity of this dataset refers to the characteristics of the dataset having multiple classes with an almost equal distribution of target class features, a large number of samples, a small number of features, and features of the categorical type.

Bank Marketing Dataset

The bank marketing dataset specifies the direct marketing campaigns of a banking institution based on phone calls. The dataset consists of 20 multivariate features that describe the characteristics of a bank client to classify if the client makes a bank term deposit or not. These features include information such as client age, education, type of job and marital status, among others (Moro et al., 2014).

The complexity of this dataset refers to the characteristics of the dataset consisting of a binary classification problem with an unbalanced class distribution, a moderate number of features, a large number of samples, and multiple input feature types.

Censor Income Dataset

The censor income dataset, also known as the "*Adult*" dataset, contains 13 multivariate input features. The target feature describes whether the sample, representing an adult, earns more or less than $50,000 a year (Kohavi, 1996; Dua and Graff, 2019). The input features include information such as the age of the adult, native country, education and

work class, among other features.

The complexity of this dataset refers to the characteristics of the dataset having a binary classification problem, an unbalanced target feature distribution, a large number of samples, a small number of features, and multiple input feature types.

7.3.2 Regression Problems

The selected regression problems for this research are discussed in this section. The characteristics of these regression problems are summarized in Table 7.3.

Table 7.3: Characteristics of Selected Regression Problems

Dataset	Samples	Features	Feature Type
Yacht Hydrodynamics	308	7	Continuous
Residential Building	372	105	Continuous
Student Performance	395	33	Continuous
Real Estate	414	7	Multivariate
Energy Efficiency	768	8	Multivariate
Concrete	1030	9	Multivariate
Parkinsons Disease	5875	26	Multivariate
Air Quality	9358	15	Multivariate
Bike Sharing	17389	16	Continuous
Gas Turbine	36733	11	Multivariate

Yacht Hydrodynamics Dataset

The yacht hydrodynamics dataset consists of seven input features used to estimate the hydrodynamic performance of sailing yachts. The regression problem in the dataset is to predict the residuary resistance of sailing yachts at initial design stage (Ortigosa et al., 2007). The complexity of this dataset refers to the characteristics of the dataset having a small number of samples, a small number of features, the presence of outliers in feature values, and continuous input feature type.

Residential Building Dataset

The residential building dataset describes 105 physical, financial, and economic features to construct different residential buildings. These features are used to estimate the sales price of a residential apartment (Rafiei and Adeli, 2016). The complexity of this dataset refers to the characteristics of the dataset consisting of a small number of samples, a large number of features, and continuous input features types.

Student Performance Dataset

The student performance dataset provides information about student performances in secondary education. The regression problem is to predict student performance in a high school mathematics subject (Cortez and Silva, 2008). The complexity of this dataset refers to the characteristics of the dataset including a small-moderate number of samples, a large number of features and multiple input features types.

Real Estate Dataset

In the real estate dataset, a regression problem is solved by determining the monetary valuation of a given real estate. The dataset consists of features such as transaction date, house age, geographical coordinates, among others (Yeh and Hsu, 2018). The complexity of the real estate dataset refers to the characteristics of the dataset having a small-moderate number of samples, a small number of features, and multiple features types.

Energy Efficiency Dataset

The energy efficiency dataset has eight features used to estimate the energy efficiency of the shapes of different buildings. These features, which include relative compactness, surface area, well area, roof area, among others, are used to predict two real-valued outputs corresponding to the heating load and cooling load requirements of the buildings (Tsanas and Xifara, 2012).

The complexity of this dataset refers to the characteristics of the dataset consisting of a small-moderate number of samples, a small number of features, and multiple input features types.

Concrete Dataset

The concrete dataset is made up of eight features that quantitatively determine concrete compressive strengths (Yeh, 1998). Such features include cement, blast furnace slag, fly ash, water, among others. The complexity of this dataset refers to the characteristics of the dataset, including a moderate number of samples, a small number of features, presence of outliers in atttribute values, and features of the continuous type.

Parkinsons Disease Telemonitoring Dataset

The Parkinsons disease telemonitoring dataset provides information about biomedical voice signal recordings captured by a telemonitoring device to detect early-stage symptoms of parkinsons disease in clinical subjects. The dataset is made up of 26 features describing the prediction of clinical motor and unified Parkinson's disease rating scale (UPDRS) scores of each clinical subject (Tsanas et al., 2009).

The complexity of this dataset refers to the characteristics of the dataset having a large number of samples, a moderate number of features, and multiple input features types.

Air Quality Dataset

The air quality dataset consists of hourly averaged features used to predict the net hourly concentrations of Nitrogen Dioxide (NO2) of a chemical multi-sensor device to determine the air quality of the device. Such features include temperature, absolute humidity, relative humidity, among others (De Vito et al., 2008). The complexity of this dataset refers to the characteristics of the dataset consisting of a large number of samples, a moderate number of features, and input features of the continuous type.

Bike Sharing Dataset

The regression problem of the bike sharing dataset is to predict the hourly count of bike rentals for environmental and seasonal features. Such features present in the dataset include season, holiday, weekday, workingday, weather, temperature, wind speed, number of registered users and others (Fanaee and Gama, 2013).

The complexity of this dataset refers to the characteristics of the dataset having a large number of samples, a moderate number of features, and multiple input features types.

Gas Turbine Dataset

The gas turbine dataset provides information regarding the study of flue gas emissions. The regression problem of this dataset is to use the eleven ambient variables in the dataset to predict the turbine energy yield of a gas plant with the regression outputs corresponding to carbon monoxide and nitrogen oxides gases respectively (Kaya et al., 2019).

The complexity of this dataset refers to the characteristics of the dataset, including a large number of samples, a small number of features, the presence of outliers in features, and input features of continuous type.

7.4 Pre-processing of Datasets

Data pre-processing is an important step to ensure that optimal performance of ML algorithms is achieved. Different algorithms require different representations of data due to algorithm-specific assumptions about data types, the number of classes, and data quality aspects. Therefore, this section describes the approaches to pre-processing applicable to all of the datasets and algorithms for the development of ensembles in the research. Section 7.4.1 discusses the general data preprocessing methods applicable to all selected algorithms in the research, while Sections 7.4.2 and 7.4.3 present the methods used to handle outliers and class imbalance in the datasets. The implementation of the data sampling to obtain the bagged and feature subsets is discussed in Section 7.4.4, while Section 7.5 provides algorithm specific preprocessing methods.

7.4.1 General Pre-processing

Training datasets are bootstrapped replicates drawn from an algorithm-specific pre-processed dataset. The pre-processed training datasets contain the same samples and input features for each algorithm. The problem of missing values was resolved for all modelling studies in both classification and regression problems.

The next section discusses the methods used to handle missing values, feature selection, feature scaling, and data encoding in the research.

Missing Values

The presence of missing values is often attributed to either an error in data collection, data integration or the process of generating values for a feature. Most ML algorithms, such as NNs and SVMs, suffer from missing values and require the implementation of a strategy to deal with missing values. On the other hand, ML algorithms such as DTs and *k*NNs are robust to the presence of missing values. However, for consistency, missing values are dealt with the same way for all algorithms. Therefore, missing values are imputed according to Kelleher et al. (2015).

In the case that a large proportion of values for a required feature or sample is missing, usually over 30%, the information within the feature or sample is deemed irrelevant, and the feature or sample was removed; otherwise, missing values were imputed. Missing values in categorical features were imputed with the mode of the feature values, while numeric features were imputed with the mean of the feature values. If outliers are present in the feature, the outliers were removed.

Feature Selection

Feature selection, also known as *"attribute selection"*, is a process of selecting features that provide significant contribution to the prediction of a ML model. It is important to perform feature selection because the construction of a model on irrelevant features in the datasets tends to decrease the accuracy of the model, increase training time, and allows the model to easily learn the noise in the data which allows overfitting of the model (Kantardzic, 2011).

The Pearson correlation coefficient, a statistical filter-based feature selection method (Hastie et al., 2001), was used to evaluate the relationship between each input feature and the target feature in the datasets. The relationship is captured in a correlation matrix annotated using a heatmap to visually identify the most correlated features to the target feature. The matrix also displays the correlation coefficients between each input feature and the target feature in a tabular form. The input features that significantly correlate with the target feature were selected while irrelevant features were not considered.

The strength of the relationship is defined in terms of direction (i.e., high or low) and

magnitude (i.e., positive or negative) of association between the input and target features. The Pearson correlation coefficient, r, for a set of input feature X and target features Y is calculated as

$$r = \frac{\sum[(X - \mu_X)(Y - \mu_Y)]}{\sqrt{\sigma_X \cdot \sigma_Y}} \tag{7.1}$$

where μ and σ represent the mean and standard deviation of the input and target feature values.

As described by Obilor and Amadi (2018) and Schober et al. (2018), the interpretation of the Pearson correlation coefficient shows that a value of r = -1 and +1 indicates perfect negative and perfect positive correlation coefficients, respectively. A value of r = 0 implies no correlation (i.e., zero relationship), r values lower than ±0.40 are said to be low, r values between ±0.40 and ±0.60 are moderate, while r values above ±0.60 are high. It is important to note that the relevance of categorical features with reference to the target feature was analyzed prior to selection for model development.

Feature Scaling

Feature scaling is performed using min-max normalization and standardization. For min-max normalization, feature values are shifted and rescaled to ensure that the values are in range between 0 and 1. The min-max normalization is often used when the data distribution does not assume a Gaussian distribution (Suarez-Alvarez et al., 2012). Feature values are normalized using

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{7.2}$$

where X' represents the normalized value, X_{max} and X_{min} represent the previous range (i.e maximum and minimum values of a feature) of the unnormalized feature values while X represents the original unnormalized value, i.e the feature value to be scaled.

Standardization, also referred to as *"z-normalization"*, is another scaling technique where feature values are centered around the mean with a unit standard deviation. This means that the mean of the feature becomes zero and the resultant distribution has a standard deviation of one.

Standardization is applied on datasets having numeric feature values that assume a Gaussian distribution (Suarez-Alvarez et al., 2012). The transformed feature values are not restricted to a particular range. Feature values are standardized using

$$X' = \frac{X - \mu}{\sigma} \tag{7.3}$$

where μ is the mean of the feature values and σ is the standard deviation of the feature values. The effect of outliers in the dataset is somewhat minimized when the feature values are scaled or normalized to a specific range of values before model development.

Encoding Categorical Features

Most real-world problems consist of categorical, numeric, or a mix of categorical and numeric feature values. As discussed in Section 7.3, a number of the selected ML algorithms perform well on datasets with categorical features while others work well on numeric features. As a result of this, there is a challenge to obtain optimal performance from the combination of these ML algorithms on datasets consisting of a mix of categorical and numeric features. However, research has shown that ML algorithms that work well on categorical features have also produced remarkable performance on numeric features through different extensions and modifications. Therefore, it becomes imperative to encode the categorical feature values in a dataset into numeric values. The encoding process is performed using the one-hot encoding scheme in this research (Potdar et al., 2017)

One-hot encoding transforms a single variable with i samples and j distinct values, to j binary variables with i samples each. Each sample shows the presence (1) or absence (0) of the binary variable (Potdar et al., 2017). One-hot encoding is illustrated in Table 7.4.

Although, the procedure of the one-hot encoding may lead to the generation of a large number of features which may increase the dimension of the dataset and the multicollinearity among features that may lower the accuracy of a model. However, the process of feature selection and scaling is expected to minimize the downsides of the one-hot encoding because only the relevant features are selected in the dataset, leading to the reduction in the dimension of the dataset.

Table 7.4: Illustration of One-Hot Encoding

Original Dataset		One-Hot Encoding			
Samples	color	Samples	color_red	color_blue	color_green
1	red	1	1	0	0
2	blue	2	0	1	0
3	green	3	0	0	1
4	blue	4	0	1	0

7.4.2 Outliers

Outliers are classified as values that lie far away from the central tendency of a feature. Outliers can be smaller or larger than the vast majority of the samples. Outliers bias the predictive model towards the outlying values. A few outliers are sometimes enough to distort the predictions of an ensemble either by altering the mean performance of base learners or by increasing variability (Cousineau and Chartier, 2010).

Research has shown that a number of ML algorithms exhibit different sensitivities to outliers during the inductive learning of the algorithms while others do not. The classification trees, RF, NB and kNN algorithms with large values of k are robust to outliers in a dataset. On the other hand, regression trees, kNN for regression, kNN with small values of k, NNs that do not use a robust estimator, and SVMs with soft margin approaches often show sensitivity to outliers and require that outliers be removed from the dataset (Bandaragoda et al., 2018; Wang et al., 2019a).

Many techniques have been developed to detect outliers. The techniques include graphical methods, parametric and non-parametric statistical methods, distance-based method, density-based methods, clustering methods and learning-based methods (Wang et al., 2019a). Automatic methods have also been developed for outlier detection (Bandaragoda et al., 2018). Such methods include isolation forests, minimum covariance determinant, local outlier factor and one-class SVM (Liu et al., 2008; Bandaragoda et al., 2018; Wang et al., 2019b).

For consistent treatment of outliers in the clean data study, all outliers were removed using the inter-quartile range (IQR) method proposed by Tukey (1977). The IQR method

was employed due to the data distribution of the selected classification and regression problems in this research. The datasets are skewed and do not adequately satisfy the assumption of a normal or Gaussian distribution. The IQR method has been shown to efficiently handle outliers in non-Gaussian data distribution (Tukey, 1977; Mandić-Rajčević and Colosio, 2019). The IQR range in a dataset is given as

$$IQR = Q3 - Q1 \tag{7.4}$$

where $Q1$ is the first quartile of the data, i.e. 25% of the data lies between minimum value in the data and $Q1$. The $Q3$ is the third quartile of the data, i.e. 75% of the data lies between minimum value in the data and $Q3$, while the IQR is the middle i.e 50% of the data.

Therefore, the IQR method detects outliers as feature values less than the lower bound or more than the upper bound, as follows:

$$LowerBound = (Q1 - 1.5 * IQR) \tag{7.5}$$

$$UpperBound = (Q3 + 1.5 * IQR) \tag{7.6}$$

The number of outliers study considers the implementation of the isolation forest method proposed by Liu et al. (2008). An isolation forest separates outliers through a recursive generation of partitions on datasets and a random selection of a feature from a given set of features. This is followed by the random selection of a split value between the maximum and minimum values of the selected feature is performed. The random partitioning generates shorter paths for outliers. Thus, when a forest of random trees collectively generates shorter path lengths for a given sample, outliers are detected (Liu et al., 2008).

The isolation forest method provides a *"contamination"* argument that defines the percentage proportion of outliers in a dataset, which accepts a range of floating values [0.1-0.5]. The values are used to implement the magnitude of outliers in the modelling study of the number of outliers in the research.

For the severity of outlier study, the inter-quartile range (IQR) method was also used. Tukey (1977) stated that the default value of 1.5 in equations (7.5) and (7.6) defines the

standard deviation from the mean that controls the sensitivity of the outlier range and the decision rule for outliers in a dataset. In this research, the 1.5 value was increased from 2.0, 2.5, 3.0, 3.5 to 4 standard deviations from the mean for the severity of outliers study.

7.4.3 Handling Class Imbalance

The class imbalance problem occurs in a dataset when the number of samples is not evenly distributed among the classes in the dataset (Vluymans et al., 2016). This problem drastically affects the performance of ML algorithms that are sensitive to skewed class distributions, because the prediction of an algorithm is usually biased towards the majority classes (Krawczyk et al., 2016). For instance, SVM, NN, DT, and *k*NN algorithms intrinsically showed sensitivity to skewed class distributions in a dataset.

In this research, the class imbalance for the clean data study was resolved using the synthetic minority oversampling technique (SMOTE) and bagging termed as *"SMOTEBagging"*. SMOTE increases the size of minority class samples by duplicating the samples or creating artificial samples (Chawla et al., 2002; Vluymans et al., 2016). While SMOTE oversampling has been reported to be susceptible to overfitting (Vluymans et al., 2016), the bagging approach implemented in the study serves as complement to SMOTE by providing a benefit to overcome the problem of overfitting. As discussed in Section 3.3.1, the bagging approach randomly samples datasets with replacement to generate new subsets. Each subset is then balanced by SMOTE before modeling. The two parameters required for the implementation of the SMOTE include *k*-nearest neighbors and the total number of over-sampling from minority class. The *"SMOTE"* method is available in the *"imbalanced-learn"* library in Python.

Furthermore, the introduction of skewed class distributions in the clean data was performed using a random undersampling approach. The approach has also been reported to balance training datasets by stochastically removing samples of the majority class without any influence on the minority samples (Vluymans et al., 2016). However, the approach was specifically used in this research to undersample one of the classes in the clean training dataset of any binary or multi-class problem to conduct the skewed class distribution study. The *"Random Undersampler"* can be assessed in the *"imbalanced-learn"* library in Python, and has a *"sampling strategy"* argument which accepts floating

values that defined the percentage of skeweness in the classes of the training dataset. The floating values are included in the ratios of 10-90%, 15-85%, 20-80%, ... ,45-55%, and 50-50% as discussed in Section 7.2.

7.4.4 Bagged and Feature Subsets Sampling

To conduct the modelling studies of the bagged and feature subsets, the *"sample"* function in the pandas dataframe library in Python was employed. For the bagged subsets study, the *"sample"* function was used to randomly resample the training dataset with replacement to create bagged subsets from 10 to 100%. On the other hand, the features in the training dataset were also randomly resampled with replacement to create feature subsets from 10 to 100%. The *"sample"* function has a *"frac"* argument that accepts floating values (in this case 0.1 to 1.0 for the bagged and feature subset studies) to represent the percentage of the samples or features to sample out from the clean training dataset.

7.4.5 Algorithm Specific Preprocessing

Table 7.5 summarizes the techniques relevant to the individual algorithms with respect to the selected classification and regression problems. For regression problems, the actual target feature was used.

Table 7.5: Algorithm Specific Preprocessing

Algorithm	Numeric Feature	Binary Class Labels	Multi-class Labels
DT	Scaling	None	None
NN	Scaling	0, 1	n output values
kNN	Normalization	None	None
SVM	Normalization	0, 1	n output values
NB	Normalization	None	None

Decision Trees

DTs are known to handle data appropriately because DTs work well on categorical and numeric features. Additionally, DTs are robust to the presence of outliers in data. Hence, while normalization of the input features has been reported to be unnecessary for DT algorithms, scaling was performed for the input features in classification and regression

problems in this research. No preprocessing was carried out on the target features in binary and multi-class classification problems. Therefore, pre-processing is not necessary for the application of DTs.

Neural Networks

NNs require that both input and target features be reformatted as shown in Table 7.5. For numeric features, input feature values are linearly scaled to a range of -1 and 1 using min-max normalization for classification and regression problems. The scaling of numeric feature values supports the optimization of the gradient descent algorithm to converge much faster than when the feature values are unscaled. When the feature values are not scaled, the gradient descent algorithm will take more numbers of iterations to converge, which increases the training time of NNs.

The target features were reformatted as (0, 1) for binary classification problems and n output values for multi-class classification problems, where n indicates the number of classes in the multi-class classification problems.

k-Nearest Neighbour

During the computation of distance, kNN algorithm requires the normalization of feature values to ensure large feature values do not dominate small feature values. Therefore, numeric features were standardized using z-normalization for classification and regression problems. No pre-processing of the target feature was performed for binary and multi-class classification problems.

Support Vector Machines

The data processing performed for SVM and kNN are similar except in the preprocessing of target features. The specific output values of (0, 1) were used to represent the target features of binary classification problems. The SVM model represented the negative class as 0 for values below the separating hyperplane, while a positive class is represented as 1 for values above the separating hyperplane. For multi-class classification problems, the target features were encoded as n different output values.

Naïve Bayes

The pre-processing required for a NB classifier is the discretization of numeric feature values. However, the discretization of numeric feature values is unnecessary due to the different probabilistic distributions made by the NB algorithm in Python (i.e. Gaussian, Bernoulli, and multinomial distribution). Numeric feature values were normalized, and no pre-processing of the target feature was performed for binary and multi-class classification problems.

7.5 Performance Measures

This section discusses the performance measures used to compare the developed HTEs and homogeneous ensembles across the selected classification and regression problems. Section 7.5.1 describes the performance measures used to compare the ensembles on classification problems, while Section 7.5.2 discusses the performance measures for regression problems. Quantification of overfitting of the individual ensembles is discussed in Section 7.5.3.

7.5.1 Performance Measures for Classification Problems

The measures used to evaluate the performance of ensembles on classification problems are as follows:

- **Ensemble Accuracy**: The accuracy of an ensemble was measured by calculating the percentage of correct predictions made by the ensemble. An ensemble produced the best generalization performance with an accuracy of 1.0 or 100%. The accuracy of an ensemble is calculated as

$$ENS_{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.7}$$

  where TP, TN, FP, and FN are computed from the confusion matrix (Igual and Seguí, 2017) and defined as follows:

  - True positives (TP): The ensemble predicts a sample as positive, and the actual sample label is positive.

  - False positives (FP): The ensemble predicts a sample as positive, but it is negative.
  - True negatives (TN): The ensemble predicts a sample as negative, and the actual label is negative.
  - False negatives (FN): The ensemble predicts a sample as negative, but the true label is positive.

  Accuracy has been reported to be an inefficient performance measure, especially when the dataset is imbalanced (Veropoulos et al., 1999; Wu, 2003). For instance, in the diagnosis of a medical disease, where the goal is to obtain low false negatives (*i.e. high prediction for unhealthy patients*) than false positives (*i.e. low prediction for healthy patients*), accuracy may provide a misleading result. This is attributed to the fact that accuracy simply assigns the majority of the samples to the negative class while ignoring the positive samples (Wu, 2003). Also, in spam detection, where obtaining low false positives (*i.e. high prediction for relevant mails*) is more important than false negatives (*i.e. low prediction for irrelevant mails*), accuracy may not be a suitable measure. Thus, other performance measures such as precision, recall, F1-Score, or ROC-AUC score, computed from the confusion matrix, are required.

- **Precision**: Precision measures the exactness in the predictions of an ensemble. Precision means that what proportion of positive predictions made by the ensemble was actually correct? That is, when the ensemble predicted a class label as positive, how often was this prediction correct?. The precision value of an ensemble lies in the range of [0, 1]. Thus, an ensemble with a precision value close to 1 indicates better performance, while a precision value close to 0 shows that the performance of the ensemble is not reliable. The precision of an ensemble is calculated as

$$Precision = \frac{TP}{TP + FP} \tag{7.8}$$

- **Recall**: Recall, also referred to as *true positive rate (TPR) or sensitivity*, measures the completeness in the predictions of an ensemble; that is, the proportion of positive samples that are correctly identified as positives (Sumathi and Poorna, 2017). The recall value of an ensemble lies in the range of [0, 1]. An ensemble with a recall value

close to 1 illustrates better performance, while a recall value close to 0 indicates poor performance by the ensemble. The recall of an ensemble is calculated as

$$Recall = \frac{TP}{TP + FN} \tag{7.9}$$

- **F1-Score**: F1-score, also known as the *F-measure*, computes the harmonic mean of the precision and recall performance of an ensemble. While an ensemble may not generate high precision and recall values concurrently, there is a cost associated with tweaking the two measures. The trend of this cost is captured by the F1-score in a single value. The F1-score of an ensemble lies in the range of [0, 1]. F1-score close to 1 means that the ensemble produces a reliable performance, while F1-score close to 0 indicate unreliable performance. The F1-Score is calculated as

$$\text{F1-score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{7.10}$$

7.5.2 Performance Measures for Regression Problems

This section discusses the measures used to evaluate the performance of ensembles on regression problems, which are as follows:

- **Mean Squared Error**: The mean squared error (MSE) determines the quality of an ensemble on a dataset by computing the average of the squares of errors produced by the ensemble during prediction. The error is obtained as the difference between the actual values in a dataset and predicted values by the ensemble. A low MSE indicates that the ensemble prediction is closer to the actual value and vice versa. Thus, an ensemble with a low MSE shows a better fit of the regression line than another ensemble with a high MSE. The MSE is calculated as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{7.11}$$

  where N represents the total number of test samples, y_i is the true value, and $\hat{y}_i$ is the predicted value over the N samples.

For easy interpretation of the final ensemble prediction, the root mean square error

(RMSE) is considered to evaluate the performance of the ensembles on regression problems. The RMSE is preferred to the MSE because the RMSE is measured in the same units as the target feature, while the MSE is measured in units that are square of the target feature, and penalizes larger errors more severely (Dua et al., 2017). The RMSE computes the square root of the MSE as

$$RMSE = \sqrt{MSE} \tag{7.12}$$

where MSE is defined in equation (7.11).

7.5.3 Measurement of Overfitting

An essential aspect of an ensemble is to effectively learn the mapping between the input and target space in the training dataset, while still providing good generalization to the test dataset. The essence is to ensure the ensemble minimizes overfitting of the training datasets. Overfitting was measured using a generalization factor (GF) proposed by Röbel (1994). The GF is calculated for every classification and regression problem.

For regression problems, the GF of an ensemble, ρ, is given as

$$\rho = \frac{MSE\_test_{err}}{MSE\_train_{err}} \tag{7.13}$$

where MSE_test_{err} is the MSE of test dataset and MSE_train_{err} is the MSE of training dataset. As MSE was used to measure accuracy for regression problems, the expectation is to minimize the GF such that $\rho \leq 1$. When $\rho \leq 1$, the ensemble generated a smaller test error than the training error, which is desirable. However, as ρ becomes larger, i.e. $\rho > 1$, the difference between the test and training error increases. The difference illustrates an increase in test error and a decrease in training error, which indicates overfitting.

For classification problems, the classification error is used as a measure of incorrect predictions made by an ensemble. Similarly, the expectation is to minimize the GF, i.e $\rho < 1$ of the ensemble. The generalization factor of an ensemble, ρ, for classification problems is given as

$$\rho = \frac{test_{err}}{train_{err}} \tag{7.14}$$

where $test_{err}$ is the classification error on testing dataset and $train_{err}$ is the classification error on training dataset. The classification error on testing dataset $test_{err}$ for an ensemble is calculated as

$$test_{err} = 1 - test\ accuracy \tag{7.15}$$

The classification error on training dataset $train_{err}$ for an ensemble is calculated as

$$train_{err} = 1 - training\ accuracy \tag{7.16}$$

Therefore, the GFs, ρ, for the classification and regression problems are interpreted in the same way, i.e the possibility for overfitting is indicated when $\rho > 1$.

7.6 k-Fold Cross Validation

k-Fold cross-validation is not a performance measure, but used to validate the generalization performance of an ensemble. k-Fold cross-validation reports the mean accuracy and associated standard deviation of the predictive accuracies of an ensemble over a number of independent runs. The selected number of independent runs is 10.

k-Fold cross-validation splits the entire dataset into k equally sized subsets. Then, an ensemble is trained on $k-1$ of the subsets and tested on the k^{th} subset. The ensemble is trained and tested k times, each time with a different test subset and a different combination of the $k-1$ subsets for training. The obtained results are presented in the form of $\bar{x} \pm \sigma_x$, where $\bar{x}$ represents mean prediction of the ensemble and σ_x denotes standard deviation of the prediction from the mean.

The number of folds, k, and the number of runs used in cross-validation are listed in Table 7.1. For k-fold validation, the scoring parameters used for the ensembles validated on classification and regressions problems are classification accuracy and MSE respectively. Then, for each problem type, the mean, $\bar{x}$, and standard deviation, σ_x, of the classification accuracies and MSEs of the ensembles over a number of independent runs were obtained as

$$\bar{x} = \frac{1}{k}\sum_{i=1}^{k} x_i \tag{7.17}$$

and

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{k} (x_i - \bar{x})^2}{k(k-1)}} \tag{7.18}$$

7.7 Hyperparameter Optimization of Base Algorithms

This section provides the control parameter values and processes used to develop the HTEs and homogeneous ensembles in this research. The control parameters stipulated for each base algorithm are set to be the same for all training processes. The goal of this research is to compare the ensembles over a range of classification and regression problems. Therefore, the control parameters were tuned to find values that work well for the individual base learners on each problem. The obtained values were set for the multiple instances of the base ML algorithms in the HTEs and homogeneous ensembles in the research. Section 7.7.1 discusses random search optimization used to obtained suitable control parameter values for the base learners in HTEs and homogeneous ensemble.

7.7.1 Random Search Optimization

As discussed in Section 2.3.4, the selection of appropriate control parameter configurations for the base algorithms is significant to balancing the bias-variance tradeoff. Therefore, the goal is to obtain optimal control parameter values that will provide a good tradeoff to minimize underfitting and overfitting in the prediction performance of an ensemble. It is important to note that the choice of the degree and range of values for the control parameters is dependent on the control parameter of a specific algorithm to be tuned.

For this reason, given the large numbers of control parameters available for individual ML algorithms used to develop the ensembles in this research, the random search optimization algorithm (discussed in Section 3.3.5) was used to search for suitable control parameters for each ML algorithm. The random search algorithm samples random combinations of control parameter values. The control parameter configuration that is best for base learners within the ensembles were selected. Hence, the control parameter search space was reduced to the control parameters that only contribute to the final

ensemble results.

The random search function evaluates each base learner for a given control parameter using cross-validation. The function requires four arguments during hyperparameter tuning. The first is an instance of the base learner required to be optimized. The second is the hyperparameter search space which accepts the control parameters of a base learner and a distribution of values to sample for each control parameter. The third argument is the scoring parameter, which includes classification accuracy and MSE for classification and regression problems respectively. The last argument is the cross validation *cv* argument which allows the specification of an integer number of folds, i.e. *k*-folds.

Prior to random search optimization, the stipulated range of control parameter values is provided in Table 7.6.

Table 7.6: Algorithm Control Parameters

Base Algorithms	**Control Parameters**	**Parameter values**
NB	Laplace Smoothing	0.001 - 1.0
*k*NN	*k* Values	1,3,5,7,9
	Distance Metric	1,2,5
DT	Split Criterion (Class/Regress)	Gini or Entropy / MSE or MAE
	Max. Depth	1 - 25
	Split Strategy	Best or Random
SVM	Kernel	RBF or poly
	Penalty Cost, *C*	0.01 - 1.0
	degree	3 - 9
	Kernel Width σ	0.001-1.0
	Multi-class decision function	one-vs-rest
MLP	Hidden Layers	2
	Hidden Nodes	(25,25,25), (50,50,50)
	Solver	adam or sgd
	Maximum Number of Iteration	500 - 1000
	Learning Rate	0.0001 - 1.0
	Activation function	ReLu or tanh

The Laplace smoothing values selected for the NB algorithm ranges between 0.1 and 1.0. Odd number of k neighbours $(k = 1,3,5,7,9)$ was considered for the *k*NN algorithm to

avoid ties for binary and non-binary classifications. Then a choice among the Manhattan $(p = 1)$, Euclidean $(p = 2)$ or Minkowski (any p larger than 2) distance was made as the distance metric.

For DTs, the Gini impurity and entropy for information gain were used as splitting criteria for classification problems. The mean squared error and mean absolute error were used for regression problems. A depth range of 1 to 25 was used as the maximum depth of the tree, while the strategy to choose the split at each node is a choice between the best or a random strategy. The minimum number of samples required to split an internal node was set at 2, and the minimum number of samples needed to be at a leaf node was set to 1. The RF models were developed using default parameter values available in the *sklearn* library in Python, but with 10 base DTs.

For SVMs, the radial basis function (RBF) and polynomial kernels were selected as the kernel functions. A range of 0.01 to 1.0 was set for the cost of misclassification, C, degree of polynomials was set to a range of 3 to 9, while the the kernel width, σ, was set to a range of 0.001 to 1.0. In the case of multi-class classification problems, the one-versus-rest method was selected to define the shape of the decision function.

For MLPs, two hidden layers of different numbers of hidden nodes were set for each network due to the different complexities and sizes of the datasets. Activation functions used include *relu* or *tanh*. Then *adam*, a stochastic gradient-based optimizer, and the stochastic gradient descent *sgd* optimization algorithm were selected for weight optimization, because the algorithms work well on small and large datasets. The maximum number of iterations, i.e. number of epochs was set between 500 to 1000 to ensure appropriate convergence of the networks. The learning rate value was set to a range of 0.0001 to 1.0 to ensure effective adaptation of each model to individual problems at different epochs. Also, the range was set to provide a suitable learning rate for the minimization of the long training time of the network and to ensure optimal convergence of the models for a given problem. The default value for momentum was used.

The random search optimization was implemented a number of times to obtain different best control parameter configurations from the control parameter values in Table 7.6 in order to construct the different types of ensembles identified in this research. To develop each ensemble for classification and regression problems, the random search optimization

algorithm was first implemented to obtain the control parameters configured for the multiple instances of the base algorithms in an ensemble.

For the first ensemble type, i.e. ensembles consisting of multiple instances of the same ML algorithm with the same configuration, the random search optimization was run once on each dataset from different initial conditions. The best control parameter where an algorithm produced the highest classification accuracy or lowest MSE score was selected and configured for all the 10 component members in each ensemble. Hence all the 10 base learners in each ensemble consist of the same configuration.

For the second ensemble type, i.e. ensembles consisting of multiple instances of the same ML algorithm with different configurations, the random search optimization was run 10 times on each dataset from different initial conditions. For each run of the optimization algorithm on a dataset, an algorithm induced a control parameter which illustrates a specific assumption made by the algorithm on the dataset for the run. Thus, 10 different best control parameters for each run were derived and configured for the 10 individual component members in each ensemble.

Furthermore, for the third ensemble type, i.e. ensembles consisting of multiple instances of the different ML algorithms with the same configuration, the random search optimization was run once for each algorithm from different initial conditions. The best control parameter obtained from each algorithm was set for the two instances of each ML algorithm in each ensemble. Thus, two instances of each ML algorithm in every ensemble consist of the same control parameter.

For the fourth ensemble type, i.e. ensembles consisting of multiple instances of the different ML algorithms with different configurations, the random search optimization was run twice for each algorithm from different initial conditions. The two runs of the optimization produced different control parameters that were set for the two instances of each ML algorithm in an ensemble. As a result, the two instances of each ML algorithm in every ensemble consist of different control parameters.

While the tuning process of the control parameter was rigorous and demanded careful attention, the process was necessary to avoid manual setting of the control parameters of each instance of the algorithm, and to avoid bias. Also, the tuning process was done to ensure that the base learners in a particular ensemble induce different experts based on

the control parameters configured for the component learners. Additionally, these control parameters result in different base experts making different assumptions after learning the trends and patterns in the datasets.

7.8 Statistical Tests

This section presents the statistical tests used to compare the performance of the ensembles to determine if differences in performance are statistically significant or not. The comparison is achieved by selection of statistical tests that are appropriate for a dataset. While parametric tests are often performed for datasets that assume a Gaussian distribution, non-parametric tests are performed on datasets with unknown distribution. In recent ML literature, non-parametric tests have usually been employed to compare the performance of ML algorithms on multiple datasets due to the unknown data distribution (Demsar, 2006; García et al., 2009; Singh et al., 2016). This research considers the selection of non-parametric tests to statistically compare the developed ensembles. Section 7.8.1 discusses Friedman test, while Section 7.8.2 describes Bonferroni-Dunn test which is a posthoc test for Friedman test.

7.8.1 Friedman Test

For the purpose of this research, the Friedman test (Friedman, 1940), a non-parametric statistical test, was performed for each modelling study to determine whether there exists a statistically significant difference between the performance of the ensembles or not, i.e. whether the results of the ensembles happened by chance or not. If there exists a statistical significant difference between ensembles, then a posthoc test, i.e. the Bonferroni-Dunn test (Dunn, 1961) was performed to verify which ensembles significantly differ in performance. These statistical tests are used to compare the performance of the developed HTEs and homogeneous ensembles on multiple datasets and are described next.

In the implementation of the Friedman test, all ensembles were ranked in terms of generalization performance, i.e. testing accuracy for classification problems and testing RMSE for regression problems. Then the average ranks of the ensembles were computed and compared. The null hypothesis of the Friedman statistic illustrates the assumption that all ensembles perform equally on all datasets and, therefore, the ranks of the

ensembles should be equal. Formally, the null hypothesis of the Friedman test is defined as

$$H_0 : j_1 = j_2 = j_3 = ... = j_k \tag{7.19}$$

The rejection of the null hypothesis results in the acceptance of the alternative hypothesis, i.e. there is a significant difference in the generalization performance of the ensembles, defined as

$$H_1 : \textit{Not all } j_i \textit{ are equal; } i = 1,2,3,...,k; \textit{ i is a list of ensembles} \tag{7.20}$$

The Friedman statistic is calculated as

$$\chi_F^2 = \frac{12N}{j(j+1)} \left[\sum_k R_k^2 - \frac{j(j+1)^2}{4} \right] \tag{7.21}$$

where N is the number of datasets, j is the number of ensemble compared, while R is the rank of the ensemble k. The value calculated in equation (7.21) is distributed according to χ_F^2 with $k-1$ degrees of freedom. However, due to the undesirability in the conservativeness of the Friedman statistic, the Iman-Davenport statistic (Iman and Davenport, 1980) was employed as an extension to test the null hypothesis, given as

$$F_F = \frac{(N-1)\chi_F^2}{N(j-1) - \chi_F^2} \tag{7.22}$$

The value of the Iman-Davenport F_F statistic in equation (7.22) is distributed according to the F-distribution with $j-1$ and $(j-1)(N-1)$ degrees of freedom obtained from a statistical handbook. The value of F_F is then compared to the critical value associated with these degrees of freedom. When the value F_F is greater than the critical value, the null hypothesis is rejected. The rejection of the null hypothesis leads to the acceptance of the alternative hypothesis, which is followed by conducting a posthoc test to determine which ensembles are significantly different.

7.8.2 Bonferroni-Dunn Test

For the Bonferroni-Dunn test, the HTEdf is designated as the control ensemble and is compared with other HTEs and homogeneous ensembles. The differences between the average rank of the HTEdf and other ensembles were compared with a critical difference (CD) value (Demsar, 2006) calculated using

$$CD = q_\alpha \sqrt{\frac{j(j+1)}{6N}} \tag{7.23}$$

where q_α is a critical value associated with the number of ensembles and significance level $\alpha = 0.05$ for a two-tailed Bonferroni-Dunn test. A significant difference in performance is detected between the HTEdf and any ensemble if the difference in average rank between the HTEdf and the ensemble is higher than the CD. The results of the Bonferroni-Dunn tests are presented using the critical difference plot proposed in Demsar (2006).

7.9 Chapter Summary

This chapter discussed the empirical process used to compare the developed ensembles. The processes were performed to provide answers to the research question identified in this research. Section 7.2 discussed the modelling studies on which the different types of ensembles developed were evaluated, while Section 7.3 discussed the selected classification and regression problems problems for the research. Section 7.4 described the data pre-processing methods focusing on general preprocessing methods, methods used to preprocess outliers, skewed class distributions, bagged subsets and feature subsets as well as algorithm-specific preprocessing methods. The performance measures used to compare the developed ensembles were presented in Section 7.5. Sections 7.6 and 7.7 described the k-fold cross-validation and algorithm control parameter values set for the base learners in an ensemble, while Section 7.8 described the statistical tests used to determine whether a statistically significant difference exists between ensembles.

Chapter 8

Empirical Analysis of Results for Classification Problems

8.1 Introduction

This chapter provides an empirical analysis of ensemble models for classification problems. The comparative analysis of the developed ensembles implemented on the six modelling studies presented in Section 7.2 is discussed in this chapter.

The analysis of the results of these ensembles considers the inductive biases of the base ML algorithms in each ensemble model, bias-variance tradeoff and the characteristics and problem complexities of the classification datasets described in Section 7.3. The ensemble models were run on high performance computing (HPC) environment. All reference to the results of training and testing accuracy, GF, and F1-score are mean averages of these performance measures for the classification datasets in the modelling studies.

Sections 8.2 and 8.3 discuss the results of the ensembles for clean data and skewed class distributions studies respectively. The results of the ensembles for the number and severity of outliers studies are described in Sections 8.4 and 8.5. Sections 8.6 and 8.7 discuss the results of the ensembles for bagged subsets and feature subsets

studies respectively. The formal statistical tests performed for each modelling study are also described. Following this, an overall discussion of the outcome of the ensemble performance for all modelling studies is presented in Section 8.8. Section 8.9 concludes the chapter with a summary of the findings.

8.2 Clean Data Study

This section discusses the results of the HTEs and homogeneous ensembles developed in this research on clean classification datasets. Each clean dataset consists of full sample and features where the skewed classes were balanced and outliers removed. The results of the ensembles for each dataset are described separately.

Clean Sonar Dataset

The problem is to classify if a sonar sample is labelled as a mine metal cylinder (M) or a rock (R) sonar signal. Plots of the testing and training accuracies of the ensembles for this dataset are provided in Figures 8.1 and 8.2. The results of the testing and training accuracy, GF and F1-score of the ensembles for the clean Sonar dataset are summarized in Table 8.1.

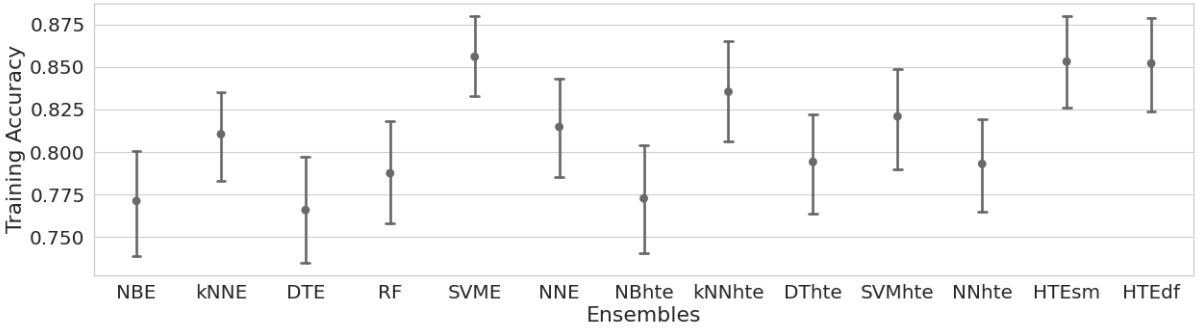

Figure 8.1: Training Accuracy of Ensembles for Clean Sonar Dataset

Illustrated in Figure 8.1, the SVME produced the highest training accuracy of 85.6%, followed by the HTEsm (85.3%) and the HTEdf (85.2%). The DTE offered the worst training performance with a training accuracy of 76.5%, followed by the NBE with an accuracy of 77.1%.

In Figure 8.2, the HTEdf offers the best generalization performance with a testing accuracy

of 81.6% and is ranked as the most accurate ensemble.

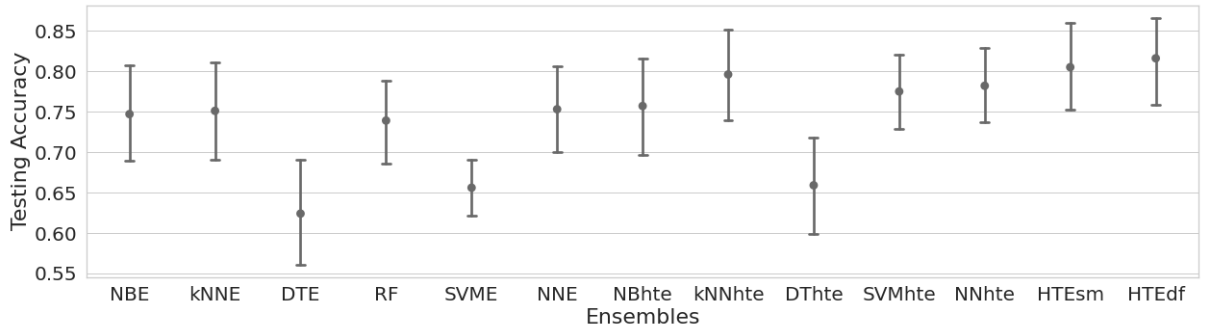

Figure 8.2: Testing Accuracy of Ensembles for Clean Sonar Dataset

The HTEsm (80.5) and *k*NNhte (79.6%) are ranked as the second and third most accurate ensembles. The DTE (62.4%) is the least accurate ensemble, followed by the SVME (65.6%) and DThte (65.9) ensembles. Also, as observed in Table 8.1, all HTEs (i.e. NBhte, *k*NNhte, DThte, SVMhte, NNhte, HTEsm, HTEdf) outperformed the pure homogeneous ensembles (i.e. NBE, *k*NNE, DTE, and SVME) and the RF algorithm in terms of generalization performance.

The generalization performance of the DTE and DThte showed that both ensembles, compared to other ensembles, struggled with the characteristics of the Sonar dataset consisting of small samples, large number of continuous-valued features, and binary classes. The DThte provided better generalization performance than the DTE, which illustrates the advantage of the mixtures of heterogeneous experts over homogeneous mixtures. Also, the choice of the same control parameter configuration for the SVME did result in worst generalization performance, while the SVMhte benefitted from the different configurations used by the base members.

Illustrated by the GF of the ensembles, the SVME showed more overfitting of the training dataset than other ensembles. The NBhte and NNhte ensembles provided GFs just slightly above the expected GF value of 1.0 as proposed by Röbel (1994). For the F1-score performance, the HTEdf and HTEsm are the best performing ensembles. The HTEdf and HTEsm are more precise in providing 87% accuracy of the mine or rock samples, while being robust to identifying 87% of all rock samples. This means that the HTEdf and HTEsm produced the lowest misclassification rate of 13%, i.e. only a small number of rock samples were misclassified as mine samples in the Sonar dataset. It is important

to note that the interpretation of the F1-score for the Sonar dataset indicates a similar interpretation for other datasets in the clean data study and other modelling studies in this chapter. The generalization performance of the HTEdf and HTEsm indicates that a combination of different ML algorithms produced a better mixture of heterogeneous experts than using the same ML algorithm to predict the binary labels of mine and rock samples in the Sonar dataset.

Table 8.1: Ensemble Results for Clean Sonar Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.74700 ± 0.22168	0.77134 ± 0.11643	1.10595	0.76
*k*NNE	0.75100 ± 0.22880	0.81048 ± 0.09474	1.31382	0.86
DTE	0.62400 ± 0.23027	0.76590 ± 0.11238	1.60618	0.70
RF	0.73900 ± 0.17980	0.78752 ± 0.10503	1.22837	0.81
SVME	0.65600 ± 0.12870	0.85600 ± 0.08587	2.38889	0.72
NNE	0.75300 ± 0.19011	0.81476 ± 0.10311	1.33342	0.86
NBhte	0.75700 ± 0.21000	0.77276 ± 0.11429	1.06936	0.74
*k*NNhte	0.79600 ± 0.19871	0.83543 ± 0.10458	1.23958	0.81
DThte	0.65900 ± 0.21810	0.79419 ± 0.10165	1.65687	0.74
SVMhte	0.77500 ± 0.16651	0.82095 ± 0.10773	1.25665	0.79
NNhte	0.78200 ± 0.17139	0.79305 ± 0.09446	1.05338	0.81
HTEsm	0.80500 ± 0.18310	0.85314 ± 0.09884	1.32782	0.87
HTEdf	0.81600 ± 0.19299	0.85210 ± 0.09816	1.24404	0.87

Clean Breast Cancer Dataset

The problem is to predict whether breast cancer in a patient is recurrent or not. Figures 8.3 and 8.4 illustrate the testing and training accuracy of the ensembles. The results of the testing and training accuracy, GF and F1-score of the ensembles are further provided in Table 8.2.

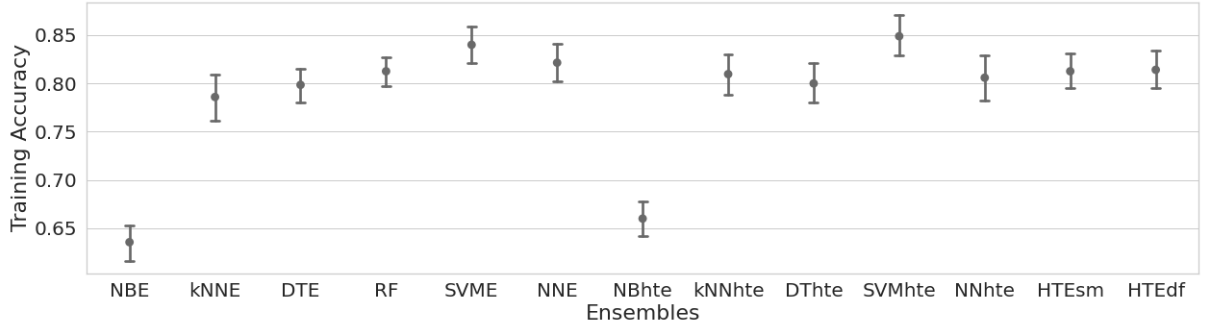

Figure 8.3: Training Accuracy of Ensembles for Clean Breast Cancer Dataset

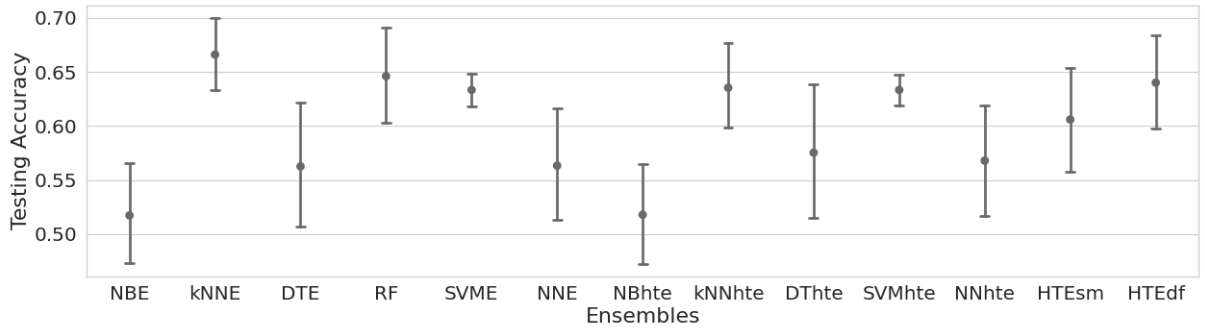

Figure 8.4: Testing Accuracy of Ensembles for Clean Breast Cancer Dataset

As shown in Figure 8.3, the SVMhte generated 84.9% training accuracy, which is followed by the SVME (84%), NNE (82.1%), HTEdf (81.4%), and HTEsm (81.3%). The NBE and NBhte provided the worst training accuracy compared to other ensembles.

Illustrated by the testing accuracy in Figure 8.4, the *k*NNE is ranked as the most accurate ensemble (66.7%), while the *k*NNhte offered the highest F1-score of 61%. The RF algorithm is the second most accurate ensemble (64.6%), outperforming the HTEdf (64%) with a slight difference of 0.6%. The HTEdf is ranked as the third most accurate ensemble. The low standard deviations of the HTEdf and HTEsm provide evidence that the ensembles achieved a good level of stability.

The generalization performance of the *k*NNE indicates the suitability of the ensemble to a small number of features in the Breast Cancer dataset, because *k*NN algorithms do not perform well on a large input dimension due to the cost associated with the distance computation among features. For the SVME (63.3%) and SVMhte (63.3%), the generalization performance of these ensembles showed the ability of SVM algorithms to perform well on small datasets. The NBE and NBhte are ranked as the least performing ensembles based on training and testing accuracies. Illustrated by the GFs of the ensembles, the NBE and NBhte slightly overfit the training dataset, but resulted in low generalization performance compared to other ensembles. The generalization performance of the NBE and NBhte was expected to be efficient because NB algorithms perform well on categorical features compared to numeric features where an assumption of data distribution is intrinsically made.

Also, the DTE, DThte, NNE and NNhte underperformed other ensembles in terms of testing accuracy, while the GFs of these ensembles indicate that the ensembles showed

more overfitting to the training dataset than other ensembles. The generalization performance of the NNE and NNhte showed that the ensembles also struggled with the characteristics of the breast cancer dataset. Thus, the generalization performance of the homogeneous ensembles is competitive with the HTEs for the complexity of the Breast Cancer dataset.

Table 8.2: Ensemble Results for Clean Breast Cancer Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.51733 ± 0.16279	0.63556 ± 0.06301	1.32439	0.53
*k*NNE	0.66600 ± 0.12338	0.78593 ± 0.08225	1.56021	0.53
DTE	0.56267 ± 0.19660	0.79852 ± 0.06422	2.17059	0.56
RF	0.64600 ± 0.16054	0.81250 ± 0.05661	1.88893	0.57
SVME	0.63333 ± 0.05578	0.84000 ± 0.06602	2.29167	0.52
NNE	0.56333 ± 0.19175	0.82148 ± 0.07093	2.44606	0.56
NBhte	0.51800 ± 0.16441	0.66000 ± 0.06004	1.41765	0.54
*k*NNhte	0.63533 ± 0.14320	0.80963 ± 0.07371	1.91556	0.61
DThte	0.57533 ± 0.21497	0.80000 ± 0.07220	2.12333	0.57
SVMhte	0.63333 ± 0.05578	0.84889 ± 0.07799	2.42647	0.57
NNhte	0.56800 ± 0.18475	0.80593 ± 0.08624	2.22595	0.53
HTEsm	0.60600 ± 0.16901	0.81259 ± 0.06559	2.10237	0.55
HTEdf	0.64000 ± 0.16069	0.81407 ± 0.06968	1.93625	0.58

Clean Indian Liver Dataset

The task is to predict the presence or absence of liver disease in a patient. Figures 8.5 and 8.6 illustrate the testing and training accuracy of the ensembles, while Table 8.3 provides the results of the testing and training accuracy, GF and F1-score of the ensembles for the clean Indian Liver dataset.

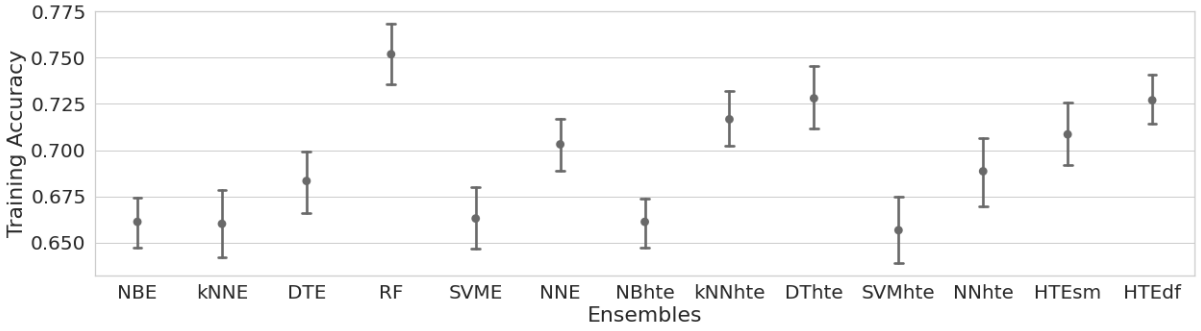

Figure 8.5: Training Accuracy of Ensembles for Clean Indian Liver Dataset

As depicted in Figure 8.5, the RF algorithm achieved the best training performance with an accuracy of 75.2%. The DThte (72.8%) and HTEdf (72.7%) are ranked as the second and third most accurate ensembles in terms of training accuracy. The SVMhte (65.6%) provided the worst training accuracy compared to other ensembles.

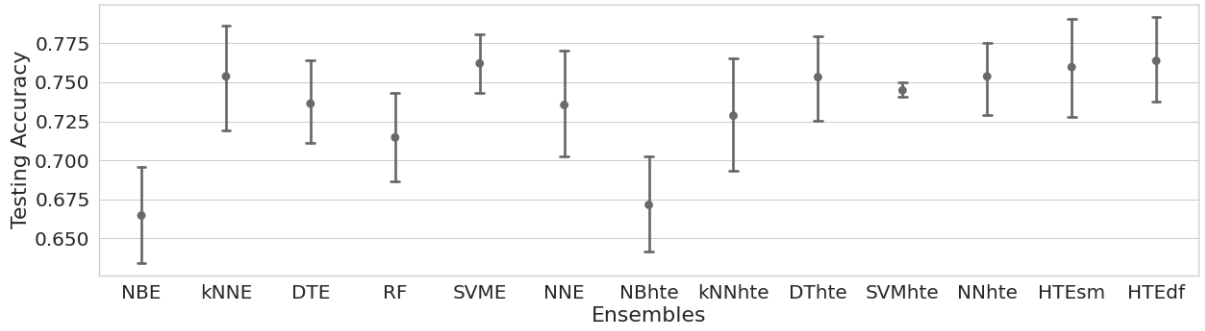

Figure 8.6: Testing Accuracy of Ensembles for Clean Indian Liver Dataset

From Figure 8.6, the HTEdf outperformed all other ensembles achieving a testing accuracy of 76.4%. The standard deviations of the HTEdf (σ=0.07788) and HTEsm (σ=0.09717) also illustrate that the HTEdf and HTEsm achieved the third and fourth best stability of predictions across independent runs. The SVME is the second most accurate (76.2%) ensemble, while the HTEsm is ranked third with an accuracy of 75.9%. The generalization performance of the HTEdf and HTEsm illustrates the benefit of combining different ML algorithms to obtain effective diverse experts in the heterogeneous mixture model compared to using the same ML algorithms for other ensembles. Also, the generalization performance of the SVME indicates the suitability of SVM algorithms to datasets of small sample sizes.

The NBE is ranked as the least accurate ensemble (66.5%), followed by the NBhte ensemble (67.2%) in testing and training accuracy. The generalization performance of the NBE and NBhte ensembles highlights the possibility that both ensembles struggled with the multivariate input features in the Indian Liver dataset compared to other ensembles.

From Table 8.3, the GF of the RF algorithm indicates that the RF algorithm slightly overfitted the training data. The GFs of other ensembles indicate that no overfitting issue was experienced by the ensembles. It can be observed that the ensembles did not show drastic degradation in the training and testing performance.

In terms of F1-score, the NNhte and HTEdf are ranked as the best performing ensembles

offering a 72% F1-score, while the NBE and NBhte are ranked as the worst performing ensembles with a 58% F1-score respectively. It can be concluded that the HTEs and homogeneous ensembles performed well for the characteristics of the Indian Liver dataset consisting of a binary classification problem with an unbalanced class distribution, a small-moderate number of samples, a small number of features, and multiple input features types.

Table 8.3: Ensemble Results for Clean Indian Liver Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.66470 ± 0.11353	0.66115 ± 0.05071	0.98954	0.58
*k*NNE	0.75379 ± 0.12294	0.66009 ± 0.06401	0.72435	0.61
DTE	0.73636 ± 0.09747	0.68330 ± 0.05850	0.83244	0.70
RF	0.71470 ± 0.10223	0.75194 ± 0.05827	1.15015	0.71
SVME	0.76212 ± 0.06557	0.66297 ± 0.06023	0.70581	0.69
NNE	0.73545 ± 0.12536	0.70314 ± 0.05180	0.89114	0.71
NBhte	0.67152 ± 0.11184	0.66115 ± 0.05008	0.96942	0.58
*k*NNhte	0.72864 ± 0.13072	0.71670 ± 0.05210	0.95788	0.69
DThte	0.75333 ± 0.10135	0.72807 ± 0.05900	0.90710	0.70
SVMhte	0.74485 ± 0.01635	0.65662 ± 0.06616	0.74306	0.70
NNhte	0.75379 ± 0.11330	0.68855 ± 0.06705	0.79053	0.72
HTEsm	0.75970 ± 0.09717	0.70854 ± 0.05910	0.82448	0.71
HTEdf	0.76379 ± 0.07788	0.72702 ± 0.0481	0.86530	0.72

Clean Credit Approval Dataset

The classification problem is concerned with approved or declined requests for credit cards. Plots of the testing and training accuracy of the ensembles for the Credit Approval dataset are given in Figures 8.7 and 8.8. Table 8.4 provides the results of the testing and training accuracy, GF and F1-score of the ensembles for the clean Credit Approval dataset.

Illustrated by the training performance in Figure 8.7, the training accuracies of the ensembles showed that all ensembles are well trained except the NBE. The benefit of combining the advantage of different ML algorithms with different control parameter configurations is illustrated in the training performance of the HTEdf, achieving the highest training accuracy with 90.2%. The RF algorithm (89.2%) and HTEsm (89.1%) are respectively ranked as the second and third best-trained ensembles, while the NBE

performed poorly in training with an accuracy of 63.4%.

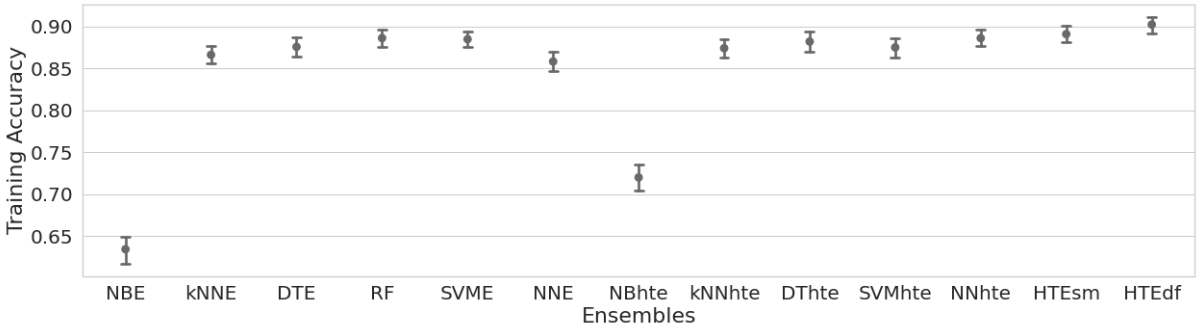

Figure 8.7: Training Accuracy of Ensembles for Clean Credit Approval Dataset

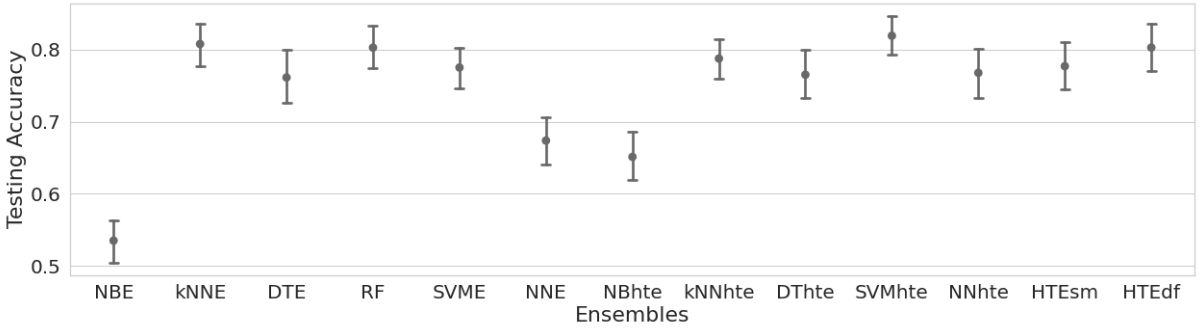

Figure 8.8: Testing Accuracy of Ensembles for Clean Credit Approval Dataset

In Figure 8.8, the SVMhte outperformed all ensembles highlighting the suitability of SVM algorithms to small datasets. The HTEdf is ranked as the second most accurate ensemble (81%), followed by the *k*NNE (80.7%), *k*NNhte (78.7%), and HTEsm (77.8%). The NBE is the least accurate ensemble, revealing the possibility that the NBE struggled with the multivariate input features in the Credit Approval dataset compared to other ensembles. The NBhte performed better than the NBE due to the benefit of the mixtures of heterogeneous experts obtained using different configurations for base NB learners in the NBhte.

While the SVMhte achieved the best testing accuracy of 82.2%, the training and generalization performance of the HTEdf illustrates the benefits of the combination of different algorithms and different configurations to learn characteristics of the credit approval dataset. On the other hand, despite being configured with base learners having the same control parameter values, the generalization performance of the *k*NNE indicates that the experts induced by the base learners were effective for the ensembles.

A further observation of the GFs of the ensembles showed that the NBE and NBhte slightly overfitted the training dataset compare to other ensembles, but resulted in poor generalization performance for the NBE. While a number of other ensembles showed more overfitting of the training dataset, the generalization performance of SVMhte, HTEdf and *k*NNE indicates that the ensembles are not adversely influenced by the problem of overfitting.

For the F1-score performance, the *k*NNhte achieved an 85% F1-score to be ranked as best performing ensemble, while the SVMhte (84%) is ranked second. The HTEdf and RF algorithm offer an equal F1-score of 83% to be ranked as the third-best performing ensembles. Thus, it can be concluded that it is beneficial to construct ensembles using a mixture of heterogeneous experts in comparison to homogeneous mixtures for this dataset.

Table 8.4: Ensemble Results for Clean Credit Approval Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.53484 ± 0.10246	0.63429 ± 0.05836	1.27194	0.45
*k*NNE	0.80780 ± 0.10413	0.86635 ± 0.03621	1.43806	0.80
DTE	0.76264 ± 0.12552	0.87714 ± 0.04161	1.93202	0.78
RF	0.77407 ± 0.12151	0.89238 ± 0.03994	2.09939	0.83
SVME	0.77670 ± 0.10249	0.88508 ± 0.03368	1.94305	0.82
NNE	0.67824 ± 0.11514	0.85619 ± 0.04505	2.23739	0.74
NBhte	0.65110 ± 0.12060	0.72000 ± 0.05860	1.24608	0.63
*k*NNhte	0.78758 ± 0.10416	0.87429 ± 0.03847	1.68969	0.85
DThte	0.77407 ± 0.11470	0.88254 ± 0.03927	1.92349	0.79
SVMhte	0.82209 ± 0.10439	0.87556 ± 0.04058	1.42965	0.84
NNhte	0.77231 ± 0.13059	0.88730 ± 0.03623	2.02037	0.80
HTEsm	0.77846 ± 0.11360	0.89143 ± 0.03410	2.04049	0.81
HTEdf	0.81000 ± 0.12202	0.90222 ± 0.03454	1.94318	0.83

Clean Red Wine Dataset

The goal is to predict the quality of red wines. The testing and training accuracies of the ensembles for this dataset are illustrated in Figures 8.9 and 8.10. Table 8.5 summarizes the results of the testing and training accuracy, GF and F1-score of the ensembles for the clean Red Wine dataset.

From Figure 8.9, the RF algorithm achieved the highest training performance with an accuracy of 86.2%, followed by the DThte (86%) and NNE (85.6%). The HTEdf is ranked as the fifth-best performing ensemble with a training accuracy of 83.7%. The training performance of the NBE (50.6%), NBhte (50.6%) and SVMhte (59.7%) showed that the ensembles did not train well on the dataset.

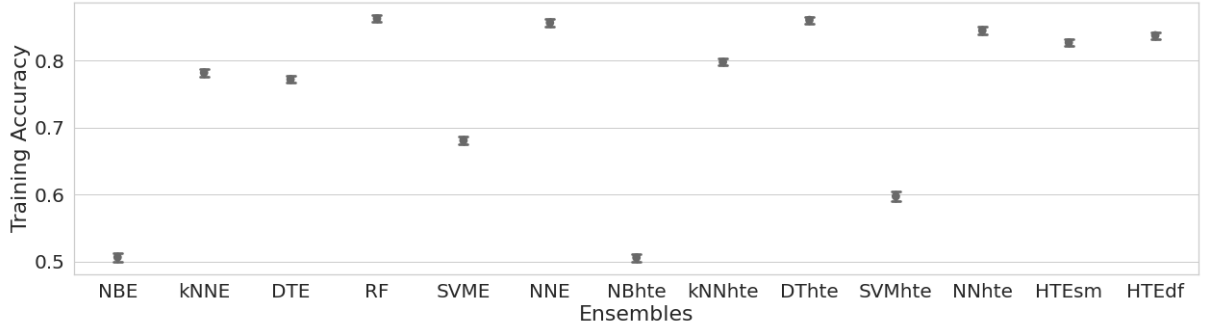

Figure 8.9: Training Accuracy of Ensembles for Red Wine Dataset

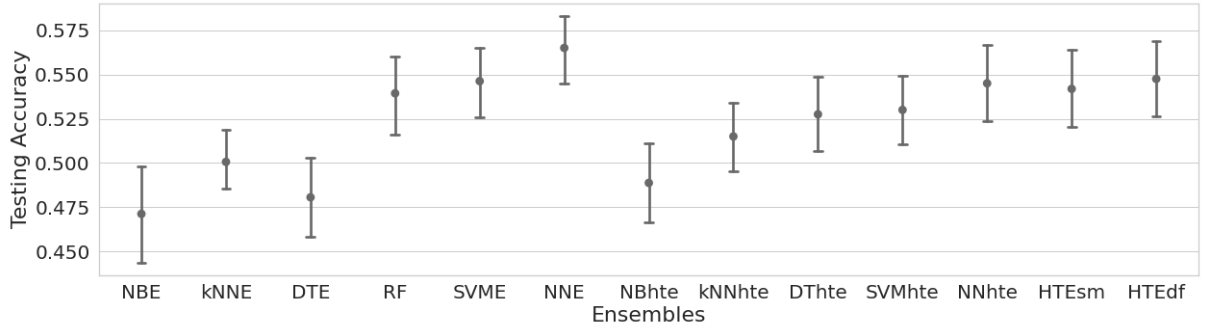

Figure 8.10: Testing Accuracy of Ensembles for Red Wine Dataset

As shown in Figure 8.10, the NNE is the most accurate ensemble (56.5%) in terms of testing accuracy, followed by the HTEdf (54.7%), SVME (54.6%), NNhte (54.5%), and HTEsm (54.1%). The NBE (47.1%) and NBhte (48.8%) are the least accurate ensembles. The generalization performance of the NNE indicates the suitability of NN algorithms to perform well on the characteristics of the Red Wine datasets with moderate sample and feature sizes, multiple classes, and continuous input features.

The NNhte also achieved a good level of testing accuracy due to the benefit of the mixtures of heterogeneous experts. The NBE is the least accurate ensemble (47.1%), followed by the DTE (48%).

Table 8.5: Ensemble Results for Clean Red Wine Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.47125 ± 0.09455	0.50660 ± 0.02228	1.07164	0.45
*k*NNE	0.50062 ± 0.06265	0.78197 ± 0.02046	2.29042	0.48
DTE	0.48063 ± 0.08122	0.77193 ± 0.01948	2.27727	0.51
RF	0.53938 ± 0.07650	0.86273 ± 0.01838	3.35569	0.60
SVME	0.54625 ± 0.06994	0.68082 ± 0.02155	1.42162	0.47
NNE	0.56500 ± 0.06756	0.85650 ± 0.01898	3.03139	0.58
NBhte	0.48875 ± 0.08472	0.50605 ± 0.02229	1.03503	0.47
*k*NNhte	0.51500 ± 0.06966	0.79789 ± 0.01889	2.39964	0.50
DThte	0.52750 ± 0.07847	0.86049 ± 0.01874	3.38684	0.61
SVMhte	0.53000 ± 0.06643	0.59751 ± 0.02526	1.16774	0.52
NNhte	0.54500 ± 0.07556	0.84488 ± 0.02198	2.93330	0.63
HTEsm	0.54187 ± 0.07970	0.82692 ± 0.01939	2.64691	0.59
HTEdf	0.54750 ± 0.07635	0.83763 ± 0.01805	2.78676	0.62

Illustrated by the GFs of the ensembles, the NBE, NBhte, SVME, and SVMhte slightly overfitted the training dataset, but resulted in low generalization performance for the NBE. On the other hand, all other ensembles showed more overfitting of the training dataset, but the effect of overfitting did not severely limit the generalization performance of the ensembles except for the DTE that achieved low generalization performance. For the F1-score performance, the NNhte outperformed other ensembles by achieving the highest F1-score of 63%. However, the difference between the F1-scores of the NNhte and HTEdf (62%) is just 1%.

Hence, it can be concluded that the homogeneous ensembles provided competitive performance with the HTEs across all performance measures to predict multi-class labels of the red wine quality samples in the Red Wine dataset.

Clean Car Evaluation Dataset

The problem is to predict one of the four classes representing the condition of a car in the dataset. Plots of the testing and training accuracies of the ensembles for this dataset are illustrated in Figures 8.11 and 8.12. Table 8.6 summarizes the results of the testing and training accuracy, GF and F1-score of the ensembles for the clean Car Evaluation dataset.

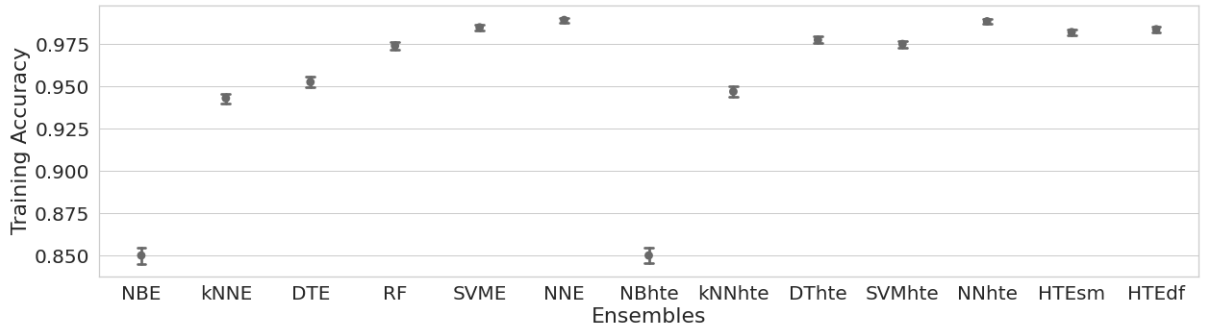

Figure 8.11: Training Accuracy of Ensembles for Car Evaluation Dataset

Illustrated in Figure 8.11, the NNE achieved the best training performance with an accuracy of 98.9%, followed by the NNhte (98.8%), SVME (98.5%), HTEdf (98.3%), and HTEsm (98.2%). The NBE and NBhte are ranked as the least trained ensemble offering 85% training accuracies respectively. Although, the training accuracies and the low standard deviations for the training dataset achieved by all ensembles provide evidence that the ensembles are well trained on the Car Evaluation dataset.

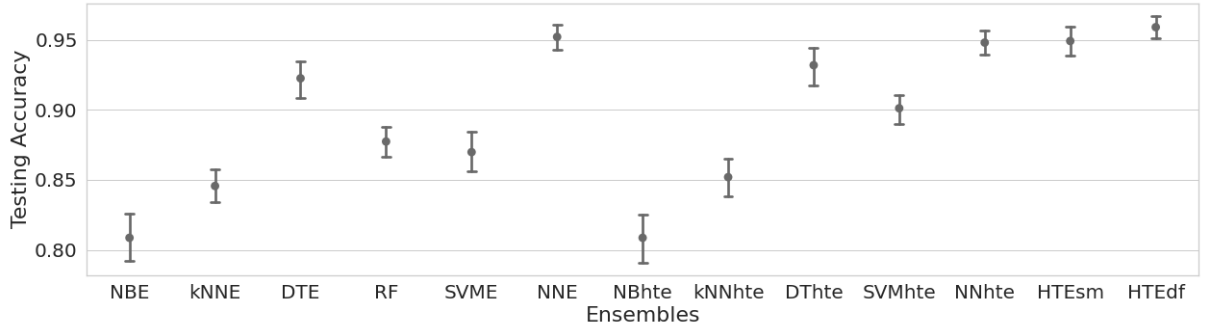

Figure 8.12: Testing Accuracy of Ensembles for Car Evaluation Dataset

In Figure 8.12, the HTEdf is the most accurate ensemble (95.8%), achieving the best stability over the independent runs as illustrated by the standard deviation of 0.02778 obtained. The NNE (95.2%) and HTEsm (94.9%) are the second and third most accurate ensembles. Also, the NBE and NBhte underperformed other ensembles with a testing accuracy of 80.8% respectively. The generalization performance of the HTEdf indicates the advantage of combining multiple instances of different ML algorithms with different configurations compared to other ensembles for the characteristics of the Car Evaluation dataset.

The GFs of the NBE, DTE, and NBhte illustrates that the ensembles slightly overfit the

training dataset, but the testing accuracies of the NBE and NBhte are adversely influenced by the problem of overfitting. Also, while other ensembles, except SVME, showed more overfitting of the training dataset, the SVME produced severe overfitting of the training dataset more than the ensembles.

Table 8.6: Ensemble Results for Clean Car Evaluation Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.80866 ± 0.06045	0.85000 ± 0.01682	1.27563	0.82
*k*NNE	0.84563 ± 0.04543	0.94308 ± 0.01090	2.71190	0.83
DTE	0.92245 ± 0.04730	0.95267 ± 0.01008	1.63830	0.96
RF	0.87734 ± 0.03900	0.97415 ± 0.00820	4.74560	0.95
SVME	0.86980 ± 0.05016	0.98503 ± 0.00627	8.69498	0.92
NNE	0.95200 ± 0.03110	0.98938 ± 0.00484	4.52174	0.98
NBhte	0.80866 ± 0.06045	0.85000 ± 0.01682	1.27563	0.82
*k*NNhte	0.85195 ± 0.04899	0.94703 ± 0.01073	2.79476	0.83
DThte	0.93182 ± 0.04916	0.97759 ± 0.00699	3.04257	0.95
SVMhte	0.90108 ± 0.03619	0.97518 ± 0.00761	3.98559	0.97
NNhte	0.94793 ± 0.03066	0.98862 ± 0.00526	4.57347	0.99
HTEsm	0.94906 ± 0.03683	0.98221 ± 0.00586	2.86269	0.98
HTEdf	0.95894 ± 0.02778	0.98390 ± 0.00571	2.54983	0.99

The generalization performance of the HTEdf and HTEsm indicates that the combination of multiple ML algorithms with the same and different control parameter configurations generated diverse experts that effectively learned and generalized well for the characteristics of the Car Evaluation dataset. Also, the generalization performance of the NNE and NNhte highlights that the ensembles produced a smooth interpolation among the training samples to achieve high generalization performances.

For the F1-score, the HTEdf and NNhte outperformed other ensembles by achieving a 99% F1-score, respectively. Although, other ensembles also performed well in terms of F1-score. Thus, the characteristics and complexity of the Car Evaluation dataset illustrate the suitability of the HTEs and homogeneous ensembles to the dataset. However, for all performance measures, it can be concluded that the HTEs generalizes well for the Car Evaluation dataset than the pure homogeneous ensembles.

Clean White Wine Dataset

The goal is to predict the quality of white wine. The prediction of wine quality labels for the White Wine dataset is the same as the Red wine dataset, except that the White Wine dataset contains an additional class label. Plots of the testing and training accuracies of the ensembles for this dataset are provided in Figures 8.13 and 8.14. Table 8.7 further summarizes the results of the testing and training accuracy, GF and F1-score of the ensembles for the clean White Wine dataset.

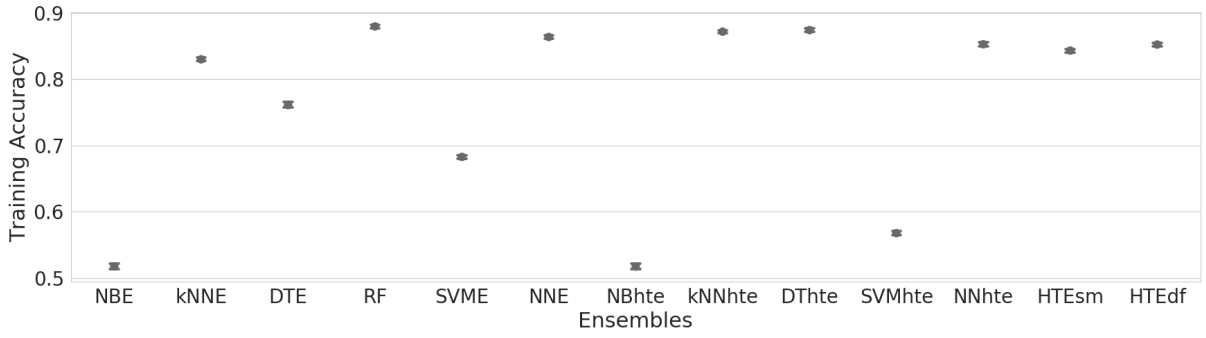

Figure 8.13: Training Accuracy of Ensembles for White Wine Dataset

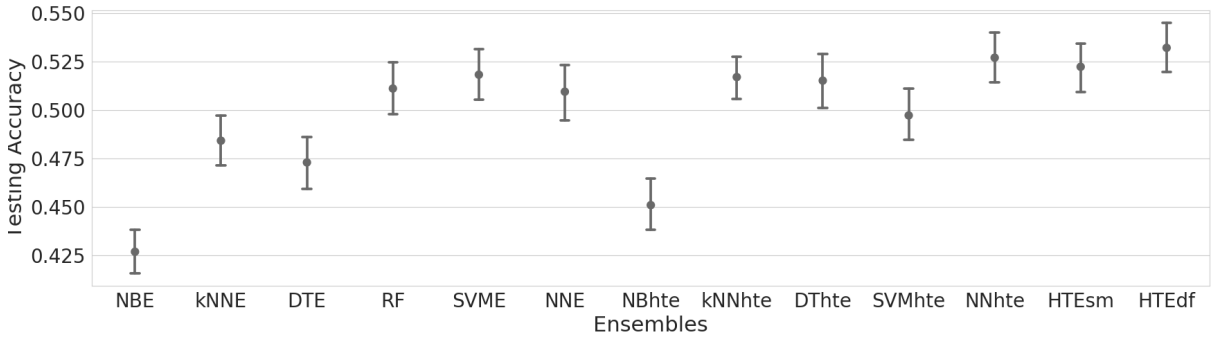

Figure 8.14: Testing Accuracy of Ensembles for White Wine Dataset

As shown in Figure 8.13 and 8.14, it can be observed that all ensembles reacted almost equivalently in training and prediction to the red wine dataset. However, while the ensembles performed better during training for the White Wine dataset, all the ensembles generated reduced testing accuracies compared to the Red Wine dataset. The training and testing performance of the ensembles are attributed to characteristics of the White Wine dataset consisting of more observations, an extra label value and a different distribution of the labels.

From Figure 8.13, the RF algorithm achieved the best training performance with an accuracy of (87.9%), followed by the DThte (87.4%), *k*NNhte (87.1%), NNE (86.3%), NNhte (85.27%), and HTEdf (85.22%). The NBE (51.7%) and NBhte (51.7%) struggled to effectively capture the trend in the dataset, and the SVMhte (56.7%) also provided the lowest training accuracy compared to other ensembles.

Table 8.7: Ensemble Results for Clean White Wine Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.42673 ± 0.04126	0.51754 ± 0.01431	1.18821	0.31
*k*NNE	0.48408 ± 0.04425	0.82999 ± 0.00850	3.03457	0.47
DTE	0.47286 ± 0.04888	0.76103 ± 0.01441	2.20592	0.45
RF	0.51102 ± 0.04675	0.87963 ± 0.00963	4.06236	0.58
SVME	0.51816 ± 0.04891	0.68273 ± 0.01040	1.51868	0.45
NNE	0.50939 ± 0.05036	0.86367 ± 0.00895	3.59862	0.55
NBhte	0.45082 ± 0.04767	0.51729 ± 0.01429	1.13771	0.32
*k*NNhte	0.51694 ± 0.03810	0.87178 ± 0.00747	3.76750	0.55
DThte	0.51510 ± 0.05408	0.87406 ± 0.00899	3.85031	0.60
SVMhte	0.49714 ± 0.04751	0.56777 ± 0.01323	1.16340	0.40
NNhte	0.52694 ± 0.04539	0.85270 ± 0.01117	3.21165	0.56
HTEsm	0.52224 ± 0.04621	0.84319 ± 0.00929	3.04677	0.57
HTEdf	0.53204 ± 0.04716	0.85228 ± 0.00782	3.16790	0.59

In Figure 8.14, the HTEdf outperformed other ensembles with a testing accuracy of 53.2%. The NNhte (52.6%) is the second most accurate ensemble, while the HTEsm (52.2%) is ranked as the third most accurate ensemble. The generalization performance of the HTEdf and HTEsm provides evidence that it is beneficial to combine heterogeneous experts induced from different ML algorithms consisting of the same and different control parameter configurations for this dataset. The NBE also performed poorly in prediction with testing accuracies of 42.6%, in which case the NBhte outperformed the NBE due to advantage of the mixtures of heterogeneous experts derived from the different configurations set for the base learners within the NBhte.

Illustrated by the GFs of the ensembles, the NBE and NBhte as well as the SVME and SVMhte slightly overfitted the training dataset, but resulted in a low generalization performance for the NBE and NBhte compared to other ensembles. On the other hand, the GFs of other ensembles indicates that the ensembles showed more overfitting of

the training dataset. The DThte (60%) slightly outperformed the HTEdf (59%) with a difference of 1% in terms of F1-score to be ranked as the best performing ensemble. The NBE and NBhte performed poorly in terms of F1-score. Therefore, it can be concluded that the HTEs trained and generalized better than the pure homogeneous ensembles (except for SVME and SVMhte) for the characteristics of the White Wine dataset consisting of a moderate number of samples, a small number of features, multi-class labels, and input features of the continuous type.

Clean Nursery Dataset

The goal is to predict one of five levels of recommendation for a nursery school admission application. The testing and training accuracy of the ensembles are illustrated in Figures 8.15 and 8.16. Table 8.8 summarizes the results of the testing and training accuracy, GF and F1-score of the ensembles for the clean Nursery dataset.

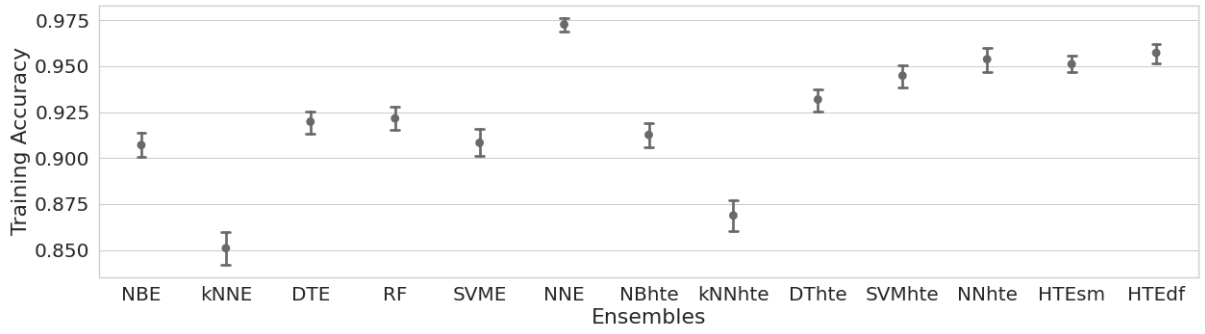

Figure 8.15: Training Accuracy of Ensembles for Nursery Dataset

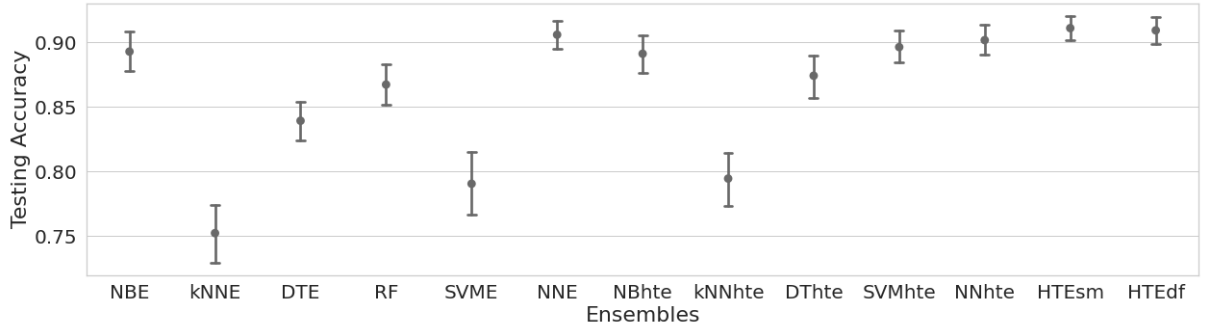

Figure 8.16: Testing Accuracy of Ensembles for Nursery Dataset

In Figure 8.15, the HTEdf is ranked as the best trained ensemble with a training accuracy of 95.7%. The NNhte (95.3%) and HTEsm (95.1%) are ranked as the second and third

best ensembles based on training accuracy. Although, other ensembles achieved efficient training performance and stability which is illustrated in the training accuracies and low standard deviation recorded by the ensembles in Table 8.8.

Table 8.8: Ensemble Results for Clean Nursery Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.89293 ± 0.05739	0.90703 ± 0.02403	1.15155	0.90
*k*NNE	0.75214 ± 0.08192	0.85092 ± 0.03237	1.66263	0.80
DTE	0.83928 ± 0.05480	0.91977 ± 0.02014	2.00338	0.93
RF	0.86736 ± 0.05732	0.92157 ± 0.02242	1.69132	0.93
SVME	0.79043 ± 0.08935	0.90826 ± 0.02662	2.28432	0.86
NNE	0.90601 ± 0.03924	0.97267 ± 0.01342	3.43858	0.97
NBhte	0.89127 ± 0.05543	0.91261 ± 0.02336	1.24416	0.90
*k*NNhte	0.79446 ± 0.07131	0.86872 ± 0.03097	1.56570	0.81
DThte	0.87420 ± 0.05927	0.93183 ± 0.02211	1.84532	0.94
SVMhte	0.89641 ± 0.04346	0.94479 ± 0.02200	1.87621	0.94
NNhte	0.90181 ± 0.04278	0.95375 ± 0.02378	2.12292	0.96
HTEsm	0.91112 ± 0.03505	0.95108 ± 0.01609	1.81682	0.97
HTEdf	0.90938 ± 0.03607	0.95719 ± 0.01814	2.11676	0.98

As shown in Figure 8.16, the HTEsm is ranked as the most accurate ensemble with a testing accuracy of 91.1%, achieving the best stability as illustrated by the lowest standard deviation of 0.03505. The HTEdf (90.9%) is the second most accurate ensemble, achieving the second-best stability with a standard deviation of 0.03607, and the NNE (90.6%) is the third most accurate ensembles. The *k*NNE (75.2%) provides the least testing accuracy, followed by the SVME (79.0%) and *k*NNhte (79.4%).

The generalization performance of the HTEdf and HTEsm shows that the combination of multiple instances of different ML algorithms generated different experts that effectively generalized well on the characteristics of the Nursery dataset. The generalization performance of the *k*NNE illustrates the possibility that the ensemble struggled with the characteristics of the Nursery dataset because *k*NN algorithms intrinsically give a preference to numeric features over categorical features. The *k*NNhte outperformed the *k*NNE due to the different configurations of the base *k*NN learners within the *k*NNhte resulting in a better generalization performance. On the other hand, the generalization performance of the SVME is explained by the inability of the base learners within the

SVME to handle a large number of samples in the Nursery dataset due to the complexity of the optimization equation to be solved by the base learners.

As illustrated by the GFs of the ensembles, the NNE produced more overfitting of the training dataset than other ensembles. Other ensembles slightly overfitted the training dataset. The F1-scores of the ensembles are competitive to show preciseness and robustness in prediction. The HTEdf achieved the highest F1-score of 98%, while the HTEsm and NNE offer 97% F1-score to be jointly ranked as the second-best performing ensembles. The characteristics of the Nursery dataset illustrate the suitability of the dataset for the HTEs and homogeneous ensembles based on the results of all performance measures.

Clean Bank Marketing Dataset

The problem describes the characteristics of a bank client to determine if the client makes a bank term deposit or not. Figures 8.17 and 8.18 illustrate the testing and training accuracies of the ensembles for the Bank Marketing dataset. The results of the testing and training accuracy, GF, and F1-score of the ensembles are summarized in Table 8.9.

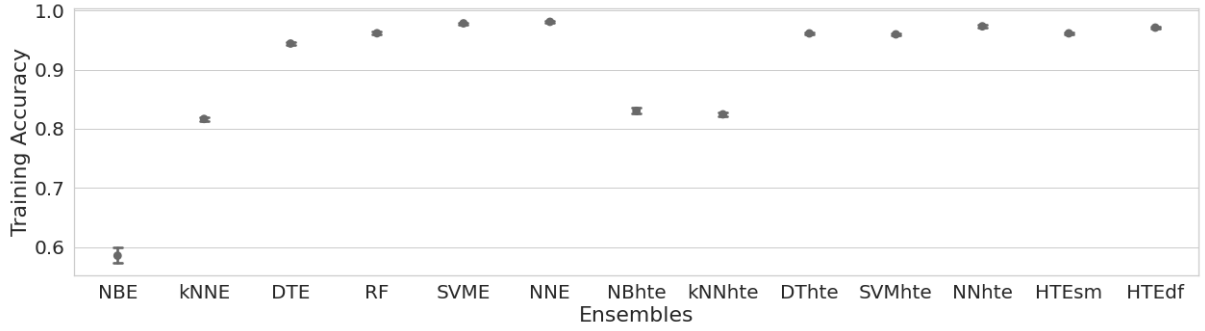

Figure 8.17: Training Accuracy of Ensembles for Clean Bank Marketing Dataset

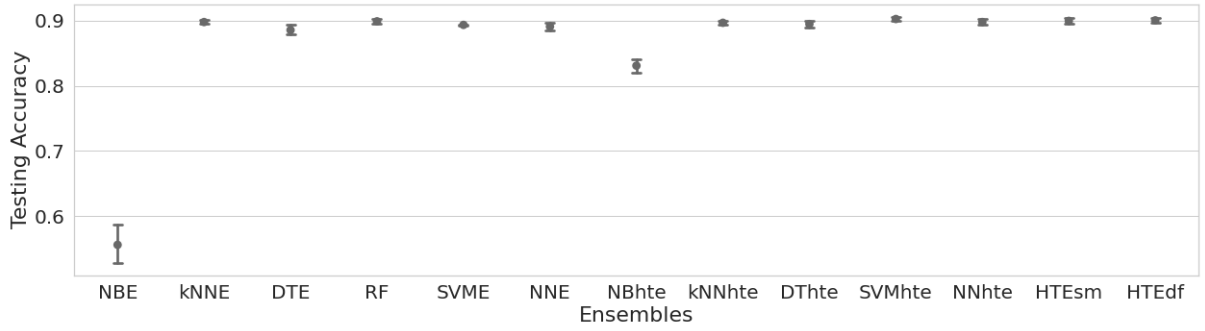

Figure 8.18: Testing Accuracy of Ensembles for Clean Bank Marketing Dataset

Table 8.9: Ensemble Results for Clean Bank Marketing Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.55560 ± 0.11109	0.58522 ± 0.04776	1.07140	0.66
*k*NNE	0.89806 ± 0.01133	0.81651 ± 0.01266	0.55557	0.69
DTE	0.88576 ± 0.02663	0.94405 ± 0.01005	2.04195	0.87
RF	0.89918 ± 0.01528	0.96151 ± 0.00882	2.61938	0.86
SVME	0.89369 ± 0.00314	0.97823 ± 0.00626	4.88388	0.84
NNE	0.89111 ± 0.01941	0.98082 ± 0.00660	5.67720	0.87
NBhte	0.83103 ± 0.03655	0.83022 ± 0.01691	0.99519	0.75
*k*NNhte	0.89693 ± 0.00907	0.82431 ± 0.01379	0.58666	0.70
DThte	0.89498 ± 0.02121	0.96121 ± 0.00841	2.70728	0.89
SVMhte	0.90274 ± 0.01235	0.95961 ± 0.00859	2.40810	0.88
NNhte	0.89806 ± 0.01577	0.97319 ± 0.00714	3.80239	0.88
HTEsm	0.89985 ± 0.01564	0.96134 ± 0.00814	2.59040	0.89
HTEdf	0.90065 ± 0.01490	0.97099 ± 0.00879	3.42483	0.90

Illustrated in Figure 8.17, the training accuracies of the ensembles provide evidence that all ensembles except the NBE trained well on the dataset. The NNE achieved the best training performance with an accuracy of 98%, followed by the SVME (97.8%), NNhte (97.3%), and HTEdf (97%). The NBE performed poorly in training with an accuracy of 58.2%.

In Figure 8.18, the SVMhte achieved a testing accuracy of 90.2% to be ranked as the most accurate ensemble. The HTEdf (90%) is the second most accurate ensemble, slightly underperforming the SVMhte with a difference of 0.2%. The HTEsm (89.98%) is ranked as the third most accurate ensemble. Also, the NBE significantly underperformed other ensembles with a testing accuracy of 55.5%.

The SVMhte outperformed the SVME due to the different configurations of the base learners in the SVMhte. The low generalization performance of the NBE is attributed to the possibility that the ensemble struggled with the characteristics of the Bank Marketing dataset compared to other ensembles.

Illustrated by the GFs of the ensembles, the NBE slightly overfitted the training dataset leading to poor training and generalization performance compared to othe ensembles. On the other hand, the *k*NNE, NBhte and *k*NNhte did not experience the problem of

overfitting, while other ensembles showed more overfitting of the training dataset.

In terms of F1-score, the HTEdf offers the highest F1-score of 90% to be rank as the best performing ensemble. Although, a number of HTEs, including HTEsm (89%), DThte (89%), SVMhte (88%), and NNhte (88%) achieved competitive F1-scores when compared to the HTEdf. Thus, considering the results of all performance measures, it can be concluded that the HTEs performed better than the pure homogeneous ensembles to train and make predictions for the characteristics of the Bank Marketing dataset.

Clean Censor Income Dataset

The task is to predict whether an adult earns more or less than $50,000 a year. Plots of the testing and training accuracies of the ensembles for the Censor Income dataset are given in Figures 8.19 and 8.20. Table 8.10 further summarizes the results of testing and training accuracy, testing and training error, and the GFs of the ensembles for the dataset.

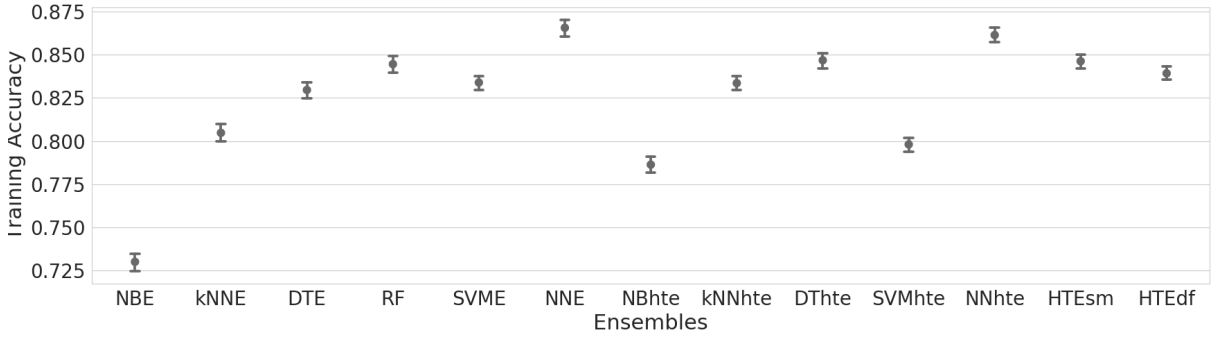

Figure 8.19: Training Accuracy of Ensembles for Clean Censor Income Dataset

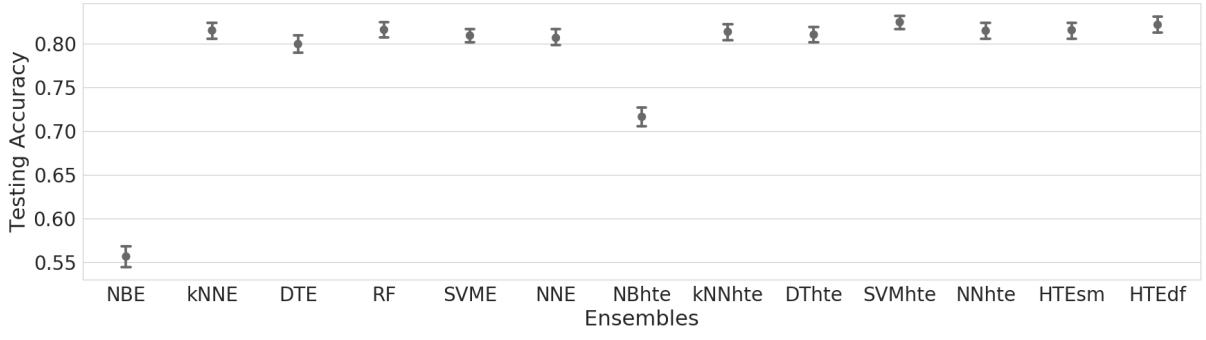

Figure 8.20: Testing Accuracy of Ensembles for Clean Censor Income Dataset

As shown in Figure 8.19, the NNE achieved the best training performance with an accuracy of 86.5%, followed by the NNhte (86.1%), DThte (84.67%), and HTEsm (84.62).

The NBE (73%) provided the worst training accuracy compared to other ensembles. The reliable training accuracies and low standard deviations of all ensembles provide evidence that the ensembles trained well to capture the characteristics of the Censor Income dataset.

Table 8.10: Ensemble Results for Clean Censor Income Dataset

Ensembles	Testing Accuracy	Training Accuracy	GF	F1-score
NBE	0.55642 ± 0.04335	0.73009 ± 0.01852	1.64346	0.74
*k*NNE	0.81527 ± 0.03381	0.80474 ± 0.01736	0.94609	0.76
DTE	0.79971 ± 0.03626	0.82957 ± 0.01693	1.17520	0.80
RF	0.81608 ± 0.03038	0.84451 ± 0.01785	1.18283	0.81
SVME	0.80941 ± 0.02628	0.83387 ± 0.01441	1.14723	0.78
NNE	0.80682 ± 0.03260	0.86555 ± 0.01622	1.43679	0.80
NBhte	0.71642 ± 0.03826	0.78633 ± 0.01563	1.32721	0.76
*k*NNhte	0.81378 ± 0.03332	0.83352 ± 0.01446	1.11861	0.78
DThte	0.81050 ± 0.03483	0.84675 ± 0.01584	1.23653	0.81
SVMhte	0.82484 ± 0.02776	0.79806 ± 0.01429	0.86741	0.79
NNhte	0.81501 ± 0.03201	0.86133 ± 0.01485	1.33401	0.80
HTEsm	0.81568 ± 0.03488	0.84622 ± 0.01480	1.19857	0.82
HTEdf	0.82183 ± 0.03272	0.83919 ± 0.01447	1.10796	0.83

From Figure 8.20, the SVMhte (82.4%) is ranked as the most accurate ensemble. The SVMhte outperforming the SVME highlights the benefit of the mixtures of heterogeneous experts achieved from the different configurations of the base learners within the SVMhte. In comparison to other ensembles, another possibility is related to the fact that the SVMhte performed well on the characteristics of the Censor Income dataset than other ensembles. The HTEdf (82.1%) is ranked as the second most accurate ensemble, while the RF algorithm (81.6%) and HTEsm (81.56%) are the third and fourth most accurate ensembles respectively. The advantage of the different ML algorithms within the HTEdf and HTEsm benefitted the predictability of the ensembles, while the generalization performance of the developed RF algorithm is due to the benefit of the intrinsic ensemble approaches (i.e. bagging and RFSM) implemented by RF algorithms.

The NBE offers the worst testing accuracy of 55.6%, illustrating the possibility that the ensemble struggled with the multivariate features in the Censor Income dataset because NB algorithms perform better on categorical features, while making assumptions for the

data distribution in numeric features.

Based on the GFs of the ensembles, the kNNE and SVMhte did not experience the problem of overfitting of the training dataset, while other ensembles slightly overfit the training dataset. Specifically, the problem of the slight overfitting adversely affect the generalization performance of the NBE.

In terms of F1-score, the HTEdf achieved the best F1-score of 84%, followed by the HTEsm (82%). The DThte and RF algorithm offer equal F1-score (81%) to be jointly ranked as the third-best performing ensembles. Thus, the HTEs performed better than the pure homogeneous ensembles for the characteristics of the Censor Income dataset to predict the binary labels of the adult income samples.

Statistical Analysis of Results

This section compares the generalization performance of the HTEs and homogeneous ensembles. The comparison is carried out to ascertain whether there exists a statistically significant difference in the performance of the ensembles. If a statistical difference exists, the ensembles that significantly differ are tested to verify the differences in the performance of the ensembles. The statistical tests used in this research were discussed in Section 7.8, including the Iman and Davenport extension of the Friedman test and Bonferroni-Dunn post hoc test. These statistical tests are used for all modelling studies.

Friedman Test

The Friedman test is used to compare the generalization performance of the 13 ensembles over the 10 datasets at a significance level $\alpha = 0.05$, which corresponds to a confidence level of 99.50%. As formally defined in Section 7.8.1, the null hypothesis of the Friedman test is that there is no significant difference between the generalization performance of the ensembles. Otherwise, the alternative hypothesis is selected.

The first step of the Friedman test is a ranking of the ensembles based on generalization performance for each dataset given in Table 8.11. The results in Table 8.11 (consisting of the testing accuracy and ranks of the ensembles for each dataset) provide the average rankings (AvR) of the ensembles, which shows an initial comparison of the ensembles in this modelling study and other modelling studies. Also, according to the Friedman test,

the ensemble presenting the lowest average rank across all datasets is the best performing ensemble.

The HTEdf is the best performing ensemble achieving the best average ranking of 1.70, followed by HTEsm (3.60), SVMhte (4.85), and NNhte (5.20). The average ranks of the HTEdf and HTEsm indicate the advantage of combining different ML algorithms to develop a mixture of heterogeneous experts. Another interesting outcome is that all HTEs were ranked better than the pure homogeneous ensembles, which illustrates that the different configurations used induced efficient base experts combined in heterogeneous mixtures compared to homogeneous mixtures.

Table 8.11: Ranking the Generalization Performance of Ensembles over Classification Datasets in the Clean Data Study

Ensemble	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor	AvR
NBE	74.7(9)	51.7(13)	66.4(13)	53.4(13)	47.1(13)	80.8(12.5)	42.6(13)	89.2(6)	55.5(13)	55.6(13)	**11.85**
*k*NNE	75.1(8)	66.6(1)	75.4(4.5)	80.7(3)	50.0(10)	84.5(11)	48.4(10)	75.2(13)	89.8(5.5)	81.52(5)	**7.10**
DTE	62.4(13)	56.2(11)	73.6(8)	76.2(10)	48.0(12)	92.2(6)	47.2(11)	83.9(10)	88.5(11)	79.9(11)	**10.30**
RF	73.9(10)	64.6(2)	71.4(11)	77.4(7.5)	53.9(6)	87.7(8)	51.1(7)	86.7(9)	89.91(4)	81.6(3)	**6.75**
SVME	65.6(12)	63.3(5.5)	76.2(2)	77.6(6)	54.6(3)	86.9(9)	51.8(4)	79.0(12)	89.3(9)	80.9(9)	**7.15**
NNE	75.3(7)	56.3(10)	73.5(9)	67.8(11)	56.5(1)	95.2(2)	50.9(8)	90.6(3)	89.1(10)	80.6(10)	**7.10**
NBhte	75.7(6)	51.8(12)	67.1(12)	65.1(12)	48.8(11)	80.8(12.5)	45.0(12)	89.1(7)	83.1(12)	71.6(12)	**10.85**
*k*NNhte	79.6(3)	63.5(4)	72.8(10)	78.7(4)	51.5(9)	85.1(10)	51.6(5)	79.4(11)	89.6(7)	81.3(7)	**7.00**
DThte	65.9(11)	57.5(8)	75.3(6)	77.4(7.5)	52.7(8)	93.1(5)	51.5(6)	87.4(8)	89.4(8)	81.0(8)	**7.55**
SVMhte	77.5(5)	63.3(5.5)	74.4(7)	82.2(1)	53.0(7)	90.1(7)	49.7(9)	89.6(5)	90.2(1)	82.4(1)	**4.85**
NNhte	78.2(4)	56.8(9)	75.4(4.5)	77.2(9)	54.5(4)	94.7(4)	52.6(2)	90.1(4)	89.8(5.5)	81.50(6)	**5.20**
HTEsm	80.5(2)	60.6(7)	75.9(3)	77.8(5)	54.1(5)	94.9(3)	52.2(3)	91.1(1)	89.98(3)	81.56(4)	**3.60**
HTEdf	81.6(1)	64.0(3)	76.3(1)	81.0(2)	54.7(2)	95.8(1)	53.2(1)	90.9(2)	90.0(2)	82.1(2)	**1.70**

Based on the average rankings of the ensembles in Table 8.11, the calculated Friedman test statistic χ_F^2 is 64.058, which is followed by the computation of the Iman-Davenport extension of the Friedman test F_F, which equals 10.305. F_F is distributed according to the F distribution of critical values with degrees of freedom equal to $(j-1) = 12$ and $(N-1) \times (j-1) = 108$. The critical value of $F(12, 108)$ for $\alpha = 0.05$ is 1.87. It is important to note that the computed degrees of freedom and critical value are used later in other modelling studies. Hence, because the value of F_F is greater than the obtained critical value, the null hypothesis that all ensembles are equal is rejected. The rejection of the null hypothesis indicates that, there is a statistically significant difference in the generalization performance of the ensembles.

Bonferroni-Dunn Test

The rejection of the null hypothesis resulted in performing a post hoc test using the Bonferroni-Dunn test. For the purpose of this research, which focuses on the development of mixtures of heterogeneous experts specifically from the combination of different ML algorithms, the HTEdf is selected as the control ensemble for the Bonferroni-Dunn test across all modelling studies. The HTEdf was selected as the control ensemble because the HTEdf maximizes behavioural diversity to obtain two benefits from the mixtures of heterogeneous experts. The first benefit is achieved by capitalizing on the inductive biases of different ML algorithms intrinsically. The second benefit takes advantage of using different control parameter configurations for the multiple instances of the different ML algorithms combined within the HTEdf. The rank of the HTEdf is compared with the rank achieved by other ensembles. The critical value, q_α, associated with the two-tailed Bonferroni-Dunn test at the significance level of $\alpha = 0.05$ with 13 ensembles is 2.87, and the computed critical difference (CD) is 4.998. The computed critical value of 2.87 and the CD value of 4.998 are also used later in other modelling studies.

For the Bonferroni-Dunn test, a significant difference is detected between the HTEdf and any ensemble if the difference between the average rank of the HTEdf and the ensemble is greater than the computed CD of 4.998. Figure 8.21 presents the critical difference plot of the significant difference in generalization performance between the HTEdf and any ensemble. The ranks outside the black marked interval indicate a significant difference; otherwise, no significant difference is considered. It is important to note that this decision rule to detect a significant difference in the Bonferroni-Dunn test and the critical difference plot is used later in other modelling studies.

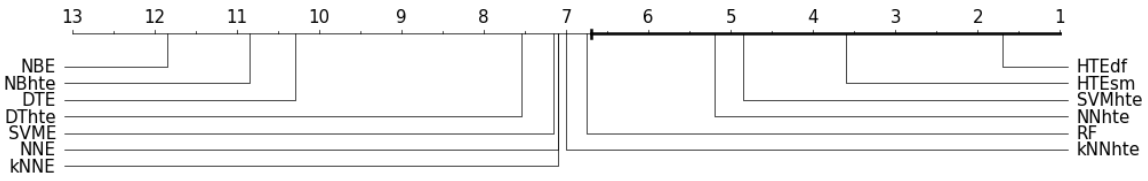

Figure 8.21: Critical Difference Plot of Ensembles for Clean Data Study in Classification Problems

As illustrated in Figure 8.21, the outcome of the Bonferroni-Dunn test showed that the HTEdf is significantly more accurate than the pure homogeneous ensembles (i.e. NBE,

*k*NNE, DTE, SVME, NNE), RF, NBhte, *k*NNhte, and DThte. On the other hand, the 10 experimental datasets did not provide sufficient evidence to show that a significant difference in generalization performance exists between the HTEdf and HTEsm, SVMhte, and NNhte.

Furthermore, it can be observed that the difference in average ranks between the HTEdf and RF is just a little above the CD = 4.998, (i.e. 6.75-1.70 = 5.05) but close to it. This is attributed to the possibility that the developed RF capitalizes on the benefit of the intrinsic ensemble approaches (i.e. bagging and RFSM) used in RF algorithms (Breiman, 2001; Bernard et al., 2009). Also, the HTEdf is significantly not different from the HTEsm because both ensembles were developed using multiple instances of different ML algorithms. In addition, the Bonferroni-Dunn test showed that there is a significant difference in generalization performance between homogeneous and heterogeneous mixtures of experts.

8.3 Skewed Class Distributions Study

This section discusses the performance of the ensembles on skewed class distributions considered from 10-90%, 15-85%, 20-80%,..., 45-55% to 50-50% in the training datasets. The summative results (over all skewness ratios) of the testing and training accuracy, GF and F1-score for each ensemble over all classification datasets are shown in Table 8.12. The best generalization performance and F1-score for each dataset are bolded, while the second-best performance for these performance measures is underlined for this modelling study and other modelling studies except the clean data study. The confusion matrices of the ensembles are provided in Appendix A to show the ensembles that achieved the highest generalization performance for the prediction of the minority class(es) in each dataset.

By observing the results in Table 8.12, the advantage of the mixtures of heterogeneous experts is evident in the overall generalization performance and F1-score of the HTEs over the homogeneous mixtures of the pure homogeneous ensembles. The HTEdf achieved the highest overall generalization performance on seven out of the 10 datasets with respect to the overall imbalanced ratios. Based on the complexity of the seven datasets to predict binary and multi-class labels, the HTEdf achieved the best overall generalization

performance for three of the six binary problems (Sonar, Indian and Credit datasets) and all multi-class classification problems (i.e. Red Wine, Car Evaluation, White Wine, and Nursery datasets). The *k*NNE offered the highest testing accuracy of 66.6% for the Breast

Table 8.12: Ensemble Results over all Classification Datasets in Skewed Class Distribution Study

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBE										
Testing accuracy	0.747	0.517	0.665	0.535	0.471	0.809	0.427	0.846	0.556	0.533
Training accuracy	0.755	0.782	0.822	0.537	0.513	0.863	0.513	0.927	0.798	0.851
GF	1.606	3.371	2.112	1.010	1.104	1.420	1.181	2.263	2.638	3.557
F1-Score	0.626	0.428	0.428	0.493	0.242	0.820	0.252	0.854	0.460	0.662
*k*NNE										
Testing accuracy	0.586	**0.666**	0.743	0.788	0.501	0.846	0.469	0.806	0.897	0.809
Training accuracy	0.672	0.732	0.787	0.829	0.608	0.870	0.622	0.850	0.819	0.865
GF	1.587	1.497	1.440	1.290	1.292	1.195	1.431	1.336	0.633	1.513
F1-Score	0.622	0.467	0.467	**0.832**	0.352	0.773	0.373	0.878	0.540	0.694
DTE										
Testing accuracy	0.630	0.566	0.739	0.759	0.486	0.923	0.475	0.897	0.886	0.784
Training accuracy	0.684	0.809	0.753	0.855	0.604	0.936	0.633	0.918	0.953	0.884
GF	1.240	4.457	1.154	2.073	1.327	1.200	1.472	1.284	2.761	2.374
F1-Score	0.664	0.510	0.510	0.718	0.399	0.888	0.364	0.881	0.784	0.671
RF										
Testing accuracy	0.709	0.625	0.736	0.792	0.537	0.883	0.523	0.895	0.899	0.801
Training accuracy	0.744	0.811	0.788	0.865	0.652	0.941	0.672	0.923	0.969	0.901
GF	1.306	3.541	1.440	1.865	1.348	2.018	1.486	1.381	3.443	2.836
F1-Score	0.659	0.461	0.461	0.637	0.422	0.848	0.394	0.868	0.841	0.694
SVME										
Testing accuracy	0.728	0.630	0.764	0.776	0.531	0.871	0.506	0.856	0.894	0.803
Training accuracy	0.744	0.709	0.823	0.814	0.568	0.959	0.633	0.939	0.973	0.874
GF	1.268	1.533	1.529	1.491	1.122	3.254	1.401	2.430	4.231	1.687
F1-Score	0.590	0.420	0.418	0.686	0.392	0.924	0.362	0.922	0.839	0.731
NNE										
Testing accuracy	0.765	0.554	0.731	0.674	0.555	0.952	0.516	0.940	0.891	0.801
Training accuracy	0.730	0.745	0.803	0.791	0.680	0.972	0.692	0.967	0.974	0.881
GF	0.898	1.963	1.591	1.718	1.406	1.988	1.603	1.921	4.627	2.112
F1-Score	0.657	0.469	0.469	0.720	0.443	0.959	0.414	0.943	0.841	0.730

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBhte										
Testing accuracy	0.762	0.518	0.672	0.651	0.489	0.809	0.450	0.851	0.831	0.703
Training accuracy	0.761	0.793	0.826	0.615	0.517	0.865	0.528	0.932	0.878	0.897
GF	1.554	3.835	2.124	0.916	1.079	1.457	1.177	2.406	2.241	3.505
F1-Score	0.592	0.442	0.442	0.554	0.236	0.824	0.234	0.856	0.533	0.703
*k*NNhte										
Testing accuracy	0.655	0.635	0.739	0.808	0.515	0.852	0.491	0.831	0.897	0.808
Training accuracy	0.656	0.735	0.783	0.820	0.604	0.882	0.629	0.868	0.803	0.866
GF	1.173	1.714	1.450	1.086	1.243	1.265	1.395	1.336	0.581	1.554
F1-Score	0.681	0.480	0.480	**0.833**	0.380	0.769	0.389	0.891	0.480	0.689
DThte										
Testing accuracy	0.671	0.591	0.756	0.762	0.530	0.935	0.521	0.912	0.895	0.805
Training accuracy	0.744	0.808	0.793	0.868	0.651	0.950	0.671	0.936	0.966	0.895
GF	1.489	4.278	1.347	2.307	1.372	1.323	1.485	1.419	3.613	2.414
F1-Score	0.649	0.527	0.527	0.723	0.437	0.907	0.393	0.909	0.801	0.682
SVMhte										
Testing accuracy	0.698	0.632	0.748	0.818	0.549	0.903	0.507	0.922	**0.903**	**0.825**
Training accuracy	0.738	0.713	0.817	0.840	0.614	0.930	0.563	0.972	0.937	0.879
GF	1.392	1.596	1.589	1.218	1.204	1.438	1.150	2.930	1.948	1.585
F1-Score	0.540	0.391	0.391	0.786	0.334	0.889	0.272	0.946	0.853	0.716
NNhte										
Testing accuracy	0.760	0.579	0.761	0.766	0.554	0.948	0.529	0.944	0.899	0.811
Training accuracy	0.735	0.746	0.826	0.838	0.665	0.970	0.665	0.974	0.965	0.882
GF	1.132	1.982	1.547	1.594	1.343	1.977	1.418	2.278	3.019	1.885
F1-Score	**0.796**	0.531	0.531	0.673	**0.471**	**0.961**	0.429	0.948	**0.860**	0.726
HTEsm										
Testing accuracy	0.769	0.644	0.750	0.811	0.548	0.952	0.519	0.957	0.897	0.802
Training accuracy	0.728	0.769	0.822	0.849	0.652	0.959	0.670	0.977	0.960	0.890
GF	1.012	2.194	1.609	1.412	1.320	1.220	1.483	1.953	2.808	2.042
F1-Score	0.761	0.549	0.539	0.802	0.449	0.946	0.477	0.952	0.812	0.724
HTEdf										
Testing accuracy	**0.775**	0.645	**0.767**	**0.814**	**0.557**	**0.955**	**0.531**	**0.964**	0.901	0.813
Training accuracy	0.738	0.773	0.821	0.867	0.640	0.963	0.664	0.983	0.966	0.897
GF	1.000	2.177	1.489	1.709	1.254	1.287	1.416	2.159	3.148	2.090
F1-Score	0.768	**0.572**	**0.562**	0.822	0.457	0.959	**0.482**	**0.963**	0.844	**0.733**

Cancer dataset, which is one of the remaining three binary classification datasets.

The SVMhte outperformed other ensembles for the Bank Marketing and Censor Income datasets. It can be observed that the HTEdf is ranked as the second most accurate ensemble for the remaining three binary classification datasets (i.e. Breast Cancer, Bank Marketing, and Censor Income datasets). The HTEsm, NNhte, SVMhte, and NNE also achieved the second-best testing performance in a number of datasets.

The NBE and NBhte are ranked as the least performing ensembles based on generalization performance for eight of the 10 datasets, followed by the *k*NNE and *k*NNhte. Also, in terms of generalization performance, it can be observed that the NBhte outperformed the NBE, and *k*NNhte performed better than *k*NNE over the eight datasets. This is attributed to the different control parameter configurations within the NBhte and *k*NNhte. This trend in performance is also observed for the other ensembles, i.e. DTE and DThte, SVME and SVMhte, as well as NNE and NNhte for most of the datasets.

Also, from the results in Appendix A, the ensembles showed different prediction behaviour based on different skewed class distributions in the training datasets over all datasets. The skewed classes are classified into extreme class distributions (10-90%, 15-85%, 20-80%), mild class distributions (25-75%, 30-70%, 35-65%), small class distributions (40-60%, 45-55%), and balanced class distribution (50-50%). The results showed that HTEdf is the most accurate ensemble over all the class distributions for five of the 10 datasets (i.e. Sonar, Indian Liver, Car Evaluation, White Wine, and Nursery dataset). While the NNE performed best for extreme class distribution, the HTEdf offered the best generalization performance from the mild to balanced class distributions for the Red Wine dataset. For all categories of the class distributions, the *k*NNE outperformed other ensembles for the Breast Cancer dataset, while the SVMhte is the most accurate ensemble for the Credit Approval, Bank Marketing, and Censor Income datasets.

Based on training performance, it is observed that all ensembles achieved competitive overall training accuracy to capture the relationship between the input features and target labels over all datasets. However, the GF of the NBE on the Credit Approval dataset illustrates that the ensemble did not experience the problem of overfitting, but still produced worst generalization performance. The GFs of other ensembles highlight that the ensembles slightly overfitted the training dataset across the classification datasets,

except the *k*NNE, *k*NNhte, NNE, and HTEdf. The *k*NNE and *k*NNhte showed no indication of overfitting the training dataset on the Bank Marketing dataset, while the NNE and HTEdf produced no evidence of overfitting for the Sonar dataset.

Further, the HTEdf expresses the effectiveness of classification by achieving the highest F1-score over five datasets, illustrating that the HTEdf correctly classified the minority and majority samples in the experimental data compared to other ensembles. The NNhte offered the best F1-score on four datasets, and the *k*NNhte performed best on the last dataset, i.e. the Credit Approval dataset. An interesting observation is that, while the NNhte achieved improved F1-score performance, the NNhte tends to sacrifice an overall generalization performance. However, this is not the case with the HTEdf, because the HTEdf performed excellently for both performance measures. The HTEsm also produced a reasonable level of performance in terms of F1-score.

Therefore, based on the results of all performance measures, the HTEdf and HTEsm illustrate the benefit of combining different ML algorithms to develop a mixture of heterogeneous experts. Specifically, by capitalizing on different control parameter configurations for the multiple instances of the different ML algorithms, the HTEdf performed excellently on multi-class classification problems and competed well against other ensembles for binary classification problems in the skewed class distribution study.

Generalization Performance of Ensembles on the Minority Class

Research has shown that using classification accuracy to evaluate the overall performance of ensembles does not adequately reveal the true prediction performance of the ensembles for imbalanced datasets considering the majority (negative) and minority (positive) classes (Veropoulos et al., 1999; Wu, 2003; Akbani et al., 2004). For instance, with a skewed class distribution of 10-90%, an ensemble that classifies all samples negative will be 90% accurate, but does not generalize accurately on the test dataset because the minority class (i.e. positive class) is significantly misclassified or ignored.

The F1-score discussed earlier has shown how the ensembles performed fairly on the minority class. The confusion matrices of the ensembles provided in Appendix A and the results in Tables 8.13 to 8.22 also showed the generalization performance of the ensembles on the minority class. For binary classification problems, the generalization performance

of the ensembles on the minority class across all class distributions is provided and the average rank of the ensembles are computed. This is shown in Tables 8.13, 8.14, 8.15, 8.16, 8.21, and 8.22.

Table 8.13: Ranking the Generalization Performance of the Ensembles on Minority Class for Sonar Dataset

Ensemble	**Skewed Classes %**									**AvR**
	10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50	
NBE	100(6.5)	100(5)	100(4.5)	100(5)	94(9)	80(13)	94(5.5)	88(6)	88(6)	**6.72**
*k*NNE	94(13)	94(11)	94(10.5)	88(12.5)	88(12.5)	81(12)	69(12.5)	69(12)	62(11.5)	**11.94**
DTE	100(6.5)	94(11)	81(13)	94(11)	94(9)	100(3.5)	69(12.5)	75(11)	56(13)	**10.06**
RF	100(6.5)	84(13)	94(10.5)	100(5)	94(9)	94(7.5)	94(5.5)	81(8.5)	69(9.5)	**8.33**
SVME	100(6.5)	100(5)	100(4.5)	100(5)	100(3.5)	100(3.5)	94(5.5)	94(4.5)	94(3.5)	**4.61**
NNE	100(6.5)	100(5)	100(4.5)	98(10)	100(3.5)	88(9.5)	88(8.5)	81(8.5)	88(6)	**6.89**
NBhte	100(6.5)	100(5)	100(4.5)	100(5)	94(9)	88(9.5)	94(5.5)	94(4.5)	88(6)	**6.17**
*k*NNhte	100(6.5)	100(5)	94(10.5)	88(12.5)	88(12.5)	82(11)	81(10.5)	62(13)	69(9.5)	**10.11**
DThte	100(6.5)	94(11)	94(10.5)	100(5)	94(9)	100(3.5)	81(10.5)	81(8.5)	62(11.5)	**8.44**
SVMhte	100(6.5)	100(5)	100(4.5)	100(5)	100(3.5)	100(3.5)	100(2)	100(2)	81(8)	**4.44**
NNhte	100(6.5)	100(5)	100(4.5)	100(5)	100(3.5)	94(7.5)	88(8.5)	81(8.5)	94(3.5)	**5.83**
HTEsm	100(6.5)	100(5)	100(4.5)	100(5)	100(3.5)	100(3.5)	100(2)	100(2)	96(1.5)	**3.72**
HTEdf	100(6.5)	100(5)	100(4.5)	100(5)	100(3.5)	100(3.5)	100(2)	100(2)	96(1.5)	**3.72**

Table 8.14: Ranking the Generalization Performance of the Ensembles on Minority Class for Breast Cancer Dataset

Ensemble	**Skewed Classes %**									**AvR**
	10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50	
NBE	95(8.5)	86(10.5)	95(4.5)	86(6)	95(1.5)	86(2.5)	95(1.5)	86(1)	95(1.5)	**4.17**
*k*NNE	95(8.5)	95(7)	62(11)	76(8.5)	52(10)	52(9.5)	38(11.5)	52(7)	29(10)	**9.22**
DTE	95(8.5)	90(9)	76(10)	71(10)	57(9)	62(7)	48(8)	52(7)	48(7)	**8.39**
RF	95(8.5)	100(3)	95(4.5)	76(8.5)	71(7)	57(8)	43(9.5)	52(7)	52(5)	**6.78**
SVME	100(3)	95(7)	90(8)	95(2)	89(3)	86(2.5)	62(6)	38(10.5)	14(12)	**6.00**
NNE	76(13)	43(13)	45(13)	38(12)	33(13)	24(13)	29(8.5)	29(12)	24(11)	**12.06**
NBhte	95(8.5)	86(10.5)	95(4.5)	86(6)	81(4)	76(6)	86(3)	81(2)	95(1.5)	**5.11**
*k*NNhte	86(12)	76(12)	57(12)	52(11)	48(11)	48(11)	43(9.5)	38(10.5)	33(8.5)	**10.83**
DThte	95(8.5)	95(7)	95(4.5)	86(6)	67(8)	52(9.5)	52(7)	48(9)	57(3)	**6.94**
SVMhte	100(3)	100(3)	100(1)	100(1)	95(1.5)	95(1)	95(1.5)	57(4)	10(13)	**3.22**
NNhte	100(3)	100(3)	81(9)	33(13)	43(12)	29(12)	38(11.5)	24(13)	33(8.5)	**9.44**
HTEsm	100(3)	100(3)	95(4.5)	88(4)	75(6)	78(5)	65(5)	55(5)	50(6)	**4.61**
HTEdf	100(3)	100(3)	95(4.5)	89(3)	78(5)	80(4)	67(4)	60(3)	55(4)	**3.72**

Table 8.15: Ranking the Generalization Performance of the Ensembles on Minority Class for Indian Liver Dataset

Ensemble	Skewed Classes %									AvR
	10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50	
NBE	100(5.5)	100(5)	100(4)	100(4.5)	100(2)	100(2)	97(4)	100(1.5)	97(2)	**3.39**
*k*NNE	100(5.5)	100(5)	97(8.5)	100(4.5)	97(5.5)	97(5)	90(9.5)	83(9.5)	77(8.5)	**6.83**
DTE	97(12)	93(10)	77(13)	83(13)	93(10)	70(13)	80(13)	70(12)	57(12)	**12.00**
RF	97(12)	90(12)	90(12)	90(11.5)	90(13)	87(11)	90(9.5)	67(13)	57(12)	**11.78**
SVME	100(5.5)	100(5)	100(4)	100(4.5)	93(10)	93(8)	93(6.5)	93(6)	93(5)	**6.06**
NNE	97(12)	90(12)	93(11)	90(11.5)	93(10)	90(10)	87(12)	80(11)	70(10)	**11.06**
NBhte	100(5.5)	100(5)	100(4.5)	100(4.5)	100(2)	100(2)	100(1)	100(1.5)	97(2)	**3.11**
*k*NNhte	100(5.5)	100(5)	94(10)	100(4.5)	97(5.5)	93(8)	90(9.5)	83(9.5)	77(8.5)	**7.33**
DThte	100(5.5)	90(12)	97(8.5)	93(10)	93(10)	83(12)	97(4)	87(8)	57(12)	**9.11**
SVMhte	100(5.5)	100(5)	100(4)	100(4.5)	100(2)	100(2)	93(6.5)	93(6)	83(7)	**4.72**
NNhte	100(5.5)	100(5)	100(4)	97(9)	93(10)	93(8)	90(9.5)	93(6)	90(6)	**7.00**
HTEsm	100(5.5)	100(5)	100(4)	100(4.5)	97(5.5)	97(5)	97(4)	95(4)	95(4)	**4.61**
HTEdf	100(5.5)	100(5)	100(4)	100(4.5)	97(5.5)	97(5)	98(2)	97(3)	97(2)	**4.06**

Table 8.16: Ranking the Generalization Performance of the Ensembles on Minority Class for Credit Approval Dataset

Ensemble	Skewed Classes %									AvR
	10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50	
NBE	56(4)	56(5.5)	60(5.5)	59(8)	59(7)	60(12)	59(12.5)	57(12)	63(11)	**8.61**
*k*NNE	74(1)	79(1)	80(1)	80(1)	80(1)	80(3)	85(2)	84(3)	81(6)	**2.11**
DTE	31(10)	39(10)	41(10.5)	53(10)	50(10.5)	66(8)	73(7)	67(10)	76(8.5)	**9.39**
RF	3(13)	7(13)	19(13)	40(13)	41(13)	67(6.5)	67(8)	81(6)	76(8.5)	**10.44**
SVME	13(12)	21(12)	36(12)	41(12)	44(12)	61(10.5)	66(9)	71(8)	81(6)	**10.39**
NNE	49(7)	56(5.5)	57(7.5)	61(6.5)	57(8.5)	61(10.5)	63(10.5)	63(11)	60(12.5)	**8.83**
NBhte	50(6)	54(7)	56(9)	57(9)	57(8.5)	59(13)	59(12.5)	54(13)	60(12.5)	**10.06**
*k*NNhte	63(2)	71(2)	74(2)	79(2)	79(2)	83(2)	83(3)	83(4)	85(4)	**2.56**
DThte	33(9)	36(11)	41(10.5)	47(11)	50(10.5)	63(9)	74(6)	80(7)	81(6)	**8.89**
SVMhte	19(11)	51(9)	66(3)	74(3)	78(3)	84(1)	87(1)	87(1)	87(2)	**3.78**
NNhte	44(8)	53(8)	60(5.5)	63(5)	61(6)	70(5)	63(10.5)	69(9)	74(10)	**7.44**
HTEsm	53(5)	57(4)	57(7.5)	61(6.5)	63(5)	67(6.5)	75(5)	82(5)	86(3)	**5.28**
HTEdf	57(3)	59(3)	62(4)	65(4)	67(4)	75(4)	77(4)	85(2)	89(1)	**3.22**

Table 8.17: Ranking the Generalization Performance of the Ensembles on Minority Class for Red Wine Dataset

Label	Skewed Classes %										
	10-90	15-85	20-80	25-75	30-70	35-65	40-60	45-55	50-50	**AvR**	**M.AvR**
NBE											
3	100(1.5)	100(2)	100(1.5)	100(1.5)	100(1.5)	100(1.5)	100(1.5)	100(1.5)	100(1.5)	1.56	
4	0(13)	0(12)	0(13)	0(12.5)	0(12.5)	0(12.5)	0(12)	0(12.5)	0(12.5)	12.5	**6.15**
8	60(2.5)	60(3.5)	60(4.5)	60(3.5)	60(4.5)	60(5)	60(4.5)	60(6.5)	60(5)	4.39	
***k*NNE**											
3	0(9)	0(9.5)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	9.06	
4	50(6)	60(4)	50(3.5)	40(8)	50(7.5)	50(5.5)	40(7)	50(7.5)	50(7)	6.22	**7.93**
8	40(8)	40(9.5)	40(9)	40(7.5)	40(8)	40(9.5)	40(9)	60(6.5)	40(9.5)	8.50	
DTE											
3	0(9)	0(9.5)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	9.06	
4	20(10.5)	30(9.5)	20(11.5)	30(10.5)	20(11)	20(10.5)	0(12)	50(7.5)	60(3.5)	9.61	**9.76**
8	40(8)	40(9.5)	40(9)	0(12.5)	20(11.5)	20(12)	40(9)	20(12.5)	20(11.5)	10.61	
RF											
3	0(9)	0(9.5)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	9.06	
4	50(6)	40(8)	30(9)	40(8)	60(4)	30(8.5)	20(9)	60(3)	40(9.5)	7.22	**9.35**
8	20(12)	0(13)	40(9)	0(12.5)	20(11.5)	20(12)	20(12)	40(11)	0(13)	11.78	
SVME											
3	0(9)	0(9.5)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	9.06	
4	50(6)	50(6.5)	30(9)	60(1.5)	60(4)	70(1.5)	60(2)	60(3)	60(3.5)	4.11	**5.06**
8	60(2.5)	60(3.5)	80(1.5)	80(1)	80(1.5)	80(1)	80(1)	80(1)	60(5)	2.00	
NNE											
3	0(9)	0(9.5)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	9.06	
4	70(1)	70(1.5)	70(1)	50(4.5)	80(1)	70(1.5)	50(5)	50(7.5)	60(3.5)	2.94	**7.85**
8	20(12)	40(9.5)	20(12.5)	20(10.5)	20(11.5)	20(12)	20(12)	20(12.5)	20(11.5)	11.56	
NBhte											
3	100(1.5)	100(2)	100(1.5)	100(1.5)	100(1.5)	100(1.5)	100(1.5)	100(1.5)	100(1.5)	1.56	
4	0(13)	0(12)	40(6.5)	0(12.5)	0(12.5)	0(12.5)	0(12)	0(12.5)	0(12.5)	11.78	**6.22**
8	60(2.5)	60(3.5)	40(9)	40(7.5)	60(4.5)	60(5)	60(4.5)	60(6.5)	60(5)	5.33	
***k*NNhte**											
3	0(9)	0(9.5)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	9.06	
4	50(6)	60(4)	50(3.5)	50(4.5)	60(4)	50(5.5)	50(5)	50(7.5)	50(7)	5.22	**7.09**
8	40(8)	40(9.5)	40(9)	40(7.5)	40(8)	60(5)	60(4.5)	60(6.5)	60(5)	7.00	
DThte											
3	0(9)	0(9.5)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	9.06	
4	40(9)	30(9.5)	20(11.5)	40(8)	40(9)	20(10.5)	10(10)	20(11)	40(9.5)	9.78	**9.74**
8	20(12)	40(9.5)	20(12.5)	20(10.5)	20(11.5)	40(9.5)	20(12)	60(6.5)	40(9.5)	10.39	

Label	Skewed Classes %										AvR
	10-90	15-85	20-80	25-75	30-70	35-65	40-60	45-55	50-50	M.Av	
SVMhte											
3	0(9)	100(2)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	8.22	
4	20(10.5)	0(12)	30(9)	30(10.5)	30(10)	30(8.5)	30(8)	50(7.5)	30(11)	9.67	**7.41**
8	40(8)	60(3.5)	80(1.5)	60(3.5)	80(1.5)	60(5)	60(4.5)	60(6.5)	60(5)	4.33	
NNhte											
3	0(9)	0(9.5)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	0(9)	9.06	
4	60(2)	70(1.5)	40(6.5)	50(4.5)	50(7.5)	40(7)	50(5)	50(7.5)	50(7)	5.39	**7.15**
8	40(8)	40(9.5)	60(4.5)	40(7.5)	40(8)	60(5)	40(9)	60(6.5)	60(5)	7.00	
HTEsm											
3	10(3.5)	10(4.5)	10(3.5)	10(3.5)	10(3.5)	10(3.5)	10(3.5)	10(3.5)	10(3.5)	3.61	
4	50(6)	50(6.5)	50(3.5)	50(4.5)	60(4)	60(3.5)	60(2)	60(3)	60(3.5)	4.06	**4.11**
8	50(5)	60(3.5)	60(4.5)	60(3.5)	60(4.5)	60(5)	60(4.5)	60(6.5)	60(5)	4.67	
HTEdf											
3	10(3.5)	10(4.5)	10(3.5)	10(3.5)	10(3.5)	10(3.5)	10(3.5)	10(3.5)	10(3.5)	3.61	
4	50(6)	60(4)	50(3.5)	60(1.5)	60(4)	60(3.5)	60(2)	64(1)	65(1)	2.94	**3.33**
8	60(2.5)	60(3.5)	60(4.5)	60(3.5)	60(4.5)	60(5)	60(4.5)	62(2)	62(1)	3.44	

Table 8.18: Ranking the Generalization Performance of the Ensembles on Minority Class for Car Evaluation Dataset

Label	Skewed Classes %										M.AvR
	10-90	15-85	20-80	25-75	30-70	35-65	40-60	45-55	50-50	AvR	
NBE											
1	82(11)	82(11)	82(10.5)	82(10.5)	82(10.5)	82(8.5)	82(11)	82(9)	82(8.5)	10.06	**7.11**
3	100(4)	100(4)	100(4.5)	100(3)	100(5)	100(4.5)	100(5)	100(4)	100(3.5)	4.17	
***k*NNE**											
1	82(11)	82(11)	82(10.5)	82(10.5)	82(10.5)	73(12.5)	91(5.5)	73(12)	64(13)	10.72	**11.67**
3	76(13)	82(13)	76(11.5)	82(11.5)	71(13)	65(13)	65(12.5)	76(13)	53(13)	12.61	
DTE											
1	91(5.5)	91(5.5)	91(5)	91(5)	91(5)	91(4)	91(5.5)	91(4)	91(4)	4.83	**6.19**
3	82(11.5)	100(4)	100(4.5)	82(11.5)	100(5)	100(4.5)	100(5)	82(11)	82(11)	7.56	
RF											
1	91(5.5)	91(5.5)	82(10.5)	82(10.5)	82(10.5)	82(8.5)	91(5.5)	91(4)	82(8.5)	7.67	**7.56**
3	100(4)	88(10.5)	94(9)	94(7.5)	82(10)	94(9.5)	100(5)	94(8)	100(3.5)	7.44	

Label	Skewed Classes %										AvR
	10-90	15-85	20-80	25-75	30-70	35-65	40-60	45-55	50-50	M.Av	
SVME											
1	82(11)	82(11)	82(10.5)	82(10.5)	82(10.5)	82(8.5)	82(11)	82(9)	73(11.5)	10.39	**10.53**
3	88(9.5)	88(10.5)	76(11.5)	76(13)	76(11.5)	82(11)	88(10)	88(9)	88(10)	10.67	
NNE											
1	91(5.5)	91(5.5)	91(5)	91(5)	91(5)	91(4)	91(5.5)	91(4)	91(4)	4.83	**4.50**
3	100(4)	100(4)	100(4.5)	100(3)	100(5)	100(4.5)	100(5)	100(4)	100(3.5)	4.17	
NBhte											
1	82(11)	82(11)	82(10.5)	82(10.5)	82(10.5)	82(8.5)	82(11)	82(9)	82(8.5)	10.06	
3	100(4)	100(4)	100(4.5)	100(3)	100(5)	100(4.5)	100(5)	100(4)	100(3.5)	4.17	**7.11**
***k*NNhte**											
1	82(11)	82(11)	82(10.5)	82(10.5)	82(10.5)	73(12.5)	82(11)	64(13)	73(11.5)	11.28	**11.42**
3	82(11.5)	88(10.5)	71(13)	88(10)	76(11.5)	71(12)	65(12.5)	82(11)	65(12)	11.56	
DThte											
1	91(5.5)	91(5.5)	91(5)	91(5)	91(5)	82(8.5)	91(5.5)	82(9)	91(4)	5.89	**7.36**
3	88(9.5)	94(8)	88(10)	94(7.5)	100(5)	94(9.5)	82(11)	82(11)	94(8)	8.83	
SVMhte											
1	91(5.5)	91(5.5)	91(5)	91(5)	91(5)	82(8.5)	82(11)	82(9)	82(8.5)	7.00	**5.58**
3	100(4)	100(4)	100(4.5)	100(3)	100(5)	100(4.5)	100(5)	100(4)	100(3.5)	4.17	
NNhte											
1	91(5.5)	91(5.5)	91(5)	91(5)	91(5)	91(4)	91(5.5)	91(4)	91(4)	4.83	**5.58**
3	94(8)	88(10.5)	100(4.5)	94(7.5)	100(5)	100(4.5)	100(5)	100(4)	94(8)	6.33	
HTEsm											
1	92(2)	92(2)	92(2)	92(2)	92(2)	92(2)	92(2)	91(4)	91(4)	2.44	**3.81**
3	100(4)	100(4)	100(4.5)	94(7.5)	100(5)	100(4.5)	100(5)	100(4)	94(8)	5.17	
HTEdf											
1	94(1)	94(1)	94(1)	94(1)	95(1)	95(1)	95(1)	92(1)	92(1)	1.00	**2.58**
3	100(4)	100(4)	100(4.5)	100(3)	100(5)	100(4.5)	100(5)	100(4)	100(3.5)	4.17	

Table 8.19: Ranking the Generalization Performance of the Ensembles on Minority Class for White Wine Dataset

Ensemble	Skewed Classes %									AvR
	10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50	
NBE	40(3)	40(4.5)	40(3.5)	40(4)	43(3.5)	40(5.5)	40(5)	40(5)	40(5)	**4.33**
*k*NNE	20(8.5)	20(11)	20(9.5)	20(10.5)	20(10.5)	20(10.5)	20(11)	20(10.5)	20(9.5)	**10.17**
DTE	0(12.5)	20(11)	20(9.5)	20(10.5)	20(10.5)	20(10.5)	20(11)	20(10.5)	0(12.5)	**10.94**
RF	0(12.5)	20(11)	0(13)	20(10.5)	20(10.5)	20(10.5)	20(11)	20(10.5)	20(9.5)	**11.00**
SVME	40(3)	40(4.5)	40(3.5)	40(4)	40(6)	20(10.5)	40(9)	20(10.5)	20(9.5)	**6.72**
NNE	20(8.5)	40(4.5)	20(9.5)	20(10.5)	20(10.5)	40(5.5)	40(5)	40(5)	40(5)	**7.11**
NBhte	40(3)	40(4.5)	40(3.5)	40(4)	40(6)	40(5.5)	40(5)	40(5)	40(5)	**4.61**
*k*NNhte	20(8.5)	20(11)	20(9.5)	20(10.5)	20(10.5)	20(10.5)	20(11)	20(10.5)	20(9.5)	**10.17**
DThte	20(8.5)	20(11)	20(9.5)	20(10.5)	20(10.5)	20(10.5)	20(11)	20(10.5)	0(12.5)	**10.50**
SVMhte	20(8.5)	40(4.5)	40(3.5)	40(4)	45(1.5)	45(1.5)	40(5)	45(1.5)	45(1.5)	**3.50**
NNhte	20(8.5)	40(4.5)	20(9.5)	40(4)	40(6)	40(5.5)	40(5)	40(5)	40(5)	**5.89**
HTEsm	40(3)	40(4.5)	40(3.5)	40(4)	43(3.5)	43(3)	40(5)	40(5)	40(5)	**4.06**
HTEdf	40(3)	40(4.5)	40(3.5)	40(4)	45(1.5)	45(1.5)	45(1)	45(1.5)	45(1.5)	**2.45**

Table 8.20: Ranking the Generalization Performance of the Ensembles on Minority Class for Nursery Dataset

Ensemble	Skewed Classes %									AvR
	10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50	
NBE	94(6.5)	100(3)	94(5.5)	100(5.5)	94(7.5)	100(4.5)	94(5.5)	100(3.5)	94(8)	**5.50**
*k*NNE	82(10.5)	86(12)	88(11.5)	86(12)	94(7.5)	86(12)	94(5.5)	86(12)	88(11.5)	**10.50**
DTE	94(6.5)	95(8)	94(5.5)	100(5.5)	94(7.5)	100(4.5)	94(5.5)	95(9)	94(8)	**6.67**
RF	88(9)	100(11)	94(5.5)	100(5.5)	94(7.5)	100(4.5)	94(5.5)	100(3.5)	94(8)	**6.67**
SVME	47(13)	68(13)	41(13)	68(13)	41(13)	68(13)	35(13)	68(13)	53(13)	**13.00**
NNE	100(2.5)	95(8)	94(5.5)	100(5.5)	100(2.5)	95(9.5)	94(5.5)	95(9)	100(3)	**5.67**
NBhte	94(6.5)	100(3)	94(5.5)	100(5.5)	94(7.5)	100(4.5)	94(5.5)	100(3.5)	94(8)	**5.50**
*k*NNhte	82(10.5)	91(11)	94(5.5)	91(11)	94(7.5)	91(11)	94(5.5)	95(9)	94(8)	**8.78**
DThte	94(6.5)	95(8)	94(5.5)	100(5.5)	88(11.5)	100(4.5)	94(5.5)	100(3.5)	100(3)	**5.94**
SVMhte	76(12)	95(8)	88(11.5)	100(5.5)	88(11.5)	100(4.5)	82(12)	95(9)	88(11.5)	**9.50**
NNhte	100(2.5)	95(8)	94(5.5)	100(5.5)	100(2.5)	95(9.5)	88(11)	95(9)	100(3)	**6.28**
HTEsm	100(2.5)	100(3)	94(5.5)	100(5.5)	100(2.5)	100(4.5)	94(5.5)	100(3.5)	100(3)	**3.94**
HTEdf	100(2.5)	100(3)	94(5.5)	100(5.5)	100(2.5)	100(4.5)	94(5.5)	100(3.5)	100(3)	**3.94**

Table 8.21: Ranking the Generalization Performance of the Ensembles on Minority Class for Bank Marketing Dataset

Ensemble	**Skewed Classes %**									**AvR**
	10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50	
NBE	79(11)	84(7)	78(7)	82(5)	76(5)	73(5)	73(4)	72(3)	66(3.5)	**5.61**
*k*NNE	89(8)	82(8)	84(5)	77(6.5)	75(6)	76(4)	70(5)	69(5)	64(5)	**5.83**
DTE	82(10)	67(11)	72(9)	63(10)	57(10)	52(9)	63(7)	53(10)	34(11)	**9.67**
RF	92(6)	76(9)	69(10.5)	53(12)	44(12)	28(12)	17(12)	21(12)	11(12)	**10.83**
SVME	3(13)	4(13)	3(13)	4(13)	3(13)	4(13)	2(13)	3(13)	4(13)	**13.00**
NNE	89(8)	85(5.5)	69(10.5)	58(11)	52(11)	49(10.5)	45(10)	54(9)	50(8)	**9.28**
NBhte	89(8)	85(5.5)	77(8)	77(6.5)	69(7.5)	67(6)	61(9)	75(2)	60(7)	**6.61**
*k*NNhte	97(1.5)	92(3)	88(2.5)	88(2)	80(3)	82(2)	74(3)	68(6.5)	66(3.5)	**3.00**
DThte	95(3.5)	74(10)	80(6)	73(8)	69(7.5)	54(8)	63(7)	63(8)	40(9.5)	**7.50**
SVMhte	73(12)	63(12)	67(12)	65(9)	59(9)	49(10.5)	43(11)	45(11)	40(9.5)	**10.67**
NNhte	94(5)	92(3)	86(4)	84(4)	80(3)	65(7)	63(7)	68(6.5)	63(6)	**5.06**
HTEsm	95(3.5)	92(3)	88(2.5)	86(3)	80(3)	80(3)	75(2)	70(4)	68(2)	**2.89**
HTEdf	97(1.5)	94(1)	90(1)	89(1)	84(1)	83(1)	78(1)	76(1)	71(1)	**1.06**

Table 8.22: Ranking the Generalization Performance of the Ensembles on Minority Class for Censor Income Dataset

Ensemble	**Skewed Classes %**									**AvR**
	10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50	
NBE	89(13)	88(12.5)	89(11.5)	88(10.5)	88(9.5)	84(9)	87(7)	85(6)	85(4.5)	**9.27**
*k*NNE	96(10)	92(7.5)	92(8.5)	90(5)	89(8)	87(4.5)	88(4)	86(5)	84(6)	**6.50**
DTE	98(5)	94(3)	95(4.5)	88(10.5)	91(2.5)	87(4.5)	82(11)	84(7)	64(8)	**6.22**
RF	99(3)	93(5)	95(4.5)	90(5)	87(11)	78(10.5)	80(12)	74(11.5)	42(13)	**8.39**
SVME	97(7.5)	88(12.5)	88(13)	76(13)	79(13)	72(13)	73(13)	69(13)	57(10.5)	**12.06**
NNE	96(10)	89(10.5)	91(10)	83(12)	88(9.5)	78(10.5)	85(8.5)	74(11.5)	47(12)	**10.50**
NBhte	90(12)	89(10.5)	89(11.5)	89(8)	90(5.5)	86(6.5)	88(4)	87(3)	86(2.5)	**7.06**
*k*NNhte	97(7.5)	93(5)	94(6.5)	91(3)	90(5.5)	89(2)	88(4)	87(3)	85(4.5)	**4.56**
DThte	98(5)	93(5)	96(2.5)	90(5)	90(5.5)	85(8)	85(8.5)	82(9)	57(10.5)	**6.56**
SVMhte	98(5)	92(7.5)	94(6.5)	89(8)	85(12)	86(6.5)	83(10)	83(8)	82(7)	**7.83**
NNhte	96(10)	90(9)	92(8.5)	89(8)	90(5.5)	77(12)	88(4)	75(10)	59(9)	**8.44**
HTEsm	100(1.5)	95(2)	96(2.5)	92(2)	91(2.5)	88(3)	88(4)	87(3)	86(2.5)	**2.56**
HTEdf	100(1.5)	96(1)	97(1)	95(1)	93(1)	90(1)	90(1)	88(1)	87(1)	**1.06**

On the other hand, the ranking of the generalization performance of the ensembles to predict the minority classes in multi-class classification problems is computed differently. For each ensemble, the rankings of the generalization performance for each minority class is provided across all class distributions. Then the average ranking for each minority class is calculated. This is followed by the mean computation of the average rankings of all minority classes as shown in Tables 8.17 and 8.18.

The White Wine dataset contains seven classes with two minority classes, i.e. "3" and "9" white wine quality labels. However, as shown in the confusion matrix of the White Wine dataset, no ensemble predicted the label "9". Hence, the generalization performance of the ensembles to predict the minority label "9" was not reported, while the minority label "3" was reported. On the other hand, the Nursery dataset contains four classes to be predicted, but has one minority class of label "2".

From the confusion matrices in Appendix A, the DTE, RF, and DThte significantly underperformed other ensembles for the prediction of the minority class. This is observed in the high average ranks of the DTE, RF, and DThte in Tables 8.13 to 8.22 compared to the HTEdf, illustrating that the predictions of the DTE, RF, and DThte are mostly biased towards the majority class. Also, the performance of the DTE, RF, and DThte is further explained by the skewed sensitivity of the information gain and Gini measure used during the tree induction in the ensembles. The findings of Dietterich et al. (1996), Flach (2003), and Liu et al. (2010) have shown that tree learning models are sensitive to imbalanced datasets when the information gain or Gini measure are used as splitting criterion. While the results of the DTE and DThte corroborates the findings of these studies, the benefits of the mixtures of heterogeneous experts are still observed in the generalization performance of the DThte over the homogeneous mixtures in the DTE. The DThte obtained lower average ranks compared to the DTE in all classification datasets except for the Car Evaluation and Censor Income datasets.

Further, the *k*NNE and *k*NNhte produced high average ranks extremely far from the average rank of the HTEdf on all datasets except for the Credit Approval dataset. Based on the inductive bias of *k*NN algorithms to classify a test sample to the class of the closest training samples, the probability that the predictions of the *k*NNE and *k*NNE are skewed towards to majority class is very high. This is attributed to the fact that *k*NN algorithms

are highly sensitive to skewed classes with large values of k because more majority samples are learned than the minority samples during training. Hence, predictions are biased toward the majority class (Archana and Elangovan, 2014). However, the kNNhte promotes the benefits of the mixtures of heterogeneous experts over the kNNE for seven of the 10 classification datasets. The kNNhte performed better than the kNNE mostly on multi-class classification problems, while the kNNE performed better for the Breast Cancer, Indian Liver, and Credit Approval datasets. Furthermore, the HTEdf is significantly different from the SVME, but not the SVMhte in terms of generalization performance to predict the minority class(es). The significant difference in performance between the HTEdf and the SVME highlights the advantage of a mixture of heterogeneous experts over a homogeneous mixture of experts based on different ML algorithms and different configurations for the base learners within the HTEdf.

The average rank of the SVME across all datasets illustrates that the predictions of the SVME are highly skewed towards the majority class. This indicates that using the same control parameter configuration for the SVME, for instance, using the same cost of misclassification C to induce base experts within the SVME, did not provide sufficient complexity to obtain hyperplanes that generated suitable margins to separate the classes effectively across all datasets. Thus, the SVME is explained to induce weak soft margins, which influence the predictions of the base experts within the SVME to bias towards the majority class. The behaviour of the SVME corroborates the findings of Wu (2003) that the SVME was unable to effectively learn positive support vectors with increasing skewed classes in the training datasets across the experimental data.

Wu (2003) explained that, as training datasets become more skewed, the ratio between the positive and negative support vectors becomes more imbalanced. As a result of the imbalance, the neighbourhood of a test sample close to the decision boundary is more likely to be dominated by negative support vectors. Therefore, the possibility of the decision function biasing towards the majority class becomes very high (Wu, 2003). On the contrary, the average rank of the SVMhte over all classification datasets showed that the different control parameter configurations of the base members in the SVMhte resulted in different suitable experts that generalized better than the SVME.

Also, the average ranks of the NBE and NBhte in Tables 8.13 to 8.22 across the 10

classification datasets showed that the NBE and NBhte are insensitive to the skewed class distributions across the datasets. The insensitivity of the NBE and NBhte to skewed class distributions is attributed to the fact that NB algorithms are considered as stable algorithms, because stable algorithms are insensitive to small changes in the sample space of the training data (Breiman, 1996a; El-Hindi et al., 2018).

Also, Hoens and Chawla (2013) empirically showed that NB algorithms are trivially skewed-insensitive because NB algorithms make predictions for the posterior probability $p(y|x_i)$ by first computing the likelihood $p(x_i|y)$ and the prior probability $p(y)$ from the training data. Therefore, due to the inductive bias of feature independence made by the NB algorithms, Hoens and Chawla (2013) concluded that the insensitivity of the NB algorithms to skewed data is attributed to the fact that the predictions are calibrated by prior probability of y or $p(y)$.

Statistical Analysis of Results

This section compares the generalization performance of the HTEs and homogeneous ensembles on the minority class(es) over all classification datasets. The comparison is achieved using statistical tests to determine whether there exists a statistically significant difference in the generalization performance of the ensembles to predict the minority class(es).

Friedman Test

The Friedman test used to compare the generalization performance of the 13 ensembles to predict the minority class(es) over the 10 datasets followed the discussion provided in Section 8.2.

Due to the skewed class distributions in each dataset, the first step is to rank the ensembles based on the generalization performance to predict the minority class(es) in each skewed class distribution for each dataset. The rankings of the ensembles have been provided in Tables 8.13 to 8.22. The last step is to rank the average ranks of the ensembles across all datasets by gathering the average ranks of the ensembles from Tables 8.13 to 8.22.

From Table 8.23, the HTEdf is ranked as the best performing ensemble achieving the lowest average ranking of 1.60, followed by the HTEsm (2.70), NBE (5.40), and

SVMhte (5.75). The average rankings of the HTEdf and HTEsm illustrate the advantage of combining different ML algorithms to develop mixtures of heterogeneous experts compared to the combination of the same ML algorithms.

Table 8.23: Ranking the Generalization Performance of Ensembles on Minority Class(es) over all Classification Datasets

Ensemble	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor	AvR
NBE	6.72(7)	4.17(3)	3.39(2)	8.61(7)	6.15(4)	7.11(7.5)	4.33(4)	5.50(3.5)	5.61(5)	9.27(11)	**5.40**
*k*NNE	11.94(13)	9.22(10)	6.83(7)	2.11(1)	7.93(10)	11.67(13)	10.17(9.5)	10.50(12)	5.83(6)	6.50(5)	**8.65**
DTE	10.06(11)	8.39(9)	12.00(13)	9.39(10)	9.76(13)	6.19(6)	10.94(12)	6.67(8.5)	9.67(10)	6.22(4)	**9.65**
RF	8.33(9)	6.78(7)	11.78(12)	10.44(13)	9.35(11)	7.56(10)	11.00(13)	6.67(8.5)	10.83(12)	8.39(9)	**10.45**
SVME	4.61(4)	6.00(6)	6.06(6)	10.39(12)	5.06(3)	10.53(11)	6.72(7)	13.00(13)	13.00(13)	12.06(13)	**8.88**
NNE	6.89(8)	12.06(13)	11.06(11)	8.83(8)	7.85(9)	4.50(3)	7.11(8)	5.67(5)	9.28(9)	10.50(12)	**8.60**
NBhte	6.17(6)	5.11(5)	3.11(1)	10.06(11)	6.22(5)	7.11(7.5)	4.61(5)	5.50(3.5)	6.61(7)	7.06(7)	**5.80**
*k*NNhte	10.11(12)	10.83(12)	7.33(9)	2.56(2)	7.09(6)	11.42(12)	10.17(9.5)	8.78(10)	3.00(3)	4.56(3)	**7.85**
DThte	8.44(10)	6.94(8)	9.11(10)	8.89(9)	9.74(12)	7.36(9)	10.50(11)	5.94(6)	7.50(8)	6.56(6)	**8.90**
SVMhte	4.44(3)	3.22(1)	4.72(5)	3.78(4)	7.41(8)	5.58(4.5)	3.50(2)	9.50(11)	10.67(11)	7.83(8)	**5.75**
NNhte	5.83(5)	9.44(11)	7.00(8)	7.44(6)	7.15(7)	5.58(4.5)	5.89(6)	6.28(7)	5.06(4)	8.44(10)	**6.85**
HTEsm	3.72(1.5)	4.61(4)	4.61(4)	5.28(5)	4.11(2)	3.81(2)	4.06(3)	3.94(1.5)	2.89(2)	2.56(2)	**2.70**
HTEdf	3.72(1.5)	3.72(2)	4.06(3)	3.22(3)	3.33(1)	2.58(1)	2.45(1)	3.94(1.5)	1.06(1)	1.06(1)	**1.60**

Also, with an exception to the NBE and NBhte, it can be observed that all HTEs provide lower average rankings compared to the pure homogeneous ensembles counterparts, highlighting the benefit of mixtures of heterogeneous experts in comparison to homogeneous mixtures.

Given the average rankings of the ensembles in Table 8.23, the calculated Friedman test statistic is $\chi_F^2 = 56.986$, while the computed Iman-Davenport extension of the Friedman test F_F gives 8.139. Because the value of F_F is greater than the obtained critical value, the null hypothesis that all ensembles are equal is rejected. The rejection of the null hypothesis shows that, there is a statistically significant difference in the generalization performance of the ensembles to predict the minority class(es) in the datasets.

Bonferroni-Dunn Test

The Bonferroni-Dunn test is performed after rejecting the null hypothesis to find out the ensembles that significantly differ from each other in the skewed class distributions study. The critical value is 2.87, and the computed critical difference (CD) equals

4.998. Figure 8.22 shows the critical difference plot indicating the significant difference in generalization performance on the minority class between the HTEdf and any other ensemble.

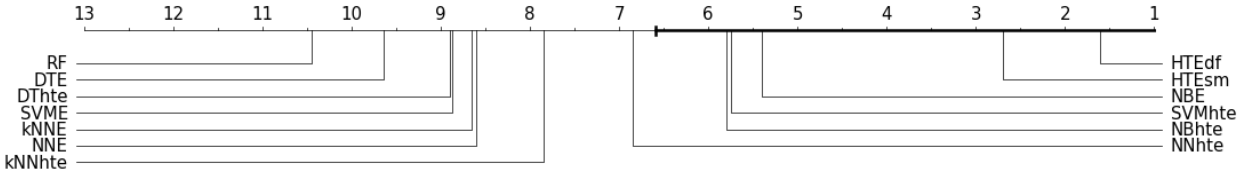

Figure 8.22: Critical Difference Plot of Ensembles for Skewed Class Distribution Study

The outcome of the Bonferroni-Dunn test in Figure 8.22 showed that the HTEdf is significantly more accurate than the *k*NNE, DTE, RF, SVME, NNE, *k*NNhte, DThte, and NNhte for the prediction of the minority classes over all datasets. The outcome illustrates that these ensembles showed sensitivity to the different skewed class distributions leading to unstable prediction performance across the 10 datasets. On the other hand, the HTEdf capitalized on the advantage of combining different ML algorithms offering different favourable assumptions in each skewed class distribution across the datasets. Also, the HTEdf maximizes the differing views obtained from the different configurations of the multiple instances of the ML algorithms combined to generate the final HTEdf prediction. Further, the outcome of the Bonferroni-Dunn test indicates that 10 experimental datasets did not provide sufficient evidence to show that a significant difference in generalization performance on the minority class exists between the HTEdf and the HTEsm, NBE, SVMhte, and NBhte.

8.4 Number of Outliers Study

This section discusses the ensemble performance across the number of outliers perturbed from 1% to 5% in the training datasets of the classification datasets. The summative results (over all outlier ratios) of the testing and training accuracy, GF and F1-score for each ensemble over all classification datasets are provided in Table 8.24.

The results presented in Table 8.24 showed that the ensembles produced mixed results in all the performance measures illustrating that the ensembles responded differently to the number of outliers across all the classification datasets.

Table 8.24: Ensemble Results over all Classification Datasets in Number of Outliers Study

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBE										
Testing Accuracy	0.747	0.517	0.665	0.535	0.471	0.809	0.427	0.835	0.556	0.505
Training Accuracy	0.721	0.649	0.683	0.617	0.548	0.837	0.432	0.873	0.798	0.807
GF	0.911	1.385	1.055	1.215	1.172	1.171	1.010	1.293	2.203	2.567
F1-score	0.788	0.544	0.572	0.476	0.384	0.820	0.316	0.846	0.790	0.750
*k*NNE										
Testing Accuracy	0.586	**0.666**	0.730	0.785	0.501	0.830	0.473	0.822	0.897	0.805
Training Accuracy	0.758	0.800	0.663	0.865	0.784	0.929	0.703	0.909	0.828	0.862
GF	1.712	1.670	0.801	1.591	2.312	2.398	1.777	1.962	0.598	1.409
F1-score	0.850	0.574	0.570	0.804	0.466	0.770	0.438	0.886	0.730	0.800
DTE										
Testing Accuracy	0.624	0.570	0.746	0.760	0.484	0.922	0.477	0.899	0.886	0.803
Training Accuracy	0.730	0.739	0.718	0.868	0.762	0.949	0.709	0.959	0.937	0.873
GF	1.396	1.653	0.901	1.814	2.173	1.518	1.796	2.437	1.809	1.552
F1-score	0.774	0.640	0.666	0.798	0.516	0.960	0.428	0.946	0.888	0.820
RF										
Testing Accuracy	0.706	0.625	0.730	0.791	0.537	0.883	0.521	0.897	0.898	0.807
Training Accuracy	0.787	0.786	0.770	0.889	0.855	0.969	0.790	0.963	0.951	0.871
GF	1.385	1.754	1.172	1.880	3.196	3.742	2.284	2.815	2.091	1.497
F1-score	0.796	0.628	**0.736**	0.826	0.618	0.928	**0.520**	0.948	0.888	0.802
SVME										
Testing Accuracy	0.723	0.623	0.747	0.777	0.531	0.871	0.500	0.879	0.894	0.805
Training Accuracy	0.843	0.800	0.690	0.882	0.569	0.981	0.470	0.969	0.973	0.869
GF	1.767	1.885	0.815	1.893	1.090	6.745	0.944	3.857	3.956	1.491
F1-score	0.854	0.510	0.668	**0.856**	0.460	0.898	0.352	0.944	0.840	0.808
NNE										
Testing Accuracy	0.764	0.556	0.730	0.670	0.545	0.949	0.515	0.933	0.891	0.803
Training Accuracy	0.840	0.793	0.722	0.857	0.830	0.975	0.781	0.983	0.971	0.871
GF	1.480	2.151	0.969	2.312	2.675	2.075	2.210	4.103	3.768	1.521
F1-score	0.898	0.610	0.662	0.744	0.570	0.976	0.506	0.978	0.890	0.812

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBhte										
Testing Accuracy	0.762	0.518	0.668	0.651	0.489	0.809	0.449	0.844	0.831	0.726
Training Accuracy	0.721	0.685	0.685	0.710	0.547	0.837	0.445	0.889	0.826	0.838
GF	0.857	1.535	1.055	1.205	1.128	1.171	0.993	1.408	0.973	1.689
F1-score	0.794	0.598	0.566	0.624	0.382	0.820	0.316	0.862	0.840	0.780
*k*NNhte										
Testing Accuracy	0.655	0.635	0.727	0.788	0.531	0.844	0.503	0.844	0.893	0.811
Training Accuracy	0.787	0.815	0.708	0.876	0.838	0.919	0.764	0.917	0.859	0.856
GF	1.633	1.974	0.936	1.716	2.895	1.918	2.110	1.871	0.759	1.310
F1-score	0.888	0.696	0.630	0.852	0.566	0.786	0.498	0.890	0.778	0.800
DThte										
Testing Accuracy	0.659	0.587	0.741	0.760	0.526	0.934	0.518	0.905	0.895	0.806
Training Accuracy	0.785	0.752	0.734	0.879	0.850	0.976	0.790	0.975	0.952	0.837
GF	1.589	1.666	0.974	1.991	3.151	2.762	2.289	3.812	2.205	1.192
F1-score	0.802	0.676	0.658	0.812	0.546	0.960	0.516	0.968	0.892	0.790
SVMhte										
Testing Accuracy	0.701	0.627	0.743	**0.816**	0.537	0.903	0.505	0.915	**0.903**	**0.828**
Training Accuracy	0.769	0.800	0.669	0.878	0.826	0.962	0.553	0.977	0.930	0.861
GF	1.296	1.873	0.774	1.504	2.662	2.519	1.108	3.633	1.408	1.237
F1-score	0.796	0.630	0.648	0.828	0.512	0.962	0.398	0.968	0.886	0.820
NNhte										
Testing Accuracy	0.762	0.571	0.757	0.767	**0.563**	0.948	0.533	0.936	0.898	0.806
Training Accuracy	0.816	0.795	0.761	0.887	0.849	0.982	0.775	0.989	0.966	0.874
GF	1.294	2.096	1.014	2.069	2.888	2.910	2.080	5.915	3.046	1.543
F1-score	0.856	0.616	0.696	0.786	0.580	0.990	0.506	0.986	0.894	0.828
HTEsm										
Testing Accuracy	0.780	0.630	0.761	0.773	0.549	0.951	0.521	0.951	0.898	0.812
Training Accuracy	0.845	0.810	0.715	0.890	0.679	0.975	0.755	0.996	0.962	0.874
GF	1.421	1.949	0.838	2.060	1.407	1.946	1.958	14.720	2.699	1.487
F1-score	0.906	0.688	0.660	0.806	0.570	0.974	0.470	**1.000**	0.890	**0.832**
HTEdf										
Testing Accuracy	**0.791**	0.634	**0.765**	0.799	0.555	**0.954**	**0.534**	**0.955**	0.900	0.814
Training Accuracy	0.845	0.785	0.749	0.901	0.842	0.985	0.770	0.997	0.955	0.862
GF	1.345	1.704	0.936	2.044	2.813	3.013	2.027	22.490	2.218	1.348
F1-score	**0.930**	**0.726**	0.700	0.830	**0.624**	**0.996**	0.494	**1.000**	**0.900**	**0.832**

The HTEdf outperformed the other ensembles for five of the 10 datasets based on generalization performance. The SVMhte is the most accurate ensemble for the Credit Approval, Bank Marketing, and Censor Income datasets. The *k*NNE and NNhte performed best for the Breast Cancer and Red Wine datasets, respectively. It is observed that the HTEdf and HTEsm remarkably ranked as the second most accurate ensemble for most of the datasets, illustrating the advantage of the mixtures of the heterogeneous experts obtained from the combination of different ML algorithms.

The NBE and NBhte provided the worst generalization performance on all datasets except for the Sonar dataset, where the *k*NNE performed worst in prediction. The generalization performance of the NBE and NBhte indicates that the ensembles struggled to capture the characteristics and complexity of experimental data compared to other ensembles in this modelling study. However, the NBhte still expressed the benefit of the mixtures of heterogeneous experts obtained from different control parameter configurations for the base NB learners within the NBhte. Other ensembles generated the same trend on most of the datasets as observed by the *k*NNE and *k*NNhte, DTE and DThte, SVME and SVMhte, as well as NNE and NNhte.

Illustrated by the high average ranks in Table 8.25, all pure homogeneous ensembles and a number of HTEs (i.e. NBhte, *k*NNhte, and DThte) demonstrated prediction behaviours that are adversely influenced by the outlier ratios over all datasets. While the DT algorithms have been shown to be robust to outliers (Breiman et al., 1993; John, 1995; Rokach and Maimon, 2014; Song and Lu, 2015; Nyitrai and Virag, 2019), the DTE and DThte developed in this modelling study exhibit inherent sensitivity to the different outlier ratios. The DTE and DThte did not generalize well on numeric features as shown in the Sonar, Red Wine and White Wine datasets, but performed well on categorical and multi-variate features observed in other datasets. This shows that the DTE and DThte prioritized categorical features over numeric features, which explains why the ensembles struggled against the different number of outliers in the study. Also, the prediction behaviour of the DTE and DThte corroborates the findings in the studies of Sebban et al. (2000) and Zeng and Cheng (2021) that the outliers in ratios adversely influence the predictive performance of DT ensembles.

Sebban et al. (2000) and Zeng and Cheng (2021) reported that there is a high possibility

that the induction algorithms producing the experts within a DT ensemble increase the depth of the base trees, which results in severe overfitting of the training dataset, leading to poor generalization performance. Also, Ch'ng and Mahat (2020) explained that outliers could influence the splitting point and affect the potential variable to be selected during tree induction. The outcome is an adverse effect on classification accuracy and the structure of the tree, which could lead to low generalization performance of the DTE for the Sonar, Red Wine, and White Wine datasets. However, the DThte achieved improved generalization performance compared to the DTE, which is due to the benefits of the mixtures of heterogeneous experts from different configurations of the base DT learners.

Given the inductive bias of *k*NN algorithms to classify a test sample to the class of the closest training samples from stored training data, Beckmann et al. (2015) explained that a large number of misclassified samples by base *k*NN learners could change the class boundaries during the distance computation. The adverse effect of the change in class boundaries due to high misclassification results in low overall generalization performance. The *k*NNE and *k*NNhte tend to show similar behaviour for most datasets, often ranked among the least performing ensembles.

Additionally, while the GFs of the *k*NNE and *k*NNhte illustrates that the ensembles showed severe overfitting of the training dataset in all datasets except for the Indian Liver dataset. Although, the *k*NNhte is better than the *k*NNE in terms of generalization performance due to the combination of different assumptions obtained from different control parameter configurations of the base *k*NN learners.

The average ranks of the NNE and SVME showed that using the same control parameter configuration for the base learners to induce base experts within these ensembles did not yield a significant improvement in generalization performance. The NNE risks overfitting the training dataset by learning from the different number of outliers in the training dataset, which impair the overall prediction performance across all datasets except for the Indian Liver dataset. On the other hand, the performance of the SVME illustrates the possibility that the predictions of the base experts were derived using soft margins. SVMs with soft margin approaches have been reported to show sensitivity to outliers (Bandaragoda et al., 2018; Wang et al., 2019a). However, the NNhte and SVMhte illustrated the benefit of using different control parameter configurations to

induce suitable experts that produced improved generalization performance compared to the SVME and NNE.

From the results presented in Appendix B, most of the ensembles, especially the pure homogeneous ensembles, offered unstable testing accuracy from different categories of the outlier ratios starting from low (1% and 2%), mild (3%) and extreme (4% and 5%) outlier ratios. However, the HTEdf achieved stable prediction performance by ranking as the best ensemble in terms of generalization performance for the low, mild, and extreme categories over five datasets (i.e. Sonar, Indian Liver, Car Evaluation, White Wine, and Nursery datasets). The SVMhte outperformed other ensembles over all the outlier ratio categories for three datasets (i.e. Credit Approval, Bank Marketing, and Censor Income datasets). The NNhte and *k*NNE are the most accurate ensembles for the Red Wine and Breast Cancer datasets. The HTEdf and HTEsm achieved the second-best performance for most datasets when the outlier ratios categories were examined.

For training performance in Table 8.24, the HTEdf and HTEsm achieved the highest training accuracy across most of the datasets, rivalled by the SVMhte, NNhte, and NNE in a number of datasets. The testing and training accuracies of the NBE and NBhte on the Red Wine and White datasets indicate that the ensembles experience slight overfitting of the training dataset, while the NBhte did not overfit for the White Wine dataset. Also, the GFs of all ensembles indicated slight overfitting of the training dataset on most datasets except for the HTEsm and HTEdf that produced severe overfitting on the Nursery dataset. On the other hand, the NBE, *k*NNE, DTE, SVME, and NNE showed no indication of overfitting for the Sonar, Indian Liver, and Bank Marketing datasets. The NBhte, *k*NNhte, DThte, SVMhte, and HTEdf also experienced no indication of overfitting the training dataset on the same datasets.

Further, despite the mixed results recorded by the ensembles in this modelling study, the advantage of the mixtures of heterogeneous experts over pure homogeneous mixtures is still evident in the F1-score performance in Table 8.24. The HTEdf achieved the highest F1-score over seven of the 10 datasets, rivalled by the HTEsm on the Nursery and Censor Income datasets. The RF algorithm capitalized on the intrinsic ensemble approaches to outperform other ensembles on the Indian Liver and White Wine datasets by offering F1-scores of 73.6% and 52.0% respectively. The SVME performed best on the Credit Approval

dataset.

Thus, the results of all performance measures provide evidence to conclude that it is beneficial to construct mixtures of heterogeneous experts to learn and generalize on datasets with outliers present in the training dataset. Specifically, the combination of different ML algorithms to construct the mixtures of heterogeneous experts in the HTEdf and HTEsm demonstrated a superior advantage over other ensembles.

Statistical Tests

This section discusses the statistical comparison of the generalization performance of the ensembles across the outlier ratios over the classification datasets.

Friedman Test

The discussion of the Friedman test is provided in Section 8.2, and is used to compare the generalization performance of the 13 ensembles across all the number of outliers over the 10 datasets in this modelling study.

Based on the number of outliers considered in each dataset, the mean average of the generalization performance of each ensemble across the number of outliers is first computed. Then the computed mean average is ranked according to the Friedman test as shown in Table 8.25.

Table 8.25: Ranking the Generalization Performance of Ensembles over all Classification Datasets in the Number of Outliers Study

Ensemble	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor	AvR
NBE	74.7(6)	51.7(13)	66.5(13)	53.5(13)	47.1(13)	80.9(12.5)	42.7(13)	83.5(12)	55.6(13)	50.5(13)	**12.15**
*k*NNE	58.6(13)	66.6(1)	73.0(9)	78.5(5)	50.1(10)	83.0(11)	47.3(11)	82.2(13)	89.7(6)	80.5(8.5)	**8.75**
DTE	62.4(12)	57.0(10)	74.6(5)	76.0(9.5)	48.4(12)	92.2(6)	47.4(10)	89.9(7)	88.6(11)	80.1(11)	**9.35**
RF	70.6(8)	62.5(6)	73.0(9)	79.1(3)	53.7(5.5)	88.3(8)	52.0(4)	89.7(8)	89.8(5)	80.7(5)	**6.15**
SVME	72.3(7)	62.2(7)	74.7(4)	77.7(6)	53.1(7.5)	87.1(9)	50.0(9)	87.9(9)	89.4(8)	80.5(8.5)	**7.50**
NNE	76.4(3)	55.6(11)	73.0(9)	66.9(11)	54.5(4)	94.9(3)	51.5(6)	93.3(4)	89.1(10)	80.3(10)	**7.10**
NBhte	76.2(4.5)	51.8(12)	68.8(12)	65.1(12)	48.9(11)	80.9(12.5)	44.9(12)	84.4(10.5)	83.1(12)	72.6(12)	**11.05**
*k*NNhte	65.5(11)	63.5(2)	72.7(11)	78.8(4)	53.1(7.5)	84.4(10)	50.3(8)	84.4(10.5)	89.3(9)	81.1(4)	**7.70**
DThte	65.9(10)	58.7(8)	74.1(7)	76.0(9.5)	52.6(9)	93.4(5)	51.8(5)	90.5(6)	89.5(7)	80.6(6.5)	**7.30**
SVMhte	70.1(9)	62.7(5)	74.3(6)	80.6(1)	53.7(5.5)	90.3(7)	50.5(7)	91.5(5)	90.1(1)	82.8(1)	**4.75**
NNhte	76.2(4.5)	57.1(9)	75.7(3)	76.7(8)	56.3(1)	94.8(4)	53.3(2)	93.6(3)	89.8(4)	80.6(6.5)	**4.50**
HTEsm	78.0(2)	63.0(4)	76.1(2)	77.3(7)	54.9(3)	95.1(2)	52.1(3)	95.1(2)	89.8(4)	81.2(3)	**3.20**
HTEdf	79.1(1)	63.4(3)	76.5(1)	79.9(2)	55.5(2)	95.4(1)	53.4(1)	95.5(1)	90.0(2)	81.4(2)	**1.60**

Illustrated by the average ranks of the ensembles in Table 8.25, the HTEdf is the best performing ensemble achieving the lowest average ranking of 1.60, followed by the HTEsm (3.20) and NNhte (4.50). The average rankings of the HTEdf and HTEsm illustrate the benefit of the combination of different ML algorithms to develop a heterogeneous mixtures of experts for the study. Another outcome is that all HTEs provide lower average rankings than the pure homogeneous ensembles counterparts, which showed the advantage of mixtures of heterogeneous experts compared to homogeneous mixtures.

From the average rankings of the ensembles in Table 8.25, the calculated Friedman test statistic χ_F^2 is 72.122, and the Iman-Davenport extension of the Friedman test is computed as F_F = 13.557. Hence, because the value of F_F is greater than the obtained critical value, the null hypothesis that all ensembles are equal is rejected. The rejection of the null hypothesis indicates that, there is a statistically significant difference in the generalization performance of the ensembles across the number of outliers for all datasets.

Bonferroni-Dunn Test

After the rejection of the null hypothesis, the Bonferroni Dunn post hoc test is performed to determine the ensembles that significantly differ from each other in the number of outlier study for classification problems. The critical value is 2.87, and the computed critical difference (CD) = 4.998. The significant difference in generalization performance between the HTEdf and any other ensemble is shown in the critical difference plot in Figure 8.23.

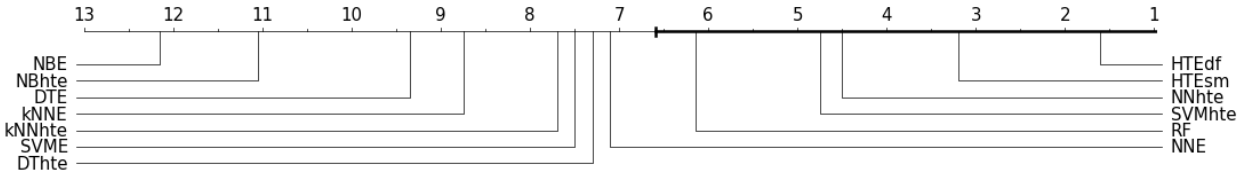

Figure 8.23: Critical Difference Plot of Ensembles for Number of Outliers Study in Classification Problems

The outcome of the Bonferroni-Dunn test showed that the HTEdf is significantly more accurate than all pure homogeneous ensembles (i.e. NBE, *k*NNE, DTE, SVME, NNE), NBhte, *k*NNhte and DThte. The HTEsm is also significantly more accurate than all these ensembles. The outcome confirms that the construction of mixtures of heterogeneous

experts using different ML algorithms as observed in HTEdf and HTEsm, provides better generalization performance than other ensembles constructed using the same ML algorithms. On the contrary, the outcome showed that the experimental datasets did not provide sufficient evidence to conclude that a significant difference in generalization performance exists between the HTEdf and the HTEsm, NNhte, SVMhte and RF algorithm.

8.5 Severity of Outliers Study

This section discusses the ensemble performance across the severity of outliers from 2 to 4 standard deviations from the estimated mean in the training datasets of the classification datasets. The summative results (over all outlier severities) of the testing and training accuracy, GF and F1-score for each ensemble over all classification datasets are provided in Table 8.26.

Illustrated in Table 8.26, the ensembles demonstrated different prediction performances over all the classification datasets. Generally, the results highlighted that the ensembles responded to the different standard deviations at which the data deviated from the estimated mean in each dataset. Unlike the number of outliers study where the HTEdf was ranked as the most accurate ensemble on five of the 10 datasets, the HTEdf outperformed other ensembles on four datasets in this modelling study.

The prediction outcome of the HTEdf on four datasets illustrates that a number of base ML algorithms combined in the HTEdf showed more sensitivity to the outlier severities as shown in the average rank of 2.25 achieved by the HTEdf, which is higher than 1.60 obtained in the number of outliers study.

The SVMhte is the most accurate ensemble for three datasets, while the NNhte offered the best generalization performance for the Red Wine and Nursery datasets. While the HTEsm rivalled the HTEdf for the Sonar dataset, both ensembles achieved the second-best generalization performance for four datasets altogether.

On the other hand, the NBE and NBhte offered the least testing accuracy for eight datasets except for the Sonar and Nursery datasets. This indicates that the NBE and NBhte struggled to generalized well on the characteristics and complexity of these datasets

compared to other ensembles, and given the outlier severities considered.

Table 8.26: Ensemble Results over all Classification Datasets in Severity of Outliers Study

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBE										
Testing Accuracy	0.747	0.517	0.665	0.535	0.471	0.809	0.427	0.854	0.556	0.591
Training Accuracy	0.732	0.655	0.661	0.628	0.546	0.945	0.432	0.877	0.798	0.774
GF	0.948	1.405	0.989	1.251	1.166	3.473	1.010	1.183	2.200	1.809
F1-score	0.826	0.562	0.662	0.478	0.378	0.820	0.314	0.840	0.788	0.680
*k*NNE										
Testing Accuracy	0.586	**0.666**	0.727	0.788	0.501	0.844	0.473	0.810	0.893	0.813
Training Accuracy	0.773	0.797	0.680	0.873	0.784	0.964	0.703	0.904	0.859	0.860
GF	1.832	1.644	0.854	1.675	2.313	4.333	1.777	1.976	0.758	1.336
F1-score	0.808	0.564	0.588	0.802	0.466	0.786	0.438	0.896	0.730	0.802
DTE										
Testing Accuracy	0.624	0.560	0.737	0.759	0.479	0.924	0.477	0.901	0.887	0.811
Training Accuracy	0.765	0.739	0.696	0.864	0.763	0.973	0.709	0.955	0.938	0.848
GF	1.609	1.692	0.866	1.778	2.198	2.815	1.796	2.178	1.818	1.244
F1-score	0.742	0.640	0.644	0.790	0.520	0.960	0.424	0.940	0.884	0.790
RF										
Testing Accuracy	0.723	0.640	0.727	0.797	0.540	0.882	0.521	0.896	0.899	0.813
Training Accuracy	0.795	0.783	0.734	0.888	0.853	0.972	0.790	0.964	0.951	0.876
GF	1.356	1.658	1.029	1.816	3.129	4.163	2.284	2.862	2.074	1.509
F1-score	0.806	0.626	**0.716**	0.812	0.592	0.928	**0.506**	0.950	**0.890**	0.810
SVME										
Testing Accuracy	0.692	0.628	0.746	0.778	0.531	0.872	0.500	0.869	0.894	0.808
Training Accuracy	0.768	0.800	0.659	0.879	0.569	0.972	0.470	0.971	0.973	0.870
GF	1.331	1.864	0.746	1.828	1.089	4.578	0.944	4.563	3.986	1.477
F1-score	0.722	0.510	0.616	0.846	0.444	0.898	0.348	0.954	0.840	0.810
NNE										
Testing Accuracy	0.766	0.558	0.729	0.674	0.539	0.948	0.515	0.953	0.890	0.815
Training Accuracy	0.854	0.786	0.673	0.855	0.824	0.967	0.781	0.997	0.971	0.877
GF	1.618	2.073	0.829	2.261	2.628	1.598	2.210	15.667	3.819	1.499
F1-score	0.900	0.632	0.670	0.734	0.542	0.990	0.498	**1.000**	**0.890**	0.814

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBhte										
Testing Accuracy	0.762	0.518	0.668	0.651	0.489	0.809	0.449	0.859	0.831	0.733
Training Accuracy	0.733	0.686	0.671	0.710	0.545	0.945	0.445	0.891	0.826	0.830
GF	0.894	1.536	1.010	1.205	1.123	3.473	0.993	1.289	0.970	1.567
F1-score	0.818	0.584	0.654	0.630	0.378	0.820	0.316	0.858	0.840	0.758
*k*NNhte										
Testing Accuracy	0.655	0.635	0.730	0.808	0.531	0.830	0.503	0.821	0.897	0.823
Training Accuracy	0.802	0.812	0.645	0.864	0.838	0.969	0.764	0.908	0.826	0.855
GF	1.751	1.941	0.760	1.412	2.895	5.484	2.110	1.946	0.593	1.221
F1-score	0.878	0.702	0.494	0.852	0.566	0.770	0.494	0.884	0.780	0.804
DThte										
Testing Accuracy	0.677	0.588	0.743	0.761	0.526	0.934	0.518	0.914	0.894	0.824
Training Accuracy	0.798	0.749	0.689	0.878	0.850	0.974	0.790	0.974	0.952	0.870
GF	1.616	1.646	0.826	1.956	3.169	2.546	2.289	3.380	2.202	1.361
F1-score	0.790	0.680	0.658	0.808	**0.624**	0.960	0.516	0.970	**0.890**	0.816
SVMhte										
Testing Accuracy	0.726	0.627	0.747	**0.818**	0.550	0.903	0.505	0.915	**0.903**	**0.829**
Training Accuracy	0.845	0.799	0.664	0.878	0.679	0.964	0.553	0.976	0.927	0.862
GF	1.779	1.856	0.754	1.485	1.401	2.683	1.108	3.592	1.340	1.236
F1-score	0.822	0.630	0.576	0.828	0.504	0.962	0.390	0.970	0.880	0.810
NNhte										
Testing Accuracy	0.764	0.571	0.759	0.768	0.551	**0.955**	0.532	**0.958**	0.898	0.818
Training Accuracy	0.831	0.790	0.676	0.885	0.832	0.971	0.774	0.998	0.963	0.879
GF	1.406	2.048	0.744	2.024	2.677	1.575	2.073	20.900	2.785	1.502
F1-score	0.872	0.652	0.592	0.782	0.550	**0.996**	0.502	**1.000**	**0.890**	0.826
HTEsm										
Testing Accuracy	**0.786**	0.632	0.761	0.771	0.555	0.949	0.521	0.940	0.898	0.824
Training Accuracy	0.849	0.813	0.727	0.888	0.845	0.973	0.755	0.979	0.956	0.876
GF	1.437	1.973	0.877	2.036	2.884	1.870	1.958	2.854	2.299	1.418
F1-score	0.922	0.690	0.654	0.850	0.566	0.990	**0.506**	0.980	**0.890**	0.826
HTEdf										
Testing Accuracy	**0.786**	0.634	**0.764**	0.800	**0.556**	0.950	**0.534**	0.941	0.898	0.827
Training Accuracy	0.850	0.787	0.716	0.902	0.843	0.973	0.771	0.985	0.967	0.876
GF	1.443	1.720	0.833	2.051	2.828	1.870	2.039	3.963	3.061	1.380
F1-score	**0.930**	**0.706**	0.672	**0.866**	0.606	0.991	0.504	0.990	**0.890**	**0.830**

The generalization performance of the *k*NNE and *k*NNhte are also unreliable following the NBE and NBhte. However, the NBhte and *k*NNhte are more accurate than the NBE and *k*NNE due to the benefit of the mixtures of heterogeneous experts from different control parameter configurations.

The high average ranks of the NBE, *k*NNE, DTE, SVME, and NBhte compared to the HTEdf in Table 8.27 indicate that these ensembles were affected the most by the outlier severities across the datasets. The DT algorithms have been shown to be robust to outliers (Breiman et al., 1993; Song and Lu, 2015; Nyitrai and Virag, 2019). However, the generalization performance of the DTE overall datasets revealed that the DTE showed sensitivity to the outlier severities in the study due to the issues stated previously in the number of outliers study in relation to the findings of Sebban et al. (2000), Ch'ng and Mahat (2020), and Zeng and Cheng (2021).

The average ranks of the NBE and NBhte indicate the possibility that the outlier severities influence the shape of the Gaussian distribution assumed by the base experts within the NBE and NBhte. Because Gaussian NB algorithms obtain the mean vectors from maximum likelihood (Ahmed et al., 2017), the base experts within these ensembles showed the possibility that the computed mean vectors are affected by the outlier severities resulting in low generalization performance compared to other ensembles. Also, the average ranks of the *k*NNhte and *k*NNE indicate that the *k*NNhte demonstrated slight sensitivity to outlier severities compare to the *k*NNE, due to the benefit of the mixtures of heterogeneous experts obtained from different control parameter configurations of the base learners in the *k*NNhte.

Also, the *k*NNE showed more overfitting of the training dataset than the *k*NNhte, and is ranked among the least accurate ensembles for most datasets, including Sonar, Indian Liver, Red Wine, White Wine, Nursery, and Bank Marketing datasets. In addition, the performance of the *k*NNE and SVME is attributed to the fact that these ensembles were constructed using the same configuration for the base learners. Thus, the *k*NNE and SVME struggled to capture the relationship between the input features and target labels due to the outlier severities.

The generalization performance of the ensembles on low (2.0 and 2.5), mild (3.0), and extreme (3.5 and 4.0) levels of outlier severities is summarized in Appendix C. The

HTEdf and HTEsm jointly outperformed other ensembles across the levels (low, mild, and extreme) of outlier severities on four datasets, i.e. Sonar, Indian Liver, Red Wine, and White Wine datasets. Also, while the HTEdf performed best on low outlier severity for the Censor Income dataset, the SVMhte is the best for extreme outlier severity. Both ensembles achieved equal generalization performance for mild outlier severity.

Further, the SVMhte and *k*NNhte are the most accurate ensembles for the Breast Cancer and Credit Approval datasets across the levels of the outlier severities. The NNhte outperformed other ensembles for the Car Evaluation and Nursery datasets across the outlier severity levels. However, the HTEdf offered the same generalization performance as the NNhte when the outlier severity was low for the Car Evaluation dataset.

Observing the training accuracy in Table 8.26, all ensembles achieved competitive training performance, illustrating mixed results across the datasets. However, the NBE, NBhte, SVME, and SVMhte seemed to struggle with respect to training accuracy for multi-class classification problems with more labels in the Red Wine and White datasets. The training accuracies of the NBE and NBhte on the Red Wine and White Wine datasets indicate that the ensembles experienced a problem of overfitting of the training dataset, resulting in poor generalization performance. The SVME and SVMhte also showed similar behaviour as the NBE and NBhte on both datasets, leading to low generalization performance.

The GFs of the ensembles showed that all ensembles achieved slight overfitting of the training dataset for most datasets except for the *k*NNE, RF, and SVME which showed severe overfitting of the training dataset on the Car Evaluation dataset. The NNE and NNhte also experienced severe overfitting on the Nursery dataset. On the contrary, there were datasets where all ensembles developed from the mixtures of heterogeneous experts, i.e. NBhte, *k*NNhte, DThte, SVMhte, NNhte, HTEsm, and HTEdf, showed no indication of overfitting. This is reflected in the Sonar, Indian Liver, and Bank Marketing datasets. Also, four ensembles of the homogeneous mixtures, i.e. NBE, *k*NNE, DTE, and SVME showed no indication of overfitting on the same datasets.

For the F1-score, The HTEdf demonstrated superiority over other ensembles by achieving the highest F1-score for five datasets, while the RF, DThte, NNhte, and HTEsm offered the best F1-score for other datasets. Although it can be observed that a number of ensembles achieved equal F1-scores with the HTEdf for the Bank Marketing dataset, the

HTEdf and HTEsm are ranked as the second best-performing ensembles for five and three datasets, respectively. Hence, the F1-score performance of these ensembles indicates that the ensembles developed as a mixture of heterogeneous experts are more accurate to classify the majority and minority samples in the experimental datasets than the pure homogeneous mixtures.

The outcome of all performance measures provides evidence to conclude that the mixtures of heterogeneous experts obtained a superior advantage over homogeneous mixtures to learn and generalize on datasets with different severities of outliers in the training dataset.

Statistical Tests

This section compares the generalization performance of the HTEs and homogeneous ensembles across the severities of outliers over the 10 classification datasets.

Friedman Test

The Friedman test used to compare the generalization performance of the 13 ensembles for the outlier severities over the 10 datasets is discussed in Section 8.2. Based on the severities of outliers considered in each dataset, i.e. 2σ to 4σ, the mean average of the generalization performance of each ensemble across outlier severities is computed. The computed mean average is then ranked according to the Friedman test as shown in Table 8.27.

The HTEdf is the best performing ensemble achieving the lowest average ranking of 2.25, followed by the HTEsm (3.65), NNhte (4.10) and SVMhte (4.40). The average ranks of the HTEdf and HTEsm further highlight the advantage of the mixtures of heterogeneous experts obtained from the combination of different ML algorithms over other ensembles. Also, all HTEs are observed to provide lower average rankings than the pure homogeneous ensembles counterparts, which illustrate the advantage of mixtures of heterogeneous experts in comparison to homogeneous mixtures.

From the average rankings of the ensembles in Table 8.27, the calculated Friedman test statistic is $\chi_F^2 = 69.214$, and the Iman-Davenport extension of the Friedman test is computed as $F_F = 12.266$. The null hypothesis that all ensembles are equal is rejected

because the value of F_F is greater than the obtained critical value.

Table 8.27: Ranking the Generalization Performance of Ensembles over all Classification Datasets in the Severity of Outliers Study

Ensemble	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor	AVR
NBE	74.7(6)	51.7(13)	66.5(13)	53.5(13)	47.1(13)	80.9(12.5)	42.7(13)	85.4(11)	55.6(13)	59.1(13)	**12.05**
*k*NNE	58.6(13)	66.6(1)	72.7(10.5)	78.8(5)	50.1(10)	84.4(10)	47.3(11)	81.0(13)	89.3(9)	81.3(8.5)	**9.10**
DTE	62.4(12)	56.0(10)	73.7(7)	75.9(10)	47.9(12)	92.4(6)	47.7(10)	90.1(7)	88.7(11)	81.1(10)	**9.50**
RF	72.3(8)	64.0(2)	72.7(10.5)	79.7(4)	54.0(5)	88.2(8)	52.1(3.5)	89.6(8)	89.9(2)	81.3(8.5)	**5.95**
SVME	69.2(9)	62.8(6)	74.6(5)	77.8(6)	53.1(7.5)	87.2(9)	50.0(9)	86.9(9)	89.4(7.5)	80.8(11)	**7.90**
NNE	76.6(3)	55.8(11)	72.9(9)	67.4(11)	53.9(6)	94.8(4)	51.5(6)	95.3(2)	89.0(10)	81.5(7)	**6.90**
NBhte	76.2(5)	51.8(12)	66.8(12)	65.1(12)	48.9(11)	80.9(12.5)	44.9(12)	85.9(10)	83.1(12)	73.3(12)	**11.05**
*k*NNhte	65.5(11)	63.5(3)	73.0(8)	80.8(2)	53.1(7.5)	83.0(11)	50.3(8)	82.1(12)	89.7(6)	82.3(5)	**7.35**
DThte	67.7(10)	58.8(8)	74.3(6)	76.1(9)	52.6(9)	93.4(5)	51.8(5)	91.4(6)	89.4(7.5)	82.4(3.5)	**6.90**
SVMhte	72.6(7)	62.7(7)	74.7(4)	81.8(1)	55.0(4)	90.3(7)	50.5(7)	91.5(5)	90.3(1)	82.9(1)	**4.40**
NNhte	76.4(4)	57.1(9)	75.9(3)	76.8(8)	55.1(3)	95.5(1)	53.2(2)	95.8(1)	89.8(4)	81.8(6)	**4.10**
HTEsm	78.6(1.5)	63.2(5)	76.1(2)	77.1(7)	55.5(2)	94.9(3)	52.1(3.5)	94.0(4)	89.8(4)	82.4(3.5)	**3.65**
HTEdf	78.6(1.5)	63.4(4)	76.4(1)	80.0(3)	55.6(1)	95.0(2)	53.4(1)	94.1(3)	89.8(4)	82.7(2)	**2.25**

Therefore, the rejection of the null hypothesis indicates that there is a statistically significant difference in the generalization performance of the ensembles across the severities of outliers for all datasets.

Bonferroni-Dunn Test

The Bonferroni Dunn post hoc test is performed after rejecting the null hypothesis results in order to determine the ensembles that significantly differ from each other in the severity of outliers study for classification problems. The critical value is 2.87, and the computed critical difference (CD) = 4.998. The differences in performance between the HTEdf and any other ensemble is shown in the critical difference plot in Figure 8.24.

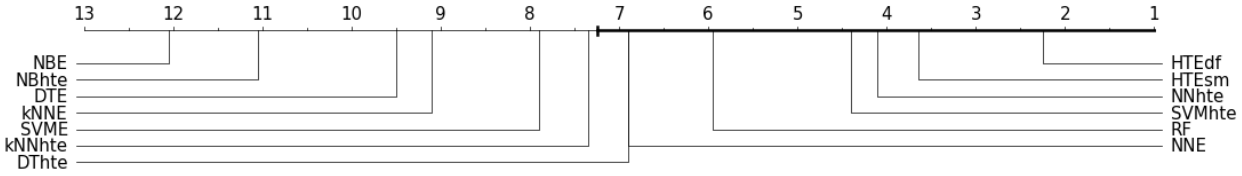

Figure 8.24: Critical Difference Plot of Ensembles for Severities of Outliers Study in Classification Problems

From Figure 8.24, the outcome of the Bonferroni-Dunn test showed that the HTEdf is significantly more accurate than the NBE, *k*NNE, DTE, SVME, NBhte, and *k*NNhte. The

difference in the average ranks of the HTEdf and *k*NNte is close to 4.998, illustrating that the *k*NNhte showed slight sensitivity to the outlier severities over all classification datasets compared to the NBE, *k*NNE, DTE, SVME, and NBhte. Also, it can be explained that the different configurations of the base *k*NN learners induced base experts that generalized to a reasonable level during prediction.

On the contrary, it was observed that the 10 experimental datasets did not provide sufficient evidence to conclude that a significant difference in generalization performance exists between the HTEdf and the HTEsm, NNhte, SVMhte, RF, DThte, and NNE.

With the exception of the NNE, the non-significant difference between the HTEdf and HTEsm, NNhte, SVMhte, RF and DThte is attributed to the fact the ensembles performed well on the outlier severities by capitalizing on the benefits of the mixtures of heterogeneous experts. The mixtures of heterogeneous experts were induced using multiple instances of the same or different ML algorithms where the instances in each ensemble consist of different control parameter configurations.

8.6 Bagged Subsets Study

This section discusses the performance of the ensembles on the different bagged subsets of the classification datasets. The bagged subsets are achieved by resampling the training samples from 10%, 20%, 30%, ... to 80%, 90%, and 100% with replacement. The discussion considers the performance of the ensembles on different input regions of the sample space in the training dataset, i.e. small (10-30%), medium (40-60%) and large (70-100%) bagged subsets. Also, the ensemble performance with respect to the bias-variance tradeoff is discussed. Table 8.28 provides the summative results (over all bagged subsets) of the testing and training accuracy, GF and F1-score of the ensembles over all classification datasets.

The results of the ensembles in Table 8.28 clearly illustrate that the mixtures of heterogeneous experts achieved better generalization performance than homogeneous mixtures. The results showed that when the ensembles were trained on different subsets of the training dataset, the HTEdf outperformed other ensembles on five of the 10 datasets, while being ranked as the second most accurate ensemble on four datasets

except for the Credit Approval dataset.

Table 8.28: Ensemble Results over all Classification Datasets in Bagged Subsets Study

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBE										
Testing accuracy	0.747	0.517	0.665	0.535	0.468	0.819	0.448	0.842	0.556	0.627
Training accuracy	0.749	0.605	0.677	0.530	0.549	0.845	0.438	0.884	0.746	0.544
GF	1.069	1.235	1.052	1.011	1.192	1.170	0.983	1.372	1.798	0.826
F1-Score	0.714	0.467	0.541	0.500	0.509	0.830	0.407	0.859	0.798	0.798
*k*NNE										
Testing accuracy	0.586	**0.666**	0.730	0.788	0.551	0.834	0.503	0.816	0.893	0.805
Training accuracy	0.737	0.752	0.680	0.876	0.698	0.922	0.516	0.851	0.910	0.813
GF	1.669	1.404	0.856	1.741	1.537	2.510	1.028	1.360	1.236	1.048
F1-Score	0.723	0.576	0.614	0.822	0.542	0.740	0.511	0.791	0.861	0.867
DTE										
Testing accuracy	0.629	0.563	0.743	0.759	0.494	0.926	0.509	0.896	0.885	0.812
Training accuracy	0.848	0.782	0.787	0.842	0.701	0.942	0.499	0.948	0.932	0.828
GF	2.771	2.199	1.282	1.586	1.766	1.381	0.986	2.407	1.752	1.110
F1-Score	0.709	0.626	0.678	0.799	0.523	0.919	0.513	0.931	0.884	0.883
RF										
Testing accuracy	0.712	0.635	0.735	0.793	0.574	0.879	0.566	0.902	0.898	0.809
Training accuracy	0.830	0.822	0.823	0.868	0.736	0.960	0.566	0.953	0.930	0.864
GF	1.954	2.280	1.705	1.578	1.810	3.832	1.011	2.991	1.556	1.470
F1-Score	0.729	0.608	**0.722**	0.817	**0.568**	0.902	0.574	0.923	0.886	0.885
SVME										
Testing accuracy	0.727	0.630	0.744	0.775	0.540	0.888	0.528	0.868	0.894	0.798
Training accuracy	0.858	0.821	0.671	0.818	0.600	0.964	0.526	0.948	0.922	0.824
GF	2.206	2.677	0.809	1.378	1.158	4.302	0.996	4.302	1.415	1.152
F1-Score	0.740	0.511	0.649	0.792	0.512	0.874	0.485	0.904	0.840	0.840
NNE										
Testing accuracy	0.766	0.551	0.728	0.675	0.566	0.942	0.556	0.943	0.891	0.808
Training accuracy	0.883	0.814	0.755	0.802	0.666	0.968	0.560	0.968	0.928	0.846
GF	2.231	2.808	1.126	1.665	1.344	2.220	1.014	2.824	1.686	1.269
F1-Score	0.798	0.616	0.671	0.744	0.558	0.936	0.564	0.960	0.884	0.884

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBhte										
Testing accuracy	0.762	0.518	0.668	0.651	0.503	0.819	0.470	0.847	0.831	0.745
Training accuracy	0.753	0.648	0.677	0.638	0.570	0.845	0.468	0.888	0.834	0.708
GF	1.017	1.383	1.043	1.012	1.166	1.170	0.998	1.377	1.025	0.885
F1-Score	0.715	0.577	0.546	0.623	0.512	0.830	0.454	0.864	0.848	0.848
*k*NNhte										
Testing accuracy	0.655	0.635	0.727	0.808	0.552	0.852	0.520	0.820	0.897	0.809
Training accuracy	0.818	0.816	0.763	0.873	0.608	0.875	0.547	0.888	0.895	0.842
GF	2.020	2.123	1.216	1.544	1.152	1.365	1.066	1.921	0.997	1.226
F1-Score	0.769	0.592	0.617	0.826	0.519	0.775	0.556	0.812	0.867	0.861
DThte										
Testing accuracy	0.681	0.588	0.756	0.761	0.556	0.927	**0.572**	0.912	0.895	0.817
Training accuracy	0.863	0.802	0.818	0.859	0.754	0.965	0.566	0.962	0.940	0.842
GF	2.658	2.390	1.491	1.738	1.999	2.814	0.999	3.542	1.889	1.174
F1-Score	0.739	0.629	0.697	0.819	0.552	0.923	**0.580**	0.943	0.890	0.887
SVMhte										
Testing accuracy	0.703	0.632	0.745	**0.818**	0.574	0.908	0.531	0.925	**0.903**	0.824
Training accuracy	0.784	0.787	0.631	0.879	0.627	0.934	0.536	0.964	0.904	0.819
GF	1.447	1.902	0.723	1.515	1.146	1.599	1.012	2.858	1.031	0.982
F1-Score	0.723	0.545	0.636	0.828	0.530	0.911	0.500	0.937	0.870	0.871
NNhte										
Testing accuracy	0.763	0.577	0.761	0.767	0.571	**0.960**	0.559	0.944	0.898	0.814
Training accuracy	0.874	0.814	0.728	0.842	0.654	0.978	0.557	0.972	0.931	0.848
GF	2.067	2.622	0.889	1.498	1.288	2.368	1.002	3.528	1.626	1.239
F1-Score	0.801	0.618	0.669	0.785	0.549	**0.969**	0.548	0.963	0.886	0.886
HTEsm										
Testing accuracy	0.797	0.645	0.745	0.771	0.562	0.942	0.567	0.952	0.898	0.819
Training accuracy	0.878	0.736	0.771	0.853	0.705	0.969	0.565	0.982	0.934	0.843
GF	1.932	1.366	1.129	1.585	1.503	2.309	1.002	5.954	1.644	1.162
F1-Score	0.800	0.653	0.668	0.818	0.561	0.951	0.560	0.978	0.892	**0.888**
HTEdf										
Testing accuracy	**0.804**	0.654	**0.768**	0.801	**0.582**	0.949	0.569	**0.962**	0.899	**0.825**
Training accuracy	0.877	0.724	0.796	0.874	0.712	0.977	0.562	0.985	0.936	0.857
GF	1.807	1.270	1.178	1.616	1.474	2.831	0.989	6.655	1.681	1.280
F1-Score	**0.824**	**0.665**	0.687	**0.838**	0.565	0.966	0.566	**0.982**	**0.893**	**0.888**

The SVMhte is the most accurate ensemble on two datasets (i.e. Credit Approval and Bank Marketing datasets), while the *k*NNE, NNNhte, and DThte offered the highest testing accuracy for the Breast Cancer, Car Evaluation, and White Wine dataset respectively.

An interesting outcome in Table 8.28 is that, unlike SVMhte, *k*NNE, NNhte, and DThte, the generalization performance of the HTEdf across the five datasets covered different characteristics and complexities ranging from datasets with small, medium, and large sample sizes to datasets with binary and multi-class labels. This outcome further illustrates that the mixtures of heterogeneous experts from different ML algorithms, where each instance of the algorithm consists of different configurations, produced different views and assumptions in the sample space that effectively generalized better than other ensembles on the test dataset.

The testing accuracies of the NBE and NBhte are similar to other modelling studies previously discussed. The NBE and NBhte achieved the worst generalization performance on eight of the 10 datasets, illustrating the possibility that the assumption of the feature independence made by the base NB learners in both ensembles is not true for the complexity of most datasets in this modelling study.

The *k*NNE and *k*NNhte also showed a similar trend in performance except for the Breast Cancer dataset, where the *k*NNE offered the highest testing accuracy of 66.6%. Another explanation for the stable generalization performance of the NBE and NBhte as well as *k*NNE and *k*NNhte being similar to other modelling studies is attributed to stability of NB and *k*NN algorithms. As reported in the empirical study of Breiman (1996a) and El-Hindi et al. (2018), when stable algorithms such as the NB and *k*NN algorithms are used as base learners in an ensemble to learn in the sample space of a training dataset, less improvement in the final ensemble prediction is often achieved. Breiman (1996a) and El-Hindi et al. (2018) explained that stable algorithms provide a small change in performance even when there is a random perturbation of sample space in the training data.

From the results in Appendix D, the HTEdf demonstrated the best generalization performance on different input space regions of the training dataset consisting of small (10-30%), medium (40-60%), and large (70-100%) subsets for five datasets (i.e. Sonar, Indian Liver, Red Wine, Nursery, and Censor Income datasets). Although, it can

be observed that the HTEsm, DThte, RF, and SVMhte achieved equal generalization performance in a number of sample space regions. While the DThte is the best ensemble for the White Wine dataset, the HTEdf competed with the DThte for small and large sample space regions. The *k*NNE and NNhte are the most accurate ensembles for all the sample space regions in the training dataset for the Breast Cancer and Car Evaluation datasets, while the SVMhte outperformed other ensembles for the Credit Approval and Bank Marketing datasets over all sample space regions.

Further, a general observation of the ensemble performance over all classification datasets revealed that the HTEdf is ranked as the most accurate and second most accurate ensemble on datasets with small sample sizes, i.e. Sonar, Breast Cancer, and Indian Liver datasets. Also, unlike other ensembles, the HTEdf is consistently ranked as the most accurate or second-most accurate ensembles on datasets with medium and large sample sizes. This is shown from the Red Wine to Censor Income dataset. The SVMhte is the only ensemble that showed competitive generalization performance with the HTEdf on large datasets as seen for the Bank Marketing and Censor Income datasets.

Illustrated by the training performance, all ensembles except the NBE achieved competitive training accuracy over all classification datasets. The NBE offered poor training accuracies of 53% and 54.4% for the Credit Approval and Censor Income datasets. The GF of the NBE indicates that the ensemble slightly overfitted the training dataset for the Credit Approval dataset, but showed no overfitting for the Censor Income dataset. However, the overfitting problem results in poor generalization performance for the Credit Approval dataset and low testing accuracy for the Censor Income dataset. All other ensembles slightly overfitted the training dataset over the classification datasets except for the Nursery dataset. The GFs of SVME, DThte, NNhte, HTEsm, and HTEdf for the Nursery dataset indicate more overfitting of the training dataset.

From the graphical results of the ensembles in Appendix D, the analysis of the ensemble performance with respect to the bias-variance tradeoff showed that the HTEs developed from the mixtures of heterogeneous experts (i.e. NBhte, DThte, *k*NNhte, DThte, SVMhte, NNhte, HTEsm, and HTEdf) generated better balance of the bias-variance tradeoff (i.e. low bias and variance errors) compared to the ensembles developed from homogeneous mixtures (i.e. NBE, *k*NNE, DTE, SVME, and NNE). The RF algorithm which is essentially

a HTE is observed to achieve competitive performance with the HTEs.

For small bagged subsets (10-30%), the HTEdf and HTEsm outperformed other ensembles to balance the bias-variance tradeoff when all classification datasets are considered. The HTEdf and HTEsm showed less overfitting on small bagged subsets compared to other ensembles, as illustrated in the graphical results in Appendix D. Also, with increasing bagged subsets (40-100%) across all datasets, the HTEdf and HTEsm still produced less overfitting of the training dataset, translating to lower bias and variance errors. This indicates that the HTEdf and HTEsm better balanced the bias-variance tradeoff than other ensembles.

A further observation of the graphical results in Appendix D reveals that the NBhte, SVMhte, and NNhte produced competitive bias-variance tradeoff performance with the HTEdf and HTEsm by generating less overfitting for small (10-30%) subsets for nine datasets, except for the Breast Cancer dataset. However, while the NBhte, SVMhte, and NNhte still produced less overfitting for medium bagged subsets (40-60%) for a number of datasets, the NBhte, SVMhte, NNhte and other ensembles showed much overfitting for large bagged subsets (70-100%) across all datasets compared to the HTEdf and HTEsm. As a result, the generalization performance of the HTEdf and HTEsm on the bias-variance tradeoff further substantiates the advantage of the mixtures of heterogeneous experts from the combination of different ML algorithms.

For the prediction of the minority and majority samples across the bagged subsets over the 10 datasets, the HTEdf achieved the highest F1-scores for six of the 10 datasets consisting of different characteristics and complexities from small, medium, and large sample sizes to binary and multi-class labels. The RF algorithm is the best performing ensemble for two datasets, i.e. Indian Liver and Red Wine datasets. The DThte and NNhte offered the highest F1-scores for the Car Evaluation and White Wine datasets. Moreso, it can be observed that the HTEdf and HTEsm are further ranked as the second-best performing ensembles for two and three datasets in terms of the F1-score.

Therefore, the outcome of all performance measures provides evidence to conclude that the mixtures of heterogeneous experts performed better than the pure homogeneous mixtures to learn and generalize on datasets with different bagged subsets of the training dataset. Specifically, the outcome provides the realization that it is beneficial to exploit the

combination of different ML algorithms to construct a mixture of heterogeneous experts.

Statistical Tests

This section discusses the statistical comparison of the generalization performance of the developed ensembles across the bagged subsets over all classification datasets.

Friedman Test

The Friedman test used to compare the generalization performance of the 13 ensembles for the bagged subsets over the 10 classification datasets is discussed in Section 8.2. For each dataset, the mean average of the generalization performance of each ensemble for all bagged subsets is calculated. Then the computed mean average of the ensembles are ranked according to the Friedman test as provided in Table 8.29.

Table 8.29: Ranking the Generalization Performance of Ensembles over all Classification Datasets in the Bagged Subsets Study

Ensemble	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor	AVR
NBE	74.7(6)	51.7(13)	66.5(13)	53.5(13)	46.8(13)	81.9(12.5)	44.8(13)	84.2(11)	55.6(13)	62.7(13)	**12.05**
*k*NNE	58.6(13)	66.6(1)	73.0(9)	78.8(5)	55.1(8)	83.4(11)	50.3(11)	81.6(13)	89.3(9)	80.5(10)	**9.00**
DTE	62.9(12)	56.3(10)	74.3(7)	75.9(10)	49.4(12)	92.6(6)	50.9(10)	89.6(8)	88.5(11)	81.2(6)	**9.20**
RF	71.2(8)	63.5(4.5)	73.5(8)	79.3(4)	57.4(2.5)	87.9(9)	56.6(4)	90.2(7)	89.8(4)	80.9(7.5)	**5.85**
SVME	72.7(7)	63.0(7)	74.4(6)	77.5(6)	54.0(9)	88.8(8)	52.8(8)	86.8(9)	89.4(8)	79.8(11)	**7.90**
NNE	76.6(3)	55.1(11)	72.8(10)	67.5(11)	56.6(4)	94.2(3.5)	55.6(6)	94.3(4)	89.1(10)	80.8(9)	**7.15**
NBhte	76.2(5)	51.8(12)	66.8(12)	65.1(12)	50.3(11)	81.9(12.5)	47.0(12)	84.7(10)	83.1(12)	74.5(12)	**11.05**
*k*NNhte	65.5(11)	63.5(4.5)	72.7(11)	80.8(2)	55.2(7)	85.2(10)	52.0(9)	82.0(12)	89.7(6)	80.9(7.5)	**8.00**
DThte	68.1(10)	58.8(8)	75.6(3)	76.1(9)	55.6(6)	92.7(5)	57.2(1)	91.2(6)	89.5(7)	81.7(4)	**5.90**
SVMhte	70.3(9)	63.2(6)	74.5(4.5)	81.8(1)	57.4(2.5)	90.8(7)	53.1(7)	92.5(5)	90.3(1)	82.4(1.5)	**4.45**
NNhte	76.3(4)	57.7(9)	76.1(2)	76.7(8)	51.4(10)	96.0(1)	55.9(5)	94.4(3)	89.8(4)	81.5(5)	**5.10**
HTEsm	79.7(2)	64.5(3)	74.5(4.5)	77.1(7)	56.2(5)	94.2(3.5)	56.7(3)	95.2(2)	89.8(4)	81.9(3)	**3.70**
HTEdf	80.4(1)	65.4(2)	76.8(1)	80.1(3)	58.2(1)	94.9(2)	56.9(2)	96.2(1)	89.9(2)	82.4(1.5)	**1.65**

The HTEdf is the best performing ensemble in terms of generalization performance achieving the lowest average ranking of 1.65 across all datasets. The HTEsm (3.70) and SVMhte (4.45) are the second and third best performing ensembles over all datasets. The average rankings of all HTEs showed that it is preferable to develop mixtures of heterogeneous experts compare to homogeneous mixtures for the bagged subsets study, especially when the mixtures of heterogeneous experts is obtained using different ML algorithms to construct HTEdf and HTEsm.

From Table 8.29, the calculated Friedman test statistic χ_F^2 from the average rankings of the ensembles is 69.056, while the Iman-Davenport extension of the Friedman test is computed as F_F = 12.199. The value of F_F is greater than the obtained critical value, and thus, the null hypothesis that all ensembles are equal is rejected. The rejection of the null hypothesis shows that, there is a statistically significant difference in the generalization performance of the ensembles across the bagged subsets for all datasets.

Bonferroni-Dunn Test

The Bonferroni-Dunn post hoc test is performed after rejecting the null hypothesis in the Friedman test. The Bonferroni-Dunn test determined the ensembles that significantly differ from each other in the bagged subsets study for classification problems. The critical value is 2.87, and the computed critical difference (CD) is 4.998. Figure 8.25 illustrates the critical difference plot for the differences in performance between the HTEdf and any other ensemble.

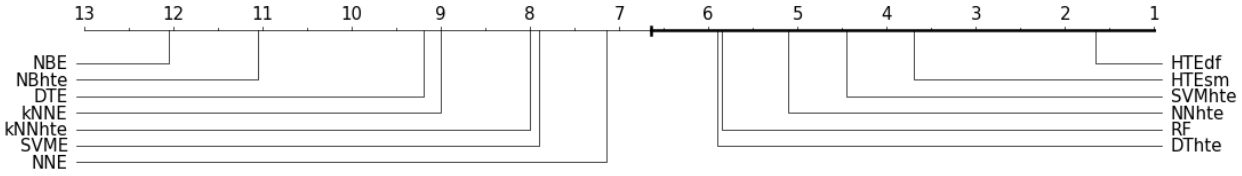

Figure 8.25: Critical Difference Plot of Ensembles for Bagged Subsets Study in Classification Problems

The outcome of the Bonferroni-Dunn test as illustrated in Figure 8.25 is that the HTEdf is significantly more accurate than all pure homogeneous ensembles, i.e. NBE, *k*NNE, DTE, SVME, and NNE, validating the advantage of the mixtures of heterogeneous experts over the pure homogeneous mixtures to learn different input regions in the sample space of each dataset. The HTEdf is also more accurate than NBhte and *k*NNhte.

On the other hand, it is observed that there is no significant difference in generalization performance between the HTEdf and the HTEsm, SVMhte, NNhte, DThte, and RF algorithm. The non-significant statistical difference between the HTEdf and HTEsm, SVMhte, NNhte, DThte, and RF algorithm showed that the ensembles induced efficient mixtures of heterogeneous experts that generalized well across all datasets.

8.7 Feature Subsets Study

This section discusses the performance of the ensembles on the different feature subsets of the classification datasets. The feature subsets are obtained by resampling the features of the training dataset from 10%, 20%, 30%, ... to 80%, 90%, and 100% with replacement. The discussion considers the performance of the ensembles on different input feature regions in the training dataset, i.e. small (10-30%), medium (40-60%), and large (70-100%) feature subsets. Also, the ensemble performance with respect to the bias-variance tradeoff is discussed. Table 8.30 provides the summative results (over all feature subsets) of the testing and training accuracy, GF and F1-score of the ensembles over all classification datasets.

The comparison of the ensembles presented in Table 8.30 illustrates that the ensembles showed different behaviours for the different feature subsets across the datasets. When the ensembles were trained on different features subsets in each dataset, the mixtures of heterogeneous experts further revealed superiority over the homogeneous mixtures of experts. The HTEdf largely dominated other ensembles based on generalization performance by achieving the highest testing accuracy for seven of the 10 datasets. Although, the HTEsm offered equal testing accuracy of 77.8% as the HTEdf for the Sonar dataset. While for the remaining datasets (i.e. Indian Liver, Car Evaluation, and Bank Marketing datasets), where the HTEdf did not achieve the highest testing accuracy, the HTEdf remarkably compensated with the second-best generalization performance on the Bank Marketing dataset and the highest F1-scores for the three datasets. Out of the three datasets, the NNhte is the most accurate ensemble for the Indian Liver and Car Evaluation datasets, while the SVMhte performed best for the Bank Marketing dataset.

A remarkable outcome of the HTEdf is that, unlike the NNhte and HTEsm, the generalization performance of the HTEdf across the seven datasets covered different characteristics and complexities. The characteristics and complexities range from datasets with small, medium and large sample sizes to datasets with small, medium and large feature sizes. This outcome further indicates that the mixtures of heterogeneous experts from different ML algorithms, where each instance of the algorithm consists of different configurations, produced different views and assumptions in the feature space that

effectively generalized better than other ensembles on the test dataset.

Table 8.30: Ensemble Results over all Classification Datasets in Feature Subsets Study

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBE										
Testing accuracy	0.710	0.490	0.632	0.513	0.466	0.595	0.436	0.719	0.671	0.454
Training accuracy	0.674	0.573	0.654	0.524	0.547	0.548	0.479	0.752	0.731	0.678
GF	0.894	1.203	1.070	1.033	1.184	0.904	1.083	1.140	1.245	1.772
F1-Score	0.740	0.492	0.484	0.432	0.479	0.641	0.407	0.695	0.753	0.562
*k*NNE										
Testing accuracy	0.641	0.611	0.727	0.711	0.483	0.747	0.446	0.719	0.892	0.773
Training accuracy	0.756	0.692	0.676	0.796	0.587	0.766	0.512	0.726	0.849	0.755
GF	1.489	1.327	0.846	1.501	1.266	1.172	1.140	1.238	0.726	1.040
F1-Score	0.827	0.594	0.636	0.740	0.505	0.729	0.511	0.742	0.811	0.745
DTE										
Testing accuracy	0.620	0.578	0.700	0.691	0.469	0.775	0.461	0.806	0.883	0.777
Training accuracy	0.719	0.689	0.716	0.774	0.561	0.791	0.526	0.816	0.889	0.781
GF	1.370	1.408	1.062	1.430	1.218	1.171	1.141	1.262	1.207	1.063
F1-Score	0.744	0.587	0.671	0.724	0.519	0.734	0.513	0.780	0.862	0.739
RF										
Testing accuracy	0.699	0.626	0.717	0.713	0.499	0.767	0.484	0.795	0.889	0.784
Training accuracy	0.739	0.719	0.751	0.772	0.615	0.785	0.595	0.818	0.921	0.799
GF	1.171	1.397	1.146	1.357	1.318	1.211	1.289	1.486	1.588	1.156
F1-Score	0.814	0.591	0.692	0.743	0.555	0.717	0.577	0.773	0.875	0.750
SVME										
Testing accuracy	0.689	0.631	0.735	0.708	0.496	0.772	0.487	0.788	0.892	0.787
Training accuracy	0.701	0.724	0.657	0.785	0.560	0.806	0.518	0.824	0.904	0.794
GF	1.090	1.475	0.778	1.438	1.150	1.524	1.068	1.873	1.959	1.109
F1-Score	0.700	0.564	0.633	0.753	0.491	0.719	0.500	0.778	0.836	0.751
NNE										
Testing accuracy	0.765	0.589	0.716	0.659	0.507	0.789	0.484	0.814	0.883	0.786
Training accuracy	0.814	0.723	0.689	0.748	0.595	0.806	0.561	0.829	0.915	0.797
GF	1.307	1.572	0.922	1.393	1.239	3.063	1.191	1.601	2.058	1.130
F1-Score	0.799	0.604	0.635	0.682	0.539	0.740	0.564	0.788	0.857	0.758

Measure	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor
NBhte										
Testing accuracy	0.725	0.497	0.651	0.554	0.479	0.600	0.451	0.731	0.853	0.626
Training accuracy	0.678	0.606	0.656	0.626	0.553	0.564	0.487	0.760	0.756	0.745
GF	0.858	1.291	1.024	1.197	1.170	0.921	1.072	1.147	0.651	1.541
F1-Score	0.736	0.531	0.489	0.545	0.481	0.646	0.454	0.709	0.809	0.673
*k*NNhte										
Testing accuracy	0.690	0.623	0.724	0.694	0.487	0.740	0.465	0.725	0.890	0.778
Training accuracy	0.783	0.707	0.689	0.771	0.549	0.762	0.558	0.726	0.878	0.753
GF	1.467	1.396	0.890	1.404	1.145	1.181	1.226	1.188	0.948	1.024
F1-Score	0.877	0.573	0.639	0.747	0.553	0.707	0.556	0.752	0.829	0.752
DThte										
Testing accuracy	0.661	0.589	0.723	0.696	0.497	0.782	0.493	0.809	0.887	0.785
Training accuracy	0.761	0.704	0.750	0.785	0.621	0.799	0.604	0.824	0.918	0.794
GF	1.458	1.442	1.118	1.500	1.345	1.238	1.296	1.448	1.569	1.105
F1-Score	0.772	0.608	0.691	0.743	**0.560**	0.730	**0.580**	0.785	0.876	0.747
SVMhte										
Testing accuracy	0.728	0.632	0.741	0.722	0.509	0.783	0.473	0.807	**0.898**	0.790
Training accuracy	0.796	0.703	0.629	0.796	0.600	0.811	0.506	0.823	0.836	0.793
GF	1.378	1.308	0.710	1.380	1.243	1.251	1.069	1.538	0.815	1.080
F1-Score	0.834	0.614	0.613	0.747	0.487	0.727	0.485	0.786	0.856	0.756
NNhte										
Testing accuracy	0.761	0.601	**0.751**	0.691	0.519	**0.793**	0.499	0.815	0.891	0.786
Training accuracy	0.793	0.717	0.686	0.772	0.596	0.810	0.557	0.828	0.908	0.793
GF	1.175	1.508	0.798	1.401	1.215	1.958	1.144	1.462	1.698	1.105
F1-Score	0.858	0.610	0.644	0.725	0.530	0.742	0.543	0.793	0.865	0.758
HTEsm										
Testing accuracy	**0.778**	0.630	0.715	0.727	0.513	0.786	0.496	0.828	0.893	0.793
Training accuracy	0.802	0.696	0.717	0.784	0.610	0.802	0.562	0.834	0.912	0.797
GF	1.157	1.263	1.012	1.347	1.263	1.254	1.158	1.682	1.586	1.097
F1-Score	0.888	0.643	0.685	0.749	0.533	0.762	0.549	**0.805**	0.872	0.760
HTEdf										
Testing accuracy	**0.778**	**0.639**	0.737	**0.732**	**0.524**	0.788	**0.501**	**0.828**	0.895	**0.794**
Training accuracy	0.804	0.688	0.732	0.780	0.569	0.804	0.574	0.834	0.904	0.785
GF	1.158	1.193	0.979	1.296	1.113	1.298	1.179	1.638	1.352	1.018
F1-Score	**0.896**	**0.659**	**0.703**	**0.776**	0.547	**0.768**	0.561	0.800	**0.878**	**0.771**

On the contrary, the NBE and NBhte are the least accurate ensembles across the classification datasets, illustrating that the NBE and NBhte struggled with the characteristics of most datasets across the different feature subsets when compared to other ensembles. However, the prediction behaviour of the NBE and NBhte in this modelling study is surprising compared to other modelling studies. The NBE and NBhte produced different testing accuracies across the datasets rather than equal testing accuracies for all datasets in other modelling studies. While a clearer explanation as to why the NBE and NBhte showed this prediction behaviour requires more in-depth investigation, it is evident that both ensembles learned differently from different input regions of the feature spaces in all datasets producing diverse base experts.

The testing accuracies of the *k*NNE and *k*NNhte are also unreliable for most datasets which are illustrated by the high average ranks of the ensembles in Table 8.31. The inadequacies in the generalization performance of *k*NNE and *k*NNhte for most datasets are possibly due to the multicollinearity in the feature space of the test dataset during prediction. This outcome leads to overfitting and poor generalization observed for most of the datasets.

Appendix E presents the generalization performance of the ensemble on different input regions of the feature space, i.e. small (10-30%), medium (40-60%), and large (70-100%) subsets in the training dataset across all datasets. It can be observed that the HTEdf and HTEsm did not win outrightly on small feature subsets, because the datasets where the HTEdf and HTEsm produced reliable performance include Credit Approval, Car Evaluation, White Wine, and Censor Income datasets. Most ensembles, especially the NNE, DThte, SVMhte, and NNhte, achieved competitive generalization performance as the HTEdf and HTEsm in the small feature subsets.

Additionally, there were a number of datasets where the SVMhte and NNhte outperformed the HTEdf and HTEsm as shown for the Sonar, Breast Cancer, Indian Liver, Nursery, and Bank Marketing datasets. For medium and large feature subsets, the HTEdf and HTEsm showed significant improvement offering the best performance on most datasets, including Sonar, Credit Approval, Red Wine, White Wine, Nursery and Censor Income datasets. Although, the SVMhte and NNhte still rivalled the HTEdf and HTEsm for most datasets, such as the Credit Approval, Car Evaluation, and Bank

Marketing datasets.

The generalization performance of the SVME and SVMhte validates the suitability of the base SVM algorithms to different feature dimensionalities of the experimental data in this modelling study. The SVME and SVMhte illustrate the possibility that the configurations of the members of the ensembles effectively mapped the input space to a higher dimensional space in the training dataset given the feature subsets. This outcome leads to a good level of generalization performance. The SVMhte specifically achieved competitive prediction performance with the HTEdf for six of the 10 datasets (i.e. Breast Cancer, Indian Liver, Credit Approval, Car Evaluation, Bank Marketing, and Censor Income datasets).

For the NNhte, the combination of different control parameter configurations for the multiple instances of the base NN learners in the NNhte resulted in efficient base experts that generalize well across the experimental datasets. Also, the configuration of the NNE delivered reasonable diverse assumptions during prediction that offered reliable generalization performance for a number of datasets. The configurations of the NNhte and NNE highlights the possibility that both ensembles learned and converged effectively to a suitable solution for most datasets.

The generalization performance of the DTE and DThte indicates that both ensembles learned the inherent noise in the training dataset resulting in severe overfitting for seven of the 10 datasets. This is shown for the Sonar, Indian Liver, Credit Approval, Red Wine, Car Evaluation, Nursery, and Bank Marketing datasets. However, the DThte is more accurate than the DTE due to the advantage of the mixtures of heterogeneous experts obtained using different configurations for the base learners within the DThte. The combination of the information gain and Gini index for feature selection benefited the DThte to learn and generalize well on the mixture of categorical and numerical features for most of the datasets, which is desirable. It is observed that the use of one feature selection criterion for the DTE did not yield significant benefit to the DTE.

Illustrated by the training performance, all ensembles achieved competitive training accuracy across all datasets. The GFs of the ensembles further showed that all ensembles slightly overfitted for most of the datasets. Specifically, five ensembles developed from the mixtures of heterogeneous experts, i.e. NBhte, *k*NNhte, SVMhte, NNhte, and HTEdf,

showed no indication of overfitting the training dataset as shown for the Sonar, Indian Liver, Car Evaluation, and Bank Marketing datasets. On the other hand, three ensembles obtained from the mixtures of homogeneous experts, i.e. NBE, *k*NNE, and NNE, showed no indication of overfitting the training dataset for the same datasets. Hence, the mixtures of heterogeneous experts performed better than the homogeneous mixtures of experts for the datasets in terms of the GFs.

The graphical results of the ensembles on the 10 datasets in Appendix E showed the ensemble performance with respect to the bias-variance tradeoff. The HTEdf consistently showed less overfitting of the training dataset for small feature subsets (10-30%) over the 10 datasets compared to other ensembles. The HTEsm is the second-best performing ensemble, achieving less overfitting of the training dataset for small feature subsets on nine of 10 datasets (except for the Red Wine dataset). It can be observed that a number of ensembles, including SVMhte, NNhte, NBhte, *k*NNhte, and RF algorithm, offered competitive performance as the HTEdf on small feature subsets.

Also, with an increase from medium (40-60%) to large (70-100%) feature subsets, the HTEdf and HTEsm showed superiority by providing less overfitting than other ensembles over all datasets. Moreso, a further observation revealed that a number of ensembles competed with the HTEdf over these input regions of the feature space across the datasets. However, another outcome is the downside to the performance of the NBE for the Bank Marketing dataset. The NBE produced inconsistent generalization performance across most datasets. Hence, the HTEdf and HTEsm are the best ensembles that showed a better balance of the bias-variance tradeoff over all datasets compared to other ensembles. This outcome validates the advantage of the mixtures of heterogeneous experts from different ML algorithms to other mixtures of the same ML algorithm with similar or different control parameter configurations.

Furthermore, the HTEdf achieved excellent performance based on the highest F1-scores to outperform other ensembles for seven of the 10 datasets, illustrating the dominance of the HTEdf over other ensembles. The seven datasets where the HTEdf is ranked as the best performing ensemble consists of different characteristics and complexities from the small, medium, and large sample size, to small, medium, and large feature size, as well as binary and multi-class labels. The DThte offered the highest F1-scores of 56% and 58%

for the Red Wine and White Wine datasets, while the HTEsm achieved the best F1-score of 80.5% for the Nursery datasets.

Thus, the outcome of all performance measures provides evidence to conclude that the HTEdf and HTEsm learned and generalize well on the different feature subsets of the training dataset over all classification datasets compare to other mixtures of experts.

Statistical Tests

This section compares the generalization performance of the developed ensembles across all feature subsets over all classification datasets.

Friedman Test

The discussion of the Friedman test is provided in Section 8.2, and the test is used to compare the generalization performance of the 13 ensembles for the feature subsets over the 10 datasets in this modelling study.

The mean average of the generalization performance for all feature subsets for each ensemble is calculated for each dataset. Then the computed mean averages of the ensembles are ranked according to the Friedman test as provided in Table 8.31.

Table 8.31: Ranking the Generalization Performance of the Ensembles over all Classification Datasets in the Feature Subsets Study

Ensemble	Sonar	Breast	Indian	Credit	Red	Car	White	Nursery	Bank	Censor	AVR
NBE	71.0(7)	49.0(13)	63.2(13)	51.3(13)	46.6(13)	59.5(13)	43.6(13)	71.9(12.5)	67.1(13)	45.4(13)	**12.35**
*k*NNE	64.1(12)	61.1(7)	72.7(5)	71.1(5)	48.3(10)	74.7(10)	44.6(12)	71.9(12.5)	89.2(4.5)	77.3(11)	**8.90**
DTE	62.1(13)	57.8(11)	70.0(11)	69.1(9.5)	46.9(12)	77.5(7)	46.1(10)	80.6(7)	88.3(10.5)	77.7(10)	**10.10**
RF	69.9(8)	62.7(5)	71.7(8)	71.3(4)	49.9(6)	76.7(9)	48.4(6.5)	79.5(8)	88.9(8)	78.4(8)	**7.05**
SVME	68.9(10)	63.1(3)	73.5(4)	70.8(6)	49.6(8)	77.2(8)	48.7(5)	78.8(9)	89.2(4.5)	78.7(4)	**6.15**
NNE	76.5(3)	58.9(9.5)	71.6(9)	65.9(11)	50.7(5)	78.9(2)	48.4(6.5)	81.4(4)	88.3(10.5)	78.6(5.5)	**6.60**
NBhte	72.5(6)	49.7(12)	65.1(12)	55.4(12)	47.9(11)	60.0(12)	45.1(11)	73.1(10)	85.3(12)	62.6(12)	**11.00**
*k*NNhte	69.0(9)	62.3(6)	72.4(6)	69.4(8)	48.7(9)	74.0(11)	46.5(9)	72.5(11)	89.0(7)	77.8(9)	**8.50**
DThte	66.1(11)	58.9(9.5)	72.3(7)	69.5(7)	49.7(7)	78.2(6)	49.3(4)	80.9(5)	88.7(9)	78.5(7)	**7.25**
SVMhte	72.8(5)	63.2(2)	74.1(2)	72.2(3)	50.9(4)	78.3(5)	47.3(8)	80.7(6)	89.8(1)	79.0(3)	**3.90**
NNhte	76.1(4)	60.1(8)	75.1(1)	69.1(9.5)	51.9(2)	79.3(1)	49.9(2)	81.5(3)	89.1(6)	78.6(5.5)	**4.20**
HTEsm	77.8(1.5)	63.0(4)	71.5(10)	72.7(2)	51.3(3)	78.6(4)	49.6(3)	82.8(1.5)	89.3(3)	79.3(2)	**3.40**
HTEdf	77.8(1.5)	63.9(1)	73.7(3)	73.2(1)	52.4(1)	78.8(3)	50.1(1)	82.8(1.5)	89.5(2)	79.4(1)	**1.60**

The HTEdf is ranked as the most accurate ensemble across all datasets by achieving the lowest average ranking of 1.60. The HTEsm (3.40) and SVMhte (3.90) are ranked as the

second and third most accurate ensembles over all datasets. The average rankings of the ensembles shows that the mixtures of heterogeneous experts are more accurate than the pure homogeneous mixtures, especially the HTEdf and HTEsm constructed from the combination of different ML algorithms compared to other ensembles developed from the same ML algorithms.

From Table 8.31, the calculated Friedman test statistic χ_F^2 from the average rankings of the ensembles = 79.523, and the Iman-Davenport extension of the Friedman test is computed as F_F = 17.682. The null hypothesis that all ensembles are equal is rejected because the value of F_F is greater than the computed critical value. The rejection of the null hypothesis indicates that, there is a statistically significant difference in the generalization performance of the ensembles across the feature subsets for all datasets.

Bonferroni-Dunn Test

The Bonferroni-Dunn post hoc test is performed after rejecting the null hypothesis in the Friedman test. The Bonferroni-Dunn test determined the ensembles that significantly differ from each other in the feature subsets study for classification problems. The critical value is 2.87, and the computed critical difference (CD) is 4.998. Figure 8.26 illustrates the critical difference plot illustrating the significant differences in generalization performance between the HTEdf and any other ensemble.

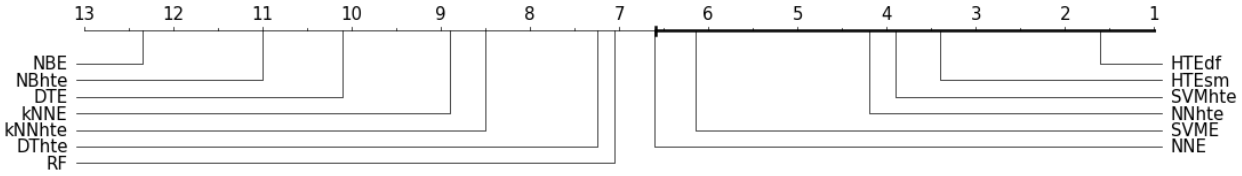

Figure 8.26: Critical Difference Plot of Ensembles for Feature Subsets Study in Classification Problems

The result of the Bonferroni-Dunn test in Figure 8.26 indicates that the HTEdf is significantly more accurate than the NBE, *k*NNE, DTE, NBhte, *k*NNhte, DThte, and RF algorithm. The difference in generalization performance between the HTEdf and these ensembles showed the advantage of the mixture of heterogeneous experts developed from the combination of different ML algorithms to learn different input regions in the feature space of each dataset compared to ensembles developed using the same ML

algorithm with the same configuration.

This trend in statistical difference in generalization performance also applies to the HTEsm and these ensembles because the HTEsm is significantly more accurate than these ensembles. It is observed that the experimental data did not provide sufficient evidence to conclude that a significant difference in generalization performance exists between the HTEdf and HTEsm, SVMhte, NNhte, SVME, and NNE.

8.8 Discussion of Results

This section presents an overall discussion of the results of the ensembles across the six modelling studies from Section 8.2 to Section 8.7. The results showed that the effect of the *No-Free-Launch Theorem* proposed by Wolpert (1996) is reflected in each modelling study, where no one ensemble performed best on all classification datasets. The difference in the performance of the ensembles over all datasets is attributed to the inductive biases of the ML algorithms, the base experts induced from the control parameter configurations obtained to construct the ensembles, and the characteristics of the datasets being examined. Also, the performance of the ensembles on all performance measures further confirm that expert-specific inductive bias leads to different experts generalizing better on individual datasets than others.

The GF values of the ensembles indicate that, for various datasets, ensembles overfittted the training data, which often lead to low testing accuracy for a number of ensembles. However, there were cases where a number of ensembles, for instance, the NBE and NBhte, overfitted the training dataset and resulted in poor generalization performance.

While the NBhte performed better than NBE, both ensembles are the worst-performing ensembles across the six modelling studies. The performance of the NBE and NBhte is attributed to the fact that the NB algorithm is a stable algorithm (Breiman, 1996a; El-Hindi et al., 2018), and struggled with the characteristics and complexity of most datasets across the modelling studies when compared to other ensembles.

The performance of the ensembles on the clean data study in Section 8.2 illustrate the potential predictive power of the HTEdf and HTEsm to create effective mixtures of heterogeneous experts. The HTEdf achieved the best average ranking over the

10 datasets, followed by the HTEsm. The performance of the HTEdf and HTEsm corroborates the findings of Kilimci et al. (2017), Xu and Zhang (2019), and Zefrehi and Hakan (2020).

Across all datasets, the HTEdf was the most accurate ensemble on four of the 10 datasets, while achieving the second-best testing accuracy on five datasets. Statistically, the HTEdf was shown to be significantly more accurate than nine ensembles, consisting of all the pure homogeneous mixtures of experts (i.e. NBE, *k*NNE, DTE, RF, SVME, and NNE) and three mixtures of heterogeneous experts (i.e. NBhte, *k*NNhte, and DThte). The HTEdf was not significantly different to the HTEsm, SVMhte, and NNhte.

For the skewed class distribution study discussed in Section 8.3, the HTEdf expresses effectiveness based on the overall generalization performance to be ranked as the most accurate ensemble on five datasets for the extreme, mild and small skewed class distributions. When one of the classes in the training dataset was undersampled to allow skewness in the class distributions, the HTEdf and HTEsm further showed superiority in predicting the minority and minority samples than other ensembles. This is illustrated by the F1-score performance of the HTEdf over all datasets.

On the contrary, the overall generalization performance is not sufficient to justify the superiority of the HTEdf and HTEsm over other ensembles. Hence, the generalization performance of all ensembles to predict the minority class in each skewed class distribution still showed that HTEdf and HTEsm are more accurate than other ensembles. The HTEdf achieved the best average ranking over the 10 datasets and was the most accurate ensemble for five datasets.

Another outcome of the skewed class distribution study is observed in the generalization performance of the NBE and NBhte to predict the minority class as illustrated by the average ranks. While NB algorithms could be seen as weak probabilistic learners, the structure and assumptions of the NBE and NBhte supported the predictability of both ensembles on the minority classes in the study.

The average rank of the SVMhte illustrates the benefit of the mixtures of experts from different configurations compared to the SVME to perform well on the skewed class distributions across the datasets. The SVME tends to deal with the weakness of the soft margin optimization problems and imbalanced support vector ratio. An increase in

skewed class distributions tends to increase the imbalance of the ratio between positive and negative support vectors. As a result, samples at the boundary of the hyperplane are more likely to be classified as negative.

The results in the number of outliers study discussed in Section 8.4 indicate that all homogeneous mixtures of experts (i.e. NBE, *k*NNE, DTE, SVME, and NNE) and two heterogeneous mixture models (i.e. *k*NNhte and DThte) showed sensitivity to the different number of outliers in the study. This is illustrated by the average ranks of the ensembles. This outcome is also observed in the severity of outliers study, except for the NNE and DThte. The average ranks of the NNE and DThte illustrate that both ensembles were not sensitive to the different standard deviations from the estimated mean across all datasets. Also, the RF algorithm capitalized on the intrinsic ensemble approaches (i.e. bagging and RFSM) to perform well in both modelling studies.

A further investigation of the outcomes of the number and severity of outlier studies illustrate the capability of the HTEdf and HTEsm to generate consistent testing accuracy and GF on most classification datasets. This outcome indicates the possibility that the HTEdf and HTEsm efficiently learned the decision boundaries during training to achieve reliable prediction accuracy. Whereas most homogeneous ensembles struggled to learn the input features properly, which leads to poor decision boundaries. The effect is inconsistent testing accuracy and higher GFs for the ensembles compared to the HTEdf and HTEsm.

The results in the bagged and feature subset studies in Sections 8.6 and 8.7 provide important insights into the performance of ensembles on different sample and feature sizes. The average ranks of the HTEdf and HTEsm indicate that both ensembles performed better than other ensembles over all datasets. This outcome highlights the excellent capability of the HTEdf and HTEsm to learn different input regions of the sample and feature spaces across the datasets. Although, the SVMhte and NNhte also achieved competitive performance in both studies.

Observing the performance of the ensembles over the different input regions, the HTEdf achieved the best performance for small, medium and large subsets of the sample and feature spaces. An interesting analysis in the modelling studies of bagged and feature subsets is to consider the ensemble performance on small bagged and feature

subsets. This is imperative to investigate the ensembles generated and trained faster, and examine the ensemble that requires less data storage for a quick computational process. Another reason is to identify the ensemble that expresses the best diversity of base learners, which maximizes classification accuracy, and the likelihood that the errors made by the induced base experts are uncorrelated. Then, due to the uncorrelated errors, incorrect classifications can be compensated by pooling the decisions of the base experts in the ensemble. Therefore, based on these premises, a critical analysis of the ensemble performance on smaller bagged and feature subsets confirmed the dominance of the HTEdf over other ensembles.

Further, the significance of the results in the six modelling studies relates to the potential benefits of the implementations of HTEdf and HTEsm respectively. From Section 6.2, the first two types of ensembles produced the construction of 11 ensemble types (i.e. NBE, *k*NNE, DTE, RF, SVME, NNE, NBhte, *k*NNhte, DThte, SVMhte, and NNhte). This process is inefficient and results in long periods of computing time when the ensembles are trained and tested. The results described across all modelling studies indicate that instead of the inefficient implementation of training the 11 ensembles developed using the same ML algorithms, a single HTEdf or HTEsm can be implemented. This single implementation is possible due to the consistent and accurate performance of the HTEdf and HTEsm.

Additionally, the combination of different ML algorithms to develop the HTEdf and HTEsm favours the HTEdf and HTEsm to better balance the bias-variance tradeoff than other ensembles. It can be observed that the ensembles developed using the same ML algorithms achieved inconsistent training and testing errors across the bagged and feature subsets, which leads to overfitting on most datasets. However, the HTEdf and HTEsm demonstrated an advantage over other ensembles by generating lower testing and training errors across the bagged and feature subsets due to the effect of the different ML algorithms combined.

Another significant outcome is that not only is the HTEdf and HTEsm more accurate on average, but there are datasets for which the HTEdf and HTEsm are the most accurate across the modelling studies. The outcomes of the HTEdf and HTEsm on the datasets confirm that through a mixture of heterogeneous experts developed using different ML

algorithms, an ensemble can be constructed that is more accurate than any ensemble developed using the same ML algorithm. The outcome also showed that by developing the ensembles of diverse mixtures of heterogeneous experts rather than homogeneous mixtures of experts, it is possible to obtain the advantages related to the inductive biases of experts while limiting the disadvantages.

Another outcome is related to the classification problem type. It is observed that the HTEdf and HTEsm performed better on binary and multi-class problems across all six modelling studies than other ensembles. The HTEdf and HTEsm were either ranked as the most accurate ensembles or rivalled the most accurate ensemble in each modelling study. The remarkable performance of HTEdf and HTEsm for binary and multi-class classification problems is attributed to the fact that the different base experts within the HTEdf and HTEsm learned and generalized efficiently for the binary and multi-class labels compared to other ensembles. The efficient generalization results in significant probability of accurate predictions produced by the base experts that formed the HTEdf and HTEsm. This outcome is in line with the findings of Guo et al. (2018).

Lastly, to verify the effectiveness of the research, the findings of this research were compared to previous works in literature. The findings showed that the HTEdf and HTEsm achieved higher generalization performance than the HTEs proposed by Tuarob et al. (2014), Sagayaraj and Santhoshkumar (2020), Tewari and Dwivedi (2020), Zain et al. (2020), and Alshdaifat et al. (2021).

For most large classification datasets, the developed HTEdf and HTEsm also outperformed the HTE proposed in the work of Nguyen et al. (2019b). While an improvement of the work in Nguyen et al. (2019b) was achieved in Nguyen et al. (2020), the work in Nguyen et al. (2020) still recorded a higher classification error rate, translating to lower classification accuracy than the HTEdf and HTEsm in this research. Through effective analysis of the inductive biases of base algorithms, the developed HTEdf and HTEsm achieved better generalization performance than the findings in these studies. The improvement is also attributed to the efficiency in the application of the random search optimization algorithm, which reliably reduced the hyperparameter search space to select only the control parameters that contributed significantly to the generalization performance of the developed HTEs.

Also, the HTEdf and HTEsm obtained better accuracy and F1-scores than the best performing HTE proposed by Zhao et al. (2020) on imbalanced classification problems. In addition, the findings of this research further provide higher generalization performances than the best performing HTE in the work of Feng et al. (2021). The developed HTEdf and HTEsm performed well across all six modelling studies. The modelling studies for skewed class distribution, number of outliers, severity of outliers are aspects not considered in the work of Feng et al. (2021).

8.9 Chapter Summary

This section presented a summary of the work done in this chapter. The results of the developed ensembles were discussed across six modelling studies, i.e. clean data, skewed class distribution, number of outliers, severity of outliers, bagged subsets, and feature subsets. The discussion considered the four performance measures, i.e. training and testing accuracy, GF and F1-score, used in this research. The ensemble performance was also discussed with respect to the bias-variance tradeoff.

A series of statistical tests were also performed to determine if the generalization performance of the ensembles were statistically significantly different. This chapter described that the mixtures of heterogeneous experts were better than homogeneous mixtures of experts. Specifically, the HTEdf and HTEsm developed through the combination of different ML algorithms were the most accurate of the ensembles in terms of average ranking of the testing accuracy. The Friedman test suggested that there is a significant difference between the performance of the ensembles. This significant difference was confirmed through the outcome of the Bonferroni-Dunn post hoc test.

For the Bonferroni-Dunn test, the HTEdf was selected as the control ensemble across the six modelling studies because the HTEdf maximizes behavioural diversity to obtain two benefits from the mixtures of heterogeneous experts. The first benefit is achieved by capitalizing on the inductive biases of different ML algorithms intrinsically, while the second benefit takes advantage of using different control parameter configurations for the multiple instances of the different ML algorithms combined within the HTEdf.

The Bonferroni-Dunn post hoc test confirmed that the HTEdf and HTEsm were

significantly more accurate than pure homogeneous ensembles and a number of heterogeneous mixtures of experts across the six modelling studies. Also, the Bonferroni-Dunn test suggested that there is no significant difference in the performance of the HTEdf and the HTEsm, SVMhte and NNhte over the six modelling studies. Lastly, the results of the ensembles and the significance of the results were discussed.

Chapter 9

Empirical Analysis of Results for Regression Problems

9.1 Introduction

This chapter provides an empirical analysis of ensemble models for regression problems. The chapter compares the performance of the developed ensembles in the five modelling studies described in Section 7.2. All reference to the results of testing RMSE, training RMSE and GFs are mean averages of the performance measures for the regression datasets in the modelling studies.

Section 9.2 discusses the results of the ensembles for the clean data study. The results of the ensembles for the number and severity of outliers studies are described in Sections 9.3 and 9.4 respectively. Sections 9.5 and 9.6 discuss the results of the ensembles for the bagged subsets and feature subsets studies. The formal statistical tests to compare the ensembles in each modelling study are also described, which is followed by an overall discussion of the outcome of the ensemble performance in all modelling studies discussed in Section 9.7. Lastly, Section 9.8 concludes the chapter with a summary of the findings.

9.2 Clean Data Study

This section presents the results of the developed ensembles for clean regression datasets. As discussed in Section 6.2, the ensembles of NB algorithms, i.e., NBE and NBhte, were excluded for regression problems. The testing and training RMSE of each ensemble for each dataset are presented using grouped barplots. The height of the each bar illustrates the performance of the ensembles. For instance, an ensemble with the shortest barplot for the testing dataset denotes that the ensemble generated the lowest testing RMSE, indicating the best generalization performance. Otherwise, the ensemble produced the highest testing RMSE, illustrating the worst generalization performance. The results of the ensembles for each dataset are described separately.

Yacht Hydrodynamics Dataset

The problem is to predict the residuary resistance of sailing yachts at initial design stage. The testing and training RMSE of the ensembles are illustrated in Figure 9.1. Table 9.1 summarizes the results of the testing RMSE, training RMSE and GFs of the ensembles for the clean Yacht Hyrodynamics dataset.

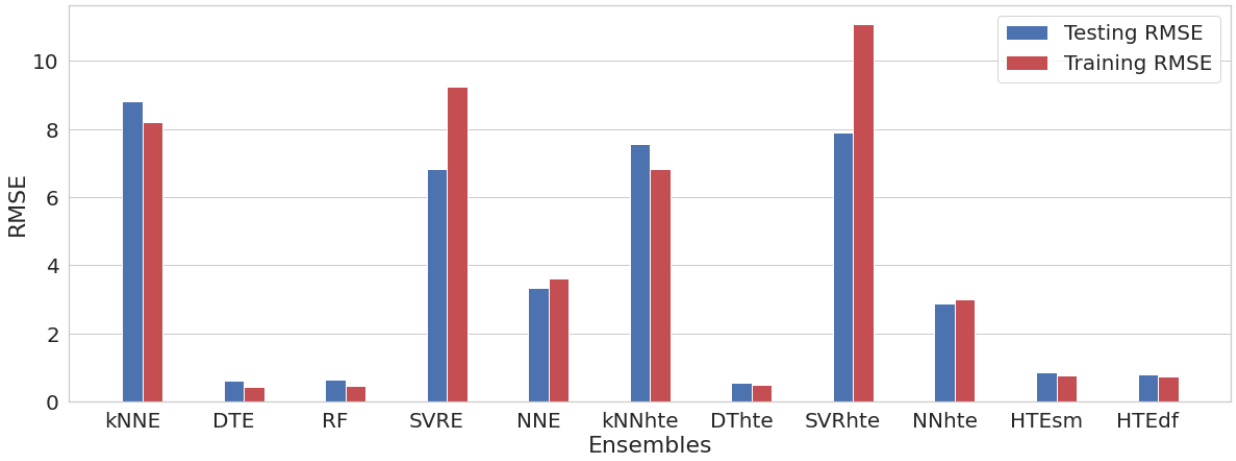

Figure 9.1: Ensemble RMSE for Clean Yacht Hydrodynamics Dataset

The results in Figure 9.1 and Table 9.1 illustrate that the complexity of the Yacht Hydrodynamics consisting of one-type input feature favours the ensembles of tree-like models DTE, RF algorithm, and DThte compared to other ensembles. The DThte outperformed other ensembles by achieving the smallest testing error of 0.55030, followed

by the DTE (0.61237) and RF algorithm (0.66274).

The HTEdf and HTEsm are ranked as the fourth and fifth most accurate ensembles in terms of testing errors. On the other hand, the generalization performance of the SVRE and SVRhte illustrates that the ensembles struggled with the complexity of the dataset. While SVR algorithms are known to perform well on small datasets (Wilson, 2008), the SVRE and SVRhte performed poorly in comparison to other ensembles.

Table 9.1: Ensemble Results for Clean Yacht Hydrodynamics Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	8.80830	8.19628	1.15492
DTE	0.61237	0.43789	1.95569
RF	0.66274	0.45526	2.11915
SVRE	6.83425	9.25483	0.54531
NNE	3.34246	3.60836	0.85805
*k*NNhte	7.54723	6.81886	1.22504
DThte	0.55030	0.49693	1.22633
SVRhte	7.90555	11.07749	0.50931
NNhte	2.88624	2.99306	0.92989
HTEsm	0.86653	0.78556	1.21677
HTEdf	0.81571	0.73621	1.22764

The DTE achieved the lowest training error for the training performance, followed by the RF algorithm, and DThte. It is observed that the HTEdf and HTEsm also offered a good level of training performance relative to the DTE, RF, and DThte. Illustrated by the GFs of all ensembles, the GFs of the NNE, NNhte, SVRE, and SVRhte indicate that these ensembles did not experience the overfitting of the training dataset. While the DTE and RF algorithm showed more overfitting based on the GFs, other ensembles slightly overfitted the training dataset.

Further, the generalization performance all ensembles also confirms the advantage of the mixtures of heterogeneous experts over the pure homogeneous experts as seen between the *k*NNE and *k*NNhte, DTE and DThte, as well as NNE and NNhte. The only exception is the SVRE and SVRhte. In addition, the heterogeneous mixtures of experts achieved less overfitting compared to the homogeneous mixtures.

Therefore, the results of all performance measures provide evidence to conclude that the mixtures of heterogeneous experts performed better than the homogeneous mixture models for the Yacht Hydrodynamics problem.

Residential Building Dataset

The goal is to estimate the sales price of residential apartments. Plots of the testing RMSE and training RMSE of the ensembles are depicted in Figure 9.2. Table 9.2 summarizes the results of the testing RMSE, training RMSE, and GFs of the ensembles.

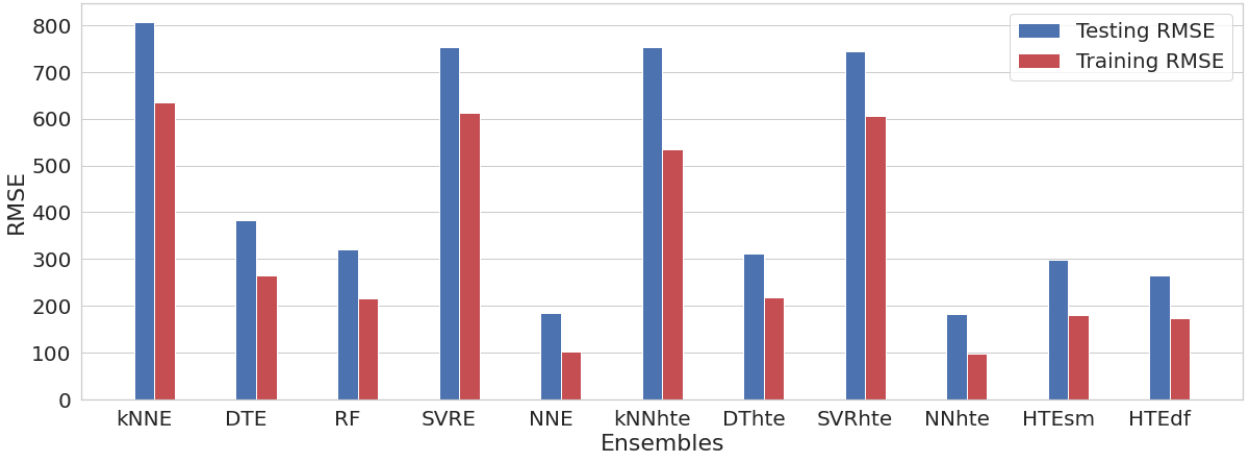

Figure 9.2: Ensemble RMSE for Clean Residential Building Dataset

Table 9.2: Ensemble Results for Clean Residential Building Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	806.10137	635.21043	1.61044
DTE	383.69436	265.11730	2.09457
RF	321.95347	216.56617	2.21006
SVRE	751.67902	611.68702	1.51010
NNE	184.91719	102.18275	3.27491
*k*NNhte	752.14817	535.44444	1.97323
DThte	311.66365	217.89727	2.04583
SVRhte	743.07273	605.88325	1.50413
NNhte	182.63824	97.55967	3.50463
HTEsm	298.84205	180.14668	2.71589
HTEdf	264.43479	173.59736	2.32034

The results illustrated in Figure 9.2 and Table 9.2 show that NNhte is the most accurate ensemble achieving the smallest testing error, while the NNE is ranked as the second most accurate ensemble. The generalization performance of the NNhte indicates that the mixtures of heterogeneous experts within the NNhte resulted in better ensemble prediction than the homogeneous mixtures in the NNE. The HTEdf and HTEsm also achieved reasonable generalization performance as the third and fourth most accurate ensembles, while the *k*NNE is the least accurate ensemble.

The results also showed that the *k*NNE and *k*NNhte generalized poorly on the complexity of the residential building dataset. The SVRE and SVRhte were slightly better than the *k*NNE and *k*NNhte. In literature, SVR algorithms have been reported to scale efficiently on datasets with small sample sizes and large features (Wilson, 2008; Nah and Lee, 2016). Thus, this characteristic presents in the residential building dataset provides the possibility that supported the prediction performance of the SVRE and SVRhte over the *k*NNE and *k*NNhte.

Furthermore, it can be observed that the training performance of all ensembles is similar to the trend in the testing behaviour of the ensembles. A number of ensembles achieved low competitive training errors, except for the *k*NNE, *k*NNhte, SVRE, and SVRhte, which performed poorly. The four ensembles produced very high training errors. While NNhte and NNE achieved the best and second-best generalization performance, both ensembles showed more overfitting of the training dataset compared to other ensembles.

Thus, the results in Table 9.2 still revealed that it is better to construct mixtures of heterogeneous experts for this problem.

Student Performance Dataset

The task is to predict the performance of students in a high school mathematics subject. The plots of the testing RMSE and training RMSE of the ensembles are given in Figure 9.3. Table 9.3 further summarizes the results of the testing RMSE, training RMSE, and GFs of the ensembles.

As shown in Figure 9.3 and Table 9.3, the RF algorithm obtained the smallest testing error of 2.00123 to be ranked as the most accurate ensemble. The DThte is the second most accurate ensemble slightly underperforming the RF algorithm with a difference of 2.83%

in testing error.

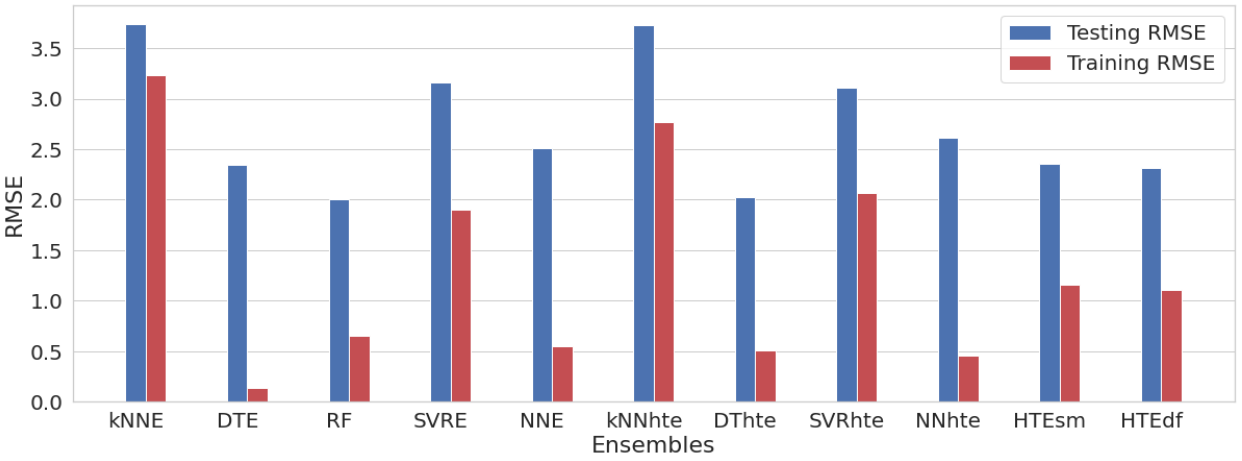

Figure 9.3: Ensemble RMSE for Student Performance Dataset

Table 9.3: Ensemble Results for Clean Student Performance Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	3.73956	3.23039	1.34008
DTE	2.34348	0.14074	277.26012
RF	2.00123	0.64978	9.48553
SVRE	3.15994	1.90374	2.75513
NNE	2.50564	0.54899	20.83122
*k*NNhte	3.72353	2.76686	1.81107
DThte	2.02953	0.50684	16.03431
SVRhte	3.10482	2.06678	2.25675
NNhte	2.60945	0.45928	32.28120
HTEsm	2.35533	1.16191	4.10922
HTEdf	2.31086	1.10234	4.39460

While the generalization performance of the DThte is competitive against the RF algorithm, the RF algorithm largely outperformed the DTE, ranked as the fourth-best performing ensemble. This observation is expected and agrees with the literature because RF combines multiple DTs trained using bagging and RFSM to achieve better predictive performance than the component DTs within the RF (Breiman, 2001; Bernard et al., 2009; Paul et al., 2018). The DTE did not capitalize on the benefit of both bagging and RFSM to induce predictive experts.

The HTEdf is the third most accurate ensemble, while the *k*NNE is the worst performing ensemble. For training performance, the DTE produced the lowest training error while the *k*NNE still performed badly in training, obtaining the highest training error. It can be observed that *k*NNhte also struggled with the complexity of the student performance dataset both in the training and testing phases. Illustrated by the GFs of the ensembles, all ensembles overfitted the training dataset. However, while measures were implemented to prevent high-level overfitting of ensembles in this research, the DTE, RF, NNE, DThte, and NNhte significantly experienced more overfitting than other ensembles.

With the exception of the NNE and NNhte, the ensembles developed from the mixture of heterogeneous experts performed better than the homogeneous mixture models.

Real Estate Dataset

The problem is to estimate the monetary valuation of a real estate. The testing RMSE and training RMSE of the ensembles are illustrated in Figure 9.4. The results of the testing RMSE, training RMSE and GF of the ensembles are provided in Table 9.4.

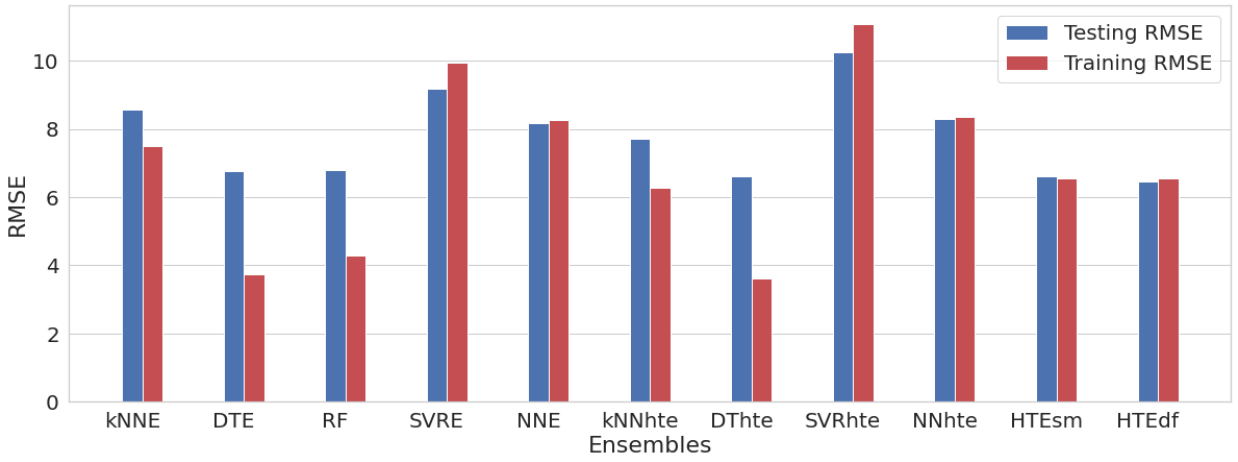

Figure 9.4: Ensemble RMSE for Real Estate Dataset

Illustrated in Figure 9.4, the most accurate ensemble is the HTEdf producing the smallest generalization error of 6.47052. The HTEsm (6.60934) is ranked as the second most accurate ensemble, followed by the DThte (6.61295). The generalization performance of the HTEdf and HTEsm showed that the ensembles generalized better than other ensembles for the characteristics of the real estate datasets. The SVRE and SVRhte underperformed by achieving high testing errors compared to other ensembles. While

the SVRE and SVRhte were expected to perform well on the real estate dataset because the SVR algorithms perform well on datasets consisting of small sample size, the SVRE and SVRhte struggled with the dataset.

Table 9.4: Ensemble Results for Clean Real Estate Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	8.57811	7.48849	1.31219
DTE	6.75240	3.74868	3.24458
RF	6.80173	4.29827	2.50410
SVRE	9.17407	9.94265	0.85137
NNE	8.17073	8.25797	0.97898
*k*NNhte	7.71519	6.28083	1.50889
DThte	6.61295	3.61572	3.34503
SVRhte	10.23970	11.07697	0.85454
NNhte	8.29605	8.34366	0.98862
HTEsm	6.60934	6.55871	1.01550
HTEdf	6.47052	6.54610	0.97704

For training performance, the DThte offered the lowest training error of 3.61572, followed by the DTE (3.74868). It can be observed that the other ensembles achieved a good level of training performance. However, as illustrated by the GF of all ensembles, the GFs of DTE and DThte indicate that the ensembles overfitted the training dataset. On the other hand, the GFs of the HTEdf, NNhte, SVRhte, NNE, SVRE, and HTEsm highlight that the ensembles did not exhibit an overfitting problem.

Also, the results in Table 9.4 reveal the advantage of the mixtures of heterogeneous experts over homogeneous mixtures of experts in terms of generalization performance achieved by the ensembles except for the SVRE which slightly outperformed the SVRhte. Although, the homogeneous mixtures of experts rivalled the heterogeneous mixtures based on overfitting.

Therefore, considering the outcome of all performance measures, it can be concluded that the HTEdf and HTEsm generalized better than other ensembles for the complexity of the real estate dataset.

Energy Efficiency Dataset

The energy efficiency of the shapes of building constructions is estimated in this dataset. Plots of the testing RMSE and training RMSE are illustrated in Figure 9.5. Table 9.5 provides the results of the testing RMSE, training RMSE, and GF of the ensembles.

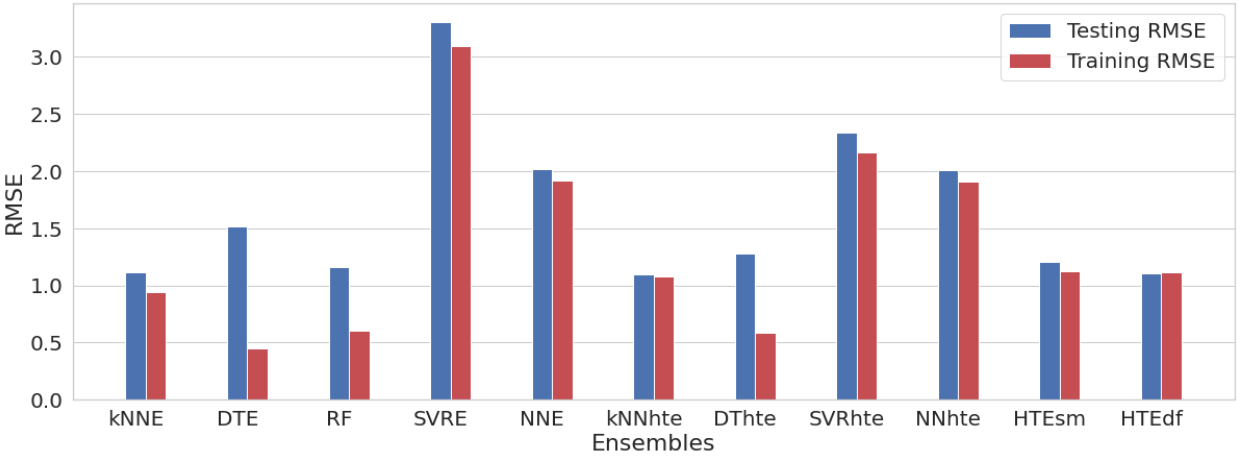

Figure 9.5: Ensemble RMSE for Energy Efficiency Dataset

Table 9.5: Ensemble Results for Clean Energy Efficiency Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	1.11056	0.94322	1.38629
DTE	1.51077	0.45076	11.23349
RF	1.16034	0.60680	3.65661
SVRE	3.30104	3.09299	1.13905
NNE	2.01520	1.91617	1.10604
*k*NNhte	1.09623	1.07436	1.04112
DThte	1.28247	0.58154	4.86330
SVRhte	2.33390	2.15913	1.16844
NNhte	2.00893	1.90534	1.11169
HTEsm	1.20575	1.12569	1.14730
HTEdf	1.10375	1.11558	0.97890

From Figure 9.5, the *k*NNhte is the most accurate ensemble for the energy efficiency dataset by generating the lowest testing error of 1.09623. The HTEdf (1.10375) provided competitive testing performance to the *k*NNhte, and is ranked as the second most accurate

ensemble. The HTEdf slightly underperformed in comparison to the *k*NNhte with a difference of 0.75% in terms of testing error, illustrating the effect of the combination of multiple instances of different ML algorithms, where the instances consist of different configurations.

It can be observed that the generalization errors of a number of ensembles, such as the *k*NNE, RF algorithm, HTEsm, and DThte are quite close to the *k*NNhte. On the other hand, the SVRE offered the highest testing error to be ranked as the worst performing ensemble. The poor generalization performance of the SVRE indicates the possibility that the configuration of the base SVR learners was inefficient for this problem. In contrast, the SVRhte benefitted from the different configurations of the base learners.

In terms of training performance, all ensembles except the SVRE achieved competitive training error, highlighting that the ensembles trained well to capture the relationship between the input and target features of this dataset. While the GFs of the DTE and DThte illustrate that both ensembles overfitted the training dataset, the DTE exhibit severe overfitting of the training dataset compared to the DThte. This means that the different configurations of the base learners in the DThte induced efficient experts that generalized on the dataset better than the base learners of the DTE.

The GF of the HTEdf indicates that the ensemble did not experience an overfitting problem. Further, another significant outcome from Table 9.5 is that the ensembles obtained from the mixtures of heterogeneous experts generalized better than the pure homogeneous mixtures of experts for this dataset.

Concrete Dataset

The regression problem is to estimate concrete compressive strengths. The testing RMSE and training RMSE are plotted in Figure 9.6. The results of the testing RMSE, training RMSE, and GF of the ensembles are provided in Table 9.6.

Illustrated in Figure 9.6 and Table 9.6, the RF algorithm is the most accurate ensemble achieving the smallest testing error of 5.93388. The generalization performance of the RF algorithms indicate that the base experts in the RF made good predictions on the test dataset compared to other ensembles. The HTEdf and DThte are ranked as the second and third most accurate ensembles. The effect of the mixtures of heterogeneous experts

is evident in the generalization performance of the HTEdf because the ensemble showed the possibility of generalizing well on the characteristics of the concrete dataset.

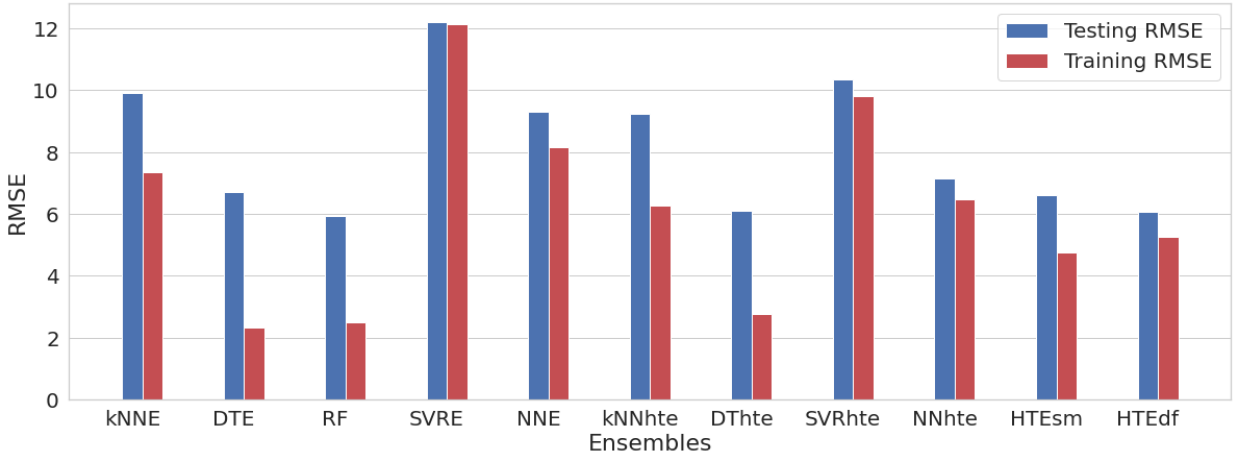

Figure 9.6: Ensemble RMSE for Concrete Dataset

Table 9.6: Ensemble Results for Clean Concrete Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	9.91529	7.33538	1.82711
DTE	6.70081	2.32002	8.34204
RF	5.93388	2.49493	5.65669
SVRE	12.20193	12.13227	1.01152
NNE	9.29890	8.14884	1.30218
*k*NNhte	9.21987	6.27842	2.15650
DThte	6.09707	2.77331	4.83333
SVRhte	10.34598	9.81562	1.11098
NNhte	7.16101	6.46362	1.22743
HTEsm	6.60890	4.75496	1.93181
HTEdf	6.05646	5.26402	1.32374

On the other hand, the SVRE underperformed in comparison to other ensembles, followed by the SVRhte. However, due to different configurations of base learners in SVRhte which induced better experts than that of the SVRE, the SVRhte was able to validate the advantage of the mixtures of heterogeneous experts over the homogeneous mixtures in the SVRE.

For training performance, the DTE delivered the smallest training error of 2.32002 to be ranked as the best performing ensemble, followed by the RF algorithm (2.49493), and DThte (2.77331). The SVRE performed poorly in training compared to other ensembles. While the GFs of all ensembles indicate that the ensembles overfitted the training dataset, the DTE and RF exhibitted more overfitting.

Hence, the results in Table 9.6 revealed that the HTEs outperformed the pure homogeneous mixtures models for the complexity of the concrete dataset.

Parkinsons Disease Dataset

The task is to detect early-stage symptoms of Parkinsons disease in clinical patients from the biomedical voice signal recordings of the patients. Plots of the testing RMSE and training RMSE are shown in Figure 9.7. Table 9.7 provides the results of the testing RMSE, training RMSE, and GF of the ensembles.

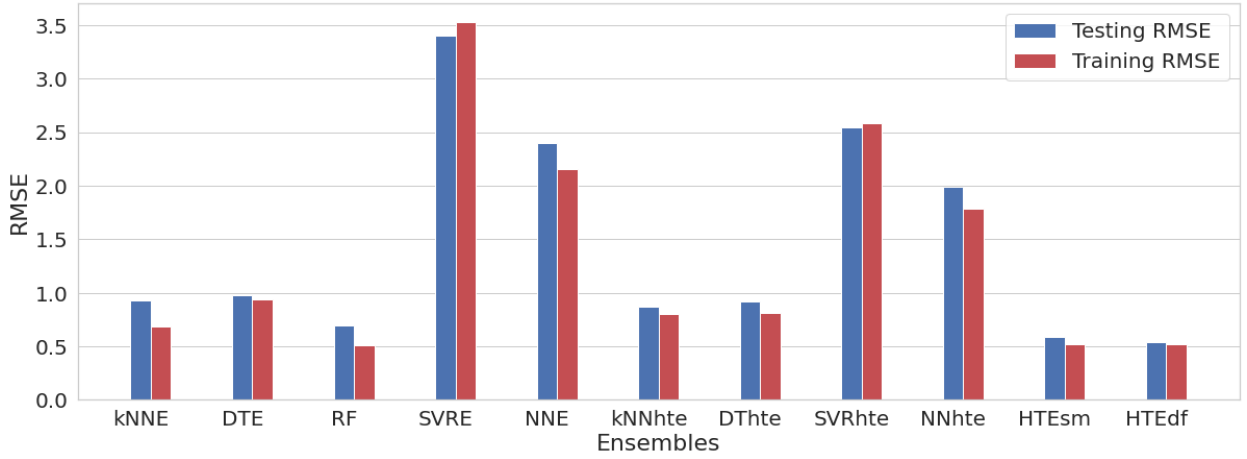

Figure 9.7: Ensemble RMSE for Parkinsons Disease Dataset

The results in Figure 9.7 and Table 9.7 show that the ensembles produced competitive training and generalization performance to predict Parkinsons Disease samples in the dataset. The HTEdf is the most accurate ensemble offering the lowest testing error of 0.53588, while the HTEsm (0.58585) is ranked as the second most accurate ensemble, followed by the RF algorithm (0.68962). The generalization performance of the HTEdf and HTEm indicate that the combination of different ML algorithms to construct a mixture of heterogeneous experts potentially lead to small prediction error and therefore provided better predictive power.

The SVRE achieved the highest training error of 3.52834, and is also ranked as the worst performing ensemble with the highest testing error of 3.39742, followed by the SVRhte for both training and testing errors. However, it can be observed that the SVRhte outperformed the SVRE due to the benefits of the mixtures of heterogeneous experts from different configurations.

Table 9.7: Ensemble Results for Clean Parkinsons Disease Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	0.92826	0.68687	1.82640
DTE	0.97891	0.93708	1.09126
RF	0.68962	0.50496	1.86512
SVRE	3.39742	3.52834	0.92717
NNE	2.40036	2.15894	1.23615
*k*NNhte	0.87349	0.80136	1.18814
DThte	0.91867	0.80720	1.29527
SVRhte	2.53995	2.58300	0.96695
NNhte	1.98650	1.78027	1.24510
HTEsm	0.58585	0.52013	1.26867
HTEdf	0.53588	0.51865	1.06755

The RF algorithm offered the best training performance, outperforming the second best-trained ensemble (i.e. HTEdf) with a slight difference of 1.37% for the training error. The HTEsm is ranked as the third best-trained ensemble with a training error of 0.52013. Illustrated by the GFs of all ensembles, the SVRE and SVRhte did not exhibit issues with overfitting, while the HTEdf and DTE showed slight overfitting of the training dataset. All other ensembles overfitted the training dataset.

The results in Table 9.7 show that the mixtures of heterogeneous experts performed better than the mixtures of homogeneous experts.

Air Quality Dataset

The problem is to predict the net hourly concentrations of Nitrogen Dioxide (NO_2) of a chemical multi-sensor device. The testing RMSE and training RMSE are plotted in Figures 9.8. Table 9.8 summarizes the result of the testing RMSE, training RMSE, and GF of the ensembles.

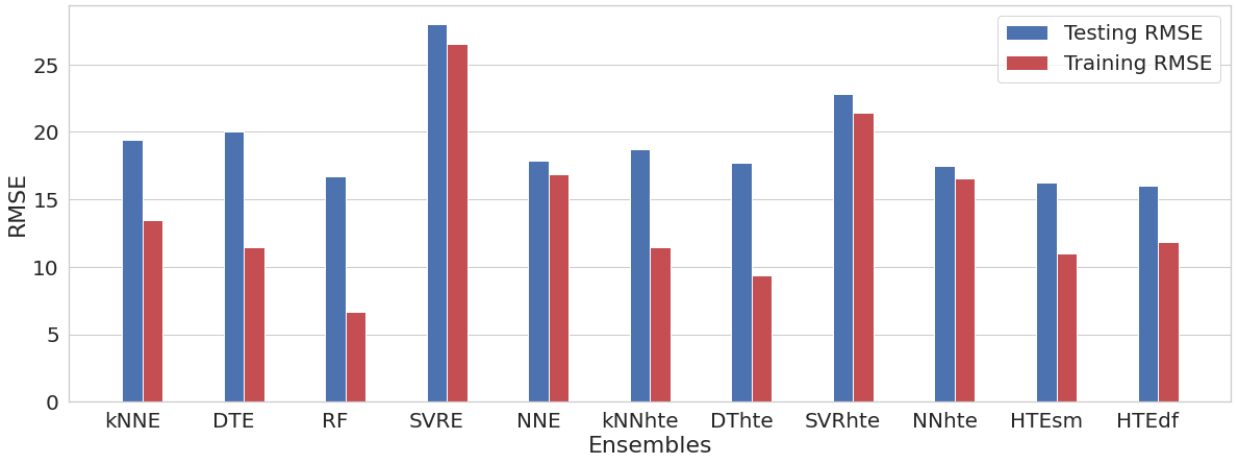

Figure 9.8: Ensemble RMSE for Air Quality Dataset

Table 9.8: Ensemble Results for Clean Air Quality Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	19.44275	13.47907	2.08063
DTE	20.05642	11.42491	3.08178
RF	16.67881	6.69179	6.21220
SVRE	27.99679	26.48880	1.11710
NNE	17.87338	16.82740	1.12818
*k*NNhte	18.71812	11.44955	2.67269
DThte	17.68167	9.38395	3.55038
SVRhte	22.78626	21.40771	1.13294
NNhte	17.48336	16.52372	1.11953
HTEsm	16.22257	10.97968	2.18303
HTEdf	16.04872	11.80934	1.84684

Illustrated in Figure 9.8 and Table 9.8, the HTEdf outperformed other ensembles

by achieving the smallest testing error of 16.04872, while the HTEsm is the second most accurate ensemble. The generalization performance of the HTEdf and HTEsm indicate that the ensembles developed from the combination of different ML algorithms generalized better on multivariate features in the dataset than other ensembles.

The RF algorithm produced competitive generalization performance close to the HTEdf, while the SVRE struggled with the characteristic of the dataset, because SVR algorithms do not perform well on datasets consisting of large samples and multivariate features. This characteristics illustrate a possible adverse influence on the SVRE achieving the highest testing error, when also compared to the SVRhte that benefited from different configurations of the base learners.

The training errors of the ensembles showed that the RF algorithm is the best trained ensemble on the training dataset achieving the lowest training error of 6.69179. The DThte and HTEsm also offered a reasonable level of training performance. However, the GF of the RF algorithm indicates that the ensemble experienced more overfitting of the training dataset than other ensembles.

Thus, the results in Table 9.8 provide evidence to conclude that the ensembles obtained from the mixtures of heterogeneous experts performed better than the pure homogeneous ensembles for the complexity of the air quality dataset.

Bike Sharing Dataset

The dataset specifies the prediction of the hourly count of bike rentals based on environmental and seasonal features. Figure 9.9 shows the plots of the testing RMSE and training RMSE. The results of the testing RMSE, training RMSE, and GF are provided in Table 9.9.

Illustrated in Figure 9.9 and Table 9.9, the advantage of the combination of the different ML algorithms is evident in the generalization performance of the HTEdf and HTEsm, because the ensembles outperformed other ensembles in terms of generalization performance. The HTEdf offered the lowest testing error of 17.63184 to be ranked as the most accurate ensemble, outperforming the HTEsm (17.68110) with a slight difference of 4.9%. The RF algorithm is the third most accurate ensemble with a testing error of 18.49675, followed by the NNhte (18.65942).

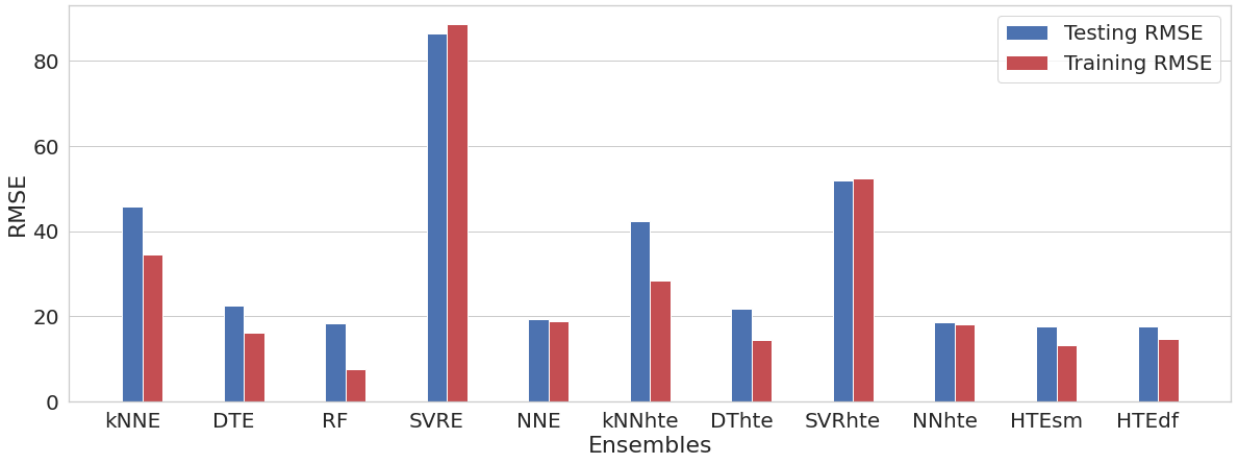

Figure 9.9: Ensemble RMSE for Bike Sharing Dataset

Table 9.9: Ensemble Results for Clean Bike Sharing Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	45.69239	34.60772	1.74318
DTE	22.50941	16.23861	1.92146
RF	18.49675	7.72522	5.73283
SVRE	86.34280	88.49999	0.95184
NNE	19.40410	18.84087	1.06068
*k*NNhte	42.31269	28.50156	2.20396
DThte	21.80844	14.45607	2.27588
SVRhte	51.88897	52.30812	0.98404
NNhte	18.65942	18.08017	1.06510
HTEsm	17.68110	13.25947	1.77814
HTEdf	17.63184	14.68068	1.44246

On the other hand, the SVRE drastically underperformed in comparison to other ensembles by achieving a very high training error of 86.34280, which illustrates poor predictive power. The poor performance of the SVRE is explained by the possibility of the ensemble being unable to handle the large number of samples in the bike sharing dataset. While the testing error of the SVRhte was unreliable being ranked as the second-worst performing ensemble, the SVRhte still outperformed the SVRE counterpart. The superiority of the SVRhte over SVRE still validates the benefit of the mixtures of

heterogeneous experts in comparison to pure homogeneous mixtures.

The RF algorithm outperformed other ensembles in training by offering the best training error of 7.72522, which is followed by the HTEsm (13.25947), DThte (14.45607), and HTEdf (14.68068).

Illustrated by the GFs of all ensembles, the GFs of the SVRE, NNE, SVRhte, and NNhte indicate that the ensembles did not overfit the training dataset. However, while the RF achieved the lowest training error, the GF of the RF algorithm highlights that the RF algorithm experienced more overfitting than other ensembles.

The results in Table 9.9 also reveal that the HTEs outperformed the pure homogeneous ensembles.

Gas Turbine Dataset

The problem is to predict the turbine energy yield of a gas plant. The plots of testing RMSE and training RMSE are illustrated in Figure 9.10, while Table 9.10 summarizes the results of the testing RMSE, training RMSE and GF of the ensembles.

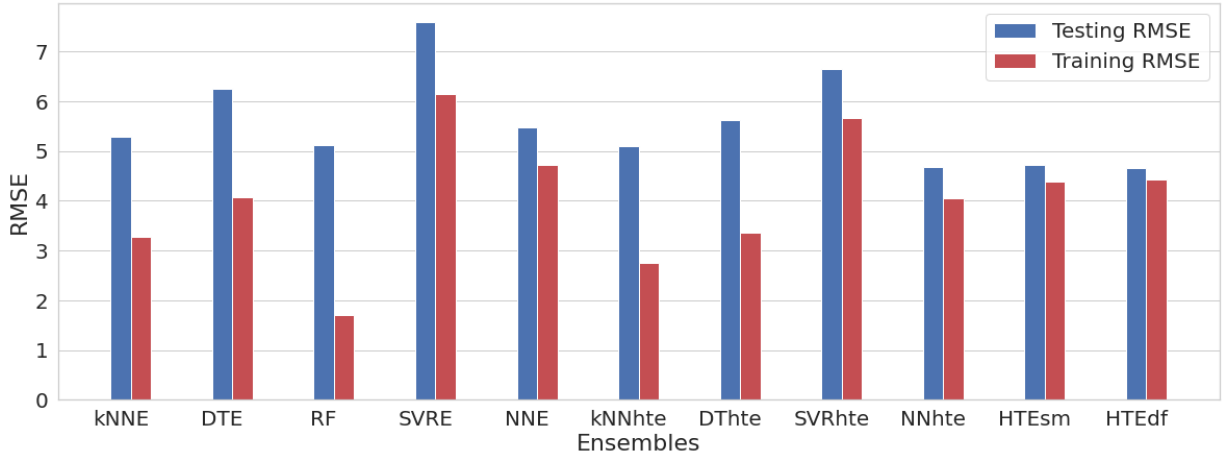

Figure 9.10: Ensemble RMSE for Gas Turbine Dataset

Figure 9.10 and Table 9.10 show that the HTEdf achieved the best generalization error of 4.66532 to be ranked as the most accurate ensemble. The NNhte is the second most accurate ensemble with a testing error of 4.67896, and slightly underperformed the HTEdf with a difference of 1.4%. The generalization performance of the NNhte illustrates that the different configurations of the base NNs resulted in efficient convergence of the base

experts within the NNhte to make good predictions.

The HTEsm, achieving a testing error of 4.72611, underperformed in comparison to the HTEdf with a slight difference of 6% to be ranked as the third most accurate ensemble. The performance of the HTEdf highlights that the ensemble constructed through the combination of different ML algorithms and different configurations for the base learners, has the potential to provide more accurate predictions than the ensembles developed using the same ML algorithms.

Table 9.10: Ensemble Results for Clean Gas Turbine Dataset

Ensemble	Testing RMSE	Training RMSE	GF
*k*NNE	5.29138	3.26653	2.62401
DTE	6.24540	4.06918	2.35563
RF	5.10826	1.71217	8.90130
SVRE	7.58885	6.13666	1.52929
NNE	5.47396	4.72415	1.34263
*k*NNhte	5.10072	2.74814	3.44498
DThte	5.62627	3.35000	2.82067
SVRhte	6.65163	5.66147	1.38038
NNhte	4.67896	4.04909	1.33531
HTEsm	4.72611	4.38701	1.16057
HTEdf	4.66532	4.42919	1.10946

The SVRE produced the worst testing error of 7.58885, highlighting the least predictive performing ensemble compared to other ensembles. The developed RF algorithm showed the possibility of the benefits of the intrinsic ensemble approaches (i.e. bagging and RFSM) implemented by RF algorithms to achieve the best training performance of 1.71217 in comparison to other ensembles. However, the GF of the RF algorithm illustrates that the ensemble showed more overfitting than other ensembles.

Additionally, while the GFs of all ensembles showed that the ensembles overfitted the training dataset, the SVRE is significantly influenced by the problem of overfitting, which results in low generalization performance compared to other ensembles.

Thus, the results in Table 9.10 provide evidence to conclude that the mixtures of heterogeneous experts performed better than homogeneous mixtures of experts for the complexity of the Gas Turbine dataset.

Statistical Analysis of Results

This section compares the performance of the developed ensembles to ascertain whether there exists a statistically significant difference in the generalization performance of the ensembles for the clean data study of regression problems. The statistical tests used include the Iman and Davenport extension of the Friedman test and Bonferroni-Dunn post hoc test.

Friedman Test

The Friedman test was discussed in Section 8.2, and is used to compare the generalization performance of the 11 ensembles developed in this chapter. The 11 ensembles are compared over the 10 regression datasets for the clean data study in this section.

The average rankings of the ensembles based on generalization performance for each dataset are provided in Table 9.11.

Table 9.11: Ranking the Generalization Performance of Ensembles over Regression Datasets in the Clean Data Study

Ensemble	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas	AvR
*k*NNE	8.81(11)	806.10(11)	3.74(11)	8.57(9)	1.11(3)	9.92(9)	0.93(6)	19.44(8)	45.69(9)	5.29(6)	**8.30**
DTE	0.61(2)	383.69(7)	2.34(4)	6.75(4)	1.51(7)	6.70(5)	0.98(7)	20.06(9)	22.51(7)	6.25(9)	**6.10**
RF	0.66(3)	321.95(6)	2.00(1)	6.80(5)	1.16(4)	5.93(1)	0.69(3)	16.68(3)	18.50(3)	5.11(5)	**3.40**
SVRE	6.83(8)	751.68(9)	3.16(9)	9.17(10)	3.30(11)	12.20(11)	3.40(11)	27.99(11)	86.34(11)	7.59(11)	**10.20**
NNE	3.34(7)	184.92(2)	2.51(6)	8.17(7)	2.02(9)	9.30(8)	2.40(9)	17.87(6)	19.40(5)	5.47(7)	**6.60**
*k*NNhte	7.55(9)	752.15(10)	3.72(10)	7.72(6)	1.09(1)	9.22(7)	0.87(4)	18.72(7)	42.31(8)	5.10(4)	**6.60**
DThte	0.55(1)	311.66(5)	2.03(2)	6.61(3)	1.28(6)	6.10(3)	0.92(5)	17.68(5)	21.81(6)	5.63(8)	**4.40**
SVRhte	7.91(10)	743.07(8)	3.11(8)	10.24(11)	2.33(10)	10.35(10)	2.54(10)	22.79(10)	51.89(10)	6.65(10)	**9.70**
NNhte	2.89(6)	182.64(1)	2.61(7)	8.30(8)	2.01(8)	7.16(6)	1.99(8)	17.48(4)	18.66(4)	4.68(2)	**5.40**
HTEsm	0.87(5)	298.84(4)	2.36(5)	6.60(2)	1.21(5)	6.61(4)	0.59(2)	16.22(2)	17.68(2)	4.73(3)	**3.40**
HTEdf	0.81(4)	264.44(3)	2.31(3)	6.47(1)	1.10(2)	6.06(2)	0.54(1)	16.05(1)	17.63(1)	4.67(1)	**1.90**

The HTEdf outperformed the other ensembles by achieving the best average rank of 1.90 over the 10 regression datasets. The HTEsm and RF algorithm are jointly ranked second

with an average rank of 3.40, followed by the DThte, ranked as the third best performing ensemble with 4.40.

While the HTEdf and HTEsm capitalize on the advantage of combining different ML algorithms to develop a mixture of heterogeneous experts, the RF provided competitive performance with the HTEsm by showing the potential benefits of the intrinsic ensembles approaches.

Another significant outcome in Table 9.11 is that the ensembles obtained from the mixtures of heterogeneous experts achieved better average rankings than the ensembles constructed from the mixtures of homogeneous experts. This is attributed to the fact that the different configurations of the base learners in heterogeneous mixtures induced efficient base experts that provided differing views on the regression datasets better than the homogeneous mixtures.

Further, based on the average rankings of the ensembles in Table 8.11, the calculated Friedman test statistic is $\chi_F^2 = 64.182$, and the Iman-Davenport extension of the Friedman test is computed as $F_F = 16.127$. F_F is distributed according to the F distribution of critical values with degrees of freedom equal to $(j-1) = 10$ and $(N-1) \times (j-1) = 90$. The critical value of $F(10, 90)$ for $\alpha = 0.05$ is 1.94.

Thus, the null hypothesis that all ensembles are equal is rejected, because the computed F_F value is greater than the critical value. The rejection of the null hypothesis indicates that there is a statistically significant difference in the generalization performance of the ensembles. It is important to note that the values of the degrees of freedom i.e., $F(10, 90)$, and the critical value of 1.94 are used later in other modelling studies of this chapter.

Bonferroni-Dunn Test

The rejection of the null hypothesis resulted in performing a post hoc test using the Bonferroni-Dunn test. The HTEdf is selected as the control ensemble for all modelling studies in this chapter, and the justification for the selection of the HTEdf as control ensemble is provided in Section 8.2.

The critical value, q_α, associated with the two-tailed Bonferroni-Dunn test at the significance level of $\alpha = 0.05$ with 11 ensembles classifiers is 2.81, and the computed critical difference (CD) is 4.168. Further, the critical value of 2.81 and the CD of 4.168 are used

later in other modelling studies of this chapter.

A significant difference is detected between the HTEdf and any other ensemble if the difference between the average rank of the HTEdf and the ensemble is greater than the computed CD of 4.168. This is illustrated in Figure 9.11.

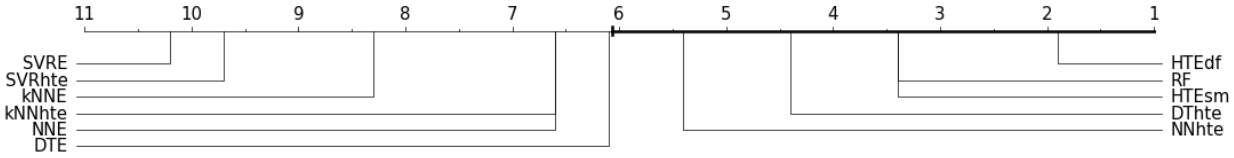

Figure 9.11: Critical Difference Plot of Ensembles for Clean Data Study in Regression Problems

From Figure 9.11, the outcome of the Bonferroni-Dunn test showed that the HTEdf is significantly more accurate than the SVRE, SVRhte, *k*NNE, *k*NNhte, NNE, and DTE. On the contrary, the 10 experimental datasets did not provide sufficient evidence to show that a significant difference in generalization performance exists between the HTEdf and HTEsm, RF algorithm, DThte, and NNhte.

Also, it can be observed that the difference in average ranks between the HTEdf and DTE is just a little above the CD = 4.168 (i.e. 6.10-1.90 = 4.20). Another significant outcome of the Bonferroni-Dunn test is that the HTEdf is significantly more accurate than all pure homogeneous ensembles (i.e SVRE, *k*NNE, DTE, and NNE), illustrating the advantage of the mixtures of heterogeneous experts over homogeneous mixtures of experts.

9.3 Number of Outliers Study

This section discusses the ensemble performance across the number of outliers perturbed from 1% to 5% in the training datasets of the regression datasets. The summative results (over all outlier ratios) of the testing RMSE, training RMSE, and GF for each ensemble over all regression datasets are provided in Table 9.12.

The results presented in Table 9.12 show that the ensembles produced different predictive performances influenced by the complexity of the datasets for the number of outliers considered. It can be observed that the HTEdf is most beneficial, capitalizing on the different ML algorithms and different base learners configurations to produce efficient

Table 9.12: Ensemble Results over all Regression Datasets in Number of Outliers Study

Measure	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas
*k*NNE										
Testing RMSE	9.014	809.366	3.742	8.696	1.425	9.811	0.949	18.417	44.610	4.627
Training RMSE	7.757	629.304	3.255	7.535	1.276	7.376	0.733	13.929	34.777	3.367
GF	1.356	1.655	1.322	1.332	1.262	1.769	1.685	1.750	1.650	1.896
DTE										
Testing RMSE	0.673	393.140	2.338	7.115	1.620	7.172	0.894	19.075	21.515	5.900
Training RMSE	0.042	4.305	0.042	3.804	0.448	2.117	0.872	11.546	15.206	4.252
GF	265.581	11153.320	3163.039	3.502	13.095	12.345	1.051	2.731	2.003	1.931
RF										
Testing RMSE	**0.502**	285.211	1.959	7.210	1.312	**5.875**	0.887	15.939	18.030	4.375
Training RMSE	0.474	98.341	0.622	4.383	0.579	2.502	0.813	6.689	7.841	1.740
GF	1.140	8.745	10.019	2.725	5.174	5.518	1.197	5.688	5.291	6.351
SVRE										
Testing RMSE	8.095	746.131	3.161	10.354	3.296	12.244	3.368	27.762	85.922	6.896
Training RMSE	10.603	592.797	1.929	11.174	3.097	12.101	3.476	27.389	89.789	6.447
GF	0.585	1.585	2.686	0.859	1.133	1.024	0.939	1.028	0.916	1.148
NNE										
Testing RMSE	3.663	373.120	2.638	8.332	2.126	7.166	2.028	17.476	18.348	4.975
Training RMSE	3.710	264.910	0.523	8.369	2.023	6.708	1.933	17.088	17.950	4.664
GF	0.978	1.986	26.222	0.991	1.110	1.142	1.101	1.046	1.045	1.138
*k*NNhte										
Testing RMSE	7.692	754.201	3.726	7.888	1.349	9.130	0.845	17.628	40.594	4.365
Training RMSE	6.521	527.528	2.795	6.315	1.215	6.250	0.577	13.061	28.568	2.827
GF	1.399	2.046	1.777	1.561	1.246	2.134	2.152	1.956	2.024	2.397
DThte										
Testing RMSE	0.634	286.191	**1.958**	**6.873**	1.305	6.208	0.865	16.861	21.454	5.079
Training RMSE	0.484	53.338	0.526	3.654	0.597	2.716	0.817	9.527	15.375	3.496
GF	1.808	29.666	14.166	3.547	4.792	5.231	1.122	3.136	1.958	2.119
SVRhte										
Testing RMSE	7.115	739.261	3.106	9.208	2.328	10.326	2.510	22.452	50.366	6.094
Training RMSE	8.926	587.597	2.092	10.035	2.163	9.793	2.556	21.974	52.225	5.836
GF	0.637	1.584	2.205	0.842	1.158	1.112	0.965	1.044	0.930	1.092
NNhte										
Testing RMSE	3.032	345.687	2.579	8.317	2.178	6.928	1.574	17.085	16.969	4.490
Training RMSE	3.019	227.321	0.489	8.427	2.064	6.444	1.509	16.731	16.504	4.168
GF	1.009	2.316	28.183	0.974	1.118	1.156	1.091	1.043	1.057	1.161
HTEsm										
Testing RMSE	0.922	184.721	2.405	7.401	1.217	6.900	0.550	15.603	17.386	4.363
Training RMSE	0.827	105.601	1.161	6.102	0.926	5.943	0.418	11.832	13.033	3.433
GF	1.239	3.174	4.294	1.472	1.733	1.349	1.735	1.779	1.797	1.618
HTEdf										
Testing RMSE	0.830	**176.982**	2.330	7.170	**1.193**	6.560	**0.396**	**15.432**	**16.808**	**4.319**
Training RMSE	0.783	99.378	1.118	6.641	1.060	5.486	0.364	12.756	13.946	3.592
GF	1.147	3.181	4.378	1.166	1.271	1.432	1.185	1.491	1.462	1.452

predictions. The HTEdf generated the lowest testing errors to outperform other ensembles for six out of the 10 datasets. The six datasets cover different characteristics and complexities ranging from small, medium, to large datasets. The generalization performance of the HTEdf also showed that the base regressors within the HTEdf give different opinions for the same input, illustrating that better fusion performance is achieved with more widely differing performance levels by the HTEdf.

The DThte achieved the best testing performance on the Student Performance and Real Estate datasets, while the RF offered the lowest prediction errors on Yacht Hydrodynamics and Concrete datasets.

The DThte capitalized on the benefit of the mixtures of heterogeneous experts from different configurations. Also, the intrinsic bagging and RFSM approaches implemented by RF algorithms illustrate a potential support for the developed RF algorithm in this modelling study. However, it can be observed that the best predictive performances of the DThte and RF algorithm did not cover all data characteristics and complexities, i.e. only covers small to medium datasets.

The HTEsm also offered promising testing performance by achieving the second-best ensemble on five datasets. On the other hand, the generalization performance of the *k*NNE, SVRE, and SVRhte are unreliable because these ensembles performed poorly on most datasets, illustrating that the ensembles struggled with the complexity of the datasets. The SVRE is the worst performing ensemble producing high prediction errors on seven datasets. The *k*NNE generated the highest testing error for three datasets (i.e. Yacht Hydrodynamics, Residential Building, and Student Performance datasets) in comparison to other ensembles.

Furthermore, the high average ranks of the pure homogeneous ensembles (i.e. *k*NNE, DTE, SVRE, and NNE) in Table 9.13 demonstrate that the ensembles showed more sensitivity to the number of outliers than the counterpart heterogeneous mixtures of experts (i.e *k*NNhte, DThte, SVRhte, and NNhte). This outcome confirms the benefit of the mixtures of heterogeneous experts over homogeneous mixtures.

Archana and Elangovan (2014) reported that *k*NN algorithms are sensitive to small values of *k*, which adversely influence the prediction performance of the algorithm. The *k*NNE showed similar prediction behaviour on the number of outliers with the possibility

that configuration of the base learners within the *k*NNE performed poorly in terms of testing error. However, the *k*NNhte performed well because the different base learner configurations induced different base experts that produced efficient predictions better than the *k*NNE.

Empirically, Sandbhor and Chaphalkar (2019) have shown that NNs are sensitive to outliers and require outlier-free data to produce reliable training and prediction performance. The average ranks of the NNE and NNhte illustrate that the NNE showed more sensitivity to the number of outliers than the NNhte. This outcome further justifies the advantage of the mixtures of heterogeneous experts over homogeneous mixtures.

Additionally, SVR algorithms have been reported to be sensitive to outliers when the soft margin method is used to construct the separating hyperplane (Fitzgerald, 2014; Kanamori et al., 2014). Thus, the average ranks of the SVRE over the datasets indicate that the base configuration produced base experts that induced soft margins during prediction. The SVRhte provided improved prediction performance than the SVRE due to the combination of different configurations for the base learners.

From Appendix F, the number of outliers is categorized into three levels, i.e. low (1% and 2%), mild (3%), and extreme (4% and 5%), and the sensitivity of the ensembles to these levels are examined. The HTEdf is consistently the most accurate ensemble in terms of generalization performance across the three outlier levels for the Energy Efficiency, Parkinsons Disease, and Air Quality datasets.

Also, the HTEdf performed best for low and extreme levels of outlier ratios for the Residential Building and Bike Sharing datasets. At the same time, the HTEsm and NNhte achieved the best generalization performance for mild level outlier ratio on the datasets. The HTEdf demonstrated superiority for the Gas Turbine dataset for low, mild and extreme levels of outlier ratio, except for the *k*NNhte that performed better for the 2% outlier ratio in the low level.

The Real Estate dataset is specifically suitable for the DThte across the low, mild and extreme levels of the outlier ratios, while performing better for the Student Performance dataset. The RF algorithm performed better for the Concrete and Yacht Hydrodynamics datasets, rivalled by the DTE and DThte for the low level of the outlier ratios.

Illustrated by the training performance of all ensembles in Table 9.12, the DTE produced the smallest training error on five datasets, i.e. Yacht Hydrodynamics, Residential Building, Student Performance, Energy Efficiency, and Concrete datasets. However, the GF of the DTE indicates that the ensemble overfitted the training dataset across the five datasets. The DTE specifically suffered from the small size of the Yacht Hydrodynamics, Residential Building and Student Performance datasets, where the ensemble experienced more overfitting.

The overfitting problem of the DTE on the Energy Efficiency and Concrete datasets is explained by the possibility that the DTE struggled with the complexity and characteristics of the datasets. Also, using the same configuration for base trees to construct the DTE did not yield sufficient benefit for the ensemble to generalize well on the two datasets.

Further, the RF algorithm outperformed other ensembles by generating the lowest training error on the Air Quality, Bike Sharing, and Gas Turbine datasets. While the GF of the RF algorithm indicates overfitting of the training dataset on these datasets, the overfitting problem was not critical compared to other ensembles.

The HTEdf and DThte offered the lowest training errors to be ranked as the best-trained ensembles on the Real Estate and Parkinsons Disease datasets. While the GFs of the HTEdf and DThte indicate that both ensembles slightly overfitted the training dataset, the HTEdf and DThte capitalized on the benefit of the mixtures of heterogeneous experts to perform well on characteristics and complexity of the Real Estate and Parkinsons Disease datasets.

Therefore, the results of all performance measures provide evidence to conclude that the mixtures of heterogeneous experts achieve better generalization performance than the homogeneous mixtures when different numbers of outliers were investigated across the regression datasets.

Statistical Analysis of Results

This section compares the generalization performance of the developed ensembles across the outlier ratios over the regression datasets.

Friedman Test

The Friedman test used to compare the generalization performance of the 11 ensembles over the 10 regression datasets followed the discussion provided in Section 8.2. Given the number of outliers considered in each dataset, the mean average of the generalization performance of each ensemble across the number of outliers is first calculated. Then the computed mean average is ranked according to the Friedman test in Table 9.13.

The HTEdf outperformed other ensembles by achieving the best average ranking of 1.90, while the RF algorithm achieved an average rank of 3.10 to be ranked as the second-best performing ensemble.

Table 9.13: Ranking the Generalization Performance of Ensembles over Regression Datasets in the Number of Outliers Study

Ensemble	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas	AvR
*k*NNE	9.01(11)	809.37(11)	3.74(11)	8.69(9)	1.42(6)	9.81(9)	0.94(7)	18.41(8)	44.61(9)	4.62(6)	**8.70**
DTE	0.67(3)	393.14(7)	2.34(4)	7.11(2)	1.62(7)	7.17(7)	0.89(6)	19.07(9)	21.51(7)	5.90(9)	**6.10**
RF	0.50(1)	285.21(3)	1.959(2)	7.21(4)	1.31(4)	5.87(1)	0.88(5)	15.93(3)	18.03(4)	4.37(4)	**3.10**
SVRE	8.09(10)	746.13(9)	3.16(9)	10.35(11)	3.29(11)	12.24(11)	3.36(11)	27.76(11)	85.92(11)	6.89(11)	**10.50**
NNE	3.66(7)	373.12(6)	2.64(7)	8.33(8)	2.12(8)	7.16(6)	2.02(9)	17.47(6)	18.34(5)	4.97(7)	**6.90**
*k*NNhte	7.69(9)	754.20(10)	3.73(10)	7.88(6)	1.34(5)	9.13(8)	0.84(3)	17.62(7)	40.59(8)	4.365(3)	**6.90**
DThte	0.63(2)	286.19(4)	1.958(1)	6.87(1)	1.30(3)	6.20(2)	0.86(4)	16.86(4)	21.45(6)	5.07(8)	**3.50**
SVRhte	7.12(8)	739.26(8)	3.11(8)	9.20(10)	2.32(10)	10.32(10)	2.51(10)	22.45(10)	50.36(10)	6.09(10)	**9.40**
NNhte	3.03(6)	345.69(5)	2.58(6)	8.31(7)	2.17(9)	6.92(5)	1.57(8)	17.08(5)	16.96(2)	4.49(5)	**5.80**
HTEsm	0.92(5)	184.72(2)	2.41(5)	7.40(5)	1.21(2)	6.90(4)	0.55(2)	15.60(2)	17.38(3)	4.363(2)	**3.20**
HTEdf	0.83(4)	176.98(1)	2.33(3)	7.17(3)	1.19(1)	6.56(3)	0.39(1)	15.43(1)	16.80(1)	4.31(1)	**1.90**

The average ranks of the HTEsm (3.20) and DThte (3.50) illustrate that the ensembles competed with the RF in terms of generalization performance over all datasets. However, the average ranking of the HTEdf indicates that the HTEdf demonstrated superiority over other ensembles due to the combination of different ML algorithms and different configurations of the base learners to develop a mixture of heterogeneous experts when experimented on different numbers of outliers.

Another outcome is that all HTEs achieved lower average rankings than the pure homogeneous mixtures. This outcome demonstrates that the different configurations of the base learners in the heterogeneous mixtures induced base experts that make better predictions in comparison to the homogeneous mixtures.

From the average rankings of the ensembles in Table 9.13, the calculated Friedman test statistic is χ_F^2 = 72.800, while the Iman-Davenport extension of the Friedman test is computed as F_F = 24.088. Because the value of F_F is greater than the obtained critical value, the null hypothesis that all ensembles are equal is rejected, illustrating that a statistically significant difference exists in the generalization performance of the ensembles.

Bonferroni-Dunn Test

After rejecting the null hypothesis, the Bonferroni-Dunn post-hoc test is performed to verify the ensembles that significantly differ from each other in the number of outliers study for regression problems. The critical value is 2.81, and the computed critical difference (CD) = 4.168.

Figure 9.12 presents the critical difference plot of the significant difference in generalization performance between the HTEdf and any other ensemble.

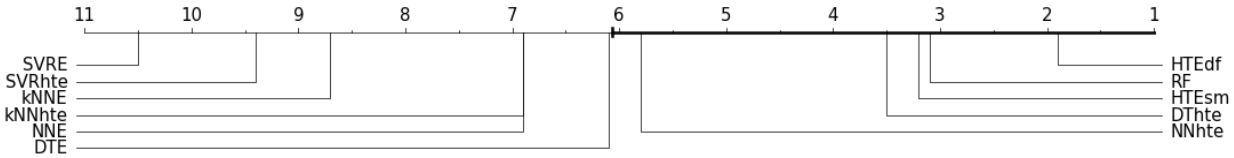

Figure 9.12: Critical Difference Plot of Ensembles for Number of Outliers Study in Regression Problems

The Bonferroni-Dunn test in this modelling study surprisingly produced the same outcome as the clean data study, as illustrated in Figures 9.11 and 9.12. While the HTEdf and DTE achieved similar average ranks in both studies, other ensembles reacted differently, producing high and low average ranks. The outcome of the Bonferroni-Dunn test reveals that the HTEdf is significantly more accurate than the SVRE, SVRhte, *k*NNE, *k*NNhte, NNE, and DTE. On the other hand, no significant difference in generalization performance exists between the HTEdf and HTEsm, RF algorithm, DThte, and NNhte.

Also, it can be observed that the difference in the average ranks between the HTEdf and DTE is just a little above the CD = 4.168 (i.e. 6.10-1.90 = 4.20), illustrating that the DTE is close to HTEdf in terms of generalization performance. Another notable outcome of the Bonferroni-Dunn test is that the HTEdf is significantly more accurate than all pure

homogeneous ensembles (i.e. SVRE, *k*NNE, DTE, NNE).

9.4 Severity of Outliers Study

This section discusses the ensemble performance across the severity of outliers for regression datasets. The severity of outliers is considered from 2 to 4 standard deviations from the estimated mean in the training datasets of each dataset. The summative results (over all outlier severities) of the testing RMSE, training RMSE, and GF for each ensemble over all regression datasets are provided in Table 9.14.

The results in Table 9.14 showed that ensembles of tree models (i.e. DTE, RF, and DThte) rivalled the HTEdf by generating low testing errors across the datasets. However, the HTEdf still leverages the benefit of different ML algorithms and different base learner configurations to achieve the best performance on five datasets, including the Student Performance, Energy Efficiency, Air Quality, Bike Sharing, and Gas Turbine datasets. The HTEsm achieved reliable testing performance to be ranked as the second-best performing ensemble on four datasets.

The RF algorithm performed best for the Concrete and Parkinsons Disease datasets, and ranked as the second most accurate ensemble for Yacht Hydrodynamics and Real Estate datasets. The DThte outperformed other ensembles specifically on the Yacht Hydrodynamics and Real Estate datasets.

Also, it can be observed that the RF and DThte competed with the HTEdf by producing the lowest testing errors on datasets consisting of mostly small and medium samples and features. In contrast, the HTEdf still demonstrated superiority over the RF and DThte by generalizing on datasets covering different complexities, i.e. small, medium, and large samples and features.

The DTE performed best for the Residential Building dataset offering the lowest prediction error, while the DThte offered the lowest prediction error for two datasets and achieved the second-best performing ensemble for three datasets (i.e. Residential Building, Concrete, and Parkinsons Disease datasets). This prediction behaviour is also observed in the generalization performance of other heterogeneous and homogeneous ensembles, for instance, SVRE and SVRhte. Further, the generalization performance of

Table 9.14: Ensemble Results over all Regression Datasets in Severity of Outliers Study

Measure	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas
*k*NNE										
Testing RMSE	7.496	1063.465	3.806	6.755	1.408	9.828	0.991	18.918	45.544	5.086
Training RMSE	7.198	474.084	2.591	6.347	1.226	7.474	0.703	13.534	34.563	3.288
GF	1.225	5.094	2.181	1.148	1.337	1.730	1.991	1.974	1.748	2.448
DTE										
Testing RMSE	1.732	**428.822**	2.784	6.990	1.515	7.138	0.978	19.940	23.364	6.145
Training RMSE	0.016	9.505	0.023	3.311	0.451	2.063	0.940	11.251	14.033	4.130
GF	21431.483	2032.394	23983.627	4.459	11.316	12.123	1.085	3.227	2.897	2.271
RF										
Testing RMSE	1.725	479.355	2.410	6.112	1.246	**5.829**	**0.410**	16.958	20.139	4.745
Training RMSE	0.398	57.559	0.955	3.992	0.603	2.495	0.197	6.453	7.766	1.693
GF	44.432	80.516	6.389	2.370	4.291	5.491	4.357	7.010	6.867	8.034
SVRE										
Testing RMSE	9.018	731.161	3.278	10.036	3.301	12.146	3.364	28.515	86.364	7.164
Training RMSE	10.080	373.390	1.839	10.297	3.093	12.045	3.504	27.087	87.341	6.184
GF	0.895	3.995	3.208	0.956	1.139	1.017	0.922	1.112	0.982	1.363
NNE										
Testing RMSE	8.475	856.267	2.539	7.038	2.038	11.368	2.224	19.149	17.961	5.523
Training RMSE	0.538	424.181	0.272	7.813	1.945	7.357	1.925	16.733	11.973	4.581
GF	246.735	4.156	87.707	0.816	1.098	2.494	1.346	1.317	2.262	1.461
*k*NNhte										
Testing RMSE	6.712	995.075	3.730	6.425	1.351	9.202	0.893	18.181	42.546	4.841
Training RMSE	6.108	384.227	2.966	5.577	1.222	6.320	0.551	11.521	28.421	2.765
GF	1.359	6.828	1.596	1.342	1.224	2.120	2.626	2.521	2.263	3.146
DThte										
Testing RMSE	**1.695**	451.888	2.383	**6.090**	1.248	6.342	0.538	17.643	23.235	5.395
Training RMSE	0.349	34.205	0.409	3.409	0.591	2.722	0.430	8.821	14.567	3.438
GF	75.018	250.203	36.061	3.196	4.460	5.428	1.573	4.117	2.592	2.513
SVRhte										
Testing RMSE	8.019	726.751	3.251	8.320	2.334	10.484	2.542	22.910	51.162	6.292
Training RMSE	8.761	388.893	1.589	9.262	2.159	9.597	2.572	21.728	51.577	5.652
GF	0.949	3.594	4.247	0.812	1.168	1.194	0.977	1.115	0.985	1.254
NNhte										
Testing RMSE	8.771	650.474	2.447	6.957	2.037	10.186	1.756	18.659	18.219	5.262
Training RMSE	0.551	333.237	0.338	7.660	1.937	6.714	1.520	16.251	10.636	4.254
GF	253.185	3.962	69.194	0.829	1.105	2.371	1.334	1.325	2.925	1.533
HTEsm										
Testing RMSE	3.391	604.212	2.330	6.232	1.111	6.869	0.964	16.336	18.796	4.611
Training RMSE	3.478	232.287	1.066	5.443	0.943	4.763	0.874	10.811	18.004	3.301
GF	1.196	6.874	4.823	1.323	1.386	2.079	1.221	2.314	1.090	1.985
HTEdf										
Testing RMSE	3.049	600.341	**2.295**	6.139	**1.096**	6.537	0.913	**15.899**	**17.436**	**4.570**
Training RMSE	2.853	207.214	1.004	5.083	1.074	4.308	0.837	9.124	16.564	3.000
GF	1.431	8.511	5.229	1.479	1.041	2.301	1.199	3.070	1.109	2.369

the SVRE, SVRhte, *k*NNE, and NNE on most datasets showed that these ensembles produced poor prediction performance in most cases. The testing performance of the SVRE, *k*NNE, and NNE illustrate that using the same configuration for the members of the ensembles did not induce efficient base experts with sufficient model complexity to fit the characteristics and complexity of the datasets given the outlier severities considered. Also, the SVRhte struggled with the characteristics and complexity of most datasets despite configuring the base learner differently. Thus, the generalization performance of the *k*NNE, SVRE and NNE corroborates the findings of the following studies: Archana and Elangovan (2014), Yang et al. (2021), Bhattacharya et al. (2017), Fitzgerald (2014), Kanamori et al. (2014), and Sandbhor and Chaphalkar (2019).

Archana and Elangovan (2014), Yang et al. (2021), and Bhattacharya et al. (2017) showed that *k*NN algorithms are sensitive to outliers with a small value of *k*, which illustrate the possibility of adversely influencing the generalization performance of the *k*NNE. In contrast, Fitzgerald (2014) and Kanamori et al. (2014) reported that SVRs using a soft margin approach are vulnerable to outliers, indicating that the base learners within the SVRE showed potential induction behaviour for soft margins. Additionally, NNs require outliers to be removed from datasets in order to perform well during training and prediction (Beliakov et al., 2011; Sandbhor and Chaphalkar, 2019). The testing performance follows the possibility that the base learners in the NNE are affected by the outlier severities introduced in the training dataset.

From Table 9.15, the average ranks of the HTEdf and HTEsm confirmed that the ensembles are much better than most ensembles across the 10 regression datasets except the RF algorithm. However, there is still much room for improvement for the HTEdf and HTEsm considering the difference in average ranks with the RF algorithm and the testing RMSE of the ensembles on a number of datasets such as the Residential Building, Air Quality, and Bike Sharing datasets.

The sensitivity of the SVRE, *k*NNE, and NNE to the outlier severities also explains how the predictive power of the HTEdf and HTEsm on the datasets are slightly influenced by the base SVR, *k*NN and NN, which are parts of the algorithms combined within the HTEdf and HTEsm. However, the generalization performance HTEdf and HTEsm is still satisfactory across all datasets, which is attributed to the different configurations set

for the multiple instances of the SVR, *k*NN and NN algorithms that induced efficient base experts, and the promising testing performance of the multiple instances of the DT algorithm configured differently.

From Appendix G, the analysis of the ensembles on low (2.0 and 2.5), mild (3.0), and extreme (3.5 and 4.0) levels of outlier severities showed that the HTEdf consistently outperformed other ensembles across all levels of outlier severities on the Energy Efficiency, Air Quality, and Bike Sharing datasets. The HTEdf also showed dominance on low and extreme outlier severities for the Student Performance and Gas Turbine datasets, while the HTEsm performed better for mild severity in both datasets.

Further, the RF algorithm achieved the best generalization performance (i.e. lowest testing error) on the Concrete and Parkinsons datasets, while the DTE performed best for the Residential Building dataset across all levels of outlier severities. The RF and DTE offered the lowest testing errors for low level of outlier severities in the Yacht Hydrodynamics dataset, while the DThte is the best ensemble on mild level severity, competing with the RF for extreme level of outlier severity. Also, the DThte and RF generalized better than other ensembles for low and mild levels of outlier severities in the Real Estate dataset, while the HTEdf performed best by achieving the lowest prediction errors for extreme level severity.

From Table 9.14, the training errors of the ensemble of base tree models, i.e. DTE, RF, and DThte, illustrate that the ensembles achieved better training performance compared to other ensembles across all datasets. The DTE offered the best training error on six datasets, while the DThte achieved the lowest training error for the Concrete and Parkinsons datasets. Also, the RF algorithm is the best-trained ensemble on the Air Quality, Bike Sharing, and Gas Turbine datasets.

However, the GFs of the DTE, RF, and DThte indicate that the ensembles overfitted the training dataset on most of the datasets. This outcome corroborates the findings in the literature that DT models usually suffer from overfitting when the models learn a training dataset to the point of high granularity, which impairs the prediction performance of the DT models (Fawagreh et al., 2014; Schonlau and Zou, 2020). Although, RF did not overfit as much as a single DT (Breiman, 2001). Hence, this training and prediction behaviour is mostly reflected by the DTE overfitting across all datasets. Further, the high GF of

the DTE specifically illustrates the severe overfitting problem of the ensemble on the Yacht Hydrodynamics, Residential Building, and Student Performance datasets. Also, the generalization performance of the DTE on six datasets showed the possibility that the DTE intrinsically prefer datasets with categorical features to numeric features, while struggling with datasets consisting of multivariate features in this modelling study. The GFs of the DThte and RF highlight that the ensembles also experienced overfitting across all datasets, but not as the DTE. On the hand, the HTEdf and HTEsm achieved lower GFs in comparison to other ensembles across the datasets.

Therefore, the performance of the ensembles illustrates that all HTEs did not perform best on all problems when compared to the pure homogeneous ensembles. However, the HTEdf and HTEsm still achieved the most consistent performance across all datasets in the severity of outliers study.

Statistical Analysis of Results

This section compares the generalization performance of the developed ensembles across the outlier severities over the 10 regression datasets.

Friedman Test

The mean average of the generalization performance of each ensemble across the outlier severities is first computed, and the mean averages are ranked according to the Friedman test. This is provided in Table 9.15.

Table 9.15: Ranking the Generalization Performance of Ensembles over Regression Datasets in the Severity of Outliers Study

Ensemble	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas	AvR
*k*NNE	7.49(7)	1063.46(11)	3.80(11)	6.75(6)	1.40(6)	9.82(7)	0.99(7)	18.91(7)	45.54(9)	5.08(5)	**7.60**
DTE	1.73(3)	428.82(1)	2.78(7)	6.99(8)	1.51(7)	7.13(5)	0.97(6)	19.94(9)	23.36(7)	6.14(9)	**6.20**
RF	1.72(2)	479.35(3)	2.41(4)	6.11(2)	1.248(4)	5.82(1)	0.41(1)	16.95(3)	20.13(5)	4.74(3)	**2.80**
SVRE	9.01(11)	731.16(8)	3.27(9)	10.03(11)	3.30(11)	12.14(11)	3.36(11)	28.51(11)	86.36(11)	7.16(11)	**10.50**
NNE	8.47(9)	856.26(9)	2.53(6)	7.03(9)	2.038(9)	11.36(10)	2.22(9)	19.14(8)	17.96(2)	5.52(8)	**7.90**
*k*NNhte	6.71(6)	995.07(10)	3.73(10)	6.42(5)	1.35(5)	9.20(6)	0.89(3)	18.18(5)	42.54(8)	4.84(4)	**6.20**
DThte	1.69(1)	451.88(2)	2.38(3)	6.09(1)	1.246(3)	6.34(2)	0.53(2)	17.64(4)	23.23(6)	5.39(7)	**3.10**
SVRhte	8.01(8)	726.75(7)	3.25(8)	8.32(10)	2.33(10)	10.48(9)	2.54(10)	22.91(10)	51.16(10)	6.29(10)	**9.20**
NNhte	8.77(10)	650.47(6)	2.44(5)	6.95(7)	2.037(8)	10.18(8)	1.75(8)	18.65(6)	18.21(3)	5.26(6)	**6.70**
HTEsm	3.39(5)	604.21(5)	2.33(2)	6.23(4)	1.11(2)	6.86(4)	0.96(5)	16.33(2)	18.79(4)	4.61(2)	**3.50**
HTEdf	3.04(4)	600.34(4)	2.29(1)	6.13(3)	1.09(1)	6.53(3)	0.91(4)	15.89(1)	17.43(1)	4.57(1)	**2.30**

The HTEdf is the best performing ensemble achieving the lowest average rank of 2.30, followed by the RF algorithm (2.80), DThte (3.10), and HTEm (3.50). The HTEdf delivered the benefits of the mixtures of heterogeneous experts through the combination of different algorithms and different configurations. However, it can be observed that the RF algorithm offered a close average ranking to that of the HTEdf. This outcome illustrates that the RF produced competitive generalization performance as the HTEdf over the 10 regression datasets. This outcome is also illustrated in Table 9.14 where the RF achieved low testing errors across the datasets.

The average ranking of the HTEsm showed that the ensemble is ranked as the fourth-best ensemble across all datasets. The average ranking of the HTEsm is explained by the fact that a number of the base learners from the different algorithms showed inherent sensitivity to the outlier severities during training. This impairs the generalization performance of the HTEsm on most datasets, as illustrated in Table 9.14. However, with respect to other ensembles, the average rankings of the DThte and HTEsm further showed that the DThte capitalizes on the advantage of the different base learner configurations, while the combination of different ML sustained the HTEsm.

The outcome of the Friedman test revealed that all HTEs achieved lower average rankings than the pure homogeneous ensembles. This indicates that different configurations of the base learners within the HTEs induced efficient base experts that generalized better than using the same configuration for the base learners in pure homogeneous ensembles.

Given the average rankings of the ensembles in Table 9.15, the calculated Friedman test statistic is $\chi_F^2 = 68.927$, while the Iman-Davenport extension of the Friedman test is computed as $F_F = 19.964$. The value of F_F is greater than the obtained critical value, illustrating the rejection of the null hypothesis that all ensembles are equal. Thus, there is a statistically significant difference in the generalization performance of the ensembles.

Bonferroni-Dunn Test

The rejection of the null hypothesis results in performing the Bonferroni-Dunn posthoc test to verfity the ensemble that is significantly different from the other in the severity of outliers study for regression problems. The critical value is 2.87, while the computed critical difference (CD) = 4.168.

Figure 9.13 presents the critical difference plot of the significant difference in generalization performance between the HTEdf and any other ensemble.

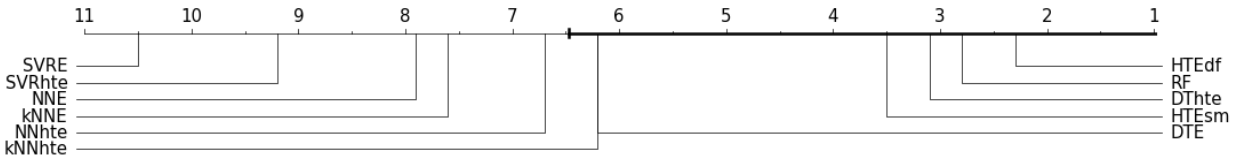

Figure 9.13: Critical Difference Plot of Ensembles for Severity of Outliers Study in Regression Problems

As illustrated in Figure 9.13, the outcome of the Bonferroni-Dunn test indicates that the HTEdf is significantly more accurate than the *k*NNE, SVRE, SVRhte, NNE, and NNhte. On the contrary, there is no significant difference in generalization performance between the HTEdf and HTEsm, RF, DThte, DTE, and *k*NNhte.

The Bonferroni-Dunn test showed that the ensembles of base tree models (i.e. DTE, RF, and DThte) achieved remarkable performance over the 10 regression datasets in the severity of outliers study. The HTEdf is significantly not different from the HTEsm because both ensembles were developed using multiple instances of different ML algorithms. Also, the different base learner configurations of the *k*NNhte resulted in base experts that delivered efficient predictions across the regression datasets.

9.5 Bagged Subsets Study

This section discusses the performance of the ensembles on different bagged subsets resampled in the training samples of the regression datasets from 10%, 20%, 30%, ... to 80%, 90%, and 100% with replacement. Table 9.16 provides the summative results (over all bagged subsets) of the testing RMSE, training RMSE, and GF of the ensembles over all regression datasets.

The results in Table 9.16 showed that when the ensembles were trained on different subsets of a training dataset, the HTEdf again outperformed other ensembles by generating the lowest testing error on five datasets. The DThte achieved the best generalization performance for three datasets, i.e. Yacht Hydrodynamics, Residential Building, and Student Performance datasets, while the RF algorithm performed best for

Table 9.16: Ensemble Results over all Regression Datasets in Bagged Subsets Study

Measure	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas
*k*NNE										
Testing RMSE	9.576	886.685	4.163	8.925	2.318	11.509	1.405	21.003	52.938	5.743
Training RMSE	7.542	670.063	3.131	7.435	1.258	8.124	1.011	14.978	41.766	3.743
GF	1.834	1.788	1.818	1.508	3.537	2.135	1.918	1.972	1.612	2.370
DTE										
Testing RMSE	1.315	434.368	2.610	9.002	2.159	9.481	1.264	22.282	24.884	6.695
Training RMSE	0.549	58.608	0.093	2.244	0.209	1.208	0.419	8.930	12.085	3.744
GF	13.413	148.861	983.987	109.760	2178.726	171.927	13.883	9.137	6.940	3.262
RF										
Testing RMSE	1.346	399.347	2.341	8.095	1.812	**7.501**	**0.986**	17.924	20.488	5.261
Training RMSE	0.679	143.463	0.590	4.201	0.539	2.636	0.422	7.111	8.683	1.871
GF	5.140	8.485	19.432	4.065	12.247	9.107	5.582	6.374	5.565	7.932
SVRE										
Testing RMSE	8.799	783.432	3.227	9.734	2.589	13.240	4.023	30.707	95.661	7.870
Training RMSE	11.656	610.620	1.839	10.050	2.215	13.157	4.094	28.864	100.430	6.396
GF	0.602	1.702	3.157	0.982	1.360	1.020	0.962	1.134	0.907	1.515
NNE										
Testing RMSE	1.674	583.124	2.756	8.774	2.252	11.119	2.545	18.876	20.668	5.907
Training RMSE	0.520	363.753	0.235	8.449	1.817	6.524	2.242	17.790	19.635	5.140
GF	13.354	66.731	859.037	1.239	1.563	3.980	1.313	1.128	1.107	1.321
*k*NNhte										
Testing RMSE	9.257	840.244	4.010	8.362	2.137	10.988	1.268	20.341	49.707	5.532
Training RMSE	6.496	569.442	2.574	6.344	1.089	6.690	0.814	12.592	34.875	3.153
GF	2.298	2.242	2.492	1.847	4.005	2.844	2.505	2.616	2.045	3.101
DThte										
Testing RMSE	**1.152**	**362.076**	**2.325**	8.026	1.796	7.636	1.224	18.803	22.715	5.752
Training RMSE	0.493	48.711	0.317	2.831	0.453	2.291	0.883	8.098	13.953	3.117
GF	12.830	72.363	78.628	18.533	17.973	12.826	2.078	5.691	2.738	3.495
SVRhte										
Testing RMSE	8.041	768.498	3.223	10.879	3.909	11.879	2.814	25.074	69.598	6.967
Training RMSE	10.325	604.154	2.170	11.298	3.482	11.228	2.765	23.523	72.494	5.829
GF	0.647	1.671	2.259	0.957	1.251	1.142	1.034	1.140	0.923	1.429
NNhte										
Testing RMSE	1.606	468.519	2.752	8.475	2.219	10.376	2.350	18.310	20.093	5.580
Training RMSE	0.534	260.612	0.204	7.829	1.881	6.061	2.086	17.163	18.995	4.900
GF	10.899	71.966	1406.385	1.478	1.473	3.854	1.300	1.141	1.119	1.296
HTEsm										
Testing RMSE	4.128	455.664	2.541	7.768	1.906	8.109	1.320	17.473	20.019	5.257
Training RMSE	3.721	281.234	1.085	5.589	1.154	5.041	1.115	10.855	12.213	3.555
GF	1.340	2.775	5.560	2.239	2.769	2.616	1.372	2.601	2.757	2.189
HTEdf										
Testing RMSE	3.777	458.836	2.466	**7.716**	**1.768**	7.971	1.260	**17.210**	**19.319**	**5.115**
Training RMSE	3.223	270.190	0.946	5.313	0.997	4.594	1.024	9.464	10.656	3.234
GF	1.531	3.018	6.906	2.382	3.292	3.038	1.481	3.310	3.335	2.500

two datasets i.e. Concrete and Parkinsons datasets.

Table 9.16 further illustrates that when multiple base regressors are trained on different subsets of the same dataset, the base regressors are more likely to have different views on the prediction of test dataset. Thus, the generalization performance of the HTEdf being better than other ensembles on half of the datasets (i.e. five out 10 datasets) confirmed this deduction that the HTEdf delivers more suitable and highly diverse regressors from the combination of different ML algorithms where the base regressors of each algorithm were configured differently.

Also, the predictive power of the HTEsm is observed on four datasets (i.e. Real Estate, Air Quality, Bike Sharing, and Gas Turbine datasets) and the HTEsm is ranked as the second most accurate ensemble. The DThte and RF algorithm also offered reliable performance on a number of datasets to be ranked as the second-best performing ensemble.

With reference to generalization performance, the DThte performed well on datasets with smaller sample sizes, i.e. Yacht Hydrodynamics, Residential Building, and Student Performance datasets, while the best generalization performance of the RF algorithm was obtained on datasets consisting of medium sample sizes, i.e. Concrete and Parkinsons Disease datasets. However, the HTEdf further demonstrated superiority over other ensembles by achieving better testing performances that covered datasets consisting of small, medium, and large datasets.

The generalization performance of the SVRE and SVRhte in this study is similar to other modelling studies where the ensembles ranked as the worst performing ensembles across the datasets. This is illustrated in the average ranks of the SVRE and SVRhte in Table 9.17. The performance of the SVRE and SVRhte demonstrate that the configurations of the base learners within each ensemble did not induce efficient experts that could generalize well on the characteristics and complexity of the datasets given the bagged subsets. Specifically, the SVRE and SVRhte were expected to perform well on the Residential Building and Student Performance datasets, because SVR algorithms perform well on datasets consisting of small samples and large features. However, the generalization performance both ensembles across all datasets still provided evidence that the mixtures of heterogeneous experts in SVRhte is better than pure homogeneous mixtures of experts in SVRE.

The trend in generalization performance between the SVRE and SVRhte is also observed among other ensembles, for instance, DTE and DThte, NNE and NNhte, as well as the *k*NNE and *k*NNhte. The *k*NNE and *k*NNhte achieved unreliable testing errors across the bagged subsets for all datasets. The outcome of the *k*NNE and *k*NNhte is explained by the intrinsic property of a *k*NN algorithm being a lazy and stable learner. Empirically, Breiman (1996a) and El-Hindi et al. (2018) have shown that stable learners do not provide reliable prediction when used to construct an ensemble, because the learners often provide little changes even when the sample space of a training dataset is randomly perturbed.

The results provided in appendix H illustrate the generalization performance of the ensembles on different input regions of the sample space, i.e. small (10-30%), medium (40-60%), and large (70-100%) subsets of the training datasets. The results indicate the differences in generalization performance among the ensemble across the three input regions.

The HTEdf and HTEsm performed best for the Air Quality, Bike Sharing, and Gas Turbine datasets across the input regions of the sample space, although rivalled by the RF algorithm and NNhte in a few cases. The RF algorithm performed best on the Parkinsons dataset across the input regions, except for the small subset region where the HTEdf offered the lowest testing error on 10% bagged size.

The NNhte and RF algorithm were the best ensembles for the Concrete dataset on small and large subsets regions of the sample space, while the DThte and RF achieved the lowest prediction errors on medium subsets. Also, the DThte and RF performed best on small subsets of the sample space in the Energy Efficiency dataset, while the HTEdf and HTEsm performed better in the medium subset region. The HTEdf averagely outperformed the DThte and RF algorithm in the large subset region.

For the Real Estate dataset, DThte and RF also performed best in the small and large subset regions of the sample space, while the HTEdf and HTEsm were better in the medium subset region. The high generalization performance of the DThte and RF across the input region of the sample space is also observed in the Student Performance dataset. The DThte also performed excellently across all input regions in the Yacht Hydrodynamics and Residential Building datasets. The only exception is on the large

subset region for the Residential Building dataset, where the NNhte and NNE performed better than the DThte.

Illustrated by the training performance of the ensembles, the ensembles of base tree models, i.e. DTE, RF algorithm, and DThte, outperformed other ensembles by producing the lowest training errors across all datasets. The DThte offered the best training performance on the Yacht Hydrodynamics and Residential datasets, while the DTE performed best on five datasets. The RF algorithm produced the smallest training error on the Air Quality, Bike Sharing, and Gas Turbine datasets. The results in Table 9.16 also revealed that the HTEdf and HTEsm achieved competitive training performance to the DTE, RF, and DThte on most datasets.

The GF of the DTE indicates that the ensemble experienced more overfitting of the training dataset than other ensembles across all datasets. Specifically, the DTE severely overfitted the training subsets for six out of the 10 datasets as shown in the Yacht Hydrodynamics, Residential Building, Student Performance, Real Estate, Energy Efficiency, and Concrete datasets.

Although the DThte also overfitted the training subsets across most datasets, the DThte still demonstrated the advantage of the mixtures of heterogeneous experts over homogeneous mixtures of experts in DTE by achieving less overfitting. This trend in GF is observed between the SVRE and SVRhte. The GFs of the NNE and NNhte also indicate how the generalization performance of the ensembles is unfavourably affected by the problem of overfitting on datasets consisting of small sample sizes, i.e. Yacht Hydrodynamics, Residential Building, and Student Performance datasets. The HTEdf and HTEsm provide slight overfitting of the training subsets across all the datasets.

From Appendix H, the performance of the ensembles on the bias-variance tradeoff showed that the ensembles developed from the mixtures of heterogeneous experts demonstrated superiority over the pure homogeneous mixtures of experts to balance the bias-variance tradeoff.

The performance of the HTEdf and HTEsm specifically illustrates low bias and variance errors across the bagged subsets over all datasets compared to other ensembles. The only exception to this outcome is for the Yacht Hydrodynamics dataset, where the DTE, RF, NNE, DThte, and NNhte outperformed the HTEdf and HTEsm across all bagged subsets

regions from small bagged subsets (10-30%), medium (40-60%) to large subset subsets (70-100%).

For the remaining nine datasets, the HTEdf and HTEsm produced smaller testing and training errors on small bagged subsets, achieving less overfitting than other ensembles, except for the Bike Sharing dataset, where the NNE and NNhte performed better. This outcome indicates that with a small dataset, it is beneficial to construct an ensemble that combines multiple decisions of different ML algorithms to achieve less overfitting compared to other ensemble types that are likely to experience more overfitting.

Despite achieving good average testing performance across all datasets, the DTE, RF and DThte overfitted on the small bagged subsets as seen for most datasets in Table 9.16. Also, the analysis of the ensemble performance with increasing bagged sizes from medium bagged subsets (40-60%) to large (70-100%) showed that the HTEdf and HTEsm further achieved a better balance of the bias-variance tradeoff than other ensembles. Although, with increasing bagged subsets, the NNE and NNhte offered reliable performances to balance the tradeoff for the Energy Efficiency, Air Quality, and Bike Sharing datasets.

Therefore, the results of the ensembles for all performance measures provide evidence to conclude that the mixtures of heterogeneous experts performed better than pure homogeneous mixtures. The outcome also deliver sufficient credence to conclude that it is beneficial to construct a mixture of heterogeneous experts from the combination of different ML algorithms to generalized better on bagged subsets of a training dataset.

Statistical Tests

This section compares the generalization performance of the developed ensembles across the bagged subsets over all regression datasets.

Friedman Test

For each dataset, the mean average of the generalization performance of each ensemble for all bagged subsets is calculated. Then the computed mean averages of the ensembles are ranked according to the Friedman test as provided in Table 9.17.

The HTEdf outperformed the other ensembles by achieving the lowest average ranking of 2.50 over the 10 regression datasets. Also, the average rank of the RF algorithm (2.60)

showed that RF achieved competitive generalization performance with the HTEdf to be ranked as the second best performing ensemble.

Table 9.17: Ranking the Generalization Performance of Ensembles over Regression Datasets in the Bagged Subsets Study

Ensemble	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas	AvR
*k*NNE	9.57(11)	886.68(11)	4.16(11)	8.92(8)	2.31(9)	11.50(9)	1.40(7)	21.00(8)	52.93(9)	5.74(6)	**8.90**
DTE	1.31(2)	434.36(3)	2.61(5)	9.00(9)	2.15(6)	9.48(5)	1.264(4)	22.28(9)	24.88(7)	6.69(9)	**5.90**
RF	1.34(3)	399.34(2)	2.34(2)	8.09(4)	1.81(3)	7.50(1)	0.98(1)	17.92(3)	20.48(4)	5.26(3)	**2.60**
SVRE	8.79(9)	783.43(9)	3.227(9)	9.73(10)	2.58(10)	13.24(11)	4.02(11)	30.70(11)	95.66(11)	7.87(11)	**10.20**
NNE	1.67(5)	583.12(7)	2.756(7)	8.77(7)	2.25(8)	11.11(8)	2.54(9)	18.87(6)	20.66(5)	5.90(8)	**7.00**
*k*NNhte	9.25(10)	840.24(10)	4.01(10)	8.36(5)	2.13(5)	10.98(7)	1.268(5)	20.34(7)	49.70(8)	5.53(4)	**7.10**
DThte	1.15(1)	362.07(1)	2.32(1)	8.02(3)	1.79(2)	7.63(2)	1.22(2)	18.80(4)	22.71(6)	5.75(7)	**2.90**
SVRhte	8.04(8)	768.49(8)	3.223(8)	10.87(11)	3.90(11)	11.87(10)	2.81(10)	25.07(10)	69.59(10)	6.96(10)	**9.60**
NNhte	1.60(4)	468.51(6)	2.752(6)	8.47(6)	2.21(7)	10.37(6)	2.35(8)	18.31(5)	20.09(3)	5.58(5)	**5.60**
HTEsm	4.12(7)	455.66(4)	2.54(4)	7.76(2)	1.90(4)	8.10(4)	1.32(6)	17.47(2)	20.01(2)	5.25(2)	**3.70**
HTEdf	3.77(6)	458.83(5)	2.46(3)	7.71(1)	1.76(1)	7.97(3)	1.260(3)	17.21(1)	19.31(1)	5.11(1)	**2.50**

The DThte (2.90) and HTEsm (3.70) are the third and fourth-best performing ensembles over all datasets. The average ranks of all ensembles showed that all HTEs achieved lower average ranks in comparison to the pure homogeneous ensembles. Also, the HTEdf is specifically the best mixture of heterogeneous experts among all HTEs.

Based on the average rankings in Table 9.17, the calculated Friedman test statistic is χ_F^2 = 65.982, while the Iman-Davenport extension of the Friedman test is computed as F_F = 17.456. The null hypothesis that all ensembles are equal is rejected on the premises that the computed F_F is greater than the critical value. Thus, a statistically significant difference in the generalization performance of the ensembles is observed.

Bonferroni-Dunn Test

After the rejection of the null hypothesis results, the Bonferroni-Dunn posthoc test is performed to determine the ensemble that statistically differs from the other in the bagged subsets study of regression problems. The critical value is 2.87, while the computed critical difference (CD) = 4.168.

Figure 9.14 illustrates the critical difference plot of the significant difference in generalization performance between the HTEdf and any other ensemble in this modelling study.

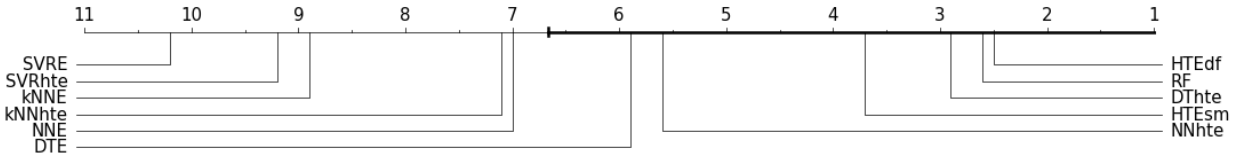

Figure 9.14: Critical Difference Plot of Ensembles for Bagged Subsets Study in Regression Problems

From Figure 9.14, the outcome of the Bonferroni-Dunn test showed that the HTEdf is significantly more accurate than all pure homogeneous ensembles (i.e. SVRE, *k*NNE, NNE), except the DTE. Also, the HTEdf is also significantly more accurate than the SVRhte and *k*NNhte, confirming that the combination of different ML algorithms to construct an ensemble is beneficial.

On the other hand, it is observed that the experimental datasets did not provide sufficient evidence to detect a significant difference in generalization performance between the HTEdf and RF algorithm, DThte, HTEsm, NNhte, and DTE.

In Table 9.17, the closeness in the average ranks of the HTEdf and RF algorithm showed that the RF algorithm performed excellently in terms of generalization performance over all datasets.

9.6 Feature Subsets Study

This section discusses the performance of the ensembles on the different feature subsets of the regression datasets. The feature subsets are resampled in the training dataset from 10%, 20%, 30%, ... to 80%, 90%, and 100% with replacement. Table 9.18 provides the summative results (over all feature subsets) of the testing RMSE, training RMSE, and GF of the ensembles over all regression datasets.

The results in Table 9.18 revealed that the ensembles achieved different performances during training and prediction, illustrating diverse behaviours on the feature subsets. It can be observed that the mixtures of heterogeneous experts generalized better than the pure homogeneous experts when the ensembles were trained on different feature subsets of the training dataset for regression problems. The HTEdf is the best performing ensemble, significantly outperforming other ensembles (i.e. achieving the lowest testing

Table 9.18: Ensemble Results over all Regression Datasets in Feature Subsets Study

Measure	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas
*k*NNE										
Testing RMSE	5.552	806.002	4.034	9.679	2.223	11.478	3.150	22.810	83.532	7.404
Training RMSE	5.779	654.532	3.488	8.849	2.137	9.543	2.459	17.246	74.018	5.591
GF	0.863	1.522	1.348	1.212	1.089	1.474	1.858	1.866	1.673	1.983
DTE										
Testing RMSE	2.141	507.418	3.425	9.824	2.323	10.839	3.075	23.862	69.166	8.218
Training RMSE	2.113	115.301	0.991	4.026	2.172	6.804	2.599	15.436	64.721	6.339
GF	282.657	735.493	537.691	14.713	1.253	4.323	1.281	2.719	1.576	1.864
RF										
Testing RMSE	2.079	435.976	**2.865**	9.021	**2.090**	10.194	**2.588**	21.261	67.635	7.238
Training RMSE	2.370	224.929	1.510	5.296	1.771	6.692	1.063	10.416	50.274	3.371
GF	1.186	5.817	5.783	3.519	2.190	3.623	6.427	5.732	4.101	6.517
SVRE										
Testing RMSE	8.738	924.636	3.451	11.687	4.480	13.876	5.720	33.687	104.809	9.645
Training RMSE	12.208	751.112	2.572	12.029	4.263	14.556	5.944	31.949	109.691	8.821
GF	0.512	1.512	2.163	0.944	1.109	0.908	0.927	1.113	0.936	1.248
NNE										
Testing RMSE	3.480	529.322	3.198	9.075	3.346	12.932	5.269	23.286	67.220	7.815
Training RMSE	4.347	368.396	1.539	9.609	3.268	10.893	4.131	21.877	68.578	7.240
GF	0.620	2.145	10.640	0.893	1.053	1.556	1.588	1.144	1.017	1.222
*k*NNhte										
Testing RMSE	4.905	780.008	3.961	9.436	2.209	10.967	3.007	22.271	81.031	7.236
Training RMSE	5.208	570.446	3.021	7.824	2.066	8.835	2.067	15.271	68.140	4.933
GF	0.831	1.889	1.768	1.510	1.231	1.605	2.349	2.365	2.080	2.547
DThte										
Testing RMSE	**2.000**	439.367	2.902	8.675	2.365	10.062	3.012	21.586	68.874	7.394
Training RMSE	2.365	158.165	1.258	4.382	2.168	6.861	2.452	13.028	60.620	5.341
GF	1.477	22.095	11.111	6.319	1.362	3.087	1.343	3.339	1.736	2.145
SVRhte										
Testing RMSE	7.244	911.096	3.526	10.414	3.503	12.444	5.054	27.990	127.613	8.989
Training RMSE	9.976	738.754	2.767	11.072	3.383	13.057	5.189	26.556	132.632	8.409
GF	0.528	1.517	1.829	0.881	1.077	0.912	0.956	1.118	0.937	1.186
NNhte										
Testing RMSE	3.105	504.956	3.229	9.034	3.127	12.116	4.565	22.997	67.215	7.432
Training RMSE	3.895	336.642	1.566	9.573	3.068	10.641	3.883	21.573	68.651	7.010
GF	0.634	2.412	11.350	0.892	1.048	1.386	1.444	1.148	1.032	1.172
HTEsm										
Testing RMSE	2.095	381.779	3.049	8.681	2.177	10.168	2.826	20.441	65.287	7.012
Training RMSE	2.639	274.442	1.828	7.228	1.700	8.656	1.979	15.171	60.338	5.735
GF	0.688	1.935	3.419	1.504	4.580	1.460	2.039	1.977	1.170	1.644
HTEdf										
Testing RMSE	2.094	**380.881**	2.972	**8.633**	2.130	**10.004**	2.768	**20.274**	**65.065**	**6.959**
Training RMSE	2.661	277.936	1.763	6.809	1.787	8.138	2.048	13.682	57.227	5.334
GF	0.686	1.878	3.583	1.706	2.372	1.642	1.827	2.508	1.102	1.898

error) on six out of the 10 datasets. Also, the HTEdf is ranked as the second most accurate ensemble for the Energy Efficiency and Parkinsons datasets. The HTEsm showed reliable generalization performance on four datasets by achieving the second-lowest prediction error.

The generalization performance of the HTEdf and HTEsm in this study illustrates the benefits of constructing an ensemble with multiple instances of different ML algorithms. By configuring the base learners differently, the HTEdf demonstrated superior generalization performance over other ensembles.

The RF algorithm offered the lowest testing error on three datasets, while achieving the second lowest prediction error on Yacht Hydrodynamics dataset. For the Yacht Hydrodynamics dataset, the DThte is more accurate than other ensembles and is ranked as the second most accurate ensemble for the Student Performance, Real Estate, and Concrete datasets. The competitive generalization performance of the RF algorithm illustrate the possibility that the ensemble leveraged on the intrinsic ensemble approaches (i.e. bagging and RFSM), while the DThte is favoured by the different configurations of the base tree learners which resulted in efficient base experts that produced low testing error.

The generalization performance of the HTEdf was best for a small dataset (i.e. Residential Building dataset), medium-sized datasets (i.e. Real Estate and Concrete datasets) and large-sized datasets (i.e. Air Quality, Bike Sharing, and Gas Turbine datasets). This outcome illustrates that the induced base experts from the differently configured multiple instances of the different ML algorithms in the HTEdf generalized better on the characteristics and complexities of datasets compared to other ensembles. The RF algorithm performed best in terms of generalization performance on small and medium-sized datasets, but struggled with large datasets. Also, the DThte offered the lowest prediction error only on one small dataset, i.e. Yacht Hydrodynamics dataset.

On the contrary, the SVRE and SVRhte are the worst performing ensembles across all datasets. The poor generalization performance of the SVRE and SVRhte over all datasets indicates that the ensembles struggled with the characteristics and complexities of most datasets. However, the SVRhte is still better than the SVRE due to effect of the differently configured base learners within the SVRhte.

Further, the results in Appendix I present the generalization performance of the ensembles on different input regions of the feature space, i.e. small (10-30%), medium (40-60%) and large (70-100%) feature subsets in the training dataset across all datasets. The HTEdf and HTEsm achieved superior generalization performance over other ensembles by offering the lowest testing errors across all feature space regions for four datasets, i.e. Residential Building, Air Quality, Bike Sharing, and Gas Turbine datasets. The RF algorithm performed best in the medium and large feature subset regions for the Parkinsons Disease dataset, while being rivalled by the HTEsm and SVRE in the low feature subset region.

The SVRhte and SVRE offered the lowest testing errors in the low feature subset region for the Student Performance dataset, and the RF performed best for the medium feature subset region. The DThte and RF algorithm jointly achieved the best predictive performance for large input regions of the feature space of the Student Performance dataset. Also, the DThte and RF offered the lowest generalization error in the medium region of the feature subsets for the Yacht Hydrodynamics dataset, while being rivalled by the *k*NNhte and HTEdf in the low feature subset egion as well as the HTEsm and DTE in the large feature subset region.

For the Real Estate dataset, the NNhte, NNE, and RF performed best in the low feature subset region, while the HTEdf, HTEsm and RF are better for the medium feature subset region. As the feature subset increases from 70% to 100%, i.e. large input feature region, the DThte and RF jointly offered the smallest testing errors. For the Energy Efficiency dataset, the HTEdf and HTEsm showed the best prediction performance in the low and medium input regions of the feature space, while the *k*NNE performed best on large feature subset region.

For the Concrete dataset, the HTEdf, HTEsm, and *k*NNhte produced the lowest testing errors in the low input region of the feature space, while the DThte, HTEsm, and NNE are better for the medium feature subset region. With increasing feature subset from 70% to 100%, The RF algorithm consistently achieved the smallest prediction error in the large input feature region.

With reference to training performance, the DTE offered the lowest training error on four datasets, i.e. Yacht Hydrodynamics, Residential Building, Student Performance, and

Real Estate datasets. The HTEsm is the best-trained ensemble for the Energy Efficiency dataset, while the RF algorithm produced the best training performance on five datasets. The training performance of the HTEdf and HTEsm illustrate the competitiveness of the ensembles with the DTE and RF algorithm due to the effect of combining different ML algorithms to learn the relationship between the input and target features of each feature subsets across the datasets.

The ensembles of base tree models, i.e. DTE, DThte, and RF showed more overfitting of the training dataset on small datasets compared to other ensembles. The high GF of the DTE illustrates the susceptibility of the ensemble to overfitting, as observed in the Yacht Hydrodynamics, Residential, Student Performance, and Real Estate datasets. The DThte capitalized on the benefit of the different base learner configurations to produce less overfitting on the datasets compared to the DTE. The less overfitting of the RF algorithm, when compared to the DTE, corroborates the findings in literature that RFs do not overfit as much as DTs because of inductive property of RF to implement both bagging and RFSM during tree induction (Breiman, 2001).

The results of the ensembles in Appendix I highlight that the HTEs outperformed the pure homogeneous ensembles with respect to the bias-variance tradeoff. The HTEdf and HTEsm specifically achieved a better balance of the tradeoff across the datasets than other ensembles. Although, there were datasets where a number of ensembles, such as the SVRE, SVRhte, NNE, and NNhte, provided competitive performance to the HTEdf and HTEsm to achieve a better tradeoff of underfitting and overfitting the training data. This is shown in the Gas Turbine, Air Quality, Concrete, and Real Estate datasets. In contrast, the DTE, DThte, and RF offered worst generalization performances to balance the tradeoff on most datasets in comparison to other ensembles. This outcome is further illustrated by the high GFs of the ensembles on most datasets, which indicate severe overfitting of the training dataset.

Furthermore, with reference to achieving a better bias-variance tradeoff, the performance of all ensembles across all feature subsets regions, i.e. from small feature subsets (10-30%), medium feature subsets (40-60%), to large feature subset (70-100%) is analyzed. The analysis showed that the HTEdf and HTEsm still offered better performance than other ensembles across all feature subsets regions on eight out of the 10 datasets. The

exceptions are for the Student Performance and Air Quality dataset, where the HTEdf and HTEsm achieved high testing and training errors from 30% to 100% feature subsets regions compared to other ensembles.

Thus, the results of all performance measures provide evidence to conclude that the HTEs performed better than pure homogeneous ensembles to learn and generalize well on different feature subsets of all experimental datasets.

Statistical Tests

This section compares the performance of the developed ensembles across the feature subsets over all regression datasets.

Friedman Test

For each dataset, the mean average of the generalization performance of each ensemble across all feature subsets is calculated. Then the computed mean averages of the ensembles are ranked according to the Friedman test as provided in Table 9.19.

Table 9.19: Ranking the Generalization Performance of Ensembles over Regression Datasets in the Feature Subsets Study

Ensemble	Yacht	Residential	Student	R.Estate	Energy	Concrete	Parkinsons	Air	Bike	Gas	AvR
*k*NNE	5.55(9)	806.00(9)	4.03(11)	9.67(8)	2.22(5)	11.47(7)	3.15(7)	22.81(6)	83.53(9)	7.40(6)	**7.70**
DTE	2.14(5)	507.41(6)	3.42(7)	9.82(9)	2.32(6)	10.83(5)	3.07(6)	23.86(9)	69.16(7)	8.21(9)	**6.90**
RF	2.07(2)	435.97(3)	2.86(1)	9.02(4)	2.09(1)	10.19(4)	2.58(1)	21.26(3)	67.63(5)	7.238(4)	**2.80**
SVRE	8.73(11)	924.63(11)	3.45(8)	11.68(11)	4.48(11)	13.87(11)	5.72(11)	33.68(11)	104.80(10)	9.64(11)	**10.60**
NNE	3.48(7)	529.32(7)	3.19(5)	9.07(6)	3.34(9)	12.93(10)	5.26(10)	23.28(8)	67.22(4)	7.81(8)	**7.40**
*k*NNhte	4.90(8)	780.00(8)	3.96(10)	9.43(7)	2.20(4)	10.96(6)	3.00(4)	22.27(5)	81.03(8)	7.236(3)	**6.30**
DThte	2.00(1)	439.36(4)	2.90(2)	8.67(2)	2.36(7)	10.06(2)	3.01(5)	21.58(4)	68.87(6)	7.39(5)	**3.80**
SVRhte	7.24(10)	911.09(10)	3.52(9)	10.41(10)	3.50(10)	12.44(9)	5.05(9)	27.99(10)	127.61(11)	8.98(10)	**9.80**
NNhte	3.10(6)	504.95(5)	3.22(6)	9.03(5)	3.12(8)	12.11(8)	4.56(8)	22.99(7)	67.21(3)	7.43(7)	**6.30**
HTEsm	2.095(4)	381.77(2)	3.04(4)	8.68(3)	2.17(3)	10.16(3)	2.82(3)	20.44(2)	65.28(2)	7.01(2)	**2.80**
HTEdf	2.094(3)	380.88(1)	2.97(3)	8.63(1)	2.13(2)	10.00(1)	2.76(2)	20.27(1)	65.06(1)	6.95(1)	**1.60**

The HTEdf achieved the lowest average rank of 1.60 to be ranked as the best performing ensemble over all datasets. The HTEsm and RF algorithm are jointly ranked as the second-best performing ensemble offering an equal average rank of 2.80. The average ranks of the HTEdf and HTEsm illustrate that maximizing the potential from the combination of different ML algorithms to construct ensembles is most beneficial.

Also, the outcome of the Friedman test reveals that all HTEs achieved lower average rankings compared to the pure homogeneous ensembles, indicating the superiority of the mixtures of heterogeneous experts over homogeneous mixtures.

Further, based on the average rankings of the ensembles in Table 9.19, the calculated Friedman test statistic is χ_F^2 is 78.291, and the Iman-Davenport extension of the Friedman test is computed as F_F = 32.457. Thus, the value of F_F is greater than the obtained critical value, which results in the rejection of the null hypothesis that all ensembles are equal. This illustrates that there is a statistically significant difference in the generalization performance of the ensembles across the feature subsets for all regression datasets.

Bonferroni-Dunn Test

After rejecting the null hypothesis, the Bonferroni-Dunn test is performed to determine the ensemble that statistically differs from the other in feature subsets study of regression problem. The critical value is 2.87, and the computed critical difference (CD) = 4.168.

Figure 9.15 illustrates the critical difference plot of the significant difference in generalization performance between the HTEdf and any other ensemble in this modelling study.

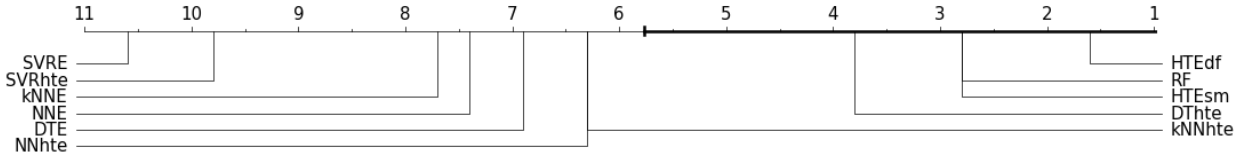

Figure 9.15: Critical Difference Plot of Ensembles for Feature Subsets Study in Regression Problems

The outcome of the Bonferroni-Dunn test in Figure 9.15 showed that the HTEdf is significantly more accurate than all pure homogeneous ensembles (i.e. SVRE, *k*NNE, NNE, and DTE), SVRhte, *k*NNhte, and NNhte, confirming the superiority of the HTEdf in this modelling study.

On the other hand, the outcome the Bonferroni-Dunn test indicates that the experimental datasets did not provide enough evidence to confirm that a significant difference in generalization performance exist between the HTEdf and HTEsm, RF algorithm, and DThte.

9.7 Discussion of Results

This section presents an overall discussion of the ensemble results across the five modelling studies from Section 9.2 to Section 9.6. The results highlight the reflection of the *No-Free-Launch Theorem* proposed by Wolpert (1996) as observed for classification problems in the previous chapter.

The GF values of the ensembles indicate that ensembles overfitted the training data on most dataets, which often leads to low testing accuracy for a number of ensembles. This is reflected in the performance of the DTE, RF, and DThte. Although, there were datasets for which the RF and DThte offered reliable generalization performance across the modelling studies.

Also, the poor outcomes of the SVRE and SVRhte in the five modelling studies highlight how the SVRE and SVRhte struggled with the characteristics and complexity of the regression datasets of this chapter. The SVRE and SVRhte were consistently ranked as the worst performing ensembles across the modelling studies. However, the average ranks of both ensembles in all the modelling studies showed that SVRhte outperformed the SVRE due to the benefits of configuring the base learner within the SVRhte differently.

In Section 9.2, the ensemble results in the clean data study illustrate the excellent prediction performance of the HTEdf over other ensembles. This is shown by the average rank of the HTEdf in Table 9.11. Across all datasets, the HTEdf offered the best generalization performance on five datasets, highlighting the benefits of using different ML algorithms where the multiple instances of the algorithms were configured differently. The equal average ranks of the HTEsm and RF algorithm demonstrate the reliability in the generalization performance of both ensembles because the average ranks of the ensembles were not extremely far from the average rank of the HTEdf. While the HTEsm also maximized the benefits of combining different ML algorithms to construct an ensemble, the RF illustrates the possibility to capitalize on the intrinsic ensemble approaches of bagging and RFSM implemented by RF algorithms.

For the number of outliers study in Section 9.3, the average rank of the HTEdf still indicates the superiority of the ensemble for six of the 10 datasets in comparison to other ensembles. A critical observation of the average ranks of all ensembles in the number

of outliers study with respect to the clean data study, showed how the introduction of the different outlier ratios to the training dataset mostly affected the pure homogeneous ensembles (i.e. *k*NNE, SVRE, and NNE) compared to other ensembles, except for the *k*NNhte and NNhte. The *k*NNhte and NNhte recorded higher average ranks in the number of outliers study compared to that of the clean dataset, illustrating that the ensembles showed sensitivity to the number of outliers in the training datasets. Again, the average ranks of the RF and HTEsm indicate that the generalization performances of the ensembles are promising across all datasets. This outcome reveals the ability of both ensembles to provide stable and consistent prediction performance across the number of outliers in each dataset.

For the severity of outliers study in Section 9.4, the performance of the ensembles reveals the sensitivity of all ensembles (except for the RF algorithm) to the severities of outliers introduced in the training dataset of each dataset. The effect of the outlier severities is reflected in the average ranks of the HTEdf and HTEsm compared to the ranks achieved for the clean data study. The prediction behaviour of the HTEdf and HTEsm corroborates the findings of Grandvalet (2004) that when the majority of the base experts in a mixture model make wrong predictions for a test observation, the overall average prediction of the mixture model is likely to be low. However, while the HTEdf recorded a high average rank of 2.30, the HTEdf still performed well on five datasets. The RF algorithm achieved competitive performance to the HTEdf, as shown in the average rank of the RF algorithm. Also, the average rank of the HTEsm provides clear evidence that the ensemble was influenced by the outlier severities, indicating that the base learners consisting of the same configurations within the HTEsm struggled to learn and generalize well on the training datasets given the outlier severities. However, the HTEdf was favoured by the different configurations of the base learners in the ensembles.

Furthermore, in Section 9.5, the results of the ensemble also reveal that when ensembles were trained on different bagged subsets of the training dataset, the HTEdf is the most accurate ensemble. The average rank of the HTEdf showed that the ensemble outperformed other ensembles on five datasets. The average rank of the RF algorithm indicates the competitiveness of the ensemble in terms of generalization performance to the HTEdf. The competitive performance of the RF algorithm to the HTEdf is attributed to the fact that a number of base experts obtained from the multiple instances of the *k*NN,

SVR, and NN algorithms configured differently within the HTEdf showed possibility of overfitting the training dataset more often. As suggested by Grandvalet (2004), a mixture model may consist of base models that are accurate in most of the input regions of the sample space, but still unnecessarily confuse the whole committee in a number of regions. The RF algorithm achieved a convincing average rank that makes the RF algorithm a candidate that could be selected as one of the best performing ensembles in the bagged subset study. The HTEsm also offered a reliable average rank across all datasets.

For the feature subsets study in Section 9.6, the HTEdf dominated other ensembles by achieving the best generalization performance for six of the 10 regression datasets. Also, the low average rank of the HTEdf (1.60) across all datasets illustrates that the HTEdf offered more consistent prediction performance than other ensembles across the input regions of the feature space in each dataset. Similarly, the average ranks of the HTEsm and RF algorithm also confirm that both ensembles offer a good level of prediction performance to compete with the HTEdf over all datasets.

Analysis of the ensemble performance over different input regions of the sample and feature space, showed that no ensemble was consistently the best over all datasets in the small bagged and feature subsets regions. However, as the sample and feature size increases from medium to large subsets, the HTEdf offered better generalization performance on more datasets than other ensembles. Although, there were cases where the RF algorithm and DThte outperformed the HTEdf in the medium subsets. Additionally, the HTEsm also showed promising predictive performance across the medium and large bagged and feature subsets over all datasets. Thus, the outcome of the ensembles on the different input regions of the sample and feature space of the training dataset demonstrate the capability of the HTEdf to generate lower prediction errors that will result in better generalization performance across all datasets than other ensembles.

The results of the five modelling studies provides evidence of the inefficiency of developing pure homogeneous ensembles (i.e. *k*NNE, DTE, SVRE, and NNE) and a number of ensembles of the same ML algorithm with different base learner configurations (i.e. *k*NNhte, SVRhte, and NNhte). The HTEdf and HTEsm are often more accurate than these ensembles across the modelling studies except for the RF algorithm and DThte. The average ranks of all ensembles in the five modelling studies showed that it is beneficial

to consider a single implementation of the HTEdf, HTEsm, and RF algorithm for the 10 regression problems identified in this research.

Furthermore, for the bagged and feature studies, the combination of different ML algorithms to construct an ensemble favours the HTEdf and HTEsm to better balance the bias-variance tradeoff in comparison to other ensembles. This is an interesting outcome where the HTEdf and HTEsm offered both training and prediction advantage over the RF algorithm, because the RF experienced the problem of overfitting on most datasets. The HTEdf and HTEsm achieved stable predictive performance due to lower testing and training errors across the bagged and feature subsets compared to other ensembles. Also, while the DTE, DThte, and NNhte offered promising generalization performance on a number of datasets, the ensembles suffered more overfitting than the HTEdf and HTEm in the modelling studies.

Conclusively, the performance of the HTEs developed in this research is compared with previous studies. The HTEdf and HTEsm achieved better generalization performance than the HTEs proposed by Elish et al. (2013). The homogeneous ensembles in the work of Elish et al. (2013) outperformed the HTEs when evaluated on a number of datasets. However, the findings of this research showed that the pure homogeneous ensembles developed did not outperform the developed HTEs. The HTEs delivered better and consistent predictive performance in comparison to the pure homogeneous ensembles across all datasets in each modelling study.

Further, Dudek (2017) empirically showed that there is no significant difference between the HTEs developed using simple averaging and weighted averaging fusion approaches. As a result of this, Dudek (2017) suggested that an efficient tuning of the control parameters of base learners in the HTEs is an essential aspect to consider in order to obtain improved performance. While this research achieved better performance in terms of low prediction errors compared to the outcome of Dudek (2017), the research further considered different ensemble approaches, including bagging, RFSM, and control parameter configuration, to improve the generalization performance of the HTEs.

Additionally, the generalization performance of the HTEs developed in this research outperform the performance of the HTEs proposed by Palaninathan et al. (2017). Palaninathan et al. (2017) developed the HTEs using two base learners and evaluated

the HTEs on a short term load forecasting dataset consisting of a small sample size. Palaninathan et al. (2017) suggested the consideration of more than two base learners to develop a HTE for better performance. The authors also pointed out the need to evaluate the HTEs on more datasets consisting of large sample sizes. The limitations of Palaninathan et al. (2017) were solved in this research through the development of HTEs using 10 instances of the base learning algorithms, and evaluated the HTEs on 10 regression datasets of different characteristics and complexities. The developed HTEs further generated low prediction errors, better than the outcome of the HTEs in Palaninathan et al. (2017).

Lastly, the work done in Hosni et al. (2018) highlights the inconsistency in the generalization performance of the HTEs proposed in their work, where a number of homogeneous ensembles significantly outperformed the proposed HTEs across all datasets. Hence, in contrast to the HTEs in Hosni et al. (2018), the HTEs developed in this research, especially the HTEdf and HTEsm, provided better and consistent generalization performance by offering low generalization errors than the pure homogeneous ensembles across all datasets.

9.8 Chapter Summary

This chapter discussed a summary of the work done. The results of the developed ensembles were discussed across five modelling studies, i.e. clean data, number of outliers, severity of outliers, bagged subsets, and feature subsets. The discussion considered the three performance measures, which include training RMSE, testing RMSE, and GF of the ensembles. The ensemble performance was also discussed with respect to the bias-variance tradeoff.

A series of statistical tests were performed to determine if the generalization performance of the ensembles were statistically significantly different. This chapter found that the mixtures of heterogeneous experts performed better than homogeneous mixtures of experts. Specifically, the HTEdf and HTEsm were the most accurate of the ensembles in terms of the average ranking of the generalization performance of the ensembles. The RF algorithm also offered competitive generalization performance to the HTEdf and HTEsm. The Friedman test illustrated that there is a significant difference between the

performance of the ensembles, confirmed through the Bonferroni-Dunn post hoc test.

For the Bonferroni-Dunn test, the HTEdf was selected as the control ensemble in the five modelling studies as performed for classification problems in Chapter 8. The outcome of the Bonferroni-Dunn test was that the HTEdf and HTEsm were significantly more accurate than pure homogeneous ensembles, and a number of ensembles developed as heterogeneous mixtures of experts across the five modelling studies. The outcome further suggested that there is no significant difference in the generalization performance of the HTEdf and the HTEsm, RF, and DThte. Lastly, the results of the ensembles and the significance of the results were discussed.

Chapter 10

Conclusions and Future Work

10.1 Introduction

This chapter provides the conclusions to this research. Section 10.2 discusses a summary of the research, while Section 10.3 describes the contributions of the research. Possible future works identified in this research are presented in Section 10.4, and Section 10.5 discusses the skills and knowledge acquired by the author during the research.

The goal of this research was to capitalize on the inductive biases of ML algorithms to develop heterogeneous mixtures of expert models that consistently produce better accuracy and generalization performance on classification and regression problems. To achieve this goal, a detailed and systematic review of the current state-of-art for the construction of heterogeneous and homogeneous ensembles was carried out. Following the review, the conceptual background of ML, bias-variance dilemma, inductive bias of ML algorithms, and ML ensemble approaches was established. Then, 13 ensemble models were developed for classification problems, while 11 ensemble models were developed for regression problems. The development of the ensembles considers four ensemble types in order to truly capitalize on the inductive biases of the ML algorithms used to construct the ensembles.

The first ensemble type involves the development of pure homogeneous ensembles using multiple instances of the same ML algorithm, where the instances consist of the same configurations. The second ensemble type considers the development of ensembles using multiple instances of the same ML algorithm, where the instances were configured differently. The third ensemble type delivers a single ensemble developed using multiple instances of different ML algorithms, where the instances consist of the same configurations. The last ensemble type provides another single ensemble developed using multiple instances of different ML algorithms, where the instances were configured differently.

All of the developed ensembles were evaluated on different modelling studies including clean data, skewed class distributions, number of outliers, severity of outliers, bagged subsets, and feature subsets. For each modelling study, the 13 classification ensemble models were evaluated on 10 classification datasets. The 11 regression ensemble models were evaluated on 10 regression datasets. All of the datasets were selected by considering different characteristics and complexities.

The performance of the ensembles was measured using classification accuracy and F1-score for classification problems, while RMSE was used for regression problems. For both problem types, a generalization factor was used to show an indication of overfitting in the performance of the ensembles. A detailed empirical analysis of the performance of the ensembles was carried for both classification and regression problems. This was followed with formal statistical tests using the Friedman test and Bonferroni-Dunn post-hoc test to statistically compare the generalization performance of the ensembles.

Thus, upon the work done in this research, the research provided the following conclusions, that in different modelling studies:

- tuning the control parameters of multiple instances of the base ML algorithms to obtain different control parameter configurations resulted in mixtures of heterogeneous experts that provided diverse ensemble predictions better than homogeneous mixtures of experts induced using the same control parameters;
- combinations of different ML algorithms resulted in a better ensemble approach that capitalized on the advantage of the inductive biases of the ML algorithms intrinsically to achieve better ensemble predictions;

- the combination of the different ML algorithms where the multiple instances of the algorithms are configured differently delivered two diversity benefits. The first is the diversity obtained from the benefit of the combination of different ML algorithms, while the second benefit was related to that diversity attributed to the advantage of capitalizing on different base learner configurations to induce efficient base experts that achieved better generalizability; and
- the ensemble developed using different ML algorithms, where the base learners were configured differently, achieved consistent and reliable generalization performance in comparison to ensembles developed using the same ML algorithm and the ensemble developed using different ML algorithms, where the base learners consist of the same configuration.

Therefore, heterogeneous mixtures of experts of different machine learning algorithms were consistently the most or one of the most accurate ensembles across all classification and regression problems. This is attributed to the advantage of capitalizing on the inductive biases of the different machine learning algorithms and the different configurations of the base members in the ensembles. The research suggests that it is beneficial to consider the implementation of a mixture of heterogeneous experts from different ML algorithms for various ML tasks. The research also recommends the need to ignore the computationally expensive process of finding the best performing homogeneous ensemble, which is often inefficient and results in long periods of computing time when the homogeneous ensembles are trained and tested. A heterogeneous ensemble provides a single implementation of different ML algorithms with lesser computational cost and better accuracy and generalization performance.

10.2 Thesis Summary

Chapter 1 presented the background of the research. This chapter provided the problem statement, research rationale, and research questions. This was followed by the presentation of the goal and objectives of the research. The research methodology, expected contributions and thesis organization were also described. This chapter described the rationale behind the development of mixtures of heterogeneous experts by capitalizing on the inductive biases of the ML algorithm forming the expert mixtures.

Chapter 2 discussed the conceptual background of ML and the bias-variance dilemma. The chapter introduced categories of learning methods, the bias-variance tradeoff, and the factors that influence an effective balance of the bias-variance tradeoff in ML. The selected ML algorithms and the inductive biases of the algorithms were discussed with respect to the bias-variance tradeoff. The selected ML algorithms in the research included NN, SVM, kNN, DT, RF, and NB algorithms. The first objective of the research stated in Section 1.5 of this thesis was achieved in this chapter.

Chapter 3 described ML ensembles and the different approaches towards developing ensembles. The chapter discussed six ensemble approaches, including bagging, boosting, stacked generalization, random feature subspace method, hyperparameter optimization, and class label manipulation. Fusion methods used to combine the individual predictions of base experts to obtain a final ensemble prediction were also presented. This chapter accomplished the second and third objectives of this research.

Chapter 4 presented a critical review of homogeneous ensembles to identify the gaps in the literature that motivated the development of homogeneous ensembles. The limitations of the developed homogeneous ensembles further motivated the investigation of the development of heterogeneous ensembles. The review of homogeneous ensembles was carried out for RF algorithm, SVM, NN, kNN, and NB ensembles. The fourth objective of this research was partly achieved in this chapter.

Chapter 5 provided a critical review of heterogeneous ensembles to investigate the performance of existing heterogeneous ensembles with respect to efficient implementation of diversity. The identified gaps in the studies were considered by the introduction and analysis of the inductive biases of ML algorithms and discussing a number of ensemble components to develop heterogeneous ensembles, and to efficiently balance the bias-variance tradeoff. The fourth objective of this research was partly achieved in this chapter.

Chapter 6 discussed the approaches to develop the mixtures of heterogeneous and homogeneous experts in this research. The ensemble types developed in this research and the practical implementations of bagging, RFSM, majority voting and averaging were described. The practical implementations of the selected ML algorithms were further explained with specific reference to the training of experts in Python. The chapter

accomplished the fifth objective of this research. Additional implementation information was provided in Chapter 7.

Chapter 7 discussed the empirical process to evaluate the developed ensembles in this research. This chapter presented the modelling studies, selected benchmark problems, data pre-processing methods, and the measures used to compare the performance of the ensembles on classification and regression problems. The k-fold cross-validation process was described with reference to the necessity of this method to ensure statistical validity. This was followed by a discussion of the hyperparameter optimization method used to obtain the algorithm-specific control parameters and associated parameter values. Lastly, the statistical tests used to determine a statistically significant difference between the performance of the ensembles were discussed. The sixth objective of the research was accomplished in this chapter.

Chapter 8 presented the empirical analysis of results for the classification problems. The chapter accomplished the seventh objective of the research to empirically compare the generalization performance of developed ensembles on six modelling studies for classification problems. The modelling studies included clean data, skewed class distributions, number of outliers, severity of outliers, bagged subsets, and feature subsets studies. The chapter concluded that the mixtures of heterogeneous experts were more accurate than homogeneous mixtures over most of the datasets across the modelling studies. Specifically, the HTEdf and HTEsm, which are mixtures of heterogeneous experts obtained from the combination of different ML algorithms in this research, were the most accurate ensembles for the datasets across the modelling studies. The overall discussion and significance of the ensemble results for the six modelling studies were also discussed. The identified research question with respect to the seventh objective for classification problems was answered in this chapter.

Chapter 9 described the empirical analysis of results for the regression problems. The chapter answered the identified research question based on the eighth objective of this research for regression problems. The eighth research objective was identified to empirically compare the generalization performance of developed ensembles on five modelling studies for regression problems. The modelling studies included clean data, number of outliers, severity of outliers, bagged subsets, and feature subsets studies. The

chapter concluded that the mixtures of heterogeneous experts were more accurate than homogeneous mixtures over most of the datasets across the modelling studies. Also, the HTEdf and HTEsm were ranked as the most or one of the most accurate ensembles for the datasets across the modelling studies. In addition, the chapter concluded that the RF algorithm is a good candidate to be selected for the regression problems, because the ensemble offered generalization performance competitive to the HTEdf and HTEsm. The overall discussion and significance of the ensemble results for the five modelling studies were also discussed.

10.3 Contributions to Knowledge

The primary contribution of this research project was the developed heterogeneous mixtures of expert models. The secondary contribution was the performance analysis of the mixtures of heterogeneous experts on a large range of problems, under different data quality conditions. While not all heterogeneous ensembles were able to outperform all homogeneous ensembles across all classification and regression problems, the identification of the limitations associated with the heterogeneous combination of experts is considered.

Furthermore, the heterogeneous ensembles, on average, outperformed the homogeneous ensembles in terms of accuracy, RMSE, GF, and other reported metrics. The performance of the heterogeneous ensembles is attributed to the introduction of diversity (due to inductive bias) into the ensembles leading to ensembles that generalized better than the homogeneous ensembles.

Specifically, the performance of the HTEdf and HTEsm showed that the diversity among heterogeneous base learners from different ML algorithms was higher than that among the heterogeneous base learners from the same ML algorithms and homogeneous base learners. The influence of the diversity among the HTEdf and HTEsm further indicates a better reduction in the deviations of base algorithms, which is attributed to the combination of the inductive biases of the algorithms. Therefore, the diverse heterogeneous base learners in the HTEdf and HTEsm are considered to generate a better integration effect than the heterogeneous base learners from the same ML algorithms and homogeneous base learners.

A significant contribution of this research is the investigation of the inductive biases and the statistical bias-variance dilemma. This research has provided a detailed analysis of the inductive biases of the selected ML algorithms. These inductive biases define the assumptions made by the selected ML algorithms to reach different conclusions on the identified modelling studies for the classification and regression problems. In addition, the research further provided an analysis of the relationship between the inductive biases of ML algorithms and the bias-variance dilemma.

Another contribution of this research involves the use of three ensemble approaches to achieve the objective of diversity within the ensembles. Bagging, RFSM, and hyperparameter optimization approaches were employed in this research. The random search optimization algorithm was used to obtain the configurations of the multiple instances of the base algorithms with respect to the inductive biases of each algorithm. Bagging and RFSM were used to create bagged and feature subsets for classification and regression problems on which the developed ensembles were also evaluated in the bagged and feature subsets studies. The mixtures of heterogeneous experts were shown to be more effective than the homogeneous mixtures of experts in predictive performance and efficiency.

10.4 Future Work

Upon accomplishing the stated research objectives in Section 1.5, potential areas of future work identified are discussed in this section. These areas of future work were not pursued in this research due to scope limitations.

The first area of potential work relates to hyperparameter optimization. The random search optimization algorithm produced suitable control parameters for the base learners during training. A possible future focus is to investigate the performance of advanced evolutionary algorithms to determine optimal control parameters quickly and efficiently. Another area is to apply these optimization algorithms to introduce diversity in different ensembles and to investigate the overall generalization performance of the ensembles.

The second area of future work involves using other ensemble approaches. Bagging, RSFM and hyperparameter optimization approaches were used in this research to

obtain the desired results that answered the research questions. Boosting and stacked generalization methods can be explored to construct ensembles by investigating the inductive biases of the selected base algorithms and other algorithms in the scope of this research.

The third potential future work is to experiment with the heterogeneous and homogeneous ensembles on more and larger (i.e. in terms of the number of samples and features) classification and regression problems, taking into account the inductive behaviours of the ensembles and the bias-variance dilemma.

Another future work is to investigate how the inductive biases of the base ML algorithms could influence the performance of the ensembles on unstructured datasets, including textual datasets used in natural language processing and sequence datasets used in bioinformatics and geo-spatial problems.

For the ensemble size, this research considered 10 base learners to develop ensembles. Another significant future work is to investigate how different ensemble sizes influence the inductive biases of the base learning algorithms and the performance of the induced base experts on the bias-variance dilemma. In addition, various pruning or voting approaches may be examined to efficiently exclude poor performing base learners in order to achieve better predictive performance on the classification and regression problems.

Further, this research considered sampling the samples and features of the training dataset with replacement in the bagged and feature subsets studies. A future work may be carried out to explore sampling with and without replacement to investigate how the ensemble will perform given the inductive biases of the base learning algorithms. Also, using sampling with replacement in the bagged and feature subsets studies demonstrated the potential to select samples and features more than once. This increases the likelihood that other samples and features may not be selected. The selected samples and features are known as *"In-the-bag set"*, while the samples and features not selected are referred to as *"Out-of-bag estimates"*. Another future work may investigate the performance of the developed ensembles on out-of-bag estimates for each bagged and feature subsets.

Another future work to consider is active learning based bagging. While the samples in the bagged subsets study were selected randomly with replacement, sample selection

can be focused on the most informative samples in a training dataset as is done in active learning.

An interesting area to explore is to map dataset characteristics to the performance of the developed ensembles. This will help to understand what dataset characteristics result in better or worst performance of the different ensembles.

Lastly, another significant future work is to capitalize on the benefits of deep learning and ensemble learning methodologies. Specifically, the analysis of inductive biases, statistical bias-variance dilemma and diversity methods can be extended to the domain of deep learning. The inductive biases of deep learning algorithms such as convolutional neural networks, deep recurrent neural networks, long short-term memory networks, deep neural networks, and beep belief networks can be studied with respect to the bias-variance dilemma and evaluated on different classification and regression problems.

10.5 Skills and Knowledge Acquired

The extensive study of the practical implementations and theoretical background of the research provided significant contributions to the skills and knowledge of the author. The theoretical background of the research broadened the understanding of the author to deeply explore and learn an interdisciplinary field of study like data science. The author has developed relevant research skills such as obtaining information, informative writing, critical literature review, model development, and critical analysis of results. Significantly, a deep understanding of how machine learning algorithms work and their inductive biases, statistical bias-variance tradeoff, and ensemble learning was achieved.

Technically, the author became proficient in Python programming for scientific programming, statistical analysis and ML model development. The technical skills also cover other aspects of data science, including data extraction, data pre-processing, data cleaning, data manipulation, data transformation, data analysis, model implementation, statistical hypothesis testing for ML, and empirical analysis of results. All of these aspects resulted in the development of the technical ability of the author in practical problem-solving and coding skills in data science.

The practical implementations required training a large number of ML algorithms for

classification and regression problems, and obtaining massive sets of results. The magnitude of the implementations required constant attentiveness to ensure the correct results were obtained. The processes leading up to the implementation taught the author the importance of consistency, because throughout the practical implementation, the consistency of the pre-processing methods, training and testing, sample and feature sets, and control parameters for the base experts was essential to ensure that the results provided a fair comparison of both the homogeneous and heterogeneous ensembles.

Another important skill developed is the use of HPC resources. The author was involved in a three-week self-development program to use HPC resources for the execution of the developed ensemble models. The use of HPC resources is essential, because of the complexity and size of the datasets and the complexity of the different models combined within the ensembles. The outcome of the self-development programme contributed to the technical skills of the author to learn and utilize different python platforms and virtual computing environments on HPC.

A significant experience of this research is the relationship of the author with the supervisor. Timely communication with the supervisor provided constructive feedback and comments that extensively aided the author in the research.

The last skill developed during this research was the importance of time management. The scope of the project required continually setting goals and deadlines and ensuring that they were met. The meeting of deadlines and setting realistic goals were complicated with the outbreak of the COVID-19 pandemic, leading to many rescheduled research activities and aligning the work plan of the author with the new normal of working remotely. However, following the research plan provided at the beginning of this research, the consistency and dedication to the daily research goals and the timely feedback of the supervisor aided the author to complete this research in record time.

List of References

Abdullah, A., Veltkamp, R. C., and Wiering, M. A. (2009). An ensemble of deep support vector machines for image categorization. *Proceedings of the International Conference of Soft Computing and Pattern Recognition.*, pages 301–306.

Abed, A. K., Al-Moukhles, H., and Abdel-Qader, I. (2018). An adaptive *k*NN based on multiple services set identifiers for indoor positioning system with an ensemble approach. *Proceedings of the IEEE 8th Annual Computing and Communication Workshop and Conference*, pages 26–32.

Ahmadian, K., Golestani, A., Analoui, M., and Jahed, M. R. (2007). Evolving ensemble of classifiers in low-dimensional spaces using multi-objective evolutionary approach. *Proceedings of the 6th IEEE International Conference on Computer and Information Science*, page 217–222.

Ahmed, E., El-Gayar, N., and El-Azab, I. A. (2010). Support vector machine ensembles using features distribution among subsets for enhancing microarray data classification. *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications*, page 1242–1246.

Ahmed, M. S., Shahjaman, M., Rana, M. M., and Mollah, M. (2017). Robustification of naïve bayes classifier and its application for microarray gene expression data analysis. *BioMed Research International*, 26(1):1–17.

Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. *Proceedings of the 15th European Conference on Machine Learning*, page 1–12.

Akila, S. (2017). Credit card fraud detection using non-overlapped risk based bagging ensemble. *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research*, page 1–4.

Aksela, M. and Laaksonen, J. (2006). Using diversity of errors for selecting members of a committee classifier. *Pattern Recognition*, 39(4):608–623.

Ala'Raj, M. and Abbod, M. (2015). A systematic credit scoring model based on heterogeneous classifier ensembles. *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications*, pages 1–7.

Alkubaisi, G. A. A. J. (2019). The role of ensemble learning in stock market classification model accuracy enhancement based on naïve bayes classifiers. *International Journal of Statistics and Probability*, 9(1):36–47.

Almeida, R., De-Mey, Y., Radmehr, G., and West, A. (2019). An ensemble based on neural networks with random weights for online data stream regression. *Soft Computing*, 224(13):9835–9855.

Alshdaifat, E., Al-hassan, M., and Aloqaily, A. (2021). Effective heterogeneous ensemble classification: An alternative approach for selecting base classifiers. *ICT Express*, 7(3):1–8.

Anand, R., Mehrotra, K. G., Mohan, C. K., and Ranka, S. (1993). An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969.

Anand, R., Mehrotra, K. G., Mohan, C. K., and Ranka, S. (1995). Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, 6(1):117–124.

Anbarasi, M. S. and Janani, V. (2017). Ensemble classifier with random forest algorithm to deal with imbalanced healthcare data. *Proceedings of the International Conference on Information Communication and Embedded Systems*, pages 1–7.

Archana, S. and Elangovan, K. (2014). Survey of classification techniques in data mining. *International Journal of Computer Science and Mobile Applications*, 2(2):65–71.

Archer, N. P. and Wang, S. (1993). Learning bias in neural networks and an approach to controlling its effects in monotonic classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):962–966.

Arefi, M. and Chowdhury, B. (2017). Ensemble adaptive neuro fuzzy support vector machine for prediction of transient stability. *Proceedings of the North American Power Symposium*, pages 1–6.

Austen-Smith, D. and Banks, J. S. (1996). Information aggregation, rationality, and the condorcet jury theorem. *American Political Science Review*, 90(1):34–45.

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of gaussian processes with model misspecification. *Computational Statistics Data Analysis*, 66(1):55– 69.

Bagheri, M. A., Gao, Q., and Escalera, S. (2014). Generic subclass ensemble: A novel approach to ensemble classification. *Proceedings of the International Conference on Pattern Recognition*, page 1254–1259.

Balogun, A. O., Balogun, A. M., Sadiku, P. O., and Adeyemo, V. E. (2017). Heterogeneous ensemble models for generic classification. *Scientific Annals of Computer Science 15(1):2017*, pages 1–8.

Bandaragoda, T. R., Ting, K. M., Albrecht, D., Liu, F. T., and Wells, J. R. (2015). Efficient anomaly detection by isolation using nearest neighbour ensemble. *Proceedings of the IEEE International Conference on Data Mining Workshops*, page 698–705.

Bandaragoda, T. R., Ting, K. M., Albrecht, D., Liu, F. T., Zhu, Y., and Wells, J. R. (2018). Isolation-based anomaly detection using nearest-neighbor ensembles. *Computational Intelligence*, 34(4):968–998.

Baneriee, C., Paul, S., and Ghoshal, M. (2018). A comparative study of different ensemble learning techniques using wisconsin breast cancer dataset. *Proceedings of the International Conference on Computer, Electrical and Communication Engineering*, pages 1–6.

Bang, S. J. and Wu, W. (2016). Naïve bayes ensemble: A new approach to classifying unlabeled multi-class asthma subjects. *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine*, page 460–465.

Batuwita, R. (2013). Class imbalance learning methods for support vector machines. In He, H. and Ma, Y., editors, *Imbalanced Learning: Foundations, Algorithms, and Applications*, pages 83–99. John Wiley Sons, Inc., Hoboken, NJ, USA.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: bagging, voting, and variants. *Machine Learning*, 36(1):105–139.

Beckmann, M., Ebecken, N. F. F., and Pires de Lima, B. L. L. (2015). A *k*NN undersampling approach for data balancing. *Journal of Intelligent Learning Systems and Applications*, 7(4):104–116.

Beliakov, G., Kelarev, A., and Yearwood, J. (2011). Robust artificial neural networks and outlier detection. *Technical report*, pages 1–39.

Beliakov, G. and Li, G. (2012). Improving the speed and stability of the *k*-nearest neighbours method. *Pattern Recognition Letters*, 33(1):1296–1301.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127.

Bergstra, J. and Bengio, Y. (2012). Random search for hyperparameter optimization. *Journal of Machine Learning Research*, 13(1):281–305.

Bergstra, J. S., Bardenet, R., Bengio, Y., and Kegl, B. (2011). Algorithms for hyperparameter optimization. *Proceedings of the 24th International Conference on Neural Information Processing Systems*, page 2546–2554.

Bernard, S., Heutte, L., and Adam, S. (2009). On the selection of decision trees in random forests. *Proceedings of the International Joint Conference on Neural Networks*, page 302–307.

Bertoni, A., Campadelli, P., and Parodi, M. (1997). A boosting algorithm for regression. *International Conference on Artificial Neural Networks*, page 343–348.

Bezdek, J., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers Geosciences*, 10(2):191–203.

Bhatnagar, V., Yede, N., Keram, R. S., and Chaurasiya, R. K. (2016). A modified approach to ensemble of svm for P300 based brain computer interface. *Proceedings of the International Conference on Advances in Human Machine Interaction*, pages 12–15.

Bhattacharya, G., Ghosh, K., and Chowdhury, A. (2017). *k*NN classification with an outlier informative distance measure. *Proceedings of the International Conference on Pattern Recognition and Machine Intelligence*, page 21–27.

Bhavan, A., Chauhan, P., Hitkul, and Shah, R. R. (2019). Bagged support vector machines for emotion recognition from speech. *Knowledge-Based Systems*, 184(3):1–7.

Bi, W., Wang, X., Tang, Z., and Tamura, H. (2005). Avoiding the local minima problem in backpropagation algorithm with modified error function. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 88(12):3645–3653.

Bian, S. and Wang, W. (2006). Investigation on diversity in homogeneous and heterogeneous ensembles. *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8.

Biggio, B. and Corona, I. (2015). One-and-a-half-class multiple classifier systems. *Proceedings of the International Workshop on Multiple Classifier Systems*, page 168–180.

Birattari, M., Yuan, Z., Balaprakash, P., and Stützle, T. (2010). F-race and iterated f-race: An overview. In Bartz-Beielstein, T., Chiarandini, M., Paquete, L., and Preuss, M., editors, *Experimental Methods for the Analysis of Optimization Algorithms*, pages 311–336. Springer, Berlin Heidelberg.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. The Clarendon Press, Oxford, UK.

Boehmke, B. and Greenwell, B. (2020). *Hands-On Machine Learning with R*. CRC Press Chapman and Hall, New York.

Bohanec, M. and Rajkovic, V. (1988). Knowledge acquisition and explanation for multi-attribute decision making. *Proceedings of the 8th International Workshop on Expert Systems and their Applications*, page 59–78.

Bonala, S. (2009). A study on neural network based system identification with application to heating, ventilating and air conditioning system. *MTech thesis, Department of Electrical Engineering National Institute of Technology, Rourkela*.

Boström, H. (2007). Feature vs. classifier fusion for predictive data mining - A case study in pesticide classification. *Proceedings of the 10th International Conference on Information Fusion*, pages 1–7.

Brady, J. and Brockmeier, D. R. (2018). Bias-variance tradeoff: A property-casualty modeler's perspective. *Variance: Casualty Actuarial Society*, 13(2):207–232.

Braga-Neto, U. and Dougherty, E. R. (2004). Bolstered error estimation. *Pattern Recognition*, 37(6):1267—1281.

Breiman, L. (1996a). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (1996b). Bias, variance and arcing classifiers. *Technical Report, Statistics Department, University of California, Berkeley, CA*.

Breiman, L. (1999a). Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36(1):85–103.

Breiman, L. (1999b). Prediction games and arcing algorithms. *Neural Computation*, 11(7):1493–1517.

Breiman, L. (2001). Random forest. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1993). *Classification and regression trees*. Chapman and Hall.

Broomhead, D. and Lowe, D. (1988). Radial basis functions, multi-variable functional interpolation and adaptive networks. *Royal Signals Radar Establishment Malvern, Malvern, U.K., Techical Report*, 4148.

Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: A survey and categorisation. *Information Fusion*, 6(1):5–20.

Buntine, W. (1990). A theory of learning classification rules. *Doctoral Dissertation, School of Computing Science, University of Technology, Sydney, Australia*.

Cachada, M. V., Abdulrahman, S. M., and Brazdil, P. (2017). Combining feature and algorithm hyperparameter selection using some metalearning methods. *Proceedings of the International Workshop on AutoML at European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 69–83.

Cai, T. and Wu, X. (2010). Selective SVM ensemble based on discretization method. *Proceedings of the International Conference on Computer Engineering and Technology*, page 148–151.

Cano, G., Garcia-Rodriguez, J., Garcia-Garcia, A., Perez-Sanchez, H., Benediktsson, J. A., Thapa, A., and Barr, A. (2017). Automatic selection of molecular descriptors using random forest: Application to drug discovery. *Expert Systems with Applications*, 72(1):151–159.

Celik, O. and Altunaydin, S. S. (2018). A research on machine learning methods and its applications. *Journal of Educational Technology and Online Learning*, 1(3):25–40.

Chali, Y., Sadid, H., and Mojahid, M. (2014). Complex question answering: Homogeneous or heterogeneous, which ensemble is better? *Proceedings of the 19th International Conference on Application of Natural Language to Information Systems*, pages 160–163.

Chaudhary, A., Kolhe, S., and Kamal, R. (2016). A hybrid ensemble for classification in multiclass datasets: An application to oilseed disease dataset. *Computers and Electronics in Agriculture*, 124(1):65–72.

Chawla, N. W. (2003). C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure. *Proceedings of the International Conference on Machine Learning, Workshop on Learning from Imbalanced Datasets II*, pages 1–8.

Chawla, N. W., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357.

Chen, H. and Yao, X. (2009). Regularized negative correlation learning for neural network ensembles. *IEEE Transactions on Neural Networks*, 20(12):1962–1979.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*, page 785–794.

Chen, T. and Wang, Y. (2016). Robotics and computer-integrated manufacturing estimating simulation workload in cloud manufacturing using a classifying artificial neural network ensemble approach. *Robotics and Computer Integrated Manufacturing*, 38(1):42–51.

Cheng, Y. Y., Chan, P. P. K., and Qiu, Z. W. (2012). Random forest based ensemble system for short term load forecasting. *Proceedings of the International Conference on Machine Learning and Cybernetics*, page 52–56.

Chiu, S. L. (1994). Fuzzy model identification based on the cluster estimation. *Journal of Intelligent Fuzzy Systems*, 2(3):267–278.

Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., and Campbell, J. P. (2020). Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science and Technology*, 9(2):1–14.

Ch'ng, C. K. and Mahat, N. I. (2020). Winsorize tree algorithm for handling outlier in classification problem. *International Journal of Operational Research*, 38(2):278–293.

Cieslak, D. A. and Chawla, N. V. (2008). Learning decision trees for unbalanced data. *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, page 241–256.

Claesen, M. and De Moor, B. (2015). Hyperparameter search in machine learning. *Proceedings of the Eleventh Metaheuristics International Conference*, page 10–14.

Clark, P. and Boswell, R. (1991). Rule induction with CN2: Some recent improvements. *Proceedings of the European working session on learning*, page 151–163.

Cohen, W. W. (1995). Fast effective rule induction. *Proceedings of the Twelfth International Conference on Machine Learning*, page 115–23.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems, Elsevier*, 47(4):547–553.

Cortez, P. and Silva, A. (2008). Using data mining to predict secondary school student performance. *Proceedings of the 5th Future Business Technology Conference*, pages 5–12.

Cousineau, D. and Chartier, S. (2010). Outliers detection and treatment: A review. *International Journal of Psychological Research*, 3(1):58–67.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society*, 20(2):215–242.

Cuzzocrea, A., Francis, S. L., and Gaber, M. M. (2013). An information-theoretic approach forsetting the optimal number of decision trees in random forests. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, page 1013–1019.

Czarnowski, I. and Jedrzejowicz, P. (2017). Stacking and rotation-based technique for machine learning classification with data reduction. *Proceedings of the IEEE International Conference on Innovations in Intelligent Systems and Applications*, page 55–60.

Das, K. and Behera, R. N. (2017). A survey on machine learning: Concept, algorithms and applications. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(3):5194–5202.

de Miranda, P., Prudencio, R., de Carvalho, A., and Soares, C. (2012). Combining a multi-objective optimization approach with meta-learning for svm parameter selection. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 2909–2914.

De Vito, S., Massera, E., Piga, M., Martinotto, L., and Di Francia, G. (2008). On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30.

Deng, H., Runger, G., and Tuv, E. (2011). Bias of importance measures for multi-valued attributes and solutions. *Proceedings of the 21st international conference on Artificial neural network*, page 293–300.

Derbeko, P., El-Yaniv, R., and Meir, R. (2002). Variance optimized bagging. *Proceedings of the European Conference on Machine Learning*, page 60–72.

Devi, R. G. and Sumanjani, P. (2015). Improved classification techniques by combining *k*NN and random forest with naïve bayesian classifier. *Proceedings of the IEEE International Conference on Engineering and Technology*, page 1–4.

Dey, A. (2016). Machine learning algorithms: A review. *International Journal of Computer Science and Information Technologies*, 7(3):1174–1179.

Dietterich, T., Kearns, M., and Mansour, Y. (1996). Applying the weak learning framework to understand and improve C4.5. *Proceedings of the 13th International Conference on Machine Learning*, page 96–104.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Proceedings of the International Workshop on Multiple Classifier Systems*, page 1–15.

Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2(1):263–286.

Dietterich, T. G. and Kong, E. B. (1995). Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. *Technical Report, Department of Computer Science, Oregon State University Corvallis*, pages 1–14.

Dodge, J., Smith, N. A., and Allen, P. G. (2017). Promoting diversity in random hyperparameter search using determinantal point processes. *Proceedings of the International Conference on Machine Learning*, page 1–10.

Dogan, A. and Birant, D. (2019). A weighted majority voting ensemble approach for classification. *Proceedings of the 4th International Conference on Computer Science and Engineering*, pages 366–371.

Domeniconi, C. and Yan, B. (2004). Nearest neighbor ensemble. *Proceedings of the International Conference on Pattern Recognition*, page 228–231.

Domingos, P. (2000). A unified bias-variance decomposition for zero-one and squared loss. *Proceedings of the 17th National Conference on Artificial Intelligence*, page 564–569.

Dong, L. J., Li, X. B., and Peng, K. (2013). Prediction of rockburst classification using random forest. *Transactions of Nonferrous Metals Society of China*, 23(2):472–477.

Dua, D. and Graff, C. (2019). UCI machine learning repository. *[http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science*.

Dua, R., Ghotra, M. S., and Pentreath, N. (2017). *Machine Learning with Spark*. Packt Publishing, second edition.

Dudek, G. (2016). Heterogeneous ensembles for short-term electricity demand forecasting. *Proceedings of the 17th International Scientific Conference on Electric Power Engineering*, pages 1–9.

Dudek, G. (2017). Ensembles of general regression neural networks for short-term electricity demand forecasting. *Proceedings of the 18th International Scientific Conference on Electric Power Engineering*, pages 1–5.

Duin, R. P. (2002). The combining classifier: to train or not to train? *Proceedings of the International Conference on Pattern Recognition*, page 765–770.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64.

Eeti, L. N. and Buddhiraju, K. M. (2018). Classification of hyperspectral remote sensing images by an ensemble of support vector machines under imbalanced data. *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, page 2659–2661.

Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., and K, L.-B. (2013). Towards an empirical foundation for assessing bayesian optimization of hyperparameters. *Proceedings of the Neural Information Processing Systems Workshop on Bayesian Optimization in Theory and Practice*, pages 1–5.

El-Habib Daho, M., Settouti, N., Lazouni, M. E. A., and Chikh, M. E. A. (2014). Weighted vote for trees aggregation in random forest. *Proceedings of the International Conference on Multimedia Computing and Systems*, page 438–443.

El-Hindi, E. (2014). Fine tuning the naïve bayesian learning algorithm. *AI Communications*, 27(2):133–141.

El-Hindi, K., Al-Salman, H., Qasem, S., and Al-Ahmadi, S. (2018). Building an ensemble of fine-tuned naïve bayesian classifiers for text classification. *Entropy*, 20(11):1–13.

Elaidi, H., Elhaddar, Y., Benabbou, Z., and Abbar, H. (2018). An idea of a clustering algorithm using support vector machines based on binary decision tree. *Proceedings of the International Conference on Intelligent Systems and Computer Vision*, page 1–5.

Elish, M. O., Helmy, T., and Hussain, M. I. (2013). Empirical study of homogeneous and heterogeneous ensemble models for software development effort estimation. *Mathematical Problems in Engineering*, 6(1):1–22.

Engelbrecht, A. P. (2007). *Computational intelligence: An introduction*. John Wiley Sons, first edition.

Ennett, C. M., Frize, M., and Walker, C. R. (2001). Influence of missing values on artificial neural network performance. *Studies in Health Technology and Informatics*, 84(1):449–453.

Fanaee, T. H. and Gama, J. (2013). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2):113–127.

Fawagreh, K., Gaber, M. M., and Elyan, E. (2014). Diversified random forests using random subspaces. *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, pages 85–92.

Fazelpour, A., Khoshgoftaar, T. M., Dittman, D. J., and Naplitano, A. (2016). Investigating the variation of ensemble size on bagging-based classifier performance in imbalanced bioinformatics datasets. *Proceedings of the IEEE 17th International Conference on Information Reuse and Integration*, page 377–383.

Feiping, N., Zhanxuan, H., and Xuelong, L. (2018). An investigation for loss functions widely used in machine learning. *Communications in Information and Systems*, 18(1):37–52.

Feng, Y., Wang, X., and Zhang, J. (2021). A heterogeneous ensemble learning method for neuroblastoma survival prediction. *IEEE Journal of Biomedical and Health Informatics*, 26(4):1472–1483.

Feng, Z., Mo, L., and Li, M. (2015). A random forest-based ensemble method for activity recognition. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, page 5074–5077.

Fernandez-Aleman, J. L., Carrillo-De-Gea, J. M., Hosni, M., Idri, A., and Garcia-Mateos, G. (2019). Homogeneous and heterogeneous ensemble classification methods in diabetes disease: A review. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, page 3956–3959.

Feurer, M., Springenberg, J. T., and Hutter, F. (2014). Using meta-learning to initialize bayesian optimization of hyperparameters. *Proceedings of the International Conference on Meta-learning and Algorithm Selection*, page 3–10.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Fitzgerald, J. (2014). Bias and variance reduction strategies for improving generalisation performance of genetic. *Doctoral thesis submitted to the Department of Computer Science and Information System, University of Limerick*, pages 48–61.

Fix, E. and Hodges, J. (1951). Disciminatory analysis. nanoparametric discrimination: Consistency properties. *Technical Report, USAF School of Aviation Medicine, Texas*.

Flach, P. A. (2003). The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. *Proceedings of the International Conference on Machine Learning*, page 194–201.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science*, 55(1):119–139.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the 13th International Conference on Machine Learning*, page 148–156.

Friedman, J. and Stuetzle, W. (1981). *Projection Pursuit Regression*, 76(376):817–823.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of *m* rankings. *Annals of Mathematical Statistics*, 11(1):86–92.

Fu, Y. and Browne, A. (2007). Using ensembles of neural networks to improve automatic relevance determination. *Proceedings of the IEEE International Conference on Neural Networks - Conference Proceedings*, page 1590–1594.

Fuchs, K., Gertheiss, J., and Tutz, G. (2015). Nearest neighbor ensembles for functional data with interpretable feature selection. *Chemometrics and Intelligent Laboratory Systems*, 146(1):186–197.

Furnkranz, J. (2002). Pairwise classification as an ensemble technique. *Proceedings of the 13th European Conference on Machine Learning*, page 97–110.

García, S., Molina, D., Lozano, M., and Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: A case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics*, 15(6):617–644.

García-Pedrajas, N. and Ortiz-Boyer, D. (2011). An empirical study of binary classifier fusion methods for multiclass classification. *Information Fusion*, 12(2):111–130.

Gastón, N. and Bagnasco, R. (2007). Don't be naive with naïve bayes. *Technical paper, Intraway*.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58.

Geurts, P. (2002). Contributions to decision tree induction; bias/variance tradeoff and time series classification. *Doctoral Thesis submitted to the Faculty of Applied Science, University of of Liege*, pages 1–260.

Gholami, R. and Fakhari, N. (2017). Support vector machine: Principles, parameters, and applications. In Samui, P., Sekhar, S., and Balas, V., editors, *Handbook of Neural Computation*, pages 515–535. Academic Press, Cambridge, MA, USA.

Glover, F. (1989). Tabu search. *ORSA Journal of Computing*, 1(3):190–206.

Goh, Y. M. and Ubeynarayana, C. U. (2017). Construction accident narrative classification: An evaluation of text mining techniques. *Accident Analysis and Prevention*, 108(1):122–130.

Gokalp, O. and Tasci, E. (2019). Weighted voting based ensemble classification with hyper-parameter optimization. *Proceedings of the Innovations in Intelligent Systems and Applications Conference*, page 1–4.

Gomes, T. A., Prudncio, R. B., Soares, C., Rossi, A. L., and Carvalho, A. (2012). Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing*, 75(1):3–13.

Gopika, D. and Azhagusundari, B. (2014). An analysis on ensemble methods in classification tasks. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(7):7423–7427.

Gorman, R. P. and Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1(1):75–89.

Grandvalet, Y. (2004). Bagging equalizes influence. *Machine Learning*, 55(3):251–270.

Guo, L., Boukir, S., and Chehata, N. (2010). Support vectors selection for supervised learning using an ensemble approach. *Proceedings of the International Conference on Pattern Recognition*, pages 37–40.

Guo, L., Wang, S., and Cao, Z. (2018). An ensemble classifier based on stacked generalization for predicting membrane protein types. *Proceedings of the 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, page 1–6.

Gupta, A. and Thakkar, A. R. (2014). Optimization of stacking ensemble configuration based on various metahueristic algorithms. *Proceedings of the IEEE International Advance Computing Conference*, page 444–451.

Haixiang, G., Yijing, L., Yanan, L., Xiao, L., and Jinling, L. (2016). BPSO-Adaboost-*k*NN ensemble learning algorithm for multi-class imbalanced data classification. *Engineering Applications of Artificial Intelligence*, 49(1):176–193.

Hamzeloo, S., Shahparast, H., and Jahromi, M. Z. (2012). A novel weighted nearest neighbor ensemble classifier. *Proceedings of the International Symposium on Artificial Intelligence and Signal Processing*, page 413–416.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer, 1st edition.

Haykin, S. (1994). Neural networks: A comprehensive foundation. *Pearson Edition Asia*.

Heath, D., Kasif, S., and Salzberg, S. (1993). Induction of oblique decision trees. *Proceedings of the International Joint Conference of Artificial Intelligence*, page 1002–1007.

Himika, Kaur, S., and Randhawa, S. (2008). Global land temperature prediction by machine learning combo approach. *Proceedings of the International Conference on Computing, Communication and Networking Technologies*, pages 1–8.

Hnoohom, N. and Jitpattanakul, A. (2018). Comparison of ensemble learning algorithms for cataract detection from fundus images. *Proceedings of the International Computer Science and Engineering Conference*, page 144–147.

Ho, T. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):823–844.

Hoens, T. R. and Chawla, N. V. (2013). Imbalanced datasets: From sampling to classifiers. In He, H. and Ma, Y., editors, *Imbalanced Learning: Foundations, Algorithms, and Applications*, pages 43–59. John Wiley Sons, Inc., Hoboken, NJ, USA.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, USA.

Hosni, M., Ali, I., Alain, A., and Ali, N. B. (2018). On the value of parameter tuning in heterogeneous ensembles effort estimation. *Soft Computing*, 22(18):5977–6010.

Huang, G., Zhu, Q., and Siew, C. K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. *Proceedings of the IEEE International Joint Conference on Neural Networks*, page 25–29.

Hui, L., Hong-qi, T., Yan-fei, L., and Lei, Z. (2015). Comparison of four adaboost algorithm based artificial neural networks in wind speed predictions. *Energy Conversion and Management*, 92(1):67–81.

Hussain, H., Benkrid, K., Hong, C., and Seker, H. (2012). An adaptive FPGA implementation of multi-core k-nearest neighbour ensemble classifier using dynamic partial reconfiguration. *Proceedings of the 22nd International Conference on Field Programmable Logic and Applications*, page 627–630.

Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *Proceedings of the International Conference on Learning and Intelligent Optimization*, page 507–523.

Iglesias, J. A., Ledezma, A., and Sanchis, A. (2014). An ensemble method based on evolving classifiers: Estacking. *Proceedings of the IEEE Symposium Series on Computational Intelligence: Evolving and Autonomous Learning Systems*, page 124–131.

Igual, L. and Seguí, S. (2017). *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applicationsm*. Springer, 1st edition.

Iman, R. L. and Davenport, J. M. (1980). Approximations of the critical region of the friedman statistic. *Communications in Statistics-Theory and Methods*, 9(6):571–595.

Ishibuchi, H., Nakashima, T., and Morisawa, T. (1999). Voting in fuzzy rules-based systems for pattern classification problems. *Fuzzy Sets Systems*, 103(2):223–238.

Iswarya, P. and Radha, V. (2015). Ensemble learning approach in improved k-nearest neighbor algorithm for text categorization. *Proceedings of the IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, page 1–5.

Jakubovitz, D., Giryes, R., and Rodrigues, M. (2019). Generalization error in deep learning. In Boche H., Caire G., C. R. K. G. M. R. P. P., editor, *Compressed Sensing and Its Applications. Applied and Numerical Harmonic Analysis*, page 153–193. Springer, Birkhäuser.

Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3):665–685.

John, G. H. (1995). Robust decision trees: Removing outliers from databases. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 174–179.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, page 338–345.

Joshi, M. S. and Kulkarni, V. Y. (2014). Analysis of methods and strategies for diversity based generation of classifier ensemble. *International Journal of Advanced Engineering and Global Technology*, 2(2):431–437.

Jurek, A., Bi, Y., Wu, S., and Nugent, C. (2011). Classification by clusters analysis - an ensemble technique in a semi-supervised classification. *Proceedings of the International Conference on Tools with Artificial Intelligence*, page 876–878.

Kadam, V. J. and Jadhav, S. M. (2020). Performance analysis of hyperparameter optimization methods for ensemble learning with small and medium sized medical datasets. *Journal of Discrete Mathematical Sciences and Cryptography*, 23(1):115–123.

Kadkhodaei, H. and Moghadam, A. M. E. (2016). An entropy based approach to find the best combination of the base classifiers in ensemble classifiers based on stack generalization. *Proceedings of the 4th International Conference on Control, Instrumentation, and Automation*, page 425–429.

Kalousis, A. and Hilario, M. (2000). Supervised knowledge discovery from incomplete data. *Proceedings of the 2nd International Conference on Data Mining*, pages 269–278.

Kalt, T. and Croft, W. B. (1996). A new probabilistic model of text classification and retrieval. *Technical Report, University of Massachusetts Center for Intelligent Information Retrieval*.

Kanamori, T., Fujiwara, S., and Takeda, A. (2014). Breakdown point of robust support vector machine. *Technical Paper, Nagoya University*, pages 1–27.

Kantardzic, M. (2011). *Data Mining: Concepts, Models, Methods, and Algorithms*. IEEE Press, A John Wiley and Sons, Inc.

Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. *M.Sc. thesis, Dept. of Mathematics, University of Chicago, Chicago, Illinois*.

Kaur, K., Kaur, G., and Jaspreet, W. (2018). Neural network ensemble and jaya algorithm based diagnosis of brain tumor using mri images. *Journal of The Institution of Engineers*, 99(5):509–517.

Kaya, H., Tufekci, P., and Uzun, E. (2019). Predicting CO and NOx emissions from gas turbines: Novel data and a benchmark PEMS. *Turkish Journal of Electrical Engineering Computer Sciences*, 27(6):4783–4796.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 3149–3157.

Kelleher, J. D., Mac, N. B., and A., D. (2015). *Fundamentals of machine learning for predictive data analytics*. MIT Press, Massachusetts.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization (pso). *Proceedings of IEEE international conference on neural networks*, page 1942–1948.

Khan, M. M., Mendes, A., and Chalup, S. K. (2018). Ensembles for breast cancer and parkinson's disease prediction. *PLoS ONE*, 13(2):1–15.

Khan, M. M., Mendes, A., Zhang, P., and Chalup, S. K. (2017). Evolving multi-dimensional wavelet neural networks for classification using cartesian genetic programming. *Neurocomputing*, 247(C):39–58.

Khan, M. U., Mushtaq, Z., Shakeel, M., Aziz, S., and Naqvi, S. Z. H. (2020). Classification of myocardial infarction using MFCC and ensemble subspace *k*NN. *Proceedings of the International Conference on Electrical, Communication, and Computer Engineering*, pages 1–5.

Khosravi, P., Vergari, A., Choi, Y. J., Liang, Y., and V., B. G. (2020). Handling missing data in decision trees: A probabilistic approach. *Proceedings of the International Conference on Machine Learning*, pages 1–8.

Kilimci, Z. H., Akyokus, S., and Omurca, S. I. (2016). The effectiveness of homogenous ensemble classifiers for turkish and english texts. *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications*, page 1–7.

Kilimci, Z. H., Akyokus, S., and Omurca, S. I. (2017). The evaluation of heterogeneous classifier ensembles for turkish texts. *Proceedings of the IEEE International Conference on Innovations in Intelligent Systems and Applications*, page 307–311.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239.

Klein, B. D. and Rossin, D. F. (1999). Data quality in neural network models: effect of error rate and magnitude of error on predictive accuracy. *Omega*, 27(5):569–582.

Klement, W., Wilk, S., Michalowski, W., Farion, K. J., Osmond, M. H., and Verter, V. (2012). Predicting the need for CT imaging in children with minor head injury using an ensemble of naïve bayes classifiers. *Artificial Intelligence in Medicine*, 54(3):163–170.

Knerr, S., Personnaz, L., and Dreyfus, G. (1990). Single-layer learning revisited: a stepwise procedure for building and training a neural network. *Neurocomputing*, 68(1):41–50.

Knox, S. W. (2018). *Machine Learning: A Concise Introduction*. John Wiley Sons, Inc, first edition.

Kohavi, R. (1996). Scaling up the accuracy of naïve bayes classifiers: A decision-tree hybrid. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page 202–207.

Kohavi, R., Becker, B., and Sommerfield, D. (1997). Improving simple bayes. *Proceedings of the European Conference on Machine Learning*, pages 1–10.

Kong, E. and Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance in machine learning. *Proceedings of the Twelfth International Conference on Machine Learning*, page 313–321.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of relief. *Proceedings of the European Conference on Machine Learning*, pages 171–182.

Krawczyk, B., Galar, M., Jeleń, L., and Herrera, F. (2016). Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. *Applied Soft Computing*, 38(1):714–726.

Krishnapuram, R., Joshi, A., and Yi, L. (1999). A fuzzy relative of the *k*-medoids algorithm with application to web document and snippet clustering. *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 1281–1286.

Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. *Proceedings of 2nd Berkeley Symposium*, page 481–492.

Kuncheva, L., Bezdek, J., and Duin, R. (2001). Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, 34(2):299–314.

Kuncheva, L. I. and Rodriguez, J. J. (2007). Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):500–508.

Kundu, S. and Ari, S. (2019). P300 based character recognition using sparse autoencoder with ensemble of svms. *Biocybernetics and Biomedical Engineering*, 39(4):956–966.

Kwak, S. K. and Kim, J. H. (2017). Statistical data preparation: Management of missing values and outliers. *Korean journal of anesthesiology*, 70(4):407–411.

Larasati, R., . K. H. (2017). Handwritten digits recognition using ensemble neural networks and ensemble decision tree. *Proceedings of the International Conference on Smart Cities, Automation and Intelligent Computing Systems*, page 99–104.

Large, J., Lines, J., and Bagnall, A. (2019). A probabilistic classifier ensemble weighting scheme based on cross-validated accuracy estimates. *Data Mining and Knowledge Discovery*, 33(6):1674–1709.

Lawi, A., Aziz, F., and Syarif, S. (2018). Ensemble GradientBoost for increasing classification accuracy of credit scoring. *Proceedings of the 4th International Conference on Computer Applications and Information Processing Technology*, page 1–4.

Lee, E. S. (2017). Exploring the performance of stacking classifier to predict depression among the elderly. *Proceedings of the IEEE International Conference on Healthcare Informatics*, page 13–20.

Lee, I. and Shin, Y. J. (2020). Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, 63(2):157–170.

Lee, J. and Lee, J. H. (2014). K-means clustering based svm ensemble methods for imbalanced data problem. *Proceedings of the Joint 7th International Conference on Soft Computing and Intelligent Systems*, page 614–617.

Lee, M., Yun, J. J., Pyka, A., Won, D., Kodama, F., Schiuma, G., Park, H., Jeon, J., Park, K., Jung, K., Yan, M., Lee, S., and Zhao, X. (2018). How to respond to the fourth industrial revolution, or the second information technology revolution? dynamic new combinations between technology, market, and society through open innovation. *Journal of Open Innovation: Technology, Market and Complexity*, 4(3):1–24.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.

Lewis, D. D. (1998). Naïve bayes at forty: The independence assumption in information retrieval. *Proceedings of the European Conference on Machine Learning*, page 4–15.

Lewis, N. D. (2015). *92 Applied Predictive Modelling Techniques in R*. CreateSpace Independent Publishing Platform.

Li, H., Wang, J., Gao, T., Lu, Y., and Su, Z. (2010). Accurate prediction of the optical absorption energies by neural network ensemble approach. *Proceedings of the 5th International Conference on Frontier of Computer Science and Technology*, page 503–507.

Li, H. G., Wu, G. Q., Hu, X. G., Zhang, J., Li, A., and Wu, X. (2011a). *K*-means clustering with bagging and MapReduce. *Proceedings of the Annual Hawaii International Conference on System Sciences*, page 1–8.

Li, J. J., Alzami, F., Gong, Y. J., and Yu, Z. (2017a). A multi-label learning method using affinity propagation and support vector machine. *IEEE Access*, 5(1):2955–2966.

Li, K. and Hao, L. (2009). Naïve bayes ensemble learning based on oracle selection. *Proceedings of the Chinese Control and Decision Conference*, page 665–670.

Li, M., Chen, W., and Zhang, T. (2017b). Biomedical signal processing and control classification of epilepsy EEG signals using DWT-based envelope analysis and neural network ensemble. *Biomedical Signal Processing and Control*, 31(1):357–365.

Li, W., Ding, S., Chen, Y., and Yang, S. (2018). Heterogeneous ensemble for default prediction of peer-to-peer lending in China. *IEEE Access*, 6:54396–54406.

Li, X. G., Yao, M. F., and Huang, W. T. (2011b). Speech recognition based on k-means clustering and neural network ensembles. *Proceedings of the 7th International Conference on Natural Computation*, page 614–617.

Liang, K. and Zhou, Z. (2012). Using an ensemble classifier on learning evaluation for e-learning system. *Proceedings of the International Conference on Computer Science and Service System*, page 538–541.

Lin, S. W., Ying, K. C., Chen, S. C., and Lee, Z. J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert systems with applications*, 35(4):1817–1824.

Lin, W., Wu, Z., Lin, L., Wen, A., and Li, J. (2017). An ensemble random forest algorithm for insurance big data analysis. *Proceedings of the IEEE International Conference on Computational Science and Engineering*, page 531–536.

Liu, B., Hao, Z. F., Lu, J., and Liu, S. Q. (2007). Apply support vector machine for CRM problem. *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, page 3288–3292.

Liu, F. T., Ting, K. M., and Zhou, Z. H. (2008). Isolation forest. *Proceedings of the Eighth IEEE International Conference on Data Mining*, pages 413–422.

Liu, W., S., C., Cieslak, D. A., and Chawla, N. V. (2010). A robust decision tree algorithm for imbalanced data sets. *Proceedings of the SIAM International Conference on Data Mining*, pages 1–12.

Liu, X., Jin, J., Wu, W., and Herz, F. (2020). A novel support vector machine ensemble model for estimation of free lime content in cement clinkers. *ISA Transactions*, 99(1):479–487.

Liu, Y. and Huang, L. (2019). A novel ensemble support vector machine model for land cover classification. *International Journal of Distributed Sensor Networks*, 15(4):1–9.

Liu, Y. and Wu, H. (2018). Prediction of road traffic congestion based on random forest. *Proceedings of the 10th International Symposium on Computational Intelligence and Design*, page 361–364.

Liu, Y., Yang, Y., and Carbonell, J. (2002). Boosting to correct inductive bias in text classification. *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, page 348–355.

Lowd, D. and Domingos, P. (2005). Naïve bayes models for probability estimation. *Proceedings of the 22nd International Conference on Machine Learning*, page 529–536.

Lu, K. and Wang, L. (2011). A novel nonlinear combination model based on support vector machine for rainfall prediction. *Proceedings of the 4th International Joint Conference on Computational Sciences and Optimization*, page 1343–1346.

Luo, G. (2016). A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1):1–16.

Luong, A. V., Vu, T. H., Nguyen, P. M., Van-Pham, N., McCall, J., Liew, A. W. C., and Nguyen, T. T. (2020). A homogeneous-heterogeneous ensemble of classifiers. *Proceedings of the International Conference on Neural Information Processing*, page 251–259.

Lutu, P. E. N. (2015). Naïve bayes classification ensembles to support modeling decisions in data stream mining. *Proceedings of the IEEE Symposium Series on Computational Intelligence*, page 335–340.

Ma, Y., Kong, X., and Wang, X. (2011). Support vector machine ensemble based on independent component analysis and fuzzy kernel clustering. *Proceedings of the Chinese Control and Decision Conference*, pages 752–755.

Maia, M. R. D. H., Plastino, A., and Freitas, A. A. (2021). An ensemble of naïve bayes classifiers for uncertain categorical data. *Proceedings of the IEEE International Conference on Data Mining*, pages 1222–1227.

Maier, H. R. and Dandy, G. C. (1998). The effect of internal parameters and geometry on the performance of back-propagation neural networks: An empirical study. *Environmental Modelling and Software*, 13(2):193–209.

Maier, H. R. and Dandy, G. C. (2000). Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environmental Modelling Software*, 15(1):101–124.

Man, W., Ji, Y., and Zhang, Z. (2018). Image classification based on improved random forest algorithm. *Proceedings of the 3rd IEEE International Conference on Cloud Computing and Big Data Analysis*, page 346–350.

Mandić-Rajčević, S. and Colosio, C. (2019). Methods for the identification of outliers and their influence on exposure assessment in agricultural pesticide applicators: A proposed approach and validation using biological monitoring. *Toxics*, 7(37):1–14.

Mantovani, R. G., Rossi, A. L. D., Alcobaça, E., Vanschoren, J., and de Carvalho, A. C. P. L. F. (2019). A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers. *Information Sciences*, 501(1):193–221.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

Marqués, A. I., García, V., and Sánchez, J. S. (2012). Exploring the behaviour of base classifiers in credit scoring ensembles. *Expert Systems with Applications*, 39(11):10244–10250.

Martinez-Muñoz, G., Hernández-Lobato, D., and Suarez, A. (2009). An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):245–259.

Mayr, A., Binder, H., Gefeller, O., and Schmid, M. (2014). The evolution of boosting algorithms–from machine learning to statistical modelling. *Methods Information in Medicine*, 53(6):419–427.

McCallum, A. and Nigam, K. (1998). A comparison of event models for naïve bayes text classification. *Proceedings of the Fifteenth National Conference on Artificial Intelligence: Workshop on Learning for Text Categorization*, pages 41–48.

Medicherla, S. (2018). Bias and variance trade-off in regression models. *International Journal of Management and Applied Science*, 4(3):1–7.

Mehta, P., Bukov, M., Wang, C. H., Day, A. G. R., Richardson, C., Fisher, C. K., and Schwab, D. J. (2019). A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810(3):1–124.

Melville, P. and Mooney, R. J. (2005). Creating diversity in ensembles using artificial data. *Information Fusion*, 6(1):99–111.

Mendes-Moreira, J., Jorge, A. M., Freire de Sousa, J., and Soares, C. (2015). Improving the accuracy of long-term travel time prediction using heterogeneous ensembles. *Neurocomputing*, 150(B):428–439.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London*, 83(559):415–446.

Merentitis, A., Debes, C., and Heremans, R. (2014). Ensemble learning in hyperspectral image classification: Toward selecting a favourable bias-variance tradeoff. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(4):1089–1102.

Mirjalili, S. (2015). The Ant Lion Optimizer. *Advances in Engineering Software*, 83(1):80–98.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Mitchell, T. M. (1980). The need for biases in learning generalizations. *Technical Report, Rutgers University*.

Montgomery, D. C. (2012). *Design and analysis of experiments*. Wiley, Hoboken, 8th edition.

Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58(302):415–434.

Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62(1):22–31.

Moudrik, J. and Neruda, R. (2015). Evolving non-linear stacking ensembles for prediction of go player attributes. *Proceedings of the IEEE Symposium Series on Computational Intelligence*, page 1673–1680.

Muhammad, I. and Yan, Z. (2015). Supervised machine learning approaches: A survey. *International Journal of Soft Computing*, 5(3):946–952.

Musumeci, F., Rottondi, C., and Nag, A. (2019). An overview on application of machine learning techniques in optical networks. *IEEE Communications Surveys and Tutorials*, 21(2):1383–1408.

Nah, S. and Lee, K. M. (2016). Random forest with data ensemble for saliency detection. *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, page 604–607.

Nakayama, H., Yun, Y. B., and Yoon, M. (2009). *Sequential Approximate Multi-objective Optimization Using Computational Intelligence*. Springer.

Nan, L. and Xiang, C. Z. (2014). A wavelet transform based support vector machine ensemble algorithm and its application in network intrusion detection. *Proceedings of the 5th International Conference on Intelligent Systems Design and Engineering Applications*, page 109–113.

Nautiyal, D. (2018). Underfitting and overfitting in machine learning. *Technical Report*.

Neal, B. (2019). On the bias-variance tradeoff: Textbooks need an update. *M.Sc Thesis submitted to Department of Computer Science and Operations Research, Faculty of Arts and Sciences, Université de Montréal*, pages 1–75.

Nguyen, C., Starek, M. J., Tissot, P. E., Cai, X., and J., G. (2019a). Ensemble neural networks for modeling DEM error. *International Journal of Geo-Information*, 8(10):1–23.

Nguyen, T., Nguyen, M. P., Pham, X. C., and Liew, A. W. C. (2018). Heterogeneous classifier ensemble with fuzzy rule-based meta learner. *Information Science*, 422(1):144–160.

Nguyen, T. T., Dang, M. T., Pham, T. D., Dao, L. P., Luong, A. V., McCall, J., and Liew, A. W. C. (2019b). Deep heterogeneous ensemble. *Proceedings of the International conference on neural information processing*, pages 1–9.

Nguyen, T. T., Pham, T. D., Dang, M. T., Luong, A. V., McCall, J., and Liew, A. W. C. (2020). Multi-layer heterogeneous ensemble with classifier and feature selection. *Proceedings of the Genetic and Evolutionary Computation Conference*, page 725–733.

Nikolić, S., Knežević, M., Ivančević, V., and Luković, I. (2014). Building an ensemble from a single naïve bayes classifier in the analysis of key risk factors for polish state fire service. *Proceedings of the Federated Conference on Computer Science and Information Systems*, page 361–367.

Nowlan, J. and Hinton, G. E. (1992). Simplifying neural networks by soft weight-sharing. *Neural Computation*, 4(4):473–493.

Nyitrai, T. and Virag, M. (2019). The effects of handling outliers on the performance of bankruptcy prediction models. *Socio-Economic Planning Sciences*, 67(1):34–42.

Obilor, E. I. and Amadi, E. C. (2018). Test for significance of pearson's correlation coefficient. *International Journal of Innovative Mathematics, Statistics, and Energy Policies*, 6(1):11–23.

Okun, O. and Priisalu, H. (2009). Dataset complexity in gene expression based cancer classification using ensembles of *k*-nearest neighbors. *Artificial Intelligence in Medicine*, 45(2–3):151–162.

Olave, M., Rajkovic, V., and Bohanec, M. (1989). An application for admission in public school systems. In Snellen, I. T. M., van de Donk, W. B. H. J., and Baquiast, J. P., editors, *Expert Systems in Public Administration*, pages 145–160. Elsevier.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11(1):169–198.

Opitz, D. and Shavlik, J. (1995). Generating accurate and diverse members of a neural network ensemble. *Proceedings of the 8th International Conference on Neural Information Processing Systems*, page 535–541.

Opitz, D. W. and Shavlik, J. W. (1996). Actively searching for an effective neural network ensemble. *Connection Science*, 8(3-4):337–354.

Ortigosa, I., Lopez, R., and Garcia, J. (2007). A neural network approach to residuary resistance of sailing yachts prediction. *Proceedings of the International Conference on Marine Engineering*, pages 223–226.

Osman, A. H. and Aljahdali, H. M. A. (2020). An effective of ensemble boosting learning method for breast cancer virtual screening using neural network model. *IEEE Access*, 8(1):39165–39174.

Osuna, E., Freund, R., and Girosi, F. (1997). An improved training algorithm for support vector machines. *Proceedings of the IEEE Signal Processing Society Workshop*, page 276–285.

Ougiaroglou, S. and Evangelidis, G. (2015). Dealing with noisy data in the context of *k*-nn classification. *Proceedings of the 7th Balkan Conference on Informatics*, pages 1–4.

Owens, M. T. and Tanner, K. D. (2017). Teaching as brain changing: Exploring connections between neuroscience and innovative teaching. *CBE—Life Sciences Education*, 16(2):1–9.

Pachange, S., Joglekar, B., and Kulkarni, P. (2016). An ensemble classifier approach for disease diagnosis using random forest. *Proceedings of the 12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, and Control*, page 1–5.

Palaninathan, A. C., Qiu, X., and Suganthan, P. N. (2017). Heterogeneous ensemble for power load demand forecasting. *Proceedings of the IEEE Region 10 Annual International Conference*, page 2040–2045.

Pandey, T. N. (2017). Credit risk analysis using machine learning classifiers. *Proceedings of the International Conference on Energy, Communication, Data Analytics, and Soft Computing*, page 1850–1854.

Pang, Y., Judd, N., Brien, J. O., and Ben-avie, M. (2017). Predicting students graduation outcomes through support vector machines. *Proceedings of the Frontiers in Education Conference*, pages 1–8.

Pao, Y. and Takefuji, Y. (1992). Functional-link net computing: theory, system architecture, and functionalities. *IEEE Computers*, 25(5):76–79.

Patel, B. N. (2012). Efficient classification of data using decision tree. *Bonfring International Journal of Data Mining*, 2(1):6–12.

Paul, A., Mukherjee, D. P., Das, P., Gangopadhyay, A., Chintha, A. R., and Kundu, S. (2018). Improved random forest for classification. *IEEE Transactions on Image Processing*, 27(8):4012–4024.

Peerlinck, A., Sheppard, J., and Senecal, J. (2019). Adaboost with neural networks for yield and protein prediction in precision agriculture. *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8.

Peng, S. and Zhu, S. (2009). Application of neural network ensemble in nonlinear time-serials forecasts. *Proceedings of the 2nd International Conference on Intelligent Computing Technology and Automation*, page 45–47.

Platt, J. C., Christiani, N., and Shawe-Taylor, J. (2000). Large margin dags for multiclass classification. *Proceedings of Neural Information Processing Systems*, page 547–553.

Podgorelec, V., Kokol, P., Stiglic, B., and Rozman, I. (2002). Decision trees: An overview and their use in medicine. *Journal of Medical System*, 26(5):445–463.

Poh, C. Q. X., Ubeynarayana, C. U., and Goh, Y. M. (2018). Safety leading indicators for construction sites: A machine learning approach. *Automation in Construction*, 93(1):375–386.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45.

Potdar, K., Pardawala, S. T., and Pai, D. C. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175(4):7–9.

Prachi, T., Mishra, S., Lakhotia, M., and Goyal, K. (2019). Bias and variance tradeoff in classification algorithm on the census income dataset. 6(3):4–8.

Prakash, J. S., Vignesh, A. K., Ashok, C., and Adithyan, R. (2012). Multiclass support vector machines classifier for machine vision application. *Proceedings of the International Conference on Machine Vision and Image*, page 197–199.

Preuveneers, D., Tsingenopoulos, I., and Joosen, W. (2020). Resource usage and performance trade-offs for machine learning models in smart environments. *Sensors*, 20(4):1–27.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3):221–234.

Rafiei, M. H. and Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering Management*, 142(2):1–10.

Rahman, A. and Verma, B. (2011). Novel layered clustering-based approach for generating ensemble of classifiers. *IEEE Transactions on Neural Networks*, 22(5):781–792.

Ramana, B. V., Babu, M. S. P., and Venkateswarlu, N. B. (2011). Critical study of selected classification algorithms for liver disease diagnosis. *International Journal of Database Management Systems*, 3(2):101–114.

Raschka, S. (2018). Machine learning. *Technical report, Department of Statistics, University of Wisconsin–Madison*.

Reif, M., Shafait, F., and Dengel, A. (2012). Meta-learning for evolutionary parameter optimization of classifiers. *Machine learning*, 87(3):357–380.

Röbel, A. (1994). The dynamic pattern selection algorithm: Effective training and controlled generalization of backpropagation neural networks. *Proceedings of the International Conference on Artificial Neural Networks*, page 643–646.

Rocha, A. and Goldenstein, S. K. (2014). Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):289–302.

Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630.

Rohman, B. P. A. and Kurniawan, D. (2017). Classification of radar environment using ensemble neural network with variation of hidden neuron number. *Jurnal Elektronika dan Telekomunikasi*, 17(1):19–24.

Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics and Data Analysis*, 53(12):4046–4072.

Rokach, L. and Maimon, O. (2014). *Data Mining with Decision Trees: Theory and Application*. World Scientific Publishing, Singapore.

Rooney, N., Patterson, D., Anand, S., and Tsymbal, A. (2004). Dynamic integration of regression models. *Proceedings of the 5th International Workshop, Multiple Classifier Systems*, pages 164–173.

Rosenblatt, F. (1957). The perceptron: A perceiving and recognizing automaton. *Technical report, Cornell Aeronautical Laboratory*.

Rubin, D. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Ruggieri, S. (2002). Efficient c4. 5 classification algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):438–444.

Rumelhart, D. E., Hinton, G., and Williams, R. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Foundations of Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362. MIT Press.

Sagayaraj, S. and Santhoshkumar, M. (2020). Heterogeneous ensemble learning method for personalized semantic web service recommendation. *International Journal of Information Technology*, 12(2):983–994.

Sandbhor, S. and Chaphalkar, N. B. (2019). Impact of outlier detection on neural networks based property value prediction. In Satapathy, S., Bhateja, V., Somanah, R., Yang, X. S., and Senkerik, R., editors, *Information Systems Design and Intelligent Applications: Advances in Intelligent Systems and Computing*, page 481–495. Springer, CSingapore.

Sarkar, S., Pramanik, A., Maiti, J., and Reniers, G. (2020). Predicting and analyzing injury severity: A machine learning-based approach using class-imbalanced proactive and reactive data. *Safety Science*, 125(1):1–23.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2):197–227.

Schober, P., Boer, M. C., and Schwarte, L. A. (2018). Correlation coefficients: Appropriate use and interpretation. *Anesthesia Analgesia*, 126(5):1763–1768.

Scholkopf, B., Smola, A., Williamson, R. C., and Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, 12(5):1207–1245.

Schonlau, M. and Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, 20(1):3–29.

Sebban, M., R., N., Chauchat, J. H., and Rakotomalala, R. (2000). Impact of learning set quality and size on decision tree performances. *International Journal of Computer Systems and Signals*, 1(1):85–105.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

Shailaja, K., Seetharamulu, B., and Jabbar, M. A. (2019). Machine learning in healthcare: A review. *Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology*, pages 910–914.

Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Sharma, R., Kamble, S. S., Gunasekaran, A., Kumar, V., and Kumar, A. (2020). A systematic literature review on machine learning applications for sustainable agriculture supply chain performance. *Computers and Operations Research*, 119(1):1–17.

Shi, H. and Lv, X. (2010). The naïve bayesian classifier learning algorithm based on adaboost and parameter expectations. *Proceedings of the 3rd International Joint Conference on Computational Sciences and Optimization: Theoretical Development and Engineering Practice*, page 377–381.

Shi, Z. (2020). Improving *k*-nearest neighbors algorithm for imbalanced data classification. *Proceedings of the IOP Conference Series on Materials Science and Engineering*.

Shilen, S. (1990). Multiple binary tree classifiers. *Pattern Recognition*, 23(7):757–763.

Simeone, O. (2018). A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*, 4(4):648–664.

Singh, P. K., Sarkar, R., and Nasipuri, M. M. (2016). Significance of non-parametric statistical tests for comparison of classifiers over multiple datasets. *International Journal of Computing Science and Mathematics*, 7(5):410–442.

Snyders, S. and Omlin, C. W. (2000). What inductive bias gives good neural network training performance? *Proceedings of the International Joint Conference on Neural Networks*, page 445–450.

Solomatine, D. P. and Shrestha, D. L. (2004). Adaboost.RT: A boosting algorithm for regression problems. *Proceedings of the IEEE International Joint Conference on Neural Networks*, page 1163–1168.

Son, H., Kim, C., Hwang, N., Kim, C., and Kang, Y. (2013). Classification of major construction materials in construction environments using ensemble classifiers. *Advanced Engineering Informatics*, 28(1):1–10.

Song, Y. Y. and Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2):130–135.

Soundarya, M. and Balakrishnan, R. (2014). Survey on classification techniques in data mining. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(7):1842–1845.

Souza, S. X., Suykens, J. A., Vandewalle, J., and Bolle, D. (2010). Coupled sim- ulated annealing. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(2):320–335.

Srisuan, J. and Hanskunatai, A. (2014). The ensemble of naïve bayes classifiers for hotel searching. *Proceedings of the International Computer Science and Engineering Conference*, page 168–173.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Staroverov, B. A. and Gnatyuk, B. A. (2016). Universal energy consumption forecasting system based on neural network ensemble. *Optical Memory and Neural Networks*, 25(3):198–202.

Steinwart, I. (2008). *Support Vector Machines*. Springer Science and Business Media.

Stewart, T. G., Zeng, D., and Wu, M. C. (2018). Constructing support vector machines with missing data. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(4):1–27.

Suarez-Alvarez, M. M., Pham, D. T., Prostov, M. Y., and Prostov, Y. I. (2012). Statistical approach to normalization of feature vectors and clustering of mixed datasets. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, page 2630–2651.

Sumathi, M. R. and Poorna, B. (2017). Design and development of ensemble of naïve bayes classifiers to predict social and communication deficiency among children. *International Journal of Applied Engineering Research*, 12(24):14190–14198.

Sun, B., Luo, J., Shu, S., and Yu, N. (2010). A new approach of random forest for multiclass classification problem. *Proceedings of the 5th International Conference on Computer Science and Education*, page 6–8.

Sun, J., Li, H., and Adeli, H. (2013). Concept drift-oriented adaptive and dynamic support vector machine ensemble with time window in corporate financial risk prediction. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 43(4):801–813.

Sun, W., Lv, Y., Li, G., and Chen, Y. (2020). Modeling river ice breakup dates by *k*-nearest neighbor ensemble. *Water*, 12(1):1–18.

Tahir, M. A. and Smith, J. (2010). Creating diverse nearest-neighbour ensembles using simultaneous metaheuristic feature selection. *Pattern Recognition Letters*, 31(11):1470–1480.

Tan, K. H., Ji, G., Lim, C. P., and Tseng, M. L. (2017). Using big data to make better decisions in the digital economy. *International Journal of Production Research*, 55(17):1–4.

Tan, M. and Eshelman, L. (1988). Using weighted networks to represent classification knowledge in noisy domains. *Proceedings of the Fifth International Conference on Machine Learning*, pages 121–134.

Tang, Y., Qiu, S., and Gui, P. (2019). Predicting housing price based on ensemble learning algorithm. *Proceedings of the International Conference on Artificial Intelligence and Data Processing*, page 1–5.

Tangirala, S. (202). Evaluating the impact of gini index and information gain on classification using decision tree classifier algorithm. *International Journal of Advanced Computer Science and Applications*, 11(2):612–619.

Tax, D. M., Duin, R. P., and Breukelen, M. V. (1997). Comparison between product and mean classifier combination rules. *Proceedings of the Workshop on Statistical Pattern Recognition*, page 165–170.

Ted, S. E. (2005). Multi-stage classification. *Proceedings of the Fifth IEEE International Conference on Data Mining*, page 386–393.

Tetko, I. V., Livingstone, D. J., and Luik, A. I. (1995). Neural network studies: Comparison of overfitting and overtraining. *Journal of Chemical Information and Modeling*, 35(5):826–833.

Tewari, S. and Dwivedi, U. D. (2020). A comparative study of heterogeneous ensemble methods for the identification of geological lithofacies. *Journal of Petroleum Exploration and Production Technology*, 10(5):1849–1868.

Thakur, D., Markandaiah, N., and Raj, D. S. (2010). Re-optimization of ID3 and C4.5 decision tree. *Proceedings of the International Conference on Computer and Communication Technology*, page 448–450.

Thewsuwan, S., Ozaki, N., and Horio, K. (2018). Svm ensemble approaches for improving texture classification performance based on complex network model with spatial information. *Proceedings of the International Workshop on Advanced Image Technology*, page 1–3.

Thorhallsson, H. and Singh, G. (2017). Visualizing the bias-variance tradeoff. *Technical Report, submitted to Computer Science Department, University of Birtish Columbia*, pages 1–9.

Tierney, N. J., Harden, F. A., Harden, M. J., and Mengersen, K. L. (2015). Using decision trees to understand structure in missing data. *BMJ Open*, 5(1):1–11.

Ting, K. M. and Witten, I. H. (1999). Issues in stacked generalization. *Journal of artificial intelligence research*, 10(1):271–289.

Torres, M. E., Colominas, M. A., Schlotthauer, G., and Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, page 4144–4147.

Tou, J. T. and Gonzalex, R. (1974). *Pattern recognition principles*. Addison Wesley, London.

Tsai, C. F., Lin, Y. C., Yen, D. C., and Chen, Y. M. (2011). Predicting stock returns by classifier ensembles. *Applied Soft Computing*, 11(2):2452–2459.

Tsai, J. T., Chou, J. H., and Liu, T. K. (2006). Tuning the structure and parameters of a neural network by using hybrid taguchi-genetic algorithm. *IEEE Transactions on Neural Network*, 17(1):69–80.

Tsanas, A., Little, M. A., McSharry, P. E., and Ramig, L. O. (2009). Accurate telemonitoring of parkinson's disease progression by non-invasive speech tests. *IEEE transactions on bio-medical engineering*, 57(4):884–893.

Tsanas, A. and Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49(1):560–567.

Tuarob, S., Tucker, C. S., Salathe, M., and Ram, N. (2014). An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages. *Journal of Biomedical Informatics*, 49(1):255–268.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison Wesley, Reading.

Vaghela, V. B., Ganatra, A., and Thakkar, A. (2009). Boost a weak learner to a strong learner using ensemble system approach. *Proceedings of the IEEE International Advance Computing Conference*, page 1432–1436.

Valentini, G. and Dietterich, T. G. (2004). Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. *Journal of Machine Learning Research*, 5(1):725–775.

Van Der Putten, P. and Van Someren, M. (2004). A bias-variance analysis of a real world learning problem: The coil challenge 2000. *Machine Learning*, 57(2):177–195.

Van Erp, M., Vuurpijl, L., and Schomaker, L. (2002). An overview and comparison of voting methods for pattern recognition. *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition*, page 195–200.

Vapnik, K., Golowich, S., and Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. *Proceedings of the 9th International Conference on Neural Information Processing Systems*, pages 281–287.

Vapnik, V. C. and Cortes, C. (1995). Support vector networks. *Machine Learning*, 20(3):273–297.

Vapnik, V. N. (2000). *The nature of statistical learning theory*. Springer, New York, second edition.

Verma, A. and Mehta, S. (2017). A comparative study of ensemble learning methods for classification in bioinformatics. *Proceedings of the 7th International Conference on Cloud Computing, Data Science and Engineering*, page 155–158.

Veropoulos, K., Campbell, C., and Cristianini, N. (1999). Controlling the sensitivity of support vector machines. *Proceedings of the International Joint Conference on AI*, page 55–60.

Victoriano, J. M., Lacatan, L. L., and Vinluan, A. A. (2020). Predicting river pollution using random forest decision tree with gis model: A case study of mmors, philippines. *International Journal of Environmental Science and Development*, 11(1):36–42.

Vinay, K., Rao, A., and Hemantha Kumar, G. (2011). Computerized analysis of classification of lung nodules and comparison between homogeneous and heterogeneous ensemble of classifier model. *Proceedings of the 3rd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, page 231–234.

Vincenzi, S., Zucchetta, M., Franzoi, P., Pellizzato, M., Pranovi, F., De-Leo, G. A., and Torricelli, P. (2011). Application of a random forest algorithm to predict spatial distribution of the potential yield of *ruditapes philippinarum* in the venice lagoon, Italy. *Ecological Modelling*, 222(8):1471–1478.

Vluymans, S., Triguero, I., Cornelis, C., and Saeys, Y. (2016). EPRENNID: An evolutionary prototype reduction based ensemble for nearest neighbor classification of imbalanced data. *Neurocomputing*, 216(1):596–610.

Wan, S. and Yang, H. (2013). Comparison among methods of ensemble learning. *Proceedings of the International Symposium on Biometrics and Security Technologies*, page 286–290.

Wang, G., Zhang, Z., Sun, J., Yang, S., and Larson, C. A. (2015). POS-RS: A random subspace method for sentiment classification based on part-of-speech analysis. *Information Processing and Management*, 51(4):458–479.

Wang, H., Bah, M. J., and Hammad, M. (2019a). Progress in outlier detection techniques: A survey. *IEEE Access*, 7(1):107964–108000.

Wang, Q. and Zhang, L. (2010). Ensemble learning based on multi-task class labels. *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, page 464–475.

Wang, S. J., Mathew, A., Chen, Y., Xi, L. F., Ma, L., and Lee, J. (2009). Empirical analysis of support vector machine ensemble classifiers. *Expert Systems with applications*, 36(3):6466–6476.

Wang, Z., Koprinska, I., Troncoso, A., and Martínez-Álvarez, F. (2018). Static and dynamic ensembles of neural networks for solar power forecasting. *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8.

Wang, Z. M., Song, G. H., and Gao, C. (2019b). An isolation-based distributed outlier detection framework using nearest neighbor ensembles for wireless sensor networks. *IEEE Access*, 7(1):96319–96333.

Webb, G. I. and Zheng, Z. (2004). Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Transactions on Knowledge and Data Engineering*, 16(8):980–991.

Wichard, J., Merkwirth, C., and Ogorzaek, M. (2002). Building ensembles with heterogeneous models. *Proceedings of the 7th Course of the International School on Neural Nets*, pages 1–8.

Wilson, M. D. (2008). Support vector machines. In Jorgensen, S. E., F. B. D., editor, *Encyclopedia of Ecology*, page 3431–3437. Academic Press, Oxford.

Windeatt, T. and Ghaderi, R. (2003). Coding and decoding strategies for multi-class learning problems. *Information Fusion*, 4(1):11–21.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2):241–259.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390.

Wu, G. Chang, E. (2003). Class-boundary alignment for imbalanced dataset learning. *Proceedings of the International Conference on Machine Learning: Workshop on Learning from Imbalanced Data Sets II*, pages 1–12.

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., and Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40.

Xie, F., Fan, H., Li, Y., Jiang, Z., Meng, R., and Bovik, A. (2017). Melanoma classification on dermoscopy images using a neural network ensemble model. *IEEE Transactions On Medical Imaging*, 36(3):849–858.

Xing, J., Luo, K., Wang, H., and Fan, J. (2019). Estimating biomass major chemical constituents from ultimate analysis using a random forest model. *Bioresource Technology*, 288(1):1–7.

Xu, J., Chen, J., and Li, B. (2009). Random forest for relational classification with application to terrorist profiling. *Proceedings of the IEEE International Conference on Granular Computing*, page 630–633.

Xu, J. and Zhang, J. (2019). An heterogeneous ensemble learning based method for ECG classification. *Proceedings of IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference*, page 439–443.

Xue, D. and Li, F. (2015). Research of text categorization model based on random forests. *Proceedings of the IEEE International Conference on Computational Intelligence and Communication Technology*, page 173–176.

Yan, T. T., Zhang, Y. P., Zhang, y. W., and Du, X. Q. (2016). A selective neural network ensemble classification for incomplete data. *International Journal of Machine Learning and Cybernetics*, 8(5):1513–1524.

Yang, J., Rahardja, S., and Fränti, P. (2021). Mean-shift outlier detection and filtering. *Pattern Recognition*, 115(1):161–171.

Yang, J., Zeng, X., Zhong, S., and Wu, S. (2013). Effective neural network ensemble approach for improving generalization performance. *IEEE Transactions on Neural Networks and Learning Systems*, 24(6):878–887.

Yao, D., Zhan, X., and Kwoh, C. K. (2019). An improved random forest-based computational model for predicting novel miRNA-disease associations. *BMC Bioinformatics*, 20(1):1–14.

Yeh, I. C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808.

Yeh, I. C. and Hsu, T. K. (2018). Building real estate valuation models with comparative approach through case-based reasoning. *Applied Soft Computing*, 65(1):260–271.

Yu, Z., Chen, H., Liuxs, J., You, J., Leung, H., and Han, G. (2016). Hybrid *k*-nearest neighbor classifier. *IEEE Transactions on Cybernetics*, 46(6):1263–1275.

Zaamout, K. and Zhang, J. Z. (2012). Improving classification through ensemble neural networks. *Proceedings of the International Conference on Natural Computation*, page 256–260.

Zain, Z. M., Yusuf, Z. M., Muthusamy, H., Kader, K. A., and Mortar, N. A. M. (2020). Heterogeneous ensemble classifiers for malay syllables classification. *Proceedings of the American Institute of Physics*, pages 1–7.

Zefrehi, H. G. and Hakan, H. A. (2020). Imbalance learning using heterogeneous ensembles. *Expert Systems with Applications*, 142(1):1–15.

Zeng, Y. and Cheng, F. (2021). Medical and health data classification method based on machine learning. *Journal of Healthcare Engineering*, 5(1):1–5.

Zhang, H., Niu, H., Ma, Z., and Zhang, S. (2022). Wind turbine condition monitoring based on bagging ensemble strategy and *k*nn algorithm. *IEEE Access*, 4(1):1–9.

Zhang, H. and Zimba, P. V. (2017). Analyzing the effects of estuarine freshwater fluxes on fish abundance using artificial neural network ensembles. *Ecological Modelling*, 359(1):103–116.

Zhang, L., Ren, Y., and Suganthan, P. N. (2013). Instance based random forest with rotated feature space. *Proceedings of the IEEE Symposium on Computational Intelligence and Ensemble Learning*, pages 31–35.

Zhang, Y. and Street, W. N. (2008). Bagging with adaptive costs. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):577–588.

Zhang, Z., Jiang, F., Li, B., and Zhang, B. (2018). A novel time difference of arrival localization algorithm using a neural network ensemble model. *International Journal of Distributed Sensor Networks*, 14(11):1–12.

Zhao, C., Xin, Y., Li, X., Yang, Y., and Chen, Y. (2020). A heterogeneous ensemble learning framework for spam detection in social networks with imbalanced data. *Applied Sciences*, 10(3):1–17.

Zhao, Z., Feng, X., Lin, Y., Wei, F., Wang, S., Xiao, T., Cao, M., and Hou, Z. (2015). Neurocomputing evolved neural network ensemble by multiple heterogeneous swarm intelligence. *Neurocomputing*, 149(A):29–38.

Zhou, Z. H. (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, New York, first edition.

Zupan, B., Bohanec, M., Bratko, I., and Demsar, J. (1997). Machine learning by function decomposition. *Proceedings of the International Conference on Machine Learning*, page 421–429.

Appendix A

Ensemble Performance on Skewed Class Distributions for Classification Problems

The results of the ensembles over the different datasets in the skewed class distribution study for classification problems are provided in this appendix. The results consist of the training and testing accuracy, GF, F1-score, and confusion matrices of the ensembles over the classification datasets.

Sonar Dataset

Table A.1: Confusion Matrices of Ensembles on Skewed Class Distributions for Sonar Dataset

10-90%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.04	0.96	0.15	0.85	0.23	0.77	0.12	0.88	0.27	0.73	0.58	0.42	0	1.00
1	0	**1.00**	0.06	0.94	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.35	0.65	0.12	0.88	0	1.00	0	1.00	0.42	0.58	0.42	0.58
1	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

15-85%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.23	0.77	0.35	0.65	0.57	0.43	0.42	0.58	0.27	0.73	0.69	0.31	0.15	0.85
1	0	**1.00**	0.06	0.94	0.06	0.94	0.06	0.84	0	**1.00**	0	**1.00**	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.38	0.62	0.50	0.50	0.12	0.88	0.46	0.54	0.42	0.58	0.27	0.73
1	0	**1.00**	0.06	0.94	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

20-80%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.23	0.77	0.38	0.62	0.54	0.46	0.35	0.65	0.31	0.69	0.65	0.35	0.23	0.77
1	0	**1.00**	0.06	0.94	0.19	0.81	0.06	0.94	0	**1.00**	0	**1.00**	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.46	0.54	0.39	0.61	0.38	0.62	0.54	0.46	0.58	0.42	0.58	0.42
1	0.06	0.94	0.06	0.94	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

25-75%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.50	0.50	0.50	0.50	0.42	0.58	0.65	0.35	0.39	0.61	0.77	0.23	0.42	0.58
1	0	**1.00**	0.12	0.88	0.06	0.94	0	**1.00**	0	**1.00**	0.12	0.98	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.46	0.54	0.50	0.50	0.31	0.69	0.27	0.73	0.50	0.50	0.42	0.58
1	0.12	0.88	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

30-70%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.58	0.42	0.62	0.38	0.62	0.38	0.58	0.42	0.38	0.61	0.62	0.38	0.5	0.50
1	0.06	0.94	0.12	0.88	0.06	0.94	0.06	0.94	0	**1.00**	0	**1.00**	0.06	0.94

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.62	0.38	0.58	0.42	0.38	0.62	0.50	0.50	0.62	0.38	0.62	0.38
1	0.12	0.88	0.06	0.94	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

35-65%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.73	0.27	0.62	0.38	0.70	0.30	0.46	0.54	0.19	0.81	0.73	0.27	0.73	0.27
1	0.12	0.80	0.19	0.81	0	**1.00**	0.06	0.94	0	**1.00**	0.12	0.88	0.12	0.88

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.77	0.23	0.70	0.30	0.42	0.58	0.58	0.42	0.70	0.30	0.62	0.38
1	0.18	0.82	0	**1.00**	0	**1.00**	0.06	0.94	0	**1.00**	0	**1.00**

40-60%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.54	0.46	0.62	0.38	0.58	0.42	0.65	0.35	0.54	0.46	0.73	0.27	0.46	0.54
1	0.06	0.94	0.31	0.69	0.31	0.69	0.06	0.94	0.06	0.94	0.12	0.88	0.06	0.94

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.69	0.31	0.62	0.38	0.42	0.58	0.81	0.19	0.88	0.12	0.77	0.23
1	0.19	0.81	0.19	0.81	0	**1.00**	0.12	0.88	0	**1.00**	0	**1.00**

45-55%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.70	0.30	0.73	0.27	0.73	0.27	0.73	0.27	0.58	0.42	0.85	0.15	0.65	0.35
1	0.12	0.88	0.31	0.69	0.25	0.75	0.19	0.81	0.06	0.94	0.19	0.81	0.06	0.94

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.77	0.23	0.69	0.31	0.46	0.57	0.81	0.19	0.77	0.23	0.85	0.15
1	0.38	0.62	0.19	0.81	0	**1.00**	0.19	0.81	0	**1.00**	0	**1.00**

50-50%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.73	0.27	0.69	0.31	0.69	0.31	0.81	0.19	0.62	0.38	0.81	0.19	0.73	0.27
1	0.12	0.88	0.38	0.62	0.44	0.56	0.31	0.69	0.06	0.94	0.12	0.88	0.12	0.88

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.81	0.19	0.62	0.38	0.54	0.46	0.69	0.31	0.88	0.12	0.85	0.15
1	0.31	0.69	0.38	0.62	0.19	0.81	0.06	0.94	0.04	**0.96**	0.04	**0.96**

Table A.2: Ensemble Performance on Skewed Class Distributions for Sonar Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.747	0.747	0.747	0.747	0.747	0.747	0.747	0.747	0.747
	Training accuracy	0.957	0.873	0.854	0.749	0.713	0.672	0.669	0.672	0.640
	GF	5.884	1.992	1.732	1.007	0.882	0.764	0.717	0.771	0.702
	F1-Score	0.26	0.47	0.47	0.68	0.71	0.79	0.69	0.77	0.79
*k*NNE	Testing accuracy	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.586
	Training accuracy	0.898	0.827	0.734	0.667	0.647	0.587	0.590	0.576	0.526
	GF	4.059	2.393	1.556	1.243	1.173	1.002	1.009	0.978	0.873
	F1-Score	0.38	0.55	0.58	0.64	0.72	0.69	0.65	0.72	0.67
DTE	Testing accuracy	0.615	0.627	0.634	0.644	0.639	0.619	0.628	0.624	0.636
	Training accuracy	0.779	0.773	0.741	0.748	0.668	0.642	0.613	0.597	0.599
	GF	1.742	1.643	1.413	1.412	1.087	1.064	0.961	0.933	0.907
	F1-Score	0.47	0.71	0.64	0.61	0.74	0.81	0.62	0.74	0.64
RF	Testing accuracy	0.698	0.701	0.731	0.721	0.735	0.688	0.697	0.677	0.730
	Training accuracy	0.895	0.817	0.801	0.775	0.719	0.708	0.673	0.653	0.655
	GF	2.876	1.633	1.352	1.240	0.943	1.068	0.926	0.931	0.783
	F1-Score	0.35	0.61	0.55	0.79	0.71	0.63	0.76	0.77	0.76
SVME	Testing accuracy	0.714	0.724	0.710	0.724	0.744	0.724	0.726	0.749	0.739
	Training accuracy	0.902	0.850	0.809	0.779	0.708	0.685	0.649	0.668	0.642
	GF	2.918	1.840	1.393	1.248	0.876	0.876	0.780	0.756	0.729
	F1-Score	0.50	0.50	0.54	0.60	0.60	0.43	0.69	0.71	0.74
NNE	Testing accuracy	0.759	0.768	0.763	0.769	0.758	0.767	0.768	0.768	0.768
	Training accuracy	0.839	0.769	0.787	0.718	0.705	0.7695	0.690	0.661	0.634
	GF	1.497	1.004	1.112	0.819	0.820	0.764	0.748	0.684	0.634
	F1-Score	0.74	0.81	0.79	0.81	0.76	0.79	0.79	0.83	0.84

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.762	0.762	0.762	0.762	0.762	0.762	0.762	0.762	0.762
	Training accuracy	0.957	0.863	0.885	0.736	0.708	0.692	0.674	0.683	0.647
	GF	5.534	1.737	2.069	0.901	0.815	0.773	0.730	0.750	0.674
	F1-Score	0.21	0.39	0.47	0.63	0.66	0.79	0.63	0.76	0.79
*k*NNhte	Testing accuracy	0.655	0.655	0.655	0.655	0.655	0.655	0.655	0.655	0.655
	Training accuracy	0.857	0.808	0.705	0.718	0.596	0.574	0.570	0.542	0.530
	GF	2.413	1.796	1.169	1.223	0.854	0.809	0.802	0.753	0.734
	F1-Score	0.57	0.60	0.63	0.61	0.72	0.79	0.74	0.71	0.76
DThte	Testing accuracy	0.667	0.692	0.692	0.637	0.687	0.647	0.666	0.671	0.678
	Training accuracy	0.899	0.807	0.818	0.777	0.710	0.709	0.657	0.681	0.637
	GF	3.297	1.595	1.692	1.627	1.079	1.213	1.035	0.973	0.889
	F1-Score	0.35	0.66	0.58	0.68	0.71	0.81	0.69	0.74	0.62
SVMhte	Testing accuracy	0.714	0.710	0.710	0.699	0.678	0.709	0.687	0.676	0.695
	Training accuracy	0.902	0.850	0.837	0.773	0.722	0.676	0.643	0.607	0.628
	GF	2.918	1.933	1.779	1.326	1.158	0.898	0.876	0.824	0.819
	F1-Score	0.21	0.35	0.60	0.54	0.60	0.63	0.63	0.66	0.64
NNhte	Testing accuracy	0.755	0.771	0.764	0.759	0.754	0.764	0.760	0.763	0.753
	Training accuracy	0.905	0.839	0.792	0.763	0.681	0.704	0.652	0.652	0.626
	GF	2.578	1.857	1.134	1.017	0.771	0.797	0.691	0.681	0.660
	F1-Score	0.21	0.66	0.71	0.50	0.68	0.71	0.84	0.81	0.79
HTEsm	Testing accuracy	0.794	0.753	0.794	0.749	0.763	0.769	0.762	0.745	0.792
	Training accuracy	0.889	0.859	0.794	0.790	0.685	0.645	0.646	0.593	0.654
	GF	1.855	1.752	1.00	1.195	0.752	0.651	0.672	0.626	0.601
	F1-Score	0.63	0.63	0.74	0.68	0.76	0.77	0.90	0.81	0.93
HTEdf	Testing accuracy	0.815	0.745	0.777	0.743	0.765	0.793	0.786	0.787	0.763
	Training accuracy	0.882	0.861	0.787	0.806	0.697	0.675	0.660	0.661	0.611
	GF	1.567	1.835	1.046	1.324	0.776	0.636	0.629	0.628	0.611
	F1-Score	0.64	0.65	0.74	0.63	0.77	0.76	0.91	0.88	0.93

Breast Dataset

Table A.3: Ensemble Performance on Skewed Class Distributions in Breast Cancer Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.517	0.517	0.517	0.517	0.517	0.517	0.517	0.517	0.517
	Training accuracy	0.952	0.904	0.880	0.833	0.793	0.747	0.684	0.643	0.601
	GF	10.062	5.031	4.025	2.887	2.333	1.909	1.528	1.352	1.210
	F1-Score	0.43	0.45	0.48	0.40	0.43	0.40	0.43	0.40	0.43
*k*NNE	Testing accuracy	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666
	Training accuracy	0.905	0.834	0.795	0.748	0.720	0.687	0.644	0.623	0.631
	GF	3.516	2.012	1.629	1.325	1.193	1.067	0.938	0.886	0.905
	F1-Score	0.35	0.43	0.34	0.57	0.51	0.47	0.46	0.56	0.51
DTE	Testing accuracy	0.587	0.549	0.580	0.579	0.569	0.559	0.565	0.543	0.565
	Training accuracy	0.976	0.927	0.921	0.837	0.846	0.773	0.721	0.681	0.597
	GF	17.208	6.178	5.337	2.582	2.798	1.942	1.559	1.432	1.079
	F1-Score	0.40	0.47	0.45	0.46	0.51	0.54	0.56	0.66	0.54
RF	Testing accuracy	0.631	0.636	0.625	0.595	0.627	0.640	0.627	0.638	0.605
	Training accuracy	0.971	0.934	0.898	0.854	0.800	0.797	0.708	0.683	0.650
	GF	12.724	5.515	3.676	2.774	1.865	1.773	1.277	1.141	1.128
	F1-Score	0.19	0.41	0.48	0.36	0.50	0.45	0.48	0.67	0.61
SVME	Testing accuracy	0.633	0.630	0.630	0.630	0.627	0.630	0.633	0.630	0.630
	Training accuracy	0.894	0.843	0.775	0.721	0.648	0.627	0.603	0.606	0.661
	GF	3.462	2.357	1.644	1.326	1.059	0.992	0.924	0.939	1.091
	F1-Score	0.26	0.32	0.31	0.29	0.42	0.43	0.58	0.61	0.56
NNE	Testing accuracy	0.560	0.560	0.543	0.554	0.542	0.561	0.543	0.555	0.565
	Training accuracy	0.874	0.833	0.808	0.764	0.764	0.687	0.676	0.649	0.653
	GF	3.492	2.635	2.380	1.889	1.941	1.402	1.410	1.267	1.253
	F1-Score	0.49	0.54	0.53	0.55	0.50	0.54	0.55	0.58	0.50

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.518	0.518	0.518	0.518	0.518	0.518	0.518	0.518	0.518
	Training accuracy	0.959	0.924	0.891	0.854	0.809	0.759	0.691	0.649	0.604
	GF	11.756	6.342	4.422	3.324	2.523	2.000	1.559	1.373	1.217
	F1-Score	0.43	0.43	0.48	0.40	0.48	0.38	0.48	0.42	0.48
*k*NNhte	Testing accuracy	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635
	Training accuracy	0.916	0.847	0.786	0.760	0.723	0.703	0.637	0.630	0.609
	GF	4.345	2.385	1.705	1.521	1.317	1.228	1.008	0.986	0.933
	F1-Score	0.35	0.45	0.39	0.49	0.44	0.54	0.52	0.58	0.56
DThte	Testing accuracy	0.589	0.599	0.590	0.597	0.596	0.593	0.595	0.581	0.582
	Training accuracy	0.977	0.923	0.917	0.835	0.817	0.787	0.721	0.683	0.611
	GF	17.869	5.207	5.00	2.457	2.207	1.911	1.451	1.321	1.075
	F1-Score	0.40	0.43	0.53	0.48	0.48	0.61	0.58	0.64	0.59
SVMhte	Testing accuracy	0.633	0.630	0.637	0.633	0.627	0.633	0.633	0.630	0.630
	Training accuracy	0.905	0.843	0.795	0.743	0.683	0.659	0.572	0.614	0.599
	GF	3.823	2.356	1.770	1.428	1.175	1.076	0.857	0.958	0.922
	F1-Score	0.26	0.36	0.33	0.30	0.35	0.45	0.38	0.56	0.53
NNhte	Testing accuracy	0.570	0.578	0.585	0.575	0.598	0.577	0.555	0.599	0.575
	Training accuracy	0.908	0.844	0.795	0.764	0.747	0.691	0.655	0.644	0.664
	GF	4.674	2.705	2.023	1.801	1.589	1.368	1.289	1.126	1.264
	F1-Score	0.19	0.19	0.55	0.52	0.57	0.50	0.61	0.56	0.53
HTEsm	Testing accuracy	0.643	0.645	0.646	0.647	0.646	0.645	0.643	0.643	0.641
	Training accuracy	0.936	0.887	0.834	0.811	0.763	0.749	0.648	0.659	0.630
	GF	5.578	3.106	3.142	2.145	1.494	1.414	1.014	0.881	0.970
	F1-Score	0.41	0.53	0.53	0.54	0.51	0.61	0.60	0.63	0.58
HTEdf	Testing accuracy	0.645	0.645	0.647	0.647	0.647	0.646	0.645	0.644	0.642
	Training accuracy	0.941	0.903	0.829	0.809	0.765	0.753	0.662	0.663	0.630
	GF	6.017	3.659	2.064	1.848	1.502	1.433	1.050	1.050	0.968
	F1-Score	0.50	0.57	0.55	0.59	0.50	0.61	0.60	0.64	0.59

Table A.4: Confusion Matrices of Ensembles on Skewed Class Distributions for Breast Cancer Dataset

10-90%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.14	0.86	0.19	0.81	0	1.00	0.05	0.95	0.65	0.65	0.22	0.78
1	0.05	0.95	0.05	0.95	0.05	0.95	0.05	0.95	0	**1.00**	0.24	0.76	0.05	0.95

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.16	0.84	0.19	0.81	0.05	0.95	0	1.00	0.19	0.81	0.22	0.78
1	0.14	0.86	0.05	0.95	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

15-85%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.27	0.73	0.22	0.78	0.27	0.73	0.19	0.81	0.11	0.89	0.59	0.41	0.24	0.76
1	0.14	0.86	0.05	0.95	0.10	0.90	0	**1.00**	0.05	0.95	0.57	0.43	0.14	0.86

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.30	0.70	0.22	0.78	0.14	0.86	0	1.00	0.38	0.62	0.38	0.62
1	0.24	0.76	0.05	0.95	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

20-80%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.27	0.73	0.22	0.78	0.30	0.70	0.27	0.73	0.11	0.89	0.54	0.46	0.27	0.73
1	0.05	0.95	0.38	0.62	0.27	0.76	0.05	0.95	0.10	0.90	0.52	0.48	0.05	0.95

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.30	0.70	0.32	0.68	0.11	0.89	0.40	0.60	0.35	0.65	0.30	0.70
1	0.43	0.57	0.05	0.95	0	**1.00**	0.19	0.81	0.05	0.95	0.05	0.95

25-75%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.46	0.54	0.32	0.68	0.19	0.81	0.08	0.92	0.65	0.35	0.22	0.78
1	0.14	0.86	0.24	0.76	0.29	0.71	0.24	0.76	0.05	0.95	0.62	0.38	0.14	0.86

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.46	0.54	0.30	0.70	0.08	0.92	0.62	0.38	0.49	0.51	0.49	0.51
1	0.48	0.52	0.14	0.86	0	**1.00**	0.67	0.33	0.12	0.88	0.11	0.89

30-70%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.43	0.57	0.46	0.54	0.38	0.62	0.24	0.76	0.59	0.41	0.32	0.68
1	0.05	**0.95**	0.38	0.62	0.43	0.57	0.29	0.71	0.19	0.89	0.67	0.33	0.19	0.81

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.41	0.59	0.38	0.62	0.65	0.86	0.65	0.35	0.49	0.51	0.46	0.54
1	0.52	0.48	0.33	0.67	0.05	**0.95**	0.57	0.43	0.25	0.75	0.22	0.78

35-65%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.43	0.57	0.49	0.51	0.38	0.62	0.24	0.76	0.76	0.24	0.22	0.78
1	0.14	0.86	0.48	0.52	0.38	0.62	0.43	0.57	0.14	0.86	0.76	0.24	0.24	0.76

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.57	0.43	0.65	0.35	0.24	0.76	0.62	0.38	0.59	0.41	0.57	0.43
1	0.52	0.48	0.48	0.52	0.05	**0.95**	0.71	0.29	0.22	0.78	0.20	0.80

40-60%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.49	0.51	0.59	0.41	0.49	0.51	0.54	0.45	0.73	0.27	0.30	0.70
1	0.05	**0.95**	0.62	0.38	0.52	0.48	0.57	0.43	0.38	0.62	0.71	0.29	0.14	0.86

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.57	0.43	0.60	0.40	0.16	0.84	0.76	0.24	0.68	0.32	0.68	0.32
1	0.57	0.43	0.48	0.52	0.05	**0.95**	0.62	0.38	0.35	0.65	0.33	0.67

45-55%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.57	0.43	0.73	0.27	0.76	0.24	0.76	0.24	0.78	0.22	0.24	0.76
1	0.14	**0.86**	0.48	0.52	0.48	0.52	0.48	0.52	0.62	0.38	0.71	0.29	0.19	0.81

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.70	0.30	0.73	0.27	0.54	0.46	0.78	0.22	0.62	0.38	0.62	0.38
1	0.62	0.38	0.52	0.48	0.43	0.57	0.76	0.24	0.45	0.55	0.40	0.60

50-50%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.65	0.35	0.57	0.43	0.65	0.35	0.89	0.11	0.68	0.32	0.27	0.73
1	0.05	**0.95**	0.71	0.29	0.52	0.48	0.48	0.52	0.86	0.14	0.76	0.24	0.05	**0.95**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.70	0.30	0.60	0.40	0.89	0.11	0.65	0.35	0.68	0.32	0.60	0.40
1	0.67	0.33	0.43	0.57	0.90	0.10	0.67	0.33	0.50	0.50	0.45	0.55

Indian Liver Dataset

Table A.5: Ensemble Performance on Skewed Class Distributions for Indian Liver Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.665
	Training accuracy	0.908	0.883	0.867	0.847	0.827	0.811	0.788	0.742	0.723
	GF	3.641	2.863	2.518	2.189	1.936	1.772	1.580	1.298	1.209
	F1-Score	0.43	0.45	0.48	0.40	0.43	0.40	0.43	0.40	0.43
*k*NNE	Testing accuracy	0.743	0.743	0.743	0.743	0.743	0.743	0.743	0.743	0.743
	Training accuracy	0.915	0.871	0.850	0.818	0.795	0.752	0.718	0.690	0.675
	GF	3.023	1.992	1.713	1.412	1.254	1.036	0.911	0.829	0.791
	F1-Score	0.35	0.43	0.34	0.57	0.51	0.47	0.46	0.56	0.51
DTE	Testing accuracy	0.740	0.740	0.740	0.738	0.740	0.735	0.742	0.740	0.738
	Training accuracy	0.874	0.808	0.807	0.771	0.764	0.723	0.705	0.665	0.657
	GF	2.063	1.354	1.347	1.144	1.102	0.9576	0.875	0.776	0.764
	F1-Score	0.40	0.47	0.45	0.46	0.51	0.54	0.56	0.66	0.54
RF	Testing accuracy	0.737	0.735	0.726	0.746	0.736	0.747	0.726	0.724	0.751
	Training accuracy	0.912	0.852	0.843	0.811	0.800	0.759	0.727	0.699	0.688
	GF	2.988	1.791	1.745	1.344	1.320	1.049	1.004	0.917	0.798
	F1-Score	0.19	0.41	0.48	0.36	0.50	0.45	0.48	0.67	0.61
SVME	Testing accuracy	0.755	0.761	0.771	0.764	0.765	0.771	0.762	0.767	0.764
	Training accuracy	0.918	0.878	0.877	0.839	0.834	0.803	0.786	0.743	0.731
	GF	3.000	1.959	1.862	1.465	1.416	1.162	1.112	0.907	0.877
	F1-Score	0.26	0.32	0.31	0.29	0.42	0.43	0.58	0.61	0.54
NNE	Testing accuracy	0.730	0.722	0.730	0.724	0.739	0.734	0.749	0.727	0.725
	Training accuracy	0.913	0.871	0.863	0.832	0.816	0.774	0.756	0.702	0.697
	GF	3.103	2.155	1.971	1.643	1.418	1.177	1.028	0.916	0.907
	F1-Score	0.49	0.54	0.53	0.55	0.50	0.54	0.55	0.58	0.50

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.672	0.672	0.672	0.672	0.672	0.672	0.672	0.672	0.672
	Training accuracy	0.912	0.886	0.874	0.845	0.833	0.813	0.791	0.746	0.730
	GF	3.727	2.877	2.603	2.116	1.964	1.754	1.569	1.291	1.215
	F1-Score	0.43	0.43	0.48	0.40	0.48	0.38	0.48	0.42	0.48
*k*NNhte	Testing accuracy	0.717	0.717	0.767	0.717	0.767	0.767	0.717	0.767	0.717
	Training accuracy	0.909	0.857	0.868	0.799	0.821	0.772	0.707	0.695	0.617
	GF	3.109	1.979	1.765	1.407	1.302	1.022	0.966	0.764	0.739
	F1-Score	0.35	0.45	0.39	0.49	0.44	0.54	0.52	0.58	0.56
DThte	Testing accuracy	0.763	0.750	0.767	0.749	0.748	0.760	0.752	0.756	0.756
	Training accuracy	0.912	0.854	0.843	0.807	0.802	0.763	0.750	0.708	0.702
	GF	2.693	1.712	1.484	1.300	1.272	1.013	0.992	0.836	0.819
	F1-Score	0.40	0.43	0.53	0.48	0.48	0.61	0.58	0.64	0.59
SVMhte	Testing accuracy	0.755	0.747	0.745	0.745	0.747	0.750	0.750	0.748	0.748
	Training accuracy	0.919	0.877	0.863	0.826	0.825	0.799	0.797	0.739	0.711
	GF	3.148	2.056	1.861	1.465	1.457	1.244	1.232	0.966	0.872
	F1-Score	0.26	0.36	0.33	0.30	0.35	0.45	0.38	0.56	0.53
NNhte	Testing accuracy	0.764	0.766	0.763	0.755	0.762	0.756	0.762	0.759	0.759
	Training accuracy	0.915	0.880	0.871	0.849	0.841	0.804	0.792	0.742	0.738
	GF	2.776	1.950	1.837	1.623	1.496	1.245	1.144	0.934	0.919
	F1-Score	0.19	0.19	0.55	0.52	0.57	0.50	0.61	0.56	0.53
HTEsm	Testing accuracy	0.751	0.737	0.766	0.744	0.750	0.747	0.744	0.758	0.751
	Training accuracy	0.917	0.875	0.878	0.839	0.836	0.803	0.792	0.732	0.723
	GF	3.000	2.104	1.918	1.590	1.552	1.284	1.231	0.903	0.898
	F1-Score	0.41	0.53	0.53	0.54	0.51	0.62	0.59	0.57	0.55
HTEdf	Testing accuracy	0.764	0.764	0.769	0.758	0.771	0.771	0.771	0.769	0.763
	Training accuracy	0.915	0.877	0.873	0.839	0.839	0.804	0.786	0.733	0.725
	GF	2.776	1.919	1.819	1.503	1.422	1.168	1.070	0.865	0.862
	F1-Score	0.50	0.57	0.56	0.59	0.51	0.58	0.59	0.59	0.57

Table A.6: Confusion Matrices of Ensembles on Skewed Class Distributions for Indian Liver Dataset

10-90%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.41	0.59	0.13	0.87	0.26	0.74	0.33	0.67	0.22	0.78	0.33	0.57	0.39	0.61
1	0	**1.00**	0	**1.00**	0.03	0.97	0.03	0.97	0	**1.00**	0.03	0.97	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.16	0.84	0.25	0.75	0.20	0.80	0.15	0.85	0.33	0.67	0.22	0.78
1	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

15-85%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.40	0.60	0.20	0.80	0.37	0.63	0.38	0.62	0.20	0.80	0.41	0.57	0.37	0.63
1	0	**1.00**	0	**1.00**	0.07	0.93	0.10	0.90	0	**1.00**	0.10	0.90	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.20	0.20	0.36	0.64	0.22	0.78	0.31	0.69	0.36	0.64	0.26	0.74
1	0	**1.00**	0.10	0.90	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

20-80%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.41	0.59	0.20	0.80	0.36	0.64	0.39	0.61	0.31	0.69	0.47	0.53	0.39	0.61
1	0	**1.00**	0.03	0.97	0.23	0.77	0.10	0.90	0	**1.00**	0.07	0.93	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.38	0.62	0.29	0.71	0.33	0.67	0.33	0.67	0.32	0.68
1	0	**1.00**	0.03	0.97	0	**1.00**	0	**1.00**	0	**1.00**	0	**1.00**

25-75%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.39	0.61	0.24	0.76	0.39	0.61	0.40	0.60	0.29	0.71	0.46	0.54	0.37	0.63
1	0	**1.00**	0	**1.00**	0.17	0.83	0.10	0.90	0	**1.00**	0.10	0.90	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.22	0.78	0.44	0.56	0.25	0.75	0.41	0.59	0.40	0.60	0.35	0.65
1	0	**1.00**	0.07	0.93	0	**1.00**	0.03	0.97	0	**1.00**	0	**1.00**

30-70%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.41	0.59	0.30	0.70	0.43	0.57	0.48	0.52	0.43	0.57	0.49	0.51	0.40	0.60
1	0	**1.00**	0.03	0.97	0.07	0.93	0.10	0.90	0.07	0.93	0.07	0.93	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.34	0.66	0.47	0.53	0.43	0.57	0.48	0.52	0.45	0.55	0.43	0.57
1	0.03	0.97	0.07	0.93	0	**1.00**	0.07	0.93	0.03	0.97	0.03	0.97

35-65%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.40	0.60	0.32	0.68	0.48	0.52	0.56	0.44	0.44	0.56	0.49	0.51	0.40	0.60
1	0	**1.00**	0.03	0.97	0.30	0.70	0.13	0.87	0.07	0.93	0.10	0.90	0	**1.00**

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.31	0.69	0.49	0.51	0.39	0.61	0.46	0.54	0.44	0.56	0.46	0.54
1	0.07	0.93	0.17	0.83	0	**1.00**	0.07	0.93	0.03	0.97	0.03	0.97

40-60%

	NBE 0	NBE 1	*k*NNE 0	*k*NNE 1	DTE 0	DTE 1	RF 0	RF 1	SVME 0	SVME 1	NNE 0	NNE 1	NBhte 0	NBhte 1
0	0.41	0.59	0.40	0.60	0.56	0.44	0.52	0.48	0.52	0.48	0.52	0.48	0.40	0.60
1	0.03	0.97	0.10	0.90	0.20	0.80	0.10	0.90	0.06	0.93	0.13	0.87	0	**1.00**

	*k*NNhte 0	*k*NNhte 1	DThte 0	DThte 1	SVMhte 0	SVMhte 1	NNhte 0	NNhte 1	HTEsm 0	HTEsm 1	HTEdf 0	HTEdf 1
0	0.43	0.57	0.54	0.46	0.48	0.52	0.48	0.52	0.45	0.55	0.45	0.55
1	0.10	0.90	0.03	0.97	0.06	0.93	0.10	0.90	0.03	0.97	0.02	0.98

45-55%

	NBE 0	NBE 1	*k*NNE 0	*k*NNE 1	DTE 0	DTE 1	RF 0	RF 1	SVME 0	SVME 1	NNE 0	NNE 1	NBhte 0	NBhte 1
0	0.40	0.60	0.41	0.59	0.60	0.40	0.62	0.38	0.54	0.46	0.60	0.40	0.40	0.60
1	0	**1.00**	0.17	0.83	0.30	0.70	0.33	0.67	0.06	0.93	0.20	0.80	0	**1.00**

	*k*NNhte 0	*k*NNhte 1	DThte 0	DThte 1	SVMhte 0	SVMhte 1	NNhte 0	NNhte 1	HTEsm 0	HTEsm 1	HTEdf 0	HTEdf 1
0	0.40	0.60	0.57	0.43	0.48	0.52	0.52	0.48	0.46	0.54	0.44	0.56
1	0.17	0.83	0.13	0.87	0.07	0.93	0.06	0.93	0.05	0.95	0.03	0.97

50-50%

	NBE 0	NBE 1	*k*NNE 0	*k*NNE 1	DTE 0	DTE 1	RF 0	RF 1	SVME 0	SVME 1	NNE 0	NNE 1	NBhte 0	NBhte 1
0	0.41	0.59	0.46	0.54	0.67	0.33	0.75	0.25	0.59	0.41	0.71	0.29	0.40	0.60
1	0.03	**0.97**	0.23	0.77	0.43	0.57	0.43	0.57	0.06	0.93	0.30	0.70	0.03	**0.97**

	*k*NNhte 0	*k*NNhte 1	DThte 0	DThte 1	SVMhte 0	SVMhte 1	NNhte 0	NNhte 1	HTEsm 0	HTEsm 1	HTEdf 0	HTEdf 1
	0.56	0.44	0.67	0.33	0.65	0.35	0.53	0.47	0.54	0.46	0.54	0.46
	0.23	0.77	0.43	0.57	0.17	0.83	0.10	0.90	0.05	0.95	0.03	**0.97**

Credit Approval Dataset

Table A.7: Ensemble Performance on Skewed Class Distributions for Credit Approval Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535
	Training accuracy	0.559	0.494	0.507	0.505	0.508	0.530	0.549	0.572	0.606
	GF	1.054	0.919	0.943	0.939	0.945	0.989	1.031	1.086	1.180
	F1-Score	0.54	0.50	0.56	0.48	0.48	0.49	0.46	0.47	0.46
*k*NNE	Testing accuracy	0.788	0.788	0.788	0.788	0.788	0.788	0.788	0.788	0.788
	Training accuracy	0.894	0.863	0.833	0.827	0.824	0.805	0.808	0.804	0.805
	GF	2.000	1.547	1.269	1.225	1.205	1.087	1.104	1.082	1.087
	F1-Score	0.81	0.85	0.83	0.83	0.82	0.84	0.85	0.84	0.83
DTE	Testing accuracy	0.765	0.757	0.755	0.755	0.758	0.761	0.763	0.761	0.754
	Training accuracy	0.948	0.917	0.914	0.870	0.859	0.815	0.793	0.783	0.799
	GF	4.519	2.928	2.849	1.885	1.716	1.292	1.145	1.101	1.224
	F1-Score	0.59	0.63	0.67	0.71	0.69	0.79	0.78	0.78	0.82
RF	Testing accuracy	0.783	0.802	0.784	0.780	0.807	0.786	0.795	0.788	0.799
	Training accuracy	0.945	0.928	0.902	0.889	0.857	0.822	0.831	0.799	0.815
	GF	3.945	2.750	2.204	1.982	1.350	1.202	1.213	1.055	1.086
	F1-Score	0.36	0.40	0.50	0.63	0.66	0.77	0.77	0.83	0.81
SVME	Testing accuracy	0.780	0.774	0.778	0.774	0.781	0.775	0.777	0.774	0.772
	Training accuracy	0.934	0.895	0.865	0.831	0.796	0.766	0.754	0.745	0.742
	GF	3.333	2.152	1.6444	1.337	1.074	0.962	0.906	1.124	0.884
	F1-Score	0.45	0.52	0.62	0.66	0.67	0.77	0.80	0.82	0.86
NNE	Testing accuracy	0.670	0.675	0.681	0.671	0.675	0.675	0.669	0.674	0.674
	Training accuracy	0.901	0.834	0.830	0.816	0.765	0.765	0.732	0.740	0.739
	GF	3.333	1.957	1.876	1.788	1.383	1.383	1.235	1.254	1.249
	F1-Score	0.67	0.70	0.69	0.69	0.67	0.68	0.67	0.67	0.62

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651
	Training accuracy	0.643	0.574	0.571	0.568	0.591	0.598	0.641	0.672	0.673
	GF	0.978	0.819	0.814	0.808	0.853	0.868	0.972	1.064	1.067
	F1-Score	0.55	0.53	0.54	0.53	0.51	0.54	0.55	0.61	0.63
*k*NNhte	Testing accuracy	0.808	0.808	0.808	0.808	0.808	0.808	0.808	0.808	0.808
	Training accuracy	0.867	0.842	0.836	0.824	0.808	0.799	0.804	0.799	0.801
	GF	1.444	1.215	1.170	1.091	1.000	0.955	0.979	0.955	0.964
	F1-Score	0.78	0.81	0.82	0.83	0.83	0.86	0.86	0.84	0.86
DThte	Testing accuracy	0.767	0.765	0.758	0.759	0.759	0.764	0.765	0.763	0.759
	Training accuracy	0.954	0.941	0.916	0.876	0.862	0.833	0.814	0.804	0.808
	GF	5.065	3.983	2.881	1.944	1.746	1.413	1.263	1.209	1.255
	F1-Score	0.60	0.62	0.67	0.68	0.70	0.77	0.81	0.83	0.83
SVMhte	Testing accuracy	0.822	0.818	0.818	0.815	0.815	0.818	0.821	0.819	0.818
	Training accuracy	0.912	0.881	0.864	0.850	0.831	0.797	0.810	0.804	0.815
	GF	2.023	1.529	1.338	1.233	1.095	0.897	0.942	0.923	0.984
	F1-Score	0.50	0.73	0.81	0.84	0.86	0.84	0.83	0.83	0.83
NNhte	Testing accuracy	0.765	0.755	0.770	0.769	0.769	0.769	0.768	0.760	0.766
	Training accuracy	0.920	0.879	0.876	0.849	0.831	0.813	0.788	0.795	0.793
	GF	2.938	2.024	1.855	1.529	1.367	1.235	1.094	1.170	1.130
	F1-Score	0.66	0.71	0.73	0.72	0.71	0.75	0.73	0.73	0.74
HTEsm	Testing accuracy	0.810	0.809	0.811	0.809	0.811	0.814	0.812	0.810	0.812
	Training accuracy	0.933	0.899	0.874	0.850	0.841	0.829	0.799	0.807	0.812
	GF	2.836	1.891	1.500	1.273	1.201	1.088	0.935	0.984	1.000
	F1-Score	0.75	0.78	0.78	0.78	0.80	0.80	0.83	0.85	0.85
HTEdf	Testing accuracy	0.817	0.815	0.813	0.810	0.813	0.816	0.818	0.815	0.812
	Training accuracy	0.957	0.920	0.895	0.866	0.857	0.834	0.818	0.829	0.831
	GF	4.256	2.313	1.781	1.418	1.308	1.108	1.000	1.082	1.112
	F1-Score	0.77	0.79	0.79	0.80	0.82	0.84	0.86	0.86	0.87

Table A.8: Confusion Matrices of Ensembles on Skewed Class Distributions for Credit Approval Dataset

10-90%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.56	0.44	**0.74**	0.26	0.31	0.69	0.03	0.97	0.13	0.87	0.49	0.51	0.50	0.50
1	0.49	0.51	0.12	0.88	0.04	0.96	0	1.00	0.01	0.99	0.12	0.88	0.40	0.60

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.63	0.37	0.33	0.67	0.19	0.81	0.44	0.56	0.53	0.47	0.57	0.43
1	0.06	0.94	0.04	0.96	0.01	0.99	0.09	0.91	0.07	0.93	0.06	0.94

15-85%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.56	0.44	**0.79**	0.21	0.39	0.61	0.07	0.93	0.21	0.79	0.56	0.44	0.54	0.46
1	0.56	0.44	0.09	0.91	0.06	0.94	0	1.00	0.01	0.99	0.15	0.85	0.49	0.51

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.71	0.29	0.36	0.64	0.51	0.49	0.53	0.47	0.57	0.43	0.59	0.41
1	0.09	0.91	0.04	0.96	0.01	0.99	0.09	0.91	0.12	0.88	0.06	0.94

20-80%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.60	0.40	**0.80**	0.20	0.41	0.59	0.19	0.81	0.36	0.64	0.57	0.43	0.56	0.44
1	0.49	0.51	0.13	0.87	0.01	0.99	0.03	0.97	0.03	0.97	0.18	0.82	0.47	0.53

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.74	0.26	0.41	0.59	0.66	0.34	0.60	0.40	0.57	0.43	0.62	0.38
1	0.10	0.90	0.01	0.99	0.03	0.97	0.12	0.88	0.03	0.97	0	1.00

25-75%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.59	0.41	**0.80**	0.20	0.53	0.47	0.40	0.60	0.41	0.59	0.61	0.39	0.57	0.43
1	0.62	0.38	0.13	0.87	0.07	0.97	0.07	0.93	0.03	0.97	0.22	0.78	0.51	0.49

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.79	0.21	0.47	0.53	0.74	0.26	0.63	0.37	0.61	0.39	0.65	0.35
1	0.12	0.88	0.07	0.93	0.06	0.94	0.18	0.82	0.05	0.95	0.03	0.97

30-70%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.59	0.41	**0.80**	0.20	0.50	0.50	0.41	0.59	0.44	0.56	0.57	0.43	0.57	0.43
1	0.62	0.38	0.16	0.84	0.10	0.90	0.04	0.96	0.04	0.96	0.22	0.78	0.54	0.46

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.79	0.21	0.50	0.50	0.78	0.21	0.61	0.39	0.63	0.37	0.67	0.33
1	0.12	0.88	0.07	0.93	0.06	0.94	0.20	0.80	0.10	0.90	0.05	0.95

35-65%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.60	0.40	0.80	0.20	0.66	0.34	0.67	0.33	0.61	0.39	0.61	0.39	0.59	0.41
1	0.62	0.38	0.12	0.88	0.07	0.93	0.12	0.88	0.06	0.94	0.25	0.75	0.50	0.50

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.83	0.17	0.63	0.37	**0.84**	0.16	0.70	0.30	0.67	0.33	0.75	0.25
1	0.10	0.90	0.07	0.93	0.16	0.84	0.19	0.81	0.12	0.88	0.05	0.95

40-60%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.59	0.41	0.85	0.15	0.73	0.27	0.67	0.33	0.66	0.31	0.63	0.37	0.59	0.41
1	0.66	0.34	0.15	0.85	0.16	0.84	0.12	0.88	0.07	0.93	0.29	0.71	0.49	0.51

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.83	0.17	0.74	0.26	**0.87**	0.13	0.63	0.37	0.75	0.25	0.77	0.23
1	0.13	0.87	0.12	0.88	0.20	0.80	0.16	0.84	0.12	0.88	0.10	0.90

45-55%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.57	0.43	0.84	0.16	0.67	0.33	0.81	0.19	0.71	0.29	0.63	0.37	0.54	0.46
1	0.63	0.37	0.13	0.87	0.10	0.90	0.16	0.84	0.07	0.93	0.29	0.71	0.32	0.68

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.83	0.17	0.80	0.20	**0.87**	0.13	0.69	0.31	0.82	0.18	0.85	0.15
1	0.15	0.85	0.13	0.87	0.21	0.79	0.22	0.78	0.15	0.85	0.10	0.90

50-50%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.63	0.37	0.81	0.19	0.76	0.24	0.76	0.24	0.81	0.19	0.60	0.40	0.60	0.40
1	0.69	0.31	0.15	0.85	0.12	0.88	0.13	0.87	0.10	0.90	0.35	0.65	0.34	0.66

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.85	0.15	0.81	0.19	0.87	0.13	0.74	0.26	0.86	0.14	**0.89**	0.11
1	0.13	0.87	0.16	0.84	0.22	0.78	0.26	0.74	0.17	0.83	0.12	0.88

Red Wine Dataset

Table A.9: Ensemble Performance on Skewed Class Distributions for Red Wine Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.471	0.471	0.471	0.471	0.471	0.471	0.471	0.471	0.471
	Training accuracy	0.430	0.443	0.457	0.490	0.515	0.535	0.555	0.586	0.610
	GF	0.928	0.949	0.974	1.037	1.091	1.138	1.188	1.277	1.356
	F1-Score	0.26	0.25	0.25	0.23	0.24	0.25	0.25	0.22	0.23
*k*NNE	Testing accuracy	0.501	0.501	0.501	0.501	0.501	0.501	0.501	0.501	0.501
	Training accuracy	0.548	0.546	0.574	0.567	0.610	0.629	0.636	0.672	0.686
	GF	1.104	1.099	1.171	1.152	1.279	1.345	1.371	1.521	1.589
	F1-Score	0.33	0.38	0.38	0.38	0.36	0.33	0.31	0.36	0.34
DTE	Testing accuracy	0.473	0.493	0.483	0.482	0.488	0.486	0.493	0.485	0.488
	Training accuracy	0.533	0.537	0.567	0.556	0.591	0.626	0.653	0.687	0.685
	GF	1.128	1.095	1.194	1.167	1.252	1.374	1.461	1.645	1.625
	F1-Score	0.46	0.41	0.41	0.41	0.38	0.38	0.38	0.38	0.38
RF	Testing accuracy	0.534	0.546	0.529	0.537	0.524	0.521	0.545	0.559	0.541
	Training accuracy	0.601	0.590	0.615	0.636	0.655	0.661	0.687	0.702	0.719
	GF	1.168	1.107	1.223	1.272	1.379	1.413	1.454	1.479	1.633
	F1-Score	0.44	0.43	0.43	0.45	0.41	0.39	0.46	0.39	0.40
SVME	Testing accuracy	0.531	0.525	0.530	0.531	0.528	0.536	0.535	0.534	0.528
	Training accuracy	0.475	0.475	0.511	0.523	0.553	0.593	0.610	0.675	0.693
	GF	0.893	0.905	0.961	0.983	1.055	1.140	1.192	1.434	1.537
	F1-Score	0.40	0.41	0.41	0.40	0.38	0.40	0.37	0.37	0.39
NNE	Testing accuracy	0.557	0.558	0.557	0.556	0.553	0.548	0.555	0.558	0.556
	Training accuracy	0.627	0.647	0.650	0.664	0.680	0.689	0.709	0.722	0.733
	GF	1.188	1.252	1.265	1.321	1.396	1.453	1.529	1.589	1.663
	F1-Score	0.49	0.51	0.51	0.48	0.45	0.45	0.45	0.45	0.45

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489	0.489
	Training accuracy	0.432	0.443	0.456	0.487	0.516	0.532	0.564	0.600	0.625
	GF	0.899	0.917	0.939	0.996	1.056	1.092	1.172	1.278	1.363
	F1-Score	0.26	0.25	0.25	0.22	0.24	0.24	0.23	0.22	0.21
*k*NNhte	Testing accuracy	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
	Training accuracy	0.547	0.553	0.560	0.567	0.607	0.620	0.637	0.662	0.683
	GF	1.071	1.085	1.102	1.120	1.234	1.276	1.336	1.435	1.529
	F1-Score	0.37	0.36	0.36	0.38	0.39	0.38	0.39	0.38	0.41
DThte	Testing accuracy	0.541	0.526	0.519	0.524	0.537	0.533	0.536	0.531	0.526
	Training accuracy	0.580	0.589	0.621	0.640	0.644	0.661	0.687	0.715	0.722
	GF	1.093	1.153	1.269	1.322	1.301	1.377	1.482	1.646	1.705
	F1-Score	0.49	0.46	0.46	0.46	0.41	0.42	0.41	0.42	0.40
SVMhte	Testing accuracy	0.551	0.549	0.549	0.548	0.548	0.551	0.549	0.549	0.549
	Training accuracy	0.512	0.544	0.569	0.594	0.622	0.642	0.644	0.692	0.708
	GF	0.920	0.989	1.046	1.113	1.195	1.254	1.307	1.464	1.544
	F1-Score	0.37	0.39	0.39	0.30	0.35	0.30	0.30	0.28	0.33
NNhte	Testing accuracy	0.556	0.556	0.550	0.556	0.561	0.548	0.551	0.554	0.553
	Training accuracy	0.614	0.635	0.629	0.657	0.659	0.672	0.697	0.701	0.717
	GF	1.150	1.216	1.213	1.294	1.287	1.378	1.482	1.492	1.579
	F1-Score	0.51	0.46	0.46	0.45	0.39	0.41	0.43	0.46	0.42
HTEsm	Testing accuracy	0.549	0.549	0.549	0.546	0.548	0.547	0.544	0.548	0.548
	Training accuracy	0.603	0.606	0.629	0.625	0.648	0.657	0.681	0.706	0.717
	GF	1.136	1.145	1.216	1.211	1.284	1.321	1.429	1.537	1.597
	F1-Score	0.47	0.44	0.44	0.44	0.45	0.45	0.45	0.45	0.45
HTEdf	Testing accuracy	0.556	0.556	0.555	0.557	0.556	0.555	0.557	0.559	0.559
	Training accuracy	0.583	0.596	0.601	0.626	0.626	0.646	0.676	0.701	0.704
	GF	1.065	1.126	1.099	1.185	1.219	1.257	1.367	1.474	1.490
	F1-Score	0.44	0.46	0.46	0.45	0.46	0.46	0.46	0.46	0.46

Table A.10: Confusion Matrices of Ensembles on Skewed Class Distributions for Red Wine Dataset

10-90%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.80	0	0.10	0	0	0.10
5	0.18	0.30	0.25	0.15	0.07	0.05
6	0.20	0.19	0.08	0.11	0.17	0.25
7	0	0.10	0.05	0.05	0.28	0.52
8	0	0	0	0	0.40	**0.60**

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.50	0.20	0.10	0	0
5	0.05	0.32	0.40	0.14	0.05	0.04
6	0.02	0.21	0.27	0.17	0.17	0.16
7	0	0.07	0.03	0.21	0.26	0.43
8	0	0	0	0	0.60	0.40

DTE

	3	4	5	6	7	8
3	0	0	0	1.00	0	0
4	0.10	0.20	0.60	0.10	0	0
5	0.06	0.22	0.45	0.22	0.05	0
6	0.05	0.16	0.21	0.36	0.16	0.06
7	0	0.02	0.10	0.26	0.52	0.10
8	0	0.20	0	0	0.40	0.40

RF

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0.10	0	0	0
5	0.03	0.30	0.49	0.15	0.03	0
6	0.02	0.22	0.18	0.32	0.20	0.06
7	0	0.10	0.07	0.36	0.31	0.16
8	0	0	0	0.20	0.60	0.20

SVME

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0	0.10	0	0
5	0.03	0.31	0.46	0.15	0.04	0.01
6	0.04	0.25	0.18	0.21	0.20	0.12
7	0	0.07	0.02	0.15	0.38	0.38
8	0	0	0	0	0.40	**0.60**

NNE

	3	4	5	6	7	8
3	0	0	1.00	0	0	0
4	0.10	**0.70**	0.20	0	0	0
5	0.02	0.24	0.48	0.21	0.05	0
6	0.02	0.14	0.15	0.36	0.28	0.05
7	0	0.02	0	0.12	0.67	0.19
8	0	0	0	0	0.80	0.20

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.80	0	0.10	0	0	0.10
5	0.18	0.31	0.24	0.15	0.07	0.05
6	0.20	0.18	0.08	0.11	0.16	0.27
7	0	0.10	0.05	0.04	0.29	0.52
8	0	0	0	0	0.40	**0.60**

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.50	0.20	0.10	0	0
5	0.03	0.22	0.44	0.21	0.07	0.03
6	0.01	0.18	0.26	0.21	0.17	0.17
7	0	0.07	0	0.14	0.34	0.45
8	0	0	0	0	0.60	0.40

DThte

	3	4	5	6	7	8
3	0	0	0	1.00	0	0
4	0.10	0.40	0.50	0	0	0.0
5	0.04	0.25	0.45	0.21	0.05	0
6	0.02	0.22	0.14	0.40	0.16	0.06
7	0	0.05	0.02	0.19	0.55	0.19
8	0	0	0	0	0.80	0.20

SVMhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.70	0.20	0.10	0	0	0
5	0.06	0.36	0.47	0.05	0.05	0.01
6	0.04	0.35	0.18	0.14	0.14	0.15
7	0	0.10	0.05	0.07	0.31	0.47
8	0	0	0	0	0.60	0.40

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.10	0.60	0	0.30	0	0
5	0.02	0.23	0.47	0.26	0.02	0
6	0.02	0.14	0.16	0.48	0.12	0.08
7	0	0.05	0.02	0.36	0.40	0.17
8	0	0	0	0.40	0.20	0.40

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.10	0.50	0.20	0.20	0	0
5	0.01	0.15	0.53	0.27	0.03	0.01
6	0.01	0.10	0.15	0.50	0.17	0.07
7	0	0.05	0	0.14	0.67	0.14
8	0	0	0	0	0.50	0.50

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.30	0.50	0.20	0	0	0
5	0.05	0.22	0.47	0.19	0.05	0.02
6	0.02	0.17	0.18	0.30	0.23	0.10
7	0	0.07	0.02	0.12	0.45	0.34
8	0	0	0	0	0.40	**0.60**

15-85%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.80	0	0.10	0	0	0.10
5	0.21	0.29	0.24	0.10	0.12	0.04
6	0.22	0.15	0.10	0.08	0.15	0.30
7	0	0.10	0.02	0.05	0.31	0.52
8	0	0	0	0	0.40	**0.60**

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.60	0.10	0.10	0	0
5	0.07	0.34	0.42	0.10	0.04	0.03
6	0.03	0.24	0.27	0.18	0.13	0.15
7	0	0.07	0	0.24	0.29	0.40
8	0	0	0	0.20	0.60	0.40

DTE

	3	4	5	6	7	8
3	0	0	1.00	0	0	0
4	0.10	0.30	0.40	0.10	0	0.10
5	0.06	0.28	0.34	0.23	0.05	0.04
6	0.05	0.21	0.18	0.27	0.21	0.08
7	0	0.07	0	0.26	0.50	0.17
8	0	0	0	0.20	0.40	0.40

RF

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.40	0.10	0.10	0	0
5	0.05	0.30	0.49	0.13	0.02	0.01
6	0.03	0.21	0.24	0.27	0.20	0.05
7	0	0.10	0	0.29	0.40	0.21
8	0	0	0	0	1.00	0

SVME

	3	4	5	6	7	8
3	0	1	0	0	0	0
4	0.40	0.50	0	0.10	0	0
5	0.04	0.36	0.39	0.17	0.04	0
6	0.04	0.29	0.14	0.23	0.19	0.11
7	0	0.05	0	0.14	0.33	0.48
8	0	0	0	0	0.40	**0.60**

NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0	**0.70**	0.20	0.10	0	0
5	0.02	0.23	0.51	0.21	0.03	0
6	0.02	0.14	0.23	0.33	0.24	0.04
7	0	0.02	0	0.19	0.64	0.15
8	0	0	0	0	0.60	0.40

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.24	0.27	0.24	0.10	0.12	0.03
6	0.25	0.14	0.08	0.08	0.16	0.29
7	0	0.10	0.02	0.05	0.31	0.52
8	0	0	0	0	0.40	**0.60**

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.60	0	0.20	0	0
5	0.05	0.29	0.42	0.14	0.05	0.05
6	0.03	0.18	0.26	0.18	0.17	0.18
7	0	0.07	0	0.19	0.31	0.43
8	0	0	0	0	0.60	0.40

DThte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.30	0.20	0.10	0	0.10
5	0.03	0.25	0.43	0.25	0.02	0.02
6	0.01	0.22	0.22	0.29	0.19	0.07
7	0	0.05	0	0.21	0.55	0.19
8	0	0	0	0	0.60	0.40

SVMhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0.10	0	0
5	0.12	0.38	0.35	0.12	0.02	0.01
6	0.06	0.36	0.14	0.16	0.14	0.14
7	0	0.10	0	0.14	0.31	0.45
8	0	0	0	0	0.40	**0.60**

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.10	**0.70**	0.10	0	0	0.10
5	0.02	0.33	0.49	0.12	0.02	0.02
6	0.02	0.24	0.17	0.32	0.17	0.08
7	0	0.05	0	0.24	0.54	0.17
8	0	0	0	0.20	0.40	0.40

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.20	0.50	0.30	0	0	0
5	0.07	0.26	0.43	0.20	0.02	0.02
6	0.03	0.22	0.17	0.24	0.23	0.11
7	0	0.05	0	0.17	0.48	0.30
8	0	0	0	0	0.40	**0.60**

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.10	0.60	0.20	0.10	0	0
5	0.05	0.30	0.43	0.16	0.04	0.02
6	0.03	0.23	0.20	0.15	0.28	0.11
7	0	0.07	0	0.12	0.45	0.36
8	0	0	0	0	0.40	**0.60**

20-80%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.80	0	0.10	0	0	0.10
5	0.27	0.22	0.26	0.10	0.10	0.05
6	0.28	0.11	0.08	0.08	0.18	0.27
7	0.02	0.07	0.03	0.05	0.31	0.52
8	0	0	0	0	0.40	0.60

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.50	0.10	0.10	0	0
5	0.08	0.32	0.38	0.15	0.05	0.02
6	0.03	0.21	0.25	0.27	0.10	0.14
7	0	0.05	0	0.29	0.31	0.35
8	0	0	0	0	0.60	0.40

DTE

	3	4	5	6	7	8
3	0	0	1.00	0	0	0
4	0.10	0.20	0.50	0.10	0.10	0
5	0.05	0.30	0.38	0.22	0.05	0
6	0.04	0.17	0.21	0.31	0.20	0.07
7	0	0.07	0.02	0.22	0.48	0.21
8	0	0	0	0	0.60	0.40

RF

	3	4	5	6	7	8
3	0	0	0	1.00	0	0
4	0.30	0.30	0.20	0.20	0	0
5	0.06	0.25	0.42	0.23	0.02	0.02
6	0.04	0.17	0.20	0.34	0.20	0.05
7	0	0.05	0.02	0.33	0.43	0.17
8	0	0	0	0	0.60	0.40

SVME

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.60	0.30	0	0.10	0	0
5	0.05	0.30	0.44	0.15	0.05	0.01
6	0.03	0.25	0.17	0.23	0.20	0.12
7	0	0.05	0.02	0.10	0.43	0.40
8	0	0	0	0	0.20	**0.80**

NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.10	**0.70**	0.20	0	0	0
5	0.02	0.20	0.55	0.21	0.02	0
6	0.02	0.13	0.23	0.38	0.20	0.04
7	0	0.02	0.02	0.19	0.60	0.17
8	0	0	0	0	0.80	0.20

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.40	0.40	0.10	0	0	0.10
5	0.08	0.27	0.37	0.18	0.05	0.05
6	0.11	0.21	0.10	0.20	0.19	0.19
7	0	0.07	0.02	0.10	0.29	0.52
8	0	0	0	0	0.60	0.40

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.50	0.20	0.10	0	0
5	0.07	0.24	0.39	0.18	0.07	0.05
6	0.03	0.17	0.24	0.22	0.17	0.16
7	0	0.05	0	0.24	0.33	0.38
8	0	0	0	0	0.60	0.40

DThte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.20	0.40	0.10	0.10	0
5	0.03	0.23	0.48	0.24	0.02	0
6	0.03	0.19	0.16	0.34	0.23	0.05
7	0	0.07	0.02	0.21	0.48	0.22
8	0	0	0	0	0.80	0.20

SVMhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.60	0.30	0	0.10	0	0
5	0.12	0.31	0.39	0.12	0.04	0.02
6	0.06	0.32	0.11	0.20	0.14	0.17
7	0	0.10	0	0.10	0.38	0.42
8	0	0	0	0	0.20	**0.80**

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.40	0.30	0.10	0	0
5	0.05	0.16	0.66	0.08	0.05	0
6	0.04	0.06	0.43	0.20	0.19	0.08
7	0	0.02	0.05	0.05	0.71	0.17
8	0	0	0	0.20	0.20	0.60

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.20	0.50	0.30	0	0	0
5	0.05	0.25	0.50	0.15	0.05	0
6	0.03	0.14	0.25	0.26	0.22	0.10
7	0	0.05	0.02	0.07	0.50	0.36
8	0	0	0	0	0.40	0.60

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.40	0.50	0	0.10	0	0
5	0.06	0.24	0.45	0.19	0.05	0.01
6	0.05	0.17	0.14	0.31	0.23	0.10
7	0	0.07	0.02	0.05	0.48	0.38
8	0	0	0	0	0.40	0.60

25-75%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0.10	0	0	0.10
5	0.34	0.21	0.23	0.07	0.10	0.05
6	0.30	0.10	0.10	0.06	0.15	0.29
7	0.02	0.07	0.02	0.08	0.29	0.52
8	0	0	0	0	0.40	0.60

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.40	0.20	0.10	0	0
5	0.08	0.28	0.42	0.17	0.03	0.02
6	0.03	0.21	0.25	0.23	0.14	0.14
7	0	0.07	0.05	0.17	0.38	0.33
8	0	0	0	0	0.60	0.40

DTE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.30	0.20	0.20	0	0
5	0.05	0.25	0.48	0.17	0.03	0.02
6	0.04	0.20	0.25	0.26	0.19	0.06
7	0.02	0.07	0.05	0.29	0.36	0.21
8	0	0	0	0.20	0.80	0

RF

	3	4	5	6	7	8
3	0	0	0	1.00	0	0
4	0.40	0.40	0.20	0	0	0
5	0.07	0.27	0.48	0.15	0.03	0
6	0.07	0.17	0.24	0.28	0.19	0.05
7	0	0.07	0.05	0.14	0.60	0.14
8	0	0	0	0.20	0.80	0

SVME

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	**0.60**	0.10	0	0	0
5	0.04	0.33	0.46	0.12	0.04	0.01
6	0.05	0.28	0.18	0.18	0.17	0.14
7	0	0.07	0.05	0.02	0.43	0.43
8	0	0	0	0	0.20	**0.80**

NNE

	3	4	5	6	7	8
3	0	0	1.00	0	0	0
4	0	0.50	0.40	0.10	0	0
5	0.02	0.19	0.53	0.20	0.06	0
6	0.03	0.11	0.20	0.33	0.28	0.05
7	0	0	0	0.14	0.60	0.26
8	0	0	0	0	0.80	0.20

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.36	0.19	0.23	0.07	0.10	0.05
6	0.34	0.08	0.09	0.06	0.14	0.19
7	0.05	0.07	0.02	0.05	0.29	0.52
8	0	0	0	0	0.60	0.40

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.50	0.30	0.30	0	0
5	0.06	0.23	0.46	0.17	0.05	0.03
6	0.03	0.16	0.20	0.21	0.20	0.20
7	0	0.05	0.02	0.22	0.26	0.45
8	0	0	0	0	0.60	0.40

DThte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.40	0.20	0.10	0	0.0
5	0.06	0.20	0.55	0.18	0.01	0
6	0.04	0.17	0.23	0.27	0.23	0.06
7	0	0.05	0	0.24	0.57	0.14
8	0	0	0	0.20	0.60	0.20

SVMhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.60	0.30	0	0	0.10	0
5	0.10	0.42	0.35	0.07	0.04	0.02
6	0.06	0.36	0.16	0.10	0.18	0.14
7	0	0.10	0.04	0	0.36	0.50
8	0	0	0	0	0.40	0.60

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.50	0.20	0	0	0.10
5	0.08	0.22	0.53	0.12	0.03	0.02
6	0.05	0.11	0.25	0.24	0.25	0.10
7	0	0.05	0.02	0.05	0.64	0.24
8	0	0	0	0.20	0.40	0.40

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.30	0.50	0.10	0.10	0	0
5	0.04	0.25	0.49	0.16	0.04	0.02
6	0.04	0.20	0.16	0.25	0.24	0.11
7	0	0.05	0.02	0.07	0.40	0.46
8	0	0	0	0	0.40	0.60

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.20	**0.60**	0.20	0	0	0
5	0.06	0.31	0.44	0.13	0.05	0.01
6	0.05	0.20	0.23	0.25	0.17	0.10
7	0	0.05	0	0.07	0.48	0.40
8	0	0	0	0	0.40	0.60

30-70%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.31	0.16	0.25	0.15	0.10	0.03
6	0.32	0.07	0.10	0.08	0.16	0.27
7	0.02	0.07	0.05	0.07	0.26	0.53
8	0	0	0	0	0.40	0.60

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0	0.10	0	0
5	0.08	0.28	0.42	0.12	0.05	0.05
6	0.03	0.20	0.25	0.23	0.14	0.15
7	0	0.07	0.07	0.31	0.21	0.34
8	0	0	0	0	0.60	0.40

DTE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.10	0.20	0.50	0.10	0.10	0
5	0.04	0.29	0.38	0.22	0.07	0
6	0.02	0.23	0.22	0.30	0.17	0.06
7	0	0.05	0.12	0.29	0.38	0.16
8	0	0	0	0.20	0.60	0.20

RF

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.60	0.20	0	0	0
5	0.05	0.28	0.44	0.21	0.01	0.01
6	0.07	0.16	0.25	0.26	0.23	0.03
7	0	0.07	0.07	0.26	0.55	0.05
8	0	0	0	0.40	0.40	0.20

SVME

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.60	0.10	0.10	0	0
5	0.03	0.34	0.44	0.15	0.03	0.01
6	0.04	0.28	0.20	0.19	0.14	0.15
7	0	0.07	0.04	0.10	0.36	0.43
8	0	0	0	0	0.20	**0.80**

NNE

	3	4	5	6	7	8
3	0	0	1.00	0	0	0
4	0.10	**0.80**	0.10	0	0	0
5	0.02	0.32	0.45	0.14	0.07	0
6	0.04	0.20	0.12	0.27	0.32	0.05
7	0	0.02	0.03	0.05	0.67	0.23
8	0	0	0	0	0.80	0.20

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.39	0.14	0.25	0.10	0.10	0.02
6	0.37	0.06	0.08	0.08	0.14	0.27
7	0.05	0.07	0.05	0.05	0.26	0.52
8	0	0	0	0	0.40	0.60

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.60	0	0.10	0	0
5	0.05	0.26	0.46	0.12	0.06	0.05
6	0.04	0.14	0.27	0.23	0.12	0.20
7	0	0.07	0.02	0.26	0.29	0.36
8	0	0	0	0.2	0.60	0.40

DThte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.40	0.20	0.10	0	0.0
5	0.05	0.28	0.39	0.20	0.08	0
6	0.02	0.17	0.22	0.28	0.24	0.07
7	0	0.05	0.07	0.10	0.64	0.14
8	0	0	0	0	0.80	0.20

SVMhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.60	0.30	0	0	0	0.10
5	0.10	0.33	0.37	0.13	0.05	0.02
6	0.06	0.29	0.14	0.17	0.17	0.17
7	0	0.10	0.05	0.02	0.33	0.50
8	0	0	0	0	0.20	**0.80**

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0.10	0	0	0
5	0.10	0.34	0.35	0.13	0.07	0.01
6	0.06	0.22	0.12	0.22	0.33	0.05
7	0	0.02	0.07	0.02	0.72	0.17
8	0	0	0	0	0.60	0.40

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.10	0.60	0.30	0	0	0
5	0.04	0.29	0.46	0.15	0.05	0.01
6	0.02	0.18	0.26	0.17	0.27	0.10
7	0	0.07	0.02	0.17	0.45	0.29
8	0	0	0	0.20	0.20	0.60

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.10	0.60	0.10	0.20	0	0
5	0.05	0.28	0.47	0.16	0.02	0.02
6	0.05	0.20	0.17	0.25	0.20	0.13
7	0	0.05	0.02	0.14	0.45	0.34
8	0	0	0	0	0.40	0.60

35-65%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.29	0.22	0.24	0.10	0.01	0.05
6	0.32	0.09	0.09	0.09	0.13	0.28
7	0.02	0.07	0.05	0.07	0.26	0.52
8	0	0	0	0.40	0	0.60

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0	0.10	0	0
5	0.08	0.29	0.41	0.14	0.03	0.05
6	0.05	0.20	0.30	0.17	0.14	0.15
7	0	0.07	0.02	0.22	0.26	0.43
8	0	0	0	0.20	0.40	0.40

DTE

	3	4	5	6	7	8
3	0	0	0	1.00	0	0
4	0.20	0.20	0.30	0.20	0.10	0
5	0.06	0.30	0.31	0.29	0.03	0.01
6	0.06	0.17	0.20	0.32	0.21	0.04
7	0	0.05	0.02	0.26	0.55	0.12
8	0	0	0	0.20	0.60	0.20

RF

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.30	0.20	0.10	0	0
5	0.07	0.32	0.44	0.14	0.02	0.01
6	0.07	0.21	0.21	0.21	0.24	0.06
7	0	0.10	0.05	0.16	0.50	0.19
8	0	0	0	0.20	0.60	0.20

SVME

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	**0.70**	0.10	0	0	0
5	0.03	0.32	0.47	0.12	0.04	0.02
6	0.04	0.27	0.20	0.18	0.15	0.16
7	0	0.07	0.05	0.05	0.38	0.45
8	0	0	0	0	0.20	**0.80**

NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.10	**0.70**	0.20	0	0	0
5	0.02	0.27	0.51	0.15	0.05	0
6	0.03	0.20	0.16	0.30	0.24	0.07
7	0	0.05	0.05	0.02	0.64	0.24
8	0	0	0	0.40	0.40	0.20

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.42	0.15	0.26	0.05	0.08	0.04
6	0.39	0.06	0.08	0.07	0.13	0.27
7	0.07	0.07	0.05	0.05	0.24	0.52
8	0	0	0	0.40	0	0.60

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.3	0.50	0.10	0.10	0	0
5	0.08	0.21	0.45	0.16	0.04	0.06
6	0.04	0.16	0.25	0.23	0.15	0.17
7	0	0.07	0.02	0.19	0.24	0.48
8	0	0	0	0	0.40	0.60

DThte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.20	0.30	0.20	0.10	0
5	0.08	0.32	0.41	0.16	0.03	0
6	0.07	0.16	0.20	0.29	0.25	0.03
7	0	0.05	0.02	0.26	0.53	0.14
8	0	0	0	0.20	0.40	0.40

SVMhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.60	0.30	0	0	0.10	0
5	0.08	0.39	0.38	0.08	0.05	0.02
6	0.07	0.35	0.17	0.08	0.17	0.16
7	0	0.10	0.07	0.02	0.31	0.50
8	0	0	0	0	0.40	0.60

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.50	0.40	0	0.10	0	0
5	0.10	0.26	0.42	0.11	0.10	0.01
6	0.06	0.24	0.11	0.21	0.29	0.09
7	0	0.05	0	0.02	0.64	0.29
8	0	0	0	0	0.40	0.60

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.20	0.60	0.10	0.10	0	0
5	0.08	0.33	0.35	0.19	0.03	0.02
6	0.09	0.20	0.17	0.25	0.18	0.11
7	0	0.05	0	0.14	0.38	0.43
8	0	0	0	0.20	0.20	0.60

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.30	0.60	0	0.10	0	0
5	0.08	0.28	0.39	0.20	0.03	0.02
6	0.08	0.16	0.21	0.28	0.18	0.09
7	0	0.07	0	0.12	0.48	0.33
8	0	0	0	0	0.40	0.60

40-60%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.34	0.21	0.25	0.07	0.08	0.05
6	0.33	0.09	0.09	0.09	0.13	0.27
7	0.02	0.07	0.05	0.07	0.26	0.53
8	0	0	0	0.40	0	0.60

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.40	0.30	0	0	0
5	0.09	0.25	0.43	0.19	0.02	0.02
6	0.05	0.18	0.33	0.11	0.17	0.16
7	0	0.05	0.05	0.21	0.29	0.40
8	0	0	0	0	0.60	0.40

DTE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.10	0	0.60	0.20	0.10	0
5	0.08	0.27	0.46	0.18	0.01	0
6	0.03	0.23	0.29	0.20	0.18	0.07
7	0	0.10	0.09	0.19	0.50	0.12
8	0	0	0	0.20	0.40	0.40

RF

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.20	0.20	0.10	0	0.10
5	0.08	0.30	0.43	0.16	0.02	0.01
6	0.05	0.18	0.20	0.36	0.17	0.04
7	0.02	0.07	0.05	0.19	0.55	0.12
8	0	0	0	0.40	0.40	0.20

SVME

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	**0.60**	0.10	0	0	0
5	0.03	0.31	0.49	0.11	0.04	0.02
6	0.04	0.27	0.24	0.12	0.18	0.15
7	0	0.07	0.05	0	0.45	0.43
8	0	0	0	0	0.20	**0.80**

NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.10	0.50	0.40	0	0	0
5	0.02	0.27	0.52	0.15	0.04	0
6	0.03	0.15	0.23	0.28	0.24	0.07
7	0	0.02	0.05	0.07	0.55	0.31
8	0	0	0	0.20	0.60	0.20

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.44	0.15	0.25	0.05	0.06	0.05
6	0.43	0.06	0.08	0.06	0.10	0.27
7	0.07	0.07	0.05	0.05	0.24	0.52
8	0	0	0	0.40	0	0.60

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.50	0.30	0	0	0
5	0.07	0.19	0.50	0.14	0.06	0.04
6	0.04	0.13	0.27	0.22	0.14	0.20
7	0	0.05	0.05	0.16	0.24	0.50
8	0	0	0	0	0.40	0.60

DThte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.10	0.40	0.10	0.10	0
5	0.07	0.26	0.45	0.20	0.01	0.01
6	0.04	0.23	0.23	0.27	0.18	0.05
7	0	0.07	0.05	0.26	0.41	0.21
8	0	0	0	0	0.80	0.20

SVMhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.60	0.30	0	0	0.10	0
5	0.13	0.38	0.36	0.06	0.05	0.02
6	0.08	0.36	0.16	0.08	0.17	0.15
7	0	0.10	0.07	0	0.33	0.50
8	0	0	0	0	0.40	0.60

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0.10	0	0	0
5	0.12	0.21	0.50	0.10	0.06	0.01
6	0.06	0.15	0.25	0.24	0.22	0.08
7	0.02	0.05	0.05	0.14	0.45	0.29
8	0	0	0	0.20	0.40	0.40

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.20	**0.60**	0.20	0	0	0
5	0.05	0.34	0.40	0.15	0.02	0.04
6	0.04	0.17	0.23	0.20	0.27	0.09
7	0	0.05	0.02	0.09	0.55	0.29
8	0	0	0	0	0.40	0.60

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.30	**0.60**	0.10	0	0	0
5	0.11	0.27	0.40	0.16	0.04	0.02
6	0.06	0.16	0.23	0.19	0.27	0.09
7	0	0.05	0.02	0.05	0.38	0.50
8	0	0	0	0.40	0	0.60

45-55%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.35	0.23	0.21	0.08	0.08	0.05
6	0.33	0.12	0.05	0.08	0.18	0.24
7	0.05	0.07	0.02	0.07	0.26	0.53
8	0	0	0	0.20	0.20	0.60

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0	0.10	0	0
5	0.09	0.33	0.35	0.17	0.03	0.03
6	0.05	0.20	0.24	0.26	0.10	0.15
7	0	0.07	0.05	0.24	0.26	0.38
8	0	0	0	0.20	0.20	0.60

DTE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.50	0.20	0.10	0	0
5	0.11	0.29	0.38	0.19	0.02	0.01
6	0.06	0.17	0.26	0.26	0.19	0.06
7	0	0.05	0.02	0.29	0.45	0.19
8	0	0	0	0.20	0.60	0.20

RF

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.60	0	0	0.10	0
5	0.08	0.32	0.45	0.09	0.04	0.02
6	0.05	0.25	0.20	0.21	0.20	0.09
7	0	0.07	0.02	0.29	0.36	0.26
8	0	0	0	0.20	0.40	0.40

SVME

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.60	0.10	0	0	0
5	0.04	0.39	0.42	0.09	0.04	0.02
6	0.04	0.30	0.17	0.16	0.19	0.14
7	0	0.07	0.05	0.05	0.38	0.45
8	0	0	0	0	0.20	**0.80**

NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.50	0.20	0	0	0
5	0.05	0.26	0.50	0.14	0.04	0.01
6	0.04	0.19	0.16	0.29	0.24	0.08
7	0	0.05	0.02	0.10	0.57	0.26
8	0	0	0	0	0.80	0.20

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.47	0.18	0.21	0.04	0.06	0.04
6	0.44	0.08	0.05	0.07	0.13	0.23
7	0.10	0.07	0.02	0.05	0.24	0.52
8	0	0	0	0	0.40	0.60

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0	0.10	0	0
5	0.08	0.28	0.36	0.20	0.05	0.03
6	0.04	0.13	0.23	0.30	0.14	0.16
7	0	0.05	0	0.16	0.24	0.55
8	0	0	0	0	0.40	0.60

DThte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.20	0.20	0.20	0.10	0
5	0.09	0.25	0.46	0.17	0.02	0.01
6	0.05	0.16	0.24	0.27	0.24	0.04
7	0	0.02	0.10	0.17	0.50	0.21
8	0	0	0	0.20	0.60	0.20

SVMhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.60	0.30	0	0	0	0.10
5	0.09	0.50	0.31	0.05	0.03	0.02
6	0.07	0.38	0.14	0.10	0.17	0.14
7	0	0.09	0.07	0.05	0.29	0.50
8	0	0	0	0	0.40	0.60

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.50	0.20	0	0	0
5	0.07	0.23	0.52	0.13	0.05	0
6	0.04	0.13	0.28	0.29	0.19	0.07
7	0	0.02	0.12	0.14	0.50	0.22
8	0	0	0	0	0.40	0.60

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.30	0.60	0	0	0.10	0
5	0.11	0.29	0.36	0.17	0.05	0.02
6	0.07	0.15	0.22	0.19	0.26	0.11
7	0	0.07	0	0.17	0.40	0.36
8	0	0	0	0.20	0.20	0.60

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.26	**0.64**	0	0.10	0	0
5	0.15	0.26	0.36	0.16	0.05	0.02
6	0.09	0.14	0.18	0.26	0.22	0.11
7	0	0.05	0	0.12	0.45	0.38
8	0	0	0	0.20	0.18	0.62

50-50%

NBE

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.37	0.19	0.26	0.05	0.08	0.05
6	0.33	0.09	0.10	0.05	0.16	0.27
7	0.05	0.07	0.02	0.07	0.26	0.53
8	0	0	0	0	0.40	0.60

*k*NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0.10	0	0	0
5	0.14	0.25	0.39	0.16	0.02	0.04
6	0.05	0.21	0.28	0.20	0.12	0.14
7	0	0.07	0	0.33	0.24	0.36
8	0	0	0	0	0.60	0.40

DTE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.20	0.60	0	0.10	0.10	0
5	0.04	0.29	0.39	0.22	0.06	0
6	0.04	0.16	0.25	0.28	0.20	0.07
7	0	0.02	0.05	0.40	0.36	0.17
8	0	0	0.20	0.40	0.20	0.20

RF

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.50	0.40	0	0.10	0	0
5	0.08	0.28	0.43	0.13	0.07	0.01
6	0.04	0.24	0.24	0.25	0.17	0.06
7	0	0.10	0.02	0.26	0.43	0.19
8	0	0	0.20	0	0.80	0

SVME

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.60	0	0.10	0	0
5	0.03	0.34	0.43	0.13	0.05	0.02
6	0.04	0.32	0.14	0.18	0.17	0.15
7	0	0.07	0.02	0.05	0.38	0.48
8	0	0	0	0	0.40	0.60

NNE

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.30	0.60	0.10	0	0	0
5	0.02	0.27	0.57	0.09	0.05	0
6	0.04	0.19	0.23	0.23	0.24	0.07
7	0	0.05	0.02	0.05	0.64	0.24
8	0	0	0	0	0.80	0.20

NBhte

	3	4	5	6	7	8
3	**1.00**	0	0	0	0	0
4	0.90	0	0	0	0	0.10
5	0.55	0.11	0.22	0.03	0.05	0.04
6	0.48	0.05	0.08	0.04	0.11	0.24
7	0.10	0.05	0.02	0.05	0.26	0.52
8	0	0	0	0	0.40	0.60

*k*NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0.10	0	0	0
5	0.11	0.17	0.46	0.18	0.04	0.04
6	0.05	0.14	0.23	0.27	0.14	0.17
7	0	0.07	0	0.24	0.24	0.45
8	0	0	0	0	0.40	0.60

DThte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.40	0.10	0	0.10	0
5	0.07	0.26	0.43	0.18	0.05	0.01
6	0.05	0.16	0.24	0.26	0.21	0.08
7	0	0	0.05	0.28	0.48	0.19
8	0	0	0	0.20	0.40	0.40

SVMhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.60	0.30	0	0	0.10	0
5	0.1	0.26	0.48	0.09	0.04	0.03
6	0.08	0.26	0.26	0.08	0.15	0.17
7	0	0.09	0.05	0.05	0.31	0.50
8	0	0	0	0	0.40	0.60

NNhte

	3	4	5	6	7	8
3	0	1.00	0	0	0	0
4	0.40	0.50	0	0.10	0	0
5	0.11	0.27	0.44	0.14	0.04	0
6	0.07	0.22	0.18	0.24	0.21	0.08
7	0.02	0.10	0.02	0.09	0.48	0.29
8	0	0	0	0.20	0.20	0.60

HTEsm

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.30	0.60	0	0	0.10	0
5	0.06	0.24	0.46	0.17	0.05	0.02
6	0.05	0.16	0.21	0.22	0.23	0.13
7	0	0.02	0.02	0.22	0.36	0.38
8	0	0	0	0	0.40	0.60

HTEdf

	3	4	5	6	7	8
3	0.10	0.90	0	0	0	0
4	0.25	**0.65**	0.10	0	0	0
5	0.08	0.25	0.39	0.18	0.07	0.03
6	0.08	0.15	0.23	0.18	0.23	0.13
7	0	0.07	0	0.17	0.38	0.38
8	0	0	0	0	0.38	**0.62**

Car Evaluation Dataset

Table A.11: Ensemble Performance on Skewed Class Distributions for Car Evaluation Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.809	0.809	0.809	0.809	0.809	0.809	0.809	0.809	0.809
	Training accuracy	0.887	0.885	0.870	0.867	0.875	0.859	0.853	0.834	0.840
	GF	1.690	1.661	1.469	1.436	1.528	1.355	1.299	1.151	1.194
	F1-Score	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82
*k*NNE	Testing accuracy	0.846	0.846	0.846	0.846	0.846	0.846	0.846	0.846	0.846
	Training accuracy	0.887	0.886	0.849	0.868	0.865	0.866	0.868	0.865	0.877
	GF	1.363	1.351	1.020	1.167	1.141	1.149	1.167	1.141	1.252
	F1-Score	0.81	0.78	0.80	0.79	0.77	0.75	0.75	0.75	0.76
DTE	Testing accuracy	0.922	0.922	0.924	0.921	0.924	0.921	0.925	0.924	0.926
	Training accuracy	0.936	0.936	0.925	0.941	0.937	0.931	0.936	0.935	0.944
	GF	1.219	1.219	1.013	1.339	1.206	1.145	1.172	1.169	1.321
	F1-Score	0.82	0.87	0.91	0.87	0.90	0.92	0.92	0.91	0.87
RF	Testing accuracy	0.884	0.880	0.876	0.894	0.878	0.881	0.880	0.884	0.891
	Training accuracy	0.953	0.945	0.938	0.941	0.948	0.942	0.938	0.931	0.936
	GF	2.468	2.182	2.000	1.797	2.346	2.052	1.935	1.681	1.703
	F1-Score	0.81	0.90	0.88	0.89	0.85	0.86	0.85	0.76	0.83
SVME	Testing accuracy	0.872	0.873	0.871	0.872	0.872	0.870	0.871	0.871	0.870
	Training accuracy	0.973	0.966	0.966	0.959	0.959	0.949	0.950	0.952	0.956
	GF	4.741	3.735	3.794	3.122	3.122	2.549	2.580	2.687	2.955
	F1-Score	0.90	0.91	0.93	0.93	0.93	0.92	0.93	0.94	0.93
NNE	Testing accuracy	0.952	0.953	0.951	0.954	0.950	0.955	0.951	0.950	0.953
	Training accuracy	0.987	0.982	0.978	0.978	0.976	0.969	0.967	0.956	0.958
	GF	3.692	2.611	2.227	2.091	2.083	1.452	1.485	1.136	1.119
	F1-Score	0.95	0.95	0.98	0.97	0.96	0.94	0.97	0.98	0.95

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.809	0.809	0.809	0.809	0.809	0.809	0.809	0.809	0.809
	Training accuracy	0.904	0.886	0.871	0.867	0.875	0.859	0.853	0.834	0.840
	GF	1.990	1.675	1.481	1.436	1.528	1.355	1.299	1.151	1.194
	F1-Score	0.86	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82
*k*NNhte	Testing accuracy	0.852	0.852	0.852	0.852	0.852	0.852	0.852	0.852	0.852
	Training accuracy	0.899	0.887	0.858	0.871	0.878	0.874	0.891	0.888	0.891
	GF	1.465	1.310	1.042	1.147	1.213	1.175	1.358	1.321	1.358
	F1-Score	0.79	0.78	0.79	0.78	0.77	0.76	0.75	0.75	0.75
DThte	Testing accuracy	0.938	0.932	0.933	0.935	0.936	0.935	0.935	0.935	0.934
	Training accuracy	0.954	0.963	0.942	0.957	0.950	0.945	0.948	0.942	0.946
	GF	1.348	1.838	1.155	1.512	1.280	1.182	1.250	1.121	1.222
	F1-Score	0.84	0.91	0.91	0.93	0.92	0.92	0.93	0.92	0.88
SVMhte	Testing accuracy	0.903	0.901	0.903	0.900	0.903	0.903	0.903	0.904	0.904
	Training accuracy	0.955	0.939	0.916	0.921	0.928	0.922	0.925	0.925	0.939
	GF	2.156	1.623	1.155	1.266	1.347	1.244	1.293	1.280	1.574
	F1-Score	0.92	0.90	0.88	0.91	0.88	0.88	0.88	0.89	0.86
NNhte	Testing accuracy	0.947	0.950	0.951	0.949	0.950	0.951	0.944	0.948	0.945
	Training accuracy	0.985	0.981	0.974	0.977	0.976	0.965	0.966	0.952	0.958
	GF	3.533	2.632	1.885	2.217	2.083	1.400	1.647	1.083	1.310
	F1-Score	0.93	0.95	0.98	0.97	0.96	0.94	0.97	0.98	0.95
HTEsm	Testing accuracy	0.952	0.949	0.953	0.953	0.953	0.951	0.951	0.951	0.951
	Training accuracy	0.972	0.967	0.958	0.960	0.962	0.952	0.953	0.949	0.958
	GF	1.714	1.545	1.119	1.175	1.237	1.021	1.043	0.961	1.167
	F1-Score	0.94	0.95	0.95	0.94	0.94	0.94	0.95	0.95	0.95
HTEdf	Testing accuracy	0.954	0.955	0.955	0.956	0.957	0.956	0.953	0.952	0.956
	Training accuracy	0.977	0.973	0.965	0.966	0.966	0.955	0.956	0.952	0.957
	GF	2.000	1.667	1.286	1.294	1.265	0.978	1.068	1.000	1.023
	F1-Score	0.96	0.96	0.96	0.96	0.96	0.95	0.96	0.96	0.96

Table A.12: Confusion Matrices of Ensembles on Skewed Class Distributions for Car Evaluation Dataset

10-90%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.42	0.35	0.06	0.17
1	0	0.82	0	0.18
2	0.03	0.04	0.91	0.02
3	0	0.24	0	0.76

DTE

	0	1	2	3
0	0.41	0.24	0.27	0.08
1	0	0.91	0	0.09
2	0.03	0.01	0.96	0
3	0	0.18	0	0.82

RF

	0	1	2	3
0	0.31	0.31	0.23	0.15
1	0	0.91	0	0.09
2	0.01	0	0.99	0
3	0	0	0	**1.00**

SVME

	0	1	2	3
0	0.64	0.05	0.31	0
1	0	0.82	0.09	0.09
2	0	0	1.00	0
3	0.06	0.06	0	0.88

NNE

	0	1	2	3
0	0.79	0.11	0.1	0
1	0	0.91	0	0.09
2	0	0	1.0	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.71	0.21	0.01	0.07
1	0	0.82	0	0.18
2	0.11	0	0.89	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.29	0.41	0.07	0.23
1	0	0.82	0	0.18
2	0.01	0.02	0.95	0.02
3	0	0.18	0	0.82

DThte

	0	1	2	3
0	0.46	0.24	0.19	0.11
1	0	0.91	0	0.09
2	0.03	0	0.97	0
3	0	0.12	0	0.88

SVMhte

	0	1	2	3
0	0.78	0.15	0.02	0.05
1	0	0.91	0	0.09
2	0.04	0	0.96	0
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.72	0.10	0.17	0.01
1	0	0.91	0	0.09
2	0	0	1.00	0
3	0	0.06	0	0.94

HTEsm

	0	1	2	3
0	0.74	0.16	0.08	0.02
1	0	0.92	0	0.08
2	0.01	0	0.99	0
3	0	0.06	0	**1.00**

HTEdf

	0	1	2	3
0	0.75	0.17	0.06	0.02
1	0	**0.94**	0	0.06
2	0.02	0	0.98	0
3	0	0	0	**1.00**

15-85%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.43	0.35	0.05	0.17
1	0	0.82	0.09	0.09
2	0.08	0.04	0.86	0.02
3	0	0.18	0	0.82

DTE

	0	1	2	3
0	0.7	0.18	0.08	0.04
1	0	0.91	0	0.09
2	0.09	0	0.91	0
3	0	0	0	**1.00**

RF

	0	1	2	3
0	0.66	0.20	0.10	0.04
1	0	0.91	0	0.09
2	0.02	0.01	0.97	0
3	0	0.12	0	0.88

SVME

	0	1	2	3
0	0.74	0.08	0.18	0
1	0	0.82	0.09	0.09
2	0.02	0	0.98	0
3	0.06	0.06	0	0.88

NNE

	0	1	2	3
0	0.85	0.10	0.05	0
1	0	0.91	0	0.09
2	0.01	0	0.99	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.36	0.35	0.07	0.22
1	0	0.82	0.09	0.09
2	0.07	0.03	0.88	0.02
3	0	0.12	0	0.88

DThte

	0	1	2	3
0	0.77	0.17	0.02	0.04
1	0	0.91	0	0.09
2	0.05	0	0.95	0
3	0	0.06	0	0.94

SVMhte

	0	1	2	3
0	0.78	0.15	0.01	0.06
1	0	0.91	0	0.09
2	0.08	0	0.92	0
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.87	0.07	0.06	0
1	0	0.91	0	0.09
2	0.01	0	0.99	0
3	0.06	0.06	0	0.88

HTEsm

	0	1	2	3
0	0.77	0.17	0.02	0.04
1	0	0.92	0	0.08
2	0.04	0	0.96	0
3	0	0	0	**1.00**

HTEdf

	0	1	2	3
0	0.80	0.14	0.02	0.04
1	0	**0.94**	0	0.06
2	0.01	0	0.99	0
3	0	0	0	**1.00**

20-80%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.48	0.35	0.01	0.16
1	0	0.82	0	0.18
2	0.08	0.02	0.88	0.02
3	0	0.24	0	0.76

DTE

	0	1	2	3
0	0.75	0.17	0.05	0.03
1	0	0.91	0	0.09
2	0.05	0	0.95	0
3	0	0	0	**1.00**

RF

	0	1	2	3
0	0.71	0.19	0.01	0.09
1	0	0.82	0	0.18
2	0.08	0	0.92	0
3	0	0.06	0	0.94

SVME

	0	1	2	3
0	0.84	0.06	0.10	0
1	0.09	0.82	0	0.09
2	0.03	0	0.97	0
3	0.12	0.12	0	0.76

NNE

	0	1	2	3
0	0.94	0.04	0.02	0
1	0	0.91	0	0.09
2	0.01	0	0.99	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.43	0.34	0.01	0.22
1	0	0.82	0	0.18
2	0.09	0.02	0.88	0.01
3	0	0.29	0	0.71

DThte

	0	1	2	3
0	0.77	0.17	0.04	0.02
1	0	0.91	0	0.09
2	0.04	0	0.96	0
3	0.06	0.06	0	0.88

SVMhte

	0	1	2	3
0	0.80	0.14	0	0.06
1	0	0.91	0	0.09
2	0.11	0	0.89	0
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.94	0.04	0.02	0
1	0	0.91	0	0.09
2	0	0	1.00	0
3	0	0	0	**1.00**

HTEsm

	0	1	2	3
0	0.78	0.18	0	0.04
1	0	0.92	0	0.08
2	0.03	0	0.97	0
3	0	0	0	**1.00**

HTEdf

	0	1	2	3
0	0.84	0.12	0	0.04
1	0	**0.94**	0	0.06
2	0.02	0	0.98	0
3	0	0	0	**1.00**

25-75%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.48	0.34	0	0.18
1	0	0.82	0	0.18
2	0.09	0.04	0.85	0.02
3	0	0.18	0	0.82

DTE

	0	1	2	3
0	0.84	0.11	0.03	0.02
1	0	0.91	0	0.09
2	0.08	0.02	0.87	0.03
3	0	0.18	0	0.82

RF

	0	1	2	3
0	0.70	0.17	0.02	0.11
1	0.09	0.82	0	0.09
2	0.06	0	0.94	0
3	0	0.06	0	0.94

SVME

	0	1	2	3
0	0.89	0.05	0.06	0
1	0.09	0.82	0	0.09
2	0.03	0	0.97	0
3	0.12	0.12	0	0.76

NNE

	0	1	2	3
0	0.92	0.06	0.01	0.01
1	0	0.91	0	0.09
2	0.01	0	0.99	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.46	0.33	0.01	0.20
1	0	0.82	0	0.18
2	0.1	0.03	0.85	0.02
3	0	0.12	0	0.88

DThte

	0	1	2	3
0	0.89	0.11	0	0
1	0	0.91	0	0.09
2	0.03	0.01	0.93	0.03
3	0	0.06	0	0.94

SVMhte

	0	1	2	3
0	0.78	0.15	0	0.07
1	0	0.91	0	0.09
2	0.07	0	0.93	0
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.92	0.06	0.01	0.01
1	0	0.91	0	0.09
2	0	0	0.99	0.01
3	0	0.06	0	0.94

HTEsm

	0	1	2	3
0	0.88	0.08	0	0.04
1	0	0.92	0	0.08
2	0.05	0	0.95	0
3	0	0.06	0	0.94

HTEdf

	0	1	2	3
0	0.89	0.10	0	0.01
1	0	**0.94**	0	0.06
2	0.02	0	0.98	0
3	0	0	0	**1.00**

30-70%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.48	0.34	0.01	0.17
1	0	0.82	0	0.18
2	0.13	0.03	0.82	0.02
3	0	0.29	0	0.71

DTE

	0	1	2	3
0	0.84	0.11	0.03	0.02
1	0	0.91	0	0.09
2	0.09	0	0.91	0
3	0	0	0	**1.00**

RF

	0	1	2	3
0	0.80	0.14	0	0.06
1	0	0.82	0	0.18
2	0.13	0.01	0.86	0
3	0.06	0.12	0	0.82

SVME

	0	1	2	3
0	0.89	0.05	0.06	0
1	0.09	0.82	0	0.09
2	0.04	0	0.96	0
3	0.12	0.12	0	0.76

NNE

	0	1	2	3
0	0.91	0.07	0.01	0.01
1	0	0.91	0	0.09
2	0.03	0	0.97	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.47	0.31	0.01	0.21
1	0	0.82	0	0.18
2	0.12	0.02	0.83	0.03
3	0	0.24	0	0.76

DThte

	0	1	2	3
0	0.88	0.11	0.01	0
1	0	0.91	0	0.09
2	0.07	0	0.93	0
3	0	0	0	**1.00**

SVMhte

	0	1	2	3
0	0.78	0.15	0	0.07
1	0	0.91	0	0.09
2	0.10	0	0.9	0
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.93	0.05	0.02	0
1	0	0.91	0	0.09
2	0.02	0	0.97	0.01
3	0	0	0	**1.00**

HTEsm

	0	1	2	3
0	0.88	0.08	0	0.04
1	0	0.92	0	0.08
2	0.11	0	0.89	0
3	0	0	0	**1.00**

HTEdf

	0	1	2	3
0	0.89	0.10	0	0.01
1	0	0.95	0	0.05
2	0.06	0	0.94	0
3	0	0	0	**1.00**

35-65%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.53	0.33	0	0.14
1	0.09	0.73	0	0.18
2	0.17	0.03	0.78	0.02
3	0.06	0.29	0	0.65

DTE

	0	1	2	3
0	0.87	0.11	0	0.02
1	0	0.91	0	0.09
2	0.08	0	0.92	0
3	0	0	0	**1.00**

RF

	0	1	2	3
0	0.87	0.11	0	0.02
1	0	0.82	0	0.18
2	0.12	0.02	0.84	0.02
3	0	0.06	0	0.94

SVME

	0	1	2	3
0	0.90	0.06	0.04	0
1	0.09	0.82	0	0.09
2	0.06	0	0.94	0
3	0.12	0.06	0	0.82

NNE

	0	1	2	3
0	0.92	0.06	0.02	0
1	0	0.91	0	0.09
2	0.04	0.01	0.95	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.53	0.29	0.01	0.17
1	0	0.73	0	0.27
2	0.15	0.02	0.80	0.03
3	0	0.29	0	0.71

DThte

	0	1	2	3
0	0.90	0.10	0	0
1	0.18	0.82	0	0
2	0.08	0	0.92	0
3	0.06	0	0	0.94

SVMhte

	0	1	2	3
0	0.81	0.14	0	0.05
1	0	0.82	0	0.18
2	0.11	0	0.88	0.01
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.89	0.07	0.04	0
1	0	0.91	0	0.09
2	0.03	0	0.96	0.01
3	0	0	0	**1.00**

HTEsm

	0	1	2	3
0	0.9	0.09	0	0.01
1	0	0.92	0	0.08
2	0.09	0	0.91	0
3	0	0	0	**1.00**

HTEdf

	0	1	2	3
0	0.92	0.08	0	0
1	0	**0.95**	0	0.05
2	0.07	0	0.93	0
3	0	0	0	**1.00**

40-60%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.64	0.28	0	0.08
1	0	0.91	0	0.09
2	0.22	0.02	0.74	0.02
3	0.12	0.23	0	0.65

DTE

	0	1	2	3
0	0.88	0.07	0.03	0.02
1	0	0.91	0	0.09
2	0.07	0	0.93	0
3	0	0	0	**1.00**

RF

	0	1	2	3
0	0.88	0.10	0.01	0.01
1	0	0.91	0	0.09
2	0.18	0.02	0.80	0
3	0	0	0	**1.00**

SVME

	0	1	2	3
0	0.88	0.06	0.06	0
1	0.09	0.82	0	0.09
2	0.04	0	0.96	0
3	0.06	0.06	0	0.88

NNE

	0	1	2	3
0	0.96	0.03	0.01	0
1	0	0.91	0	0.09
2	0.02	0	0.98	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.65	0.25	0	0.10
1	0	0.82	0	0.18
2	0.23	0.02	0.73	0.02
3	0.06	0.29	0	0.65

DThte

	0	1	2	3
0	0.87	0.12	0.01	0
1	0.09	0.91	0	0
2	0.06	0	0.94	0
3	0.06	0.12	0	0.82

SVMhte

	0	1	2	3
0	0.81	0.14	0	0.05
1	0	0.82	0	0.18
2	0.11	0	0.89	0
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.95	0.04	0.01	0
1	0	0.91	0	0.09
2	0.04	0	0.96	0
3	0	0	0	**1.00**

HTEsm

	0	1	2	3
0	0.88	0.09	0.01	0.02
1	0	0.92	0	0.08
2	0.09	0	0.91	0
3	0	0	0	**1.00**

HTEdf

	0	1	2	3
0	0.90	0.09	0	0.01
1	0	**0.95**	0	0.05
2	0.06	0	0.94	0
3	0	0	0	**1.00**

45-55%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.65	0.24	0	0.11
1	0.09	0.73	0	0.18
2	0.23	0.02	0.73	0.02
3	0	0.24	0	0.76

DTE

	0	1	2	3
0	0.88	0.09	0.01	0.02
1	0	0.91	0	0.09
2	0.09	0	0.91	0
3	0.06	0.12	0	0.82

RF

	0	1	2	3
0	0.89	0.11	0	0
1	0	0.91	0	0.09
2	0.29	0.01	0.68	0.02
3	0	0.06	0	0.94

SVME

	0	1	2	3
0	0.94	0.06	0	0
1	0.09	0.82	0	0.09
2	0.05	0	0.95	0
3	0.06	0.06	0	0.88

NNE

	0	1	2	3
0	0.94	0.06	0	0
1	0	0.91	0	0.09
2	0.01	0	0.99	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.67	0.22	0	0.11
1	0.18	0.64	0	0.18
2	0.23	0.03	0.72	0.02
3	0	0.18	0	0.82

DThte

	0	1	2	3
0	0.90	0.09	0	0.01
1	0.09	0.82	0	0.09
2	0.06	0	0.94	0
3	0.06	0.12	0	0.82

SVMhte

	0	1	2	3
0	0.83	0.15	0	0.02
1	0	0.82	0	0.18
2	0.11	0	0.89	0
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.93	0.07	0	0
1	0	0.91	0	0.09
2	0	0	1.0	0
3	0	0	0	**1.00**

HTEsm

	0	1	2	3
0	0.9	0.09	0	0.01
1	0	0.91	0	0.09
2	0.08	0	0.92	0
3	0.06	0	0	**1.00**

HTEdf

	0	1	2	3
0	0.90	0.09	0	0.01
1	0	**0.92**	0	0.08
2	0.08	0	0.92	0
3	0	0	0	**1.00**

50-50%

NBE

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNE

	0	1	2	3
0	0.77	0.18	0	0.05
1	0.18	0.64	0	0.18
2	0.22	0.03	0.73	0.02
3	0.24	0.23	0	0.53

DTE

	0	1	2	3
0	0.88	0.10	0	0.02
1	0	0.91	0	0.09
2	0.14	0	0.86	0
3	0.06	0.12	0	0.82

RF

	0	1	2	3
0	0.92	0.06	0	0.02
1	0	0.82	0	0.18
2	0.21	0.01	0.77	0.01
3	0	0	0	**1.00**

SVME

	0	1	2	3
0	0.93	0.06	0.01	0
1	0.18	0.73	0	0.09
2	0.07	0	0.93	0
3	0.06	0.06	0	0.88

NNE

	0	1	2	3
0	0.98	0.02	0	0
1	0	0.91	0	0.09
2	0.05	0	0.95	0
3	0	0	0	**1.00**

NBhte

	0	1	2	3
0	0.72	0.21	0	0.07
1	0	0.82	0	0.18
2	0.18	0	0.82	0
3	0	0	0	**1.00**

*k*NNhte

	0	1	2	3
0	0.77	0.17	0	0.06
1	0.09	0.73	0	0.18
2	0.25	0.03	0.7	0.02
3	0.12	0.23	0	0.65

DThte

	0	1	2	3
0	0.94	0.06	0	0
1	0	0.91	0	0.09
2	0.15	0	0.85	0
3	0.06	0	0	0.94

SVMhte

	0	1	2	3
0	0.83	0.15	0	0.02
1	0	0.82	0	0.18
2	0.14	0.01	0.85	0
3	0	0	0	**1.00**

NNhte

	0	1	2	3
0	0.95	0.03	0.02	0
1	0	0.91	0	0.09
2	0.05	0	0.94	0.01
3	0.06	0	0	0.94

HTEsm

	0	1	2	3
0	0.87	0.11	0	0.02
1	0	0.91	0	0.09
2	0.12	0	0.88	0
3	0.06	0	0	0.94

HTEdf

	0	1	2	3
0	0.90	0.09	0	0.01
1	0	**0.92**	0	0.08
2	0.13	0	0.87	0
3	0	0	0	**1.00**

White Wine Dataset

Table A.13: Ensemble Performance on Skewed Class Distributions for White Wine Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.427	0.427	0.427	0.427	0.427	0.427	0.427	0.427	0.427
	Training accuracy	0.479	0.478	0.485	0.490	0.498	0.517	0.539	0.568	0.560
	GF	1.100	1.098	1.113	1.124	1.141	1.186	1.243	1.326	1.302
	F1-Score	0.25	0.26	0.26	0.26	0.25	0.25	0.25	0.25	0.24
*k*NNE	Testing accuracy	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469
	Training accuracy	0.553	0.562	0.587	0.597	0.620	0.637	0.658	0.682	0.703
	GF	1.188	1.212	1.286	1.318	1.397	1.463	1.553	1.670	1.788
	F1-Score	0.37	0.37	0.40	0.37	0.38	0.37	0.38	0.37	0.35
DTE	Testing accuracy	0.476	0.475	0.473	0.477	0.473	0.475	0.473	0.476	0.474
	Training accuracy	0.529	0.569	0.578	0.616	0.636	0.658	0.679	0.707	0.722
	GF	1.113	1.218	1.249	1.362	1.448	1.535	1.642	1.788	1.892
	F1-Score	0.38	0.35	0.37	0.37	0.40	0.37	0.34	0.34	0.36
RF	Testing accuracy	0.519	0.521	0.527	0.521	0.533	0.522	0.514	0.524	0.524
	Training accuracy	0.604	0.615	0.636	0.653	0.668	0.688	0.713	0.730	0.742
	GF	1.215	1.244	1.299	1.380	1.407	1.532	1.693	1.763	1.845
	F1-Score	0.40	0.44	0.40	0.39	0.39	0.39	0.40	0.39	0.35
SVME	Testing accuracy	0.506	0.504	0.504	0.506	0.506	0.507	0.507	0.507	0.506
	Training accuracy	0.544	0.545	0.568	0.587	0.631	0.668	0.693	0.720	0.740
	GF	1.083	1.090	1.148	1.196	1.339	1.485	1.606	1.761	1.900
	F1-Score	0.37	0.37	0.37	0.37	0.36	0.34	0.35	0.36	0.37
NNE	Testing accuracy	0.517	0.510	0.519	0.520	0.516	0.517	0.516	0.513	0.512
	Training accuracy	0.635	0.641	0.656	0.679	0.693	0.707	0.726	0.737	0.756
	GF	1.323	1.365	1.398	1.495	1.577	1.648	1.766	1.852	2.000
	F1-Score	0.45	0.45	0.45	0.43	0.44	0.41	0.42	0.43	0.38

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.450	0.450	0.450	0.450	0.450	0.450	0.450	0.450	0.450
	Training accuracy	0.478	0.478	0.488	0.499	0.519	0.537	0.569	0.593	0.593
	GF	1.054	1.054	1.074	1.098	1.143	1.188	1.276	1.351	1.351
	F1-Score	0.24	0.26	0.25	0.26	0.25	0.23	0.22	0.21	0.19
*k*NNhte	Testing accuracy	0.491	0.491	0.491	0.491	0.491	0.491	0.491	0.491	0.491
	Training accuracy	0.564	0.571	0.593	0.606	0.629	0.643	0.669	0.686	0.701
	GF	1.167	1.186	1.251	1.292	1.372	1.426	1.538	1.621	1.702
	F1-Score	0.39	0.40	0.39	0.38	0.37	0.40	0.42	0.39	0.36
DThte	Testing accuracy	0.528	0.520	0.522	0.525	0.510	0.521	0.522	0.522	0.515
	Training accuracy	0.598	0.622	0.641	0.654	0.661	0.689	0.705	0.728	0.739
	GF	1.174	1.270	1.331	1.373	1.445	1.540	1.620	1.757	1.858
	F1-Score	0.44	0.39	0.42	0.42	0.41	0.38	0.37	0.35	0.36
SVMhte	Testing accuracy	0.507	0.507	0.506	0.506	0.506	0.505	0.507	0.507	0.508
	Training accuracy	0.500	0.502	0.489	0.520	0.569	0.595	0.613	0.633	0.646
	GF	0.986	0.990	0.967	1.029	1.146	1.222	1.274	1.343	1.390
	F1-Score	0.36	0.35	0.35	0.31	0.28	0.24	0.23	0.18	0.15
NNhte	Testing accuracy	0.533	0.529	0.528	0.528	0.527	0.529	0.522	0.528	0.533
	Training accuracy	0.606	0.626	0.635	0.654	0.669	0.684	0.691	0.700	0.716
	GF	1.185	1.259	1.293	1.364	1.429	1.491	1.528	1.573	1.644
	F1-Score	0.42	0.43	0.46	0.42	0.42	0.40	0.46	0.32	0.40
HTEsm	Testing accuracy	0.514	0.522	0.520	0.516	0.520	0.523	0.520	0.520	0.519
	Training accuracy	0.606	0.615	0.632	0.656	0.668	0.683	0.707	0.722	0.739
	GF	1.234	1.242	1.304	1.407	1.446	1.505	1.638	1.727	1.843
	F1-Score	0.47	0.47	0.47	0.48	0.48	0.48	0.48	0.48	0.48
HTEdf	Testing accuracy	0.534	0.532	0.533	0.530	0.530	0.532	0.524	0.533	0.534
	Training accuracy	0.605	0.615	0.631	0.645	0.662	0.679	0.697	0.712	0.728
	GF	1.180	1.216	1.266	1.324	1.391	1.458	1.571	1.622	1.713
	F1-Score	0.48	0.48	0.48	0.48	0.48	0.49	0.49	0.48	0.48

Table A.14: Confusion Matrices of Ensembles on Skewed Class Distributions for White Wine Dataset

10-90%

NBE

	3	4	5	6	7	8	9
3	**0.40**	0	0.60	0	0	0	0
4	0.04	0.44	0.24	0.04	0.16	0.08	0
5	0.05	0.19	0.47	0.08	0.15	0.06	0
6	0.02	0.08	0.27	0.07	0.32	0.23	0.01
7	0.02	0.03	0.12	0.06	0.24	0.50	0.03
8	0.03	0	0.06	0	0.34	0.49	0.08
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0.20	0.60	0	0	0	0
4	0.08	0.68	0.12	0.12	0	0	0
5	0.07	0.20	0.45	0.19	0.05	0.04	0
6	0.04	0.13	0.28	0.22	0.20	0.12	0.01
7	0.02	0.06	0.07	0.19	0.42	0.23	0.01
8	0.03	0	0.06	0.09	0.31	0.43	0.08
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0	0	0.60	0.40	0	0	0
4	0	0.28	0.36	0.16	0.16	0.04	0
5	0.05	0.17	0.41	0.27	0.07	0.03	0
6	0.03	0.07	0.21	0.33	0.20	0.15	0.01
7	0.04	0.04	0.07	0.24	0.31	0.28	0.02
8	0	0.03	0.06	0.17	0.31	0.40	0.03
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0	0	0.60	0.20	0.20	0	0
4	0.12	0.52	0.20	0.16	0	0	0
5	0.03	0.19	0.50	0.17	0.08	0.03	0
6	0.01	0.09	0.28	0.26	0.21	0.14	0.01
7	0.02	0.04	0.07	0.18	0.44	0.25	0
8	0	0	0.08	0.20	0.23	0.46	0.03
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.12	0.60	0.16	0.08	0.04	0	0
5	0.04	0.15	0.55	0.14	0.08	0.03	0.01
6	0.01	0.05	0.32	0.23	0.17	0.21	0.01
7	0	0.03	0.11	0.15	0.27	0.42	0.02
8	0	0.03	0.06	0.06	0.26	0.51	0.08
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.40	0	0	0
4	0.12	0.48	0.28	0.08	0.04	0	0
5	0.03	0.16	0.55	0.16	0.07	0.03	0
6	0.02	0.07	0.25	0.31	0.22	0.12	0.01
7	0	0.02	0.06	0.16	0.47	0.29	0
8	0	0	0.03	0.06	0.37	0.51	0.03
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.60	0	0	0	0
4	0.04	0.44	0.24	0.04	0.16	0.08	0
5	0.04	0.20	0.48	0.07	0.15	0.06	0
6	0.02	0.08	0.27	0.07	0.32	0.23	0.01
7	0.01	0.03	0.12	0.06	0.25	0.50	0.03
8	0	0.03	0.06	0	0.34	0.49	0.08
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0.20	0.60	0	0	0	0
4	0.08	0.64	0.12	0.12	0	0.04	0
5	0.06	0.17	0.48	0.19	0.06	0.04	0
6	0.03	0.09	0.25	0.24	0.23	0.14	0.02
7	0.02	0.03	0.06	0.13	0.43	0.31	0.02
8	0.03	0	0.03	0.06	0.31	0.49	0.08
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.20	0.20	0	0
4	0.08	0.44	0.32	0.08	0.04	0.04	0
5	0.02	0.16	0.48	0.21	0.09	0.04	0
6	0.01	0.07	0.22	0.31	0.22	0.16	0.01
7	0.01	0.03	0.04	0.14	0.50	0.28	0
8	0	0	0.03	0.03	0.28	0.63	0.03
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.04	0.60	0.20	0.08	0.08	0	0
5	0.06	0.16	0.57	0.10	0.07	0.03	0.01
6	0.02	0.06	0.33	0.22	0.15	0.20	0.02
7	0.02	0.03	0.13	0.14	0.22	0.41	0.05
8	0	0.03	0.06	0.11	0.20	0.49	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.20	0.20	0	0
4	0.08	0.64	0.12	0.12	0.04	0	0
5	0.01	0.18	0.48	0.20	0.11	0.02	0
6	0.02	0.07	0.22	0.26	0.34	0.08	0.01
7	0.01	0.03	0.04	0.11	0.59	0.22	0
8	0	0	0	0.09	0.31	0.54	0.06
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.12	0.52	0.20	0.08	0.08	0	0
5	0.03	0.17	0.55	0.15	0.06	0.04	0
6	0.02	0.06	0.26	0.24	0.23	0.19	0
7	0.01	0.02	0.04	0.16	0.35	0.41	0.01
8	0	0	0.09	0	0.31	0.54	0.06
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.08	0.56	0.24	0	0.12	0	0
5	0.01	0.16	0.59	0.12	0.07	0.05	0
6	0.01	0.05	0.29	0.21	0.24	0.19	0.01
7	0.01	0.03	0.04	0.09	0.40	0.43	0
8	0	0	0.08	0.03	0.26	0.57	0.06
9	0	0	0	0	0	0	0

15-85%

NBE

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.08	0.40	0.24	0.04	0.16	0.08	0
5	0.07	0.18	0.44	0.11	0.15	0.05	0
6	0.03	0.07	0.24	0.10	0.33	0.22	0.01
7	0.02	0.02	0.10	0.08	0.27	0.48	0.03
8	0.03	0	0.06	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0	0.60	0.20	0	0	0
4	0.08	0.64	0.08	0.12	0.04	0.04	0
5	0.07	0.25	0.46	0.13	0.07	0.02	0
6	0.05	0.13	0.29	0.22	0.19	0.11	0.01
7	0.03	0.04	0.09	0.18	0.43	0.21	0.02
8	0.06	0.03	0.03	0.11	0.26	0.43	0.08
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0.20	0	0	0.60	0.20	0	0
4	0.20	0.36	0.28	0.04	0.08	0.04	0
5	0.06	0.13	0.40	0.29	0.09	0.03	0
6	0.05	0.06	0.29	0.26	0.21	0.12	0.01
7	0.02	0.05	0.11	0.20	0.33	0.26	0.03
8	0.03	0.03	0.03	0.14	0.26	0.48	0.03
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.40	0	0	0
4	0.08	0.60	0.20	0.08	0.04	0	0
5	0.03	0.18	0.52	0.19	0.06	0.02	0
6	0.03	0.08	0.26	0.31	0.19	0.13	0
7	0.01	0.06	0.09	0.17	0.46	0.21	0
8	0	0.03	0.06	0.05	0.20	0.63	0.03
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	**0.40**	0	0.20	0.40	0	0	0
4	0.20	0.48	0.20	0.08	0.04	0	0
5	0.06	0.15	0.56	0.11	0.09	0.02	0.01
6	0.04	0.04	0.30	0.22	0.18	0.21	0.01
7	0.02	0.02	0.12	0.13	0.29	0.41	0.01
8	0.03	0	0.06	0.08	0.20	0.57	0.06
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	**0.40**	0	0.20	0.20	0.20	0	0
4	0.08	0.64	0.12	0.12	0.04	0	0
5	0.03	0.18	0.48	0.19	0.09	0.03	0
6	0.02	0.06	0.22	0.32	0.29	0.09	0
7	0	0.03	0.05	0.11	0.57	0.24	0
8	0	0	0.03	0.03	0.46	0.43	0.05
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.60	0	0	0	0
4	0.08	0.40	0.24	0.04	0.16	0.08	0
5	0.07	0.18	0.44	0.11	0.15	0.05	0
6	0.03	0.07	0.24	0.10	0.33	0.22	0.01
7	0.02	0.02	0.10	0.08	0.27	0.48	0.03
8	0.03	0	0.06	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0	0.60	0.20	0	0	0
4	0.08	0.56	0.20	0.08	0.04	0.04	0
5	0.07	0.18	0.50	0.13	0.08	0.04	0
6	0.04	0.08	0.28	0.23	0.20	0.15	0.02
7	0.02	0.02	0.07	0.14	0.47	0.26	0.02
8	0.06	0	0.03	0.06	0.31	0.46	0.08
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0.20	0	0.20	0.40	0.20	0	0
4	0.12	0.48	0.24	0.08	0.08	0	0
5	0.03	0.17	0.44	0.27	0.07	0.02	0
6	0.03	0.06	0.24	0.29	0.23	0.14	0.01
7	0	0.03	0.05	0.21	0.40	0.30	0.01
8	0	0	0.03	0.09	0.17	0.68	0.03
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.12	0.44	0.20	0.08	0.12	0.04	0
5	0.08	0.14	0.56	0.09	0.07	0.04	0.02
6	0.04	0.05	0.32	0.19	0.16	0.22	0.02
7	0.03	0.02	0.12	0.12	0.26	0.40	0.05
8	0.03	0	0.06	0.09	0.20	0.51	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.20	0.20	0.20	0	0
4	0.12	0.48	0.20	0.12	0.04	0.04	0
5	0.03	0.13	0.60	0.13	0.06	0.04	0.01
6	0.04	0.03	0.27	0.29	0.18	0.18	0.01
7	0.01	0.02	0.08	0.11	0.35	0.42	0.01
8	0.03	0	0.03	0.06	0.11	0.71	0.06
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.12	0.52	0.20	0.08	0.08	0	0
5	0.05	0.16	0.51	0.16	0.08	0.04	0
6	0.02	0.05	0.27	0.21	0.25	0.19	0.01
7	0.01	0.02	0.08	0.11	0.38	0.39	0.01
8	0	0	0.06	0.06	0.23	0.57	0.08
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.12	0.52	0.20	0.12	0	0.04	0
5	0.06	0.13	0.52	0.16	0.09	0.04	0
6	0.02	0.06	0.27	0.20	0.27	0.17	0.01
7	0	0.01	0.04	0.11	0.44	0.38	0.02
8	0	0	0.03	0.03	0.20	0.66	0.08
9	0	0	0	0	0	0	0

20-80%

NBE

	3	4	5	6	7	8	9
3	**0.40**	0	0.60	0	0	0	0
4	0.08	0.44	0.24	0.08	0.08	0.08	0
5	0.10	0.18	0.42	0.09	0.16	0.05	0
6	0.04	0.07	0.24	0.09	0.32	0.22	0.02
7	0.02	0.03	0.10	0.07	0.26	0.49	0.03
8	0.03	0	0.06	0	0.34	0.51	0.06
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0.20	0.40	0	0.20	0	0
4	0.08	0.60	0.20	0.12	0	0	0
5	0.08	0.21	0.47	0.15	0.05	0.04	0
6	0.06	0.14	0.25	0.26	0.16	0.11	0.02
7	0.02	0.06	0.06	0.22	0.42	0.20	0.02
8	0.03	0	0.06	0.11	0.40	0.31	0.09
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.40	0	0	0
4	0.16	0.44	0.16	0.12	0.08	0.04	0
5	0.06	0.18	0.41	0.20	0.11	0.04	0
6	0.03	0.12	0.22	0.28	0.21	0.13	0.01
7	0.02	0.06	0.07	0.24	0.34	0.26	0.01
8	0	0	0.03	0.26	0.17	0.54	0
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0	0.40	0.40	0.20	0	0	0
4	0.16	0.56	0.12	0.04	0.12	0	0
5	0.06	0.17	0.48	0.19	0.07	0.03	0
6	0.03	0.11	0.25	0.26	0.21	0.14	0
7	0.01	0.04	0.07	0.17	0.47	0.24	0
8	0	0	0.06	0.14	0.28	0.49	0.03
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.20	0.48	0.16	0.12	0.04	0	0
5	0.12	0.14	0.50	0.14	0.06	0.03	0.01
6	0.07	0.05	0.26	0.26	0.15	0.20	0.01
7	0.05	0.02	0.11	0.16	0.22	0.43	0.01
8	0.06	0	0.06	0.09	0.14	0.57	0.08
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.20	0.20	0	0
4	0.08	0.56	0.24	0.08	0.04	0	0
5	0.02	0.16	0.51	0.20	0.07	0.04	0
6	0.02	0.06	0.22	0.35	0.23	0.12	0
7	0.01	0.02	0.03	0.19	0.43	0.32	0
8	0	0	0.03	0.12	0.31	0.54	0
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.60	0	0	0	0
4	0.08	0.44	0.24	0.08	0.08	0.08	0
5	0.10	0.18	0.41	0.09	0.16	0.05	0.01
6	0.05	0.07	0.23	0.09	0.32	0.22	0.02
7	0.03	0.03	0.10	0.07	0.26	0.48	0.03
8	0.03	0	0.06	0	0.34	0.49	0.08
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0	0.60	0.20	0	0	0
4	0.08	0.52	0.20	0.08	0.08	0.04	0
5	0.07	0.16	0.48	0.18	0.07	0.04	0
6	0.04	0.09	0.26	0.23	0.22	0.14	0.02
7	0.02	0.05	0.04	0.11	0.47	0.29	0.02
8	0.06	0	0.03	0.03	0.40	0.40	0.08
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.40	0	0	0
4	0.08	0.56	0.16	0.12	0.04	0.04	0
5	0.04	0.18	0.45	0.21	0.09	0.03	0
6	0.02	0.10	0.20	0.32	0.21	0.14	0.01
7	0.01	0.04	0.03	0.18	0.39	0.34	0.01
8	0	0	0.06	0.08	0.20	0.63	0.03
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.16	0.44	0.20	0.08	0.12	0	0
5	0.17	0.12	0.50	0.10	0.06	0.04	0.01
6	0.12	0.04	0.28	0.21	0.14	0.20	0.01
7	0.04	0.03	0.11	0.16	0.21	0.40	0.05
8	0.03	0	0.06	0.11	0.20	0.49	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	0.20	0	0.20	0.60	0	0	0
4	0.12	0.56	0.12	0.20	0	0	0
5	0.03	0.12	0.40	0.41	0.01	0.03	0
6	0.02	0.05	0.16	0.49	0.15	0.12	0.01
7	0.01	0.02	0.03	0.28	0.36	0.30	0
8	0	0.03	0	0.17	0.23	0.51	0.06
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.12	0.48	0.28	0.12	0	0	0
5	0.05	0.16	0.55	0.12	0.09	0.02	0.01
6	0.03	0.06	0.26	0.24	0.25	0.15	0.01
7	0.01	0.02	0.04	0.15	0.41	0.35	0.02
8	0	0	0.03	0.11	0.20	0.57	0.09
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.12	0.52	0.28	0.04	0.04	0	0
5	0.05	0.13	0.57	0.12	0.09	0.03	0.01
6	0.02	0.05	0.26	0.25	0.26	0.15	0.01
7	0	0.02	0.06	0.11	0.41	0.38	0.02
8	0	0	0.09	0	0.28	0.57	0.06
9	0	0	0	0	0	0	0

25-75%

NBE

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.08	0.44	0.24	0.04	0.12	0.08	0
5	0.10	0.19	0.38	0.11	0.14	0.08	0
6	0.05	0.06	0.20	0.12	0.31	0.24	0.02
7	0.03	0.03	0.07	0.09	0.25	0.50	0.03
8	0.03	0	0.06	0	0.34	0.49	0.08
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0.20	0.40	0.20	0	0	0
4	0.08	0.64	0.12	0.08	0.04	0.04	0
5	0.08	0.21	0.45	0.16	0.06	0.04	0
6	0.05	0.12	0.28	0.22	0.19	0.12	0.02
7	0.03	0.05	0.07	0.17	0.45	0.21	0.02
8	0.06	0	0.06	0.11	0.34	0.34	0.09
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0.20	0	0.20	0.40	0.20	0	0
4	0.16	0.44	0.28	0.08	0.04	0	0
5	0.06	0.18	0.47	0.16	0.08	0.05	0
6	0.05	0.08	0.22	0.24	0.22	0.18	0.01
7	0.01	0.07	0.07	0.15	0.33	0.36	0.01
8	0.03	0	0.06	0.23	0.28	0.40	0
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0.20	0.20	0.40	0.20	0	0	0
4	0.16	0.40	0.24	0.12	0.04	0.04	0
5	0.05	0.21	0.41	0.24	0.06	0.03	0
6	0.05	0.09	0.22	0.28	0.23	0.13	0
7	0.02	0.04	0.08	0.15	0.44	0.27	0
8	0.03	0	0	0.11	0.26	0.54	0.06
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.20	0.48	0.20	0.08	0.04	0	0
5	0.16	0.12	0.48	0.13	0.08	0.02	0.01
6	0.11	0.04	0.22	0.26	0.14	0.22	0.01
7	0.05	0.02	0.09	0.16	0.22	0.44	0.02
8	0.03	0	0.06	0.11	0.17	0.54	0.09
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	0.20	0	0.20	0.60	0	0	0
4	0.12	0.52	0.20	0.08	0.08	0	0
5	0.04	0.10	0.59	0.16	0.09	0.02	0
6	0.02	0.04	0.26	0.26	0.31	0.10	0.01
7	0.01	0.02	0.04	0.13	0.54	0.26	0
8	0	0	0.03	0.05	0.46	0.43	0.03
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.60	0	0	0	0
4	0.16	0.36	0.24	0.04	0.12	0.08	0
5	0.15	0.16	0.36	0.11	0.14	0.08	0
6	0.10	0.05	0.19	0.11	0.30	0.23	0.02
7	0.03	0.02	0.07	0.10	0.25	0.50	0.03
8	0.03	0	0.06	0	0.34	0.49	0.08
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.40	0	0	0
4	0.08	0.60	0.12	0.16	0	0.04	0
5	0.07	0.17	0.47	0.17	0.07	0.05	0
6	0.04	0.07	0.28	0.21	0.21	0.17	0.02
7	0.04	0.05	0.03	0.10	0.50	0.26	0.02
8	0.06	0	0.06	0.06	0.31	0.43	0.08
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0.20	0.20	0.40	0	0.20	0	0
4	0.16	0.56	0.12	0.16	0	0	0
5	0.07	0.18	0.48	0.17	0.07	0.03	0
6	0.04	0.07	0.20	0.29	0.24	0.16	0
7	0.01	0.04	0.05	0.12	0.46	0.31	0.01
8	0.03	0	0.06	0.14	0.14	0.60	0.03
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.40	0.20	0	0	0
4	0.24	0.48	0.16	0.08	0.04	0	0
5	0.28	0.11	0.45	0.05	0.07	0.03	0.01
6	0.18	0.04	0.27	0.15	0.15	0.20	0.01
7	0.11	0.02	0.10	0.10	0.22	0.41	0.04
8	0.12	0	0.06	0.06	0.14	0.51	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	**0.40**	0	0.20	0	0.40	0	0
4	0.12	0.48	0.12	0.12	0.04	0.12	0
5	0.06	0.13	0.47	0.22	0.05	0.06	0.01
6	0.04	0.05	0.21	0.28	0.21	0.20	0.01
7	0.01	0.02	0.05	0.09	0.44	0.39	0
8	0	0	0.03	0.06	0.20	0.66	0.05
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	**0.40**	0	0.20	0.40	0	0	0
4	0.12	0.52	0.24	0.08	0.04	0	0
5	0.05	0.15	0.54	0.15	0.08	0.03	0
6	0.05	0.06	0.22	0.26	0.21	0.19	0.01
7	0.01	0.04	0.03	0.15	0.36	0.40	0.01
8	0	0	0.06	0.06	0.25	0.57	0.06
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.40**	0	0.20	0.40	0	0	0
4	0.12	0.52	0.24	0.04	0.04	0.04	0
5	0.06	0.17	0.51	0.14	0.07	0.04	0.01
6	0.05	0.06	0.23	0.23	0.22	0.20	0.01
7	0.01	0.03	0.02	0.14	0.37	0.42	0.01
8	0.03	0	0.06	0.03	0.28	0.54	0.06
9	0	0	0	0	0	0	0

30-70%

NBE

	3	4	5	6	7	8	9
3	0.43	0	0.57	0	0	0	0
4	0.16	0.36	0.24	0.04	0.12	0.08	0
5	0.12	0.16	0.40	0.10	0.15	0.07	0
6	0.08	0.04	0.23	0.10	0.31	0.23	0.01
7	0.03	0.02	0.08	0.09	0.25	0.50	0.03
8	0.03	0	0.06	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.20	0.20	0	0
4	0.08	0.60	0.16	0	0.12	0.04	0
5	0.09	0.20	0.47	0.15	0.06	0.03	0
6	0.05	0.11	0.29	0.22	0.21	0.10	0.02
7	0.03	0.04	0.06	0.19	0.46	0.21	0.01
8	0.06	0	0.06	0.11	0.34	0.34	0.09
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.40	0	0	0
4	0.20	0.40	0.20	0.08	0.04	0.08	0
5	0.04	0.20	0.41	0.27	0.05	0.03	0
6	0.04	0.10	0.21	0.35	0.18	0.11	0.01
7	0.02	0.06	0.08	0.19	0.33	0.31	0.01
8	0.03	0	0.06	0.20	0.23	0.48	0
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0.20	0	0.60	0.20	0	0	0
4	0.16	0.52	0.16	0.16	0	0	0
5	0.06	0.20	0.43	0.19	0.09	0.03	0
6	0.03	0.11	0.24	0.27	0.18	0.16	0.01
7	0.01	0.04	0.06	0.18	0.43	0.27	0.01
8	0.03	0	0.03	0.11	0.23	0.57	0.03
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.40	0	0	0
4	0.16	0.56	0.16	0.08	0.04	0	0
5	0.18	0.12	0.48	0.11	0.07	0.03	0.01
6	0.12	0.04	0.26	0.21	0.16	0.20	0.01
7	0.06	0.02	0.10	0.12	0.28	0.41	0.01
8	0.06	0	0.03	0.11	0.17	0.54	0.09
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	0.20	0	0.20	0.60	0	0	0
4	0.12	0.52	0.24	0.04	0.08	0	0
5	0.04	0.15	0.55	0.16	0.08	0.02	0
6	0.02	0.06	0.25	0.29	0.28	0.10	0
7	0.02	0.02	0.07	0.13	0.52	0.24	0
8	0	0	0.03	0.03	0.46	0.45	0.03
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	0.40	0	0.60	0	0	0	0
4	0.16	0.40	0.20	0.04	0.12	0.08	0
5	0.19	0.14	0.37	0.10	0.14	0.06	0
6	0.13	0.03	0.21	0.10	0.29	0.22	0.02
7	0.06	0.02	0.07	0.09	0.23	0.50	0.03
8	0.03	0	0.06	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.20	0.20	0	0
4	0.08	0.56	0.20	0.04	0.08	0.04	0
5	0.08	0.14	0.49	0.18	0.07	0.04	0
6	0.04	0.06	0.28	0.21	0.25	0.14	0.02
7	0.02	0.03	0.06	0.13	0.46	0.29	0.01
8	0.06	0	0.06	0.06	0.37	0.37	0.08
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0.20	0	0.60	0.20	0	0	0
4	0.16	0.48	0.16	0.08	0.04	0.08	0
5	0.04	0.17	0.44	0.25	0.07	0.03	0
6	0.04	0.08	0.19	0.31	0.22	0.15	0.01
7	0	0.03	0.04	0.19	0.40	0.33	0.01
8	0.03	0	0.03	0.06	0.25	0.63	0
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	**0.45**	0	0.40	0.15	0	0	0
4	0.28	0.36	0.20	0.12	0.04	0	0
5	0.37	0.08	0.41	0.05	0.05	0.03	0.01
6	0.28	0.02	0.24	0.12	0.13	0.20	0.01
7	0.21	0	0.07	0.09	0.19	0.39	0.05
8	0.2	0	0.06	0.03	0.11	0.49	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.40	0	0	0
4	0.12	0.60	0.16	0.08	0.04	0	0
5	0.11	0.13	0.52	0.15	0.06	0.03	0
6	0.08	0.05	0.26	0.28	0.17	0.14	0.02
7	0.05	0.02	0.04	0.15	0.41	0.32	0.01
8	0.03	0	0.06	0.09	0.17	0.54	0.11
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	0.43	0	0.37	0.20	0	0	0
4	0.16	0.60	0.16	0.04	0.04	0	0
5	0.07	0.15	0.55	0.13	0.07	0.03	0
6	0.05	0.04	0.27	0.23	0.23	0.17	0.01
7	0.02	0.03	0.05	0.09	0.38	0.42	0.01
8	0.03	0	0	0.03	0.37	0.51	0.06
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.45**	0	0.55	0	0	0	0
4	0.16	0.56	0.16	0.04	0.08	0	0
5	0.08	0.15	0.57	0.11	0.04	0.05	0
6	0.05	0.05	0.27	0.21	0.20	0.21	0.01
7	0.03	0.01	0.06	0.08	0.38	0.43	0.01
8	0.06	0	0.03	0	0.28	0.57	0.06
9	0	0	0	0	0	0	0

35-65%

NBE

	3	4	5	6	7	8	9
3	0.40	0	0.40	0.20	0	0	0
4	0.16	0.36	0.24	0.04	0.12	0.08	0
5	0.14	0.15	0.39	0.10	0.15	0.07	0
6	0.10	0.05	0.21	0.10	0.32	0.21	0.01
7	0.04	0.02	0.08	0.09	0.25	0.49	0.03
8	0.03	0	0.06	0	0.28	0.54	0.09
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.20	0.20	0	0
4	0.08	0.72	0.08	0.08	0.04	0	0
5	0.1	0.19	0.44	0.17	0.06	0.03	0.01
6	0.07	0.10	0.28	0.23	0.20	0.10	0.02
7	0.04	0.07	0.07	0.18	0.40	0.22	0.02
8	0.06	0	0.06	0.09	0.34	0.37	0.08
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.40	0	0	0
4	0.16	0.40	0.24	0.08	0.04	0.08	0
5	0.05	0.19	0.40	0.23	0.10	0.03	0
6	0.06	0.12	0.21	0.28	0.20	0.12	0.01
7	0.03	0.10	0.05	0.24	0.39	0.18	0.01
8	0.06	0.03	0.06	0.14	0.28	0.43	0
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.40	0	0	0
4	0.20	0.52	0.20	0.04	0.04	0	0
5	0.07	0.19	0.46	0.20	0.06	0.02	0
6	0.05	0.12	0.25	0.26	0.20	0.11	0.01
7	0.02	0.06	0.05	0.18	0.42	0.27	0
8	0.03	0	0	0.20	0.23	0.51	0.03
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.40	0	0	0
4	0.12	0.52	0.16	0.12	0.08	0	0
5	0.18	0.13	0.47	0.12	0.07	0.02	0.01
6	0.11	0.04	0.26	0.20	0.19	0.19	0.01
7	0.05	0.03	0.10	0.15	0.24	0.42	0.01
8	0.06	0	0.06	0.11	0.14	0.54	0.09
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.20	0.20	0	0
4	0.12	0.52	0.24	0.04	0.08	0	0
5	0.05	0.13	0.48	0.24	0.07	0.03	0
6	0.03	0.06	0.25	0.30	0.23	0.13	0
7	0.03	0.02	0.05	0.15	0.38	0.37	0
8	0	0	0	0.11	0.26	0.63	0
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	0.40	0	0.60	0	0	0	0
4	0.16	0.36	0.24	0.04	0.12	0.08	0
5	0.27	0.13	0.31	0.10	0.13	0.06	0
6	0.18	0.04	0.17	0.09	0.30	0.21	0.01
7	0.09	0.01	0.07	0.08	0.23	0.49	0.03
8	0.03	0	0.06	0	0.28	0.54	0.09
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.20	0.20	0	0
4	0.08	0.60	0.16	0.08	0.08	0	0
5	0.09	0.14	0.50	0.16	0.06	0.05	0
6	0.06	0.07	0.24	0.23	0.22	0.16	0.02
7	0.03	0.02	0.05	0.15	0.44	0.29	0.02
8	0.06	0	0.03	0	0.34	0.49	0.08
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0.20	0.40	0.40	0	0	0	0
4	0.16	0.52	0.16	0.12	0.04	0	0
5	0.08	0.19	0.40	0.22	0.09	0.02	0
6	0.06	0.08	0.24	0.25	0.23	0.13	0.01
7	0.03	0.05	0.05	0.14	0.45	0.28	0
8	0.06	0	0.06	0.11	0.20	0.57	0
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	**0.45**	0	0.40	0.15	0	0	0
4	0.28	0.32	0.24	0.08	0.08	0	0
5	0.42	0.08	0.36	0.06	0.04	0.03	0.01
6	0.32	0.02	0.23	0.10	0.10	0.22	0.01
7	0.24	0.01	0.07	0.10	0.14	0.40	0.04
8	0.2	0	0.06	0.03	0.09	0.51	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.20	0.20	0	0
4	0.12	0.60	0.16	0.04	0.04	0.04	0
5	0.13	0.16	0.49	0.14	0.04	0.04	0
6	0.09	0.05	0.22	0.25	0.21	0.18	0
7	0.05	0.03	0.07	0.10	0.33	0.42	0
8	0.09	0	0	0	0.20	0.71	0
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	0.43	0	0.37	0.20	0	0	0
4	0.16	0.52	0.20	0.08	0.04	0	0
5	0.08	0.19	0.50	0.12	0.08	0.03	0
6	0.06	0.06	0.25	0.21	0.24	0.17	0.01
7	0.01	0.04	0.04	0.15	0.39	0.35	0.02
8	0.03	0	0.06	0.06	0.28	0.51	0.06
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.45**	0	0.55	0	0	0	0
4	0.16	0.52	0.16	0.08	0.04	0.04	0
5	0.10	0.16	0.47	0.15	0.08	0.04	0
6	0.07	0.05	0.25	0.20	0.26	0.16	0.01
7	0.03	0.02	0.03	0.12	0.35	0.44	0.01
8	0.03	0	0.06	0	0.37	0.48	0.06
9	0	0	0	0	0	0	0

40-60%

NBE

	3	4	5	6	7	8	9
3	0.40	0	0.60	0	0	0	0
4	0.16	0.36	0.24	0.04	0.08	0.12	0
5	0.17	0.13	0.36	0.11	0.16	0.07	0
6	0.12	0.04	0.19	0.10	0.31	0.23	0.01
7	0.06	0.01	0.08	0.08	0.24	0.50	0.03
8	0.03	0	0.06	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.20	0.20	0	0
4	0.08	0.68	0.12	0.08	0	0.04	0
5	0.12	0.22	0.44	0.12	0.07	0.03	0
6	0.05	0.12	0.27	0.25	0.20	0.09	0.02
7	0.04	0.05	0.07	0.17	0.43	0.22	0.02
8	0.06	0	0.06	0.14	0.26	0.37	0.11
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0.20	0	0.60	0.20	0	0	0
4	0.12	0.40	0.44	0	0.04	0	0
5	0.02	0.21	0.46	0.16	0.11	0.04	0
6	0.03	0.11	0.31	0.18	0.19	0.17	0.01
7	0.01	0.07	0.11	0.10	0.39	0.29	0.03
8	0	0	0.09	0.20	0.17	0.54	0
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0.20	0.20	0.60	0	0	0	0
4	0.12	0.56	0.08	0.04	0.20	0	0
5	0.06	0.22	0.43	0.18	0.07	0.04	0
6	0.06	0.13	0.20	0.27	0.17	0.16	0.01
7	0.01	0.08	0.04	0.16	0.40	0.31	0
8	0	0.06	0	0.17	0.23	0.54	0
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.40	0	0	0
4	0.16	0.52	0.16	0.08	0.08	0	0
5	0.18	0.13	0.46	0.14	0.05	0.03	0.01
6	0.11	0.04	0.24	0.21	0.17	0.22	0.01
7	0.06	0.02	0.10	0.11	0.26	0.43	0.02
8	0.06	0	0.06	0.11	0.14	0.54	0.09
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.20	0.20	0	0
4	0.12	0.60	0.16	0	0.12	0	0
5	0.05	0.15	0.46	0.22	0.08	0.04	0
6	0.03	0.06	0.21	0.32	0.25	0.13	0
7	0.01	0.02	0.04	0.16	0.41	0.35	0.01
8	0	0	0	0.06	0.37	0.51	0.06
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	0.40	0	0.60	0	0	0	0
4	0.20	0.36	0.24	0.04	0.08	0.08	0
5	0.36	0.11	0.27	0.10	0.10	0.06	0
6	0.25	0.02	0.15	0.08	0.26	0.22	0.02
7	0.11	0.01	0.07	0.06	0.22	0.50	0.03
8	0.03	0	0.06	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.20	0.20	0	0
4	0.12	0.48	0.24	0.04	0.04	0.08	0
5	0.1	0.15	0.48	0.16	0.07	0.04	0
6	0.05	0.08	0.21	0.26	0.22	0.15	0.03
7	0.04	0.03	0.04	0.11	0.54	0.23	0.01
8	0.06	0	0.06	0.08	0.20	0.49	0.11
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.40	0	0	0
4	0.16	0.52	0.20	0.04	0.08	0	0
5	0.04	0.23	0.39	0.20	0.11	0.03	0
6	0.06	0.08	0.22	0.24	0.21	0.18	0.01
7	0.01	0.03	0.08	0.11	0.43	0.32	0.02
8	0	0.03	0.03	0.11	0.20	0.57	0.06
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	0.40	0	0.60	0	0	0	0
4	0.44	0.32	0.08	0.08	0.08	0	0
5	0.5	0.08	0.27	0.05	0.05	0.04	0.01
6	0.39	0.02	0.17	0.08	0.13	0.19	0.02
7	0.28	0	0.07	0.03	0.20	0.37	0.05
8	0.26	0	0.03	0	0.09	0.51	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	0.40	0	0.40	0	0.20	0	0
4	0.24	0.44	0.24	0.08	0	0	0
5	0.12	0.09	0.59	0.11	0.04	0.05	0
6	0.08	0.03	0.25	0.31	0.18	0.14	0.01
7	0.05	0.01	0.10	0.10	0.39	0.34	0.01
8	0.06	0	0.03	0.03	0.20	0.63	0.05
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	0.40	0	0.60	0	0	0	0
4	0.12	0.52	0.24	0	0.12	0	0
5	0.09	0.17	0.51	0.09	0.08	0.05	0.01
6	0.05	0.06	0.25	0.18	0.24	0.21	0.01
7	0.02	0.02	0.06	0.08	0.40	0.41	0.01
8	0.03	0	0.03	0.06	0.28	0.54	0.06
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.45**	0	0.55	0	0	0	0
4	0.16	0.48	0.28	0	0.04	0.04	0
5	0.12	0.14	0.50	0.11	0.09	0.04	0
6	0.07	0.05	0.21	0.23	0.22	0.21	0.01
7	0.04	0.01	0.05	0.09	0.37	0.41	0.03
8	0.03	0	0.06	0.03	0.23	0.60	0.05
9	0	0	0	0	0	0	0

45-55%

NBE

	3	4	5	6	7	8	9
3	0.40	0	0.60	0	0	0	0
4	0.16	0.40	0.20	0.04	0.12	0.08	0
5	0.20	0.12	0.38	0.09	0.16	0.05	0
6	0.13	0.04	0.20	0.08	0.33	0.20	0.02
7	0.05	0.02	0.09	0.06	0.27	0.48	0.03
8	0.03	0	0.06	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.20	0.20	0	0
4	0.08	0.56	0.20	0.08	0.04	0.04	0
5	0.12	0.21	0.43	0.15	0.06	0.03	0
6	0.07	0.10	0.26	0.25	0.21	0.09	0.02
7	0.04	0.05	0.06	0.27	0.37	0.18	0.03
8	0.06	0.03	0.03	0.09	0.37	0.34	0.08
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.20	0	0.20	0
4	0.16	0.44	0.28	0.08	0	0.04	0
5	0.06	0.16	0.42	0.20	0.12	0.04	0
6	0.04	0.13	0.23	0.23	0.21	0.15	0.01
7	0.04	0.08	0.08	0.18	0.29	0.32	0.01
8	0.03	0.03	0	0.20	0.28	0.43	0.03
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0.20	0.20	0.40	0.20	0	0	0
4	0.16	0.56	0.20	0.04	0.04	0	0
5	0.09	0.19	0.48	0.15	0.07	0.02	0
6	0.05	0.13	0.22	0.23	0.24	0.13	0
7	0.01	0.07	0.08	0.15	0.42	0.27	0
8	0.06	0	0.03	0.14	0.17	0.54	0.06
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	0.20	0.20	0.40	0.20	0	0	0
4	0.16	0.56	0.12	0.08	0.08	0	0
5	0.19	0.12	0.47	0.14	0.05	0.02	0.01
6	0.11	0.04	0.26	0.24	0.14	0.20	0.01
7	0.06	0.02	0.09	0.15	0.24	0.42	0.02
8	0.09	0	0.06	0.06	0.17	0.54	0.08
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.40	0	0	0
4	0.12	0.56	0.20	0.04	0.08	0	0
5	0.07	0.12	0.55	0.17	0.06	0.03	0
6	0.04	0.05	0.23	0.32	0.19	0.16	0.01
7	0.03	0.02	0.04	0.19	0.33	0.38	0.01
8	0	0	0.03	0.09	0.31	0.51	0.06
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	0.40	0	0.60	0	0	0	0
4	0.28	0.36	0.16	0.04	0.08	0.08	0
5	0.43	0.10	0.26	0.08	0.09	0.04	0
6	0.30	0.02	0.14	0.07	0.26	0.19	0.02
7	0.16	0	0.07	0.05	0.23	0.46	0.03
8	0.06	0	0.06	0	0.28	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0	0.40	0.20	0.20	0	0
4	0.12	0.52	0.20	0.08	0.04	0.04	0
5	0.12	0.13	0.50	0.15	0.05	0.05	0
6	0.07	0.06	0.24	0.23	0.24	0.14	0.02
7	0.04	0.04	0.03	0.15	0.46	0.27	0.01
8	0.06	0	0.03	0.03	0.37	0.40	0.11
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0.20	0.20	0.40	0	0.20	0	0
4	0.12	0.40	0.24	0.20	0.04	0	0
5	0.10	0.15	0.42	0.16	0.12	0.05	0
6	0.08	0.09	0.20	0.21	0.24	0.17	0.01
7	0.03	0.06	0.07	0.13	0.34	0.36	0.01
8	0.03	0	0	0.06	0.23	0.60	0.08
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	**0.45**	0.40	0.15	0	0	0	0
4	0.52	0.28	0.04	0.08	0.08	0	0
5	0.64	0.06	0.19	0.03	0.05	0.02	0.01
6	0.48	0.01	0.14	0.07	0.10	0.18	0.02
7	0.33	0	0.06	0.06	0.13	0.37	0.05
8	0.31	0	0.03	0	0.09	0.46	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	0.40	0	0.40	0	0.20	0	0
4	0.28	0.44	0.12	0.16	0	0	0
5	0.50	0.08	0.30	0.08	0.03	0.01	0
6	0.40	0.03	0.11	0.18	0.18	0.09	0.01
7	0.32	0	0.03	0.09	0.28	0.27	0.01
8	0.23	0	0	0.06	0.17	0.46	0.08
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	0.40	0	0.40	0.20	0	0	0
4	0.16	0.64	0.12	0.04	0.04	0	0
5	0.1	0.16	0.50	0.11	0.09	0.04	0
6	0.07	0.05	0.26	0.18	0.24	0.19	0.01
7	0.03	0.03	0.04	0.10	0.36	0.43	0.01
8	0.06	0	0.06	0	0.23	0.57	0.08
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.45**	0	0.55	0	0	0	0
4	0.16	0.52	0.24	0.04	0.04	0	0
5	0.15	0.13	0.48	0.13	0.07	0.04	0
6	0.09	0.03	0.24	0.18	0.26	0.19	0.01
7	0.02	0.01	0.05	0.11	0.41	0.39	0.01
8	0.06	0	0.03	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

50-50%

NBE

	3	4	5	6	7	8	9
3	0.40	0	0.40	0.20	0	0	0
4	0.16	0.40	0.24	0	0.12	0.08	0
5	0.26	0.14	0.29	0.11	0.13	0.07	0
6	0.16	0.04	0.15	0.10	0.31	0.22	0.02
7	0.06	0.02	0.07	0.08	0.26	0.48	0.03
8	0.03	0	0.06	0	0.31	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNE

	3	4	5	6	7	8	9
3	0.20	0.40	0	0.20	0.20	0	0
4	0.08	0.68	0.16	0.04	0.04	0	0
5	0.12	0.19	0.45	0.14	0.06	0.04	0
6	0.07	0.12	0.25	0.21	0.24	0.09	0.02
7	0.05	0.06	0.09	0.23	0.32	0.22	0.03
8	0.06	0	0.03	0.11	0.37	0.34	0.09
9	0	0	0	0	0	0	0

DTE

	3	4	5	6	7	8	9
3	0	0	0.20	0.40	0.20	0.20	0
4	0.20	0.40	0.24	0.12	0.04	0	0
5	0.12	0.17	0.33	0.25	0.09	0.04	0
6	0.07	0.09	0.17	0.32	0.18	0.16	0.01
7	0.02	0.06	0.05	0.30	0.29	0.28	0
8	0.08	0.03	0	0.23	0.26	0.37	0.03
9	0	0	0	0	0	0	0

RF

	3	4	5	6	7	8	9
3	0.20	0.20	0.60	0	0	0	0
4	0.24	0.48	0.16	0.04	0.08	0	0
5	0.08	0.23	0.40	0.15	0.10	0.04	0
6	0.06	0.10	0.23	0.24	0.22	0.14	0.01
7	0.02	0.03	0.12	0.16	0.34	0.31	0.02
8	0.03	0	0	0.11	0.17	0.63	0.06
9	0	0	0	0	0	0	0

SVME

	3	4	5	6	7	8	9
3	0.20	0.20	0.40	0.20	0	0	0
4	0.20	0.48	0.20	0.08	0.04	0	0
5	0.19	0.10	0.48	0.14	0.06	0.02	0.01
6	0.11	0.04	0.25	0.24	0.15	0.20	0.01
7	0.06	0.02	0.09	0.13	0.25	0.43	0.02
8	0.09	0	0.06	0.09	0.14	0.51	0.11
9	0	0	0	0	0	0	0

NNE

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.40	0	0	0
4	0.12	0.52	0.24	0.04	0.08	0	0
5	0.09	0.14	0.52	0.17	0.04	0.04	0
6	0.08	0.06	0.23	0.26	0.17	0.19	0.01
7	0.06	0.04	0.04	0.20	0.22	0.44	0
8	0.03	0	0.03	0.08	0.17	0.63	0.06
9	0	0	0	0	0	0	0

NBhte

	3	4	5	6	7	8	9
3	0.40	0	0.40	0.20	0	0	0
4	0.32	0.28	0.24	0	0.08	0.08	0
5	0.53	0.09	0.17	0.10	0.06	0.05	0
6	0.41	0.02	0.10	0.08	0.17	0.20	0.02
7	0.23	0.01	0.05	0.05	0.18	0.45	0.03
8	0.17	0	0.06	0	0.17	0.51	0.09
9	0	0	0	0	0	0	0

*k*NNhte

	3	4	5	6	7	8	9
3	0.20	0.20	0.20	0.20	0.20	0	0
4	0.12	0.52	0.20	0.04	0.08	0.04	0
5	0.1	0.13	0.48	0.17	0.07	0.05	0
6	0.07	0.08	0.24	0.18	0.25	0.15	0.03
7	0.05	0.04	0.06	0.10	0.44	0.28	0.03
8	0.06	0	0.03	0.06	0.37	0.40	0.08
9	0	0	0	0	0	0	0

DThte

	3	4	5	6	7	8	9
3	0	0	0.20	0.60	0.20	0	0
4	0.20	0.40	0.24	0.08	0.04	0.04	0
5	0.10	0.17	0.38	0.24	0.07	0.04	0
6	0.08	0.07	0.19	0.26	0.23	0.16	0.01
7	0.01	0.05	0.07	0.17	0.33	0.36	0.01
8	0.03	0	0.03	0.08	0.17	0.63	0.06
9	0	0	0	0	0	0	0

SVMhte

	3	4	5	6	7	8	9
3	**0.45**	0.35	0.20	0	0	0	0
4	0.56	0.24	0.04	0.08	0.08	0	0
5	0.73	0.05	0.11	0.03	0.05	0.02	0.01
6	0.55	0.01	0.08	0.05	0.13	0.16	0.02
7	0.37	0	0.02	0.02	0.18	0.35	0.06
8	0.31	0	0.03	0	0.09	0.46	0.11
9	0	0	0	0	0	0	0

NNhte

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.20	0.20	0	0
4	0.20	0.48	0.16	0.12	0.04	0	0
5	0.20	0.20	0.37	0.15	0.04	0.04	0
6	0.16	0.05	0.17	0.29	0.16	0.15	0.02
7	0.09	0.03	0.04	0.13	0.36	0.34	0.01
8	0.06	0	0.03	0.06	0.14	0.63	0.08
9	0	0	0	0	0	0	0

HTEsm

	3	4	5	6	7	8	9
3	0.40	0	0.20	0.40	0	0	0
4	0.16	0.48	0.20	0.08	0.08	0	0
5	0.13	0.16	0.44	0.15	0.08	0.04	0
6	0.09	0.07	0.18	0.20	0.26	0.19	0.01
7	0.03	0.03	0.04	0.13	0.33	0.43	0.01
8	0.03	0.03	0.03	0	0.26	0.57	0.08
9	0	0	0	0	0	0	0

HTEdf

	3	4	5	6	7	8	9
3	**0.45**	0.20	0	0.20	0.15	0	0
4	0.20	0.52	0.20	0.04	0.04	0	0
5	0.14	0.16	0.45	0.12	0.10	0.03	0
6	0.09	0.04	0.20	0.18	0.32	0.16	0.01
7	0.03	0.04	0.04	0.11	0.35	0.43	0
8	0.08	0	0	0	0.23	0.63	0.06
9	0	0	0	0	0	0	0

Nursery Dataset

Table A.15: Ensemble Performance on Skewed Class Distributions for Nursery Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.834	0.861	0.834	0.861	0.834	0.861	0.834	0.861	0.834
	Training accuracy	0.899	0.902	0.925	0.919	0.926	0.934	0.938	0.942	0.958
	GF	1.644	1.418	2.213	1.716	2.243	2.106	2.677	2.397	3.952
	F1-Score	0.86	0.86	0.85	0.86	0.85	0.86	0.85	0.85	0.85
*k*NNE	Testing accuracy	0.793	0.822	0.793	0.822	0.793	0.822	0.793	0.822	0.793
	Training accuracy	0.820	0.816	0.828	0.840	0.850	0.850	0.877	0.877	0.891
	GF	1.150	0.967	1.203	1.112	1.380	1.187	1.683	1.447	1.899
	F1-Score	0.88	0.89	0.89	0.88	0.87	0.90	0.86	0.88	0.85
DTE	Testing accuracy	0.894	0.900	0.896	0.900	0.893	0.903	0.895	0.902	0.892
	Training accuracy	0.897	0.893	0.912	0.911	0.921	0.931	0.931	0.927	0.935
	GF	1.029	0.935	1.182	1.124	1.354	1.406	1.522	1.342	1.662
	F1-Score	0.90	0.88	0.90	0.90	0.91	0.85	0.87	0.88	0.84
RF	Testing accuracy	0.903	0.893	0.899	0.889	0.894	0.893	0.894	0.894	0.896
	Training accuracy	0.904	0.914	0.913	0.923	0.921	0.929	0.931	0.930	0.938
	GF	1.010	1.244	1.161	1.442	1.342	1.507	1.536	1.514	1.677
	F1-Score	0.90	0.89	0.89	0.89	0.88	0.84	0.85	0.85	0.82
SVME	Testing accuracy	0.848	0.867	0.849	0.865	0.849	0.866	0.849	0.866	0.848
	Training accuracy	0.922	0.925	0.928	0.940	0.942	0.937	0.950	0.945	0.958
	GF	1.949	1.773	2.097	2.250	2.603	2.127	3.020	2.436	3.619
	F1-Score	0.80	0.93	0.94	0.94	0.93	0.94	0.93	0.95	0.94
NNE	Testing accuracy	0.936	0.945	0.934	0.945	0.935	0.944	0.939	0.946	0.935
	Training accuracy	0.955	0.953	0.963	0.966	0.969	0.972	0.973	0.968	0.980
	GF	1.422	1.170	1.784	1.618	2.097	2.000	2.259	1.688	3.250
	F1-Score	0.96	0.96	0.95	0.96	0.95	0.95	0.93	0.95	0.92

Ensemble	**Measure**	**Skewed Classes %**								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.839	0.865	0.839	0.865	0.839	0.865	0.839	0.865	0.839
	Training accuracy	0.909	0.905	0.935	0.916	0.935	0.934	0.948	0.941	0.964
	GF	1.769	1.421	2.477	1.607	2.477	2.045	3.096	2.288	4.472
	F1-Score	0.87	0.85	0.86	0.86	0.85	0.86	0.85	0.86	0.84
*k*NNhte	Testing accuracy	0.822	0.842	0.822	0.842	0.822	0.842	0.822	0.842	0.822
	Training accuracy	0.827	0.835	0.845	0.865	0.868	0.874	0.896	0.889	0.910
	GF	1.029	0.958	1.148	1.170	1.348	1.254	1.712	1.423	1.978
	F1-Score	0.89	0.90	0.89	0.89	0.88	0.91	0.88	0.90	0.88
DThte	Testing accuracy	0.910	0.916	0.910	0.915	0.91	0.915	0.908	0.914	0.910
	Training accuracy	0.920	0.917	0.928	0.930	0.94	0.944	0.949	0.943	0.951
	GF	1.125	1.012	1.250	1.214	1.50	1.518	1.804	1.509	1.837
	F1-Score	0.94	0.92	0.92	0.90	0.93	0.88	0.90	0.89	0.90
SVMhte	Testing accuracy	0.921	0.923	0.921	0.923	0.920	0.925	0.920	0.927	0.92
	Training accuracy	0.962	0.968	0.968	0.977	0.970	0.974	0.977	0.976	0.98
	GF	2.079	2.406	2.469	3.348	2.667	2.885	3.478	3.042	4.00
	F1-Score	0.95	0.96	0.94	0.96	0.93	0.95	0.93	0.95	0.94
NNhte	Testing accuracy	0.942	0.948	0.942	0.945	0.942	0.949	0.941	0.949	0.940
	Training accuracy	0.964	0.963	0.969	0.975	0.973	0.978	0.981	0.978	0.983
	GF	1.611	1.405	1.871	2.200	2.148	2.318	3.105	2.318	3.529
	F1-Score	0.95	0.96	0.95	0.95	0.94	0.95	0.92	0.95	0.92
HTEsm	Testing accuracy	0.958	0.957	0.954	0.955	0.956	0.956	0.957	0.958	0.958
	Training accuracy	0.970	0.973	0.975	0.977	0.973	0.980	0.982	0.980	0.983
	GF	1.400	1.593	1.840	1.957	1.630	2.200	2.389	2.100	2.471
	F1-Score	0.96	0.96	0.96	0.96	0.95	0.95	0.94	0.95	0.94
HTEdf	Testing accuracy	0.965	0.962	0.964	0.964	0.966	0.963	0.967	0.963	0.966
	Training accuracy	0.978	0.980	0.980	0.984	0.979	0.985	0.987	0.983	0.989
	GF	1.591	1.900	1.800	2.250	1.619	2.467	2.538	2.176	3.091
	F1-Score	0.97	0.97	0.96	0.96	0.97	0.96	0.96	0.96	0.96

Table A.16: Confusion Matrices of Ensembles on Skewed Class Distributions for Nursery Dataset

10-90%

NBE

	0	1	2	3
0	1.00	0	0	0
1	0	0.69	0.10	0.21
2	0	0.06	0.94	0
3	0	0.13	0	0.87

*k*NNE

	0	1	2	3
0	0.94	0.01	0.03	0.02
1	0	0.78	0.14	0.08
2	0	0.18	0.82	0
3	0	0.10	0.01	0.89

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.83	0.07	0.10
2	0	0	0.94	0.06
3	0	0.13	0	0.87

RF

	0	1	2	3
0	0.96	0.03	0	0.01
1	0	0.83	0.08	0.09
2	0	0.12	0.88	0
3	0	0.09	0	0.91

SVME

	0	1	2	3
0	0.58	0.22	0	0.20
1	0	0.91	0.01	0.08
2	0	0.53	0.47	0
3	0	0.06	0	0.94

NNE

	0	1	2	3
0	0.98	0.01	0.01	0
1	0	0.93	0.04	0.03
2	0	0	**1.00**	0
3	0	0.02	0	0.98

NBhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.73	0.10	0.17
2	0	0.06	0.94	0
3	0	0.13	0	0.87

*k*NNhte

	0	1	2	3
0	0.94	0.01	0.03	0.02
1	0	0.76	0.17	0.07
2	0	0.18	0.82	0
3	0	0.07	0.01	0.92

DThte

	0	1	2	3
0	1.00	0	0	0
1	0	0.85	0.06	0.09
2	0	0.06	0.94	0
3	0	0.06	0	0.94

SVMhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.05	0.06
2	0	0.24	0.76	0
3	0	0.04	0	0.96

NNhte

	0	1	2	3
0	0.98	0	0.02	0
1	0	0.92	0.04	0.04
2	0	0	**1.00**	0
3	0	0.05	0	0.95

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.91	0.04	0.05
2	0	0	**1.00**	0
3	0	0.04	0	0.96

HTEdf

	0	1	2	3
0	1.00	0	0	0
1	0	0.94	0.03	0.03
2	0	0	**1.00**	0
3	0	0.03	0	0.97

15-85%

NBE

	0	1	2	3
0	1.0	0	0	0
1	0	0.66	0.19	0.15
2	0	0	**1.00**	0
3	0	0.14	0	0.86

*k*NNE

	0	1	2	3
0	0.98	0	0.01	0.01
1	0	0.80	0.10	0.10
2	0	0.14	0.86	0
3	0	0.11	0.01	0.88

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.83	0.06	0.11
2	0	0.05	0.95	0
3	0	0.20	0	0.80

RF

	0	1	2	3
0	1.00	0	0	0
1	0	0.83	0.07	0.10
2	0	0	**1.00**	0
3	0	0.18	0	0.82

SVME

	0	1	2	3
0	0.97	0.01	0	0.02
1	0	0.91	0	0.09
2	0	0.32	0.68	0
3	0	0.07	0	0.93

NNE

	0	1	2	3
0	1.00	0	0	0
1	0	0.96	0.02	0.02
2	0	0.05	0.95	0
3	0	0.03	0	0.97

NBhte

	0	1	2	3
0	1.0	0	0	0
1	0	0.67	0.19	0.14
2	0	0	**1.00**	0
3	0	0.16	0	0.84

*k*NNhte

	0	1	2	3
0	0.97	0	0.02	0.01
1	0	0.79	0.13	0.08
2	0	0.09	0.91	0
3	0	0.08	0.01	0.91

DThte

	0	1	2	3
0	1.00	0	0	0
1	0	0.85	0.06	0.09
2	0	0.05	0.95	0
3	0	0.11	0	0.89

SVMhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.91	0.01	0.08
2	0	0.05	0.95	0
3	0	0.03	0	0.97

NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.94	0.03	0.03
2	0	0.05	0.95	0
3	0	0.04	0	0.96

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.93	0.04	0.03
2	0	0	**1.00**	0
3	0	0.05	0	0.95

HTEdf

	0	1	2	3
0	1.0	0	0	0
1	0	0.94	0.04	0.02
2	0	0	**1.00**	0
3	0	0.03	0	0.97

20-80%

NBE

	0	1	2	3
0	1.0	0	0	0
1	0	0.66	0.10	0.24
2	0	0.06	**0.94**	0
3	0	0.12	0	0.88

*k*NNE

	0	1	2	3
0	0.99	0	0.01	0
1	0	0.78	0.14	0.08
2	0	0.12	0.88	0
3	0	0.12	0.01	0.87

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.82	0.06	0.12
2	0	0.06	**0.94**	0
3	0	0.13	0	0.87

RF

	0	1	2	3
0	1.00	0	0	0
1	0	0.81	0.10	0.09
2	0	0.06	**0.94**	0
3	0	0.14	0	0.86

SVME

	0	1	2	3
0	1.00	0	0	0
1	0	0.92	0.01	0.07
2	0	0.59	0.41	0
3	0	0.06	0	0.94

NNE

	0	1	2	3
0	1.00	0	0	0
1	0	0.93	0.04	0.03
2	0	0.06	**0.94**	0
3	0	0.06	0	0.94

NBhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.68	0.11	0.21
2	0	0.06	**0.94**	0
3	0	0.13	0	0.87

*k*NNhte

	0	1	2	3
0	0.98	0	0.02	0
1	0	0.73	0.18	0.09
2	0	0.06	**0.94**	0
3	0	0.09	0	0.91

DThte

	0	1	2	3
0	1.0	0	0	0
1	0	0.85	0.04	0.11
2	0	0.06	**0.94**	0
3	0	0.09	0	0.91

SVMhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.04	0.07
2	0	0.12	0.88	0
3	0	0.05	0	0.95

NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.92	0.03	0.05
2	0	0.06	**0.94**	0
3	0	0.07	0	0.93

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.88	0.04	0.08
2	0	0.06	**0.94**	0
3	0	0.04	0	0.96

HTEdf

	0	1	2	3
0	1.00	0	0	0
1	0	0.90	0.04	0.06
2	0	0.06	**0.94**	0
3	0	0.04	0	0.96

25-75%

NBE

	0	1	2	3
0	1.0	0	0	0
1	0	0.67	0.19	0.14
2	0	0	**1.00**	0
3	0	0.14	0	0.86

*k*NNE

	0	1	2	3
0	0.98	0	0.01	0.01
1	0	0.81	0.10	0.09
2	0	0.14	0.86	0
3	0.01	0.15	0	0.84

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.82	0.1	0.08
2	0	0	**1.00**	0
3	0	0.15	0	0.85

RF

	0	1	2	3
0	1.0	0	0	0
1	0	0.84	0.08	0.08
2	0	0	**1.00**	0
3	0	0.19	0	0.81

SVME

	0	1	2	3
0	1.00	0	0	0
1	0	0.91	0	0.09
2	0	0.32	0.68	0
3	0	0.06	0	0.94

NNE

	0	1	2	3
0	1.00	0	0	0
1	0	0.93	0.02	0.05
2	0	0	**1.00**	0
3	0	0.04	0	0.96

NBhte

	0	1	2	3
0	1.0	0	0	0
1	0	0.68	0.19	0.13
2	0	0	**1.00**	0
3	0	0.15	0	0.85

*k*NNhte

	0	1	2	3
0	0.99	0	0.01	0
1	0	0.76	0.14	0.1
2	0	0.09	0.91	0
3	0	0.09	0.01	0.9

DThte

	0	1	2	3
0	1.00	0	0	0
1	0	0.81	0.1	0.09
2	0	0	**1.00**	0
3	0	0.13	0	0.87

SVMhte

	0	1	2	3
0	1.0	0	0	0
1	0	0.92	0.03	0.05
2	0	0	**1.00**	0
3	0	0.06	0	0.94

NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.04	0.07
2	0	0	**1.00**	0
3	0	0.06	0	0.94

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.91	0.06	0.03
2	0	0	**1.00**	0
3	0	0.05	0	0.95

HTEdf

	0	1	2	3
0	1.0	0	0	0
1	0	0.91	0.06	0.03
2	0	0	**1.00**	0
3	0	0.03	0	0.97

30-70%

NBE

	0	1	2	3
0	1.00	0	0	0
1	0	0.68	0.10	0.22
2	0	0.06	0.94	0
3	0	0.13	0	0.87

*k*NNE

	0	1	2	3
0	0.99	0	0.01	0
1	0	0.77	0.16	0.07
2	0	0.06	0.94	0
3	0.01	0.18	0.01	0.80

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.83	0.05	0.12
2	0	0.06	0.94	0
3	0	0.11	0	0.89

RF

	0	1	2	3
0	1.0	0	0	0
1	0	0.80	0.09	0.11
2	0	0.06	0.94	0
3	0	0.16	0	0.84

SVME

	0	1	2	3
0	1.00	0	0	0
1	0	0.90	0.01	0.09
2	0	0.59	0.41	0
3	0	0.08	0	0.92

NNE

	0	1	2	3
0	1.0	0	0	0
1	0	0.92	0.04	0.04
2	0	0	**1.00**	0
3	0	0.07	0	0.93

NBhte

	0	1	2	3
0	1.0	0	0	0
1	0	0.68	0.11	0.21
2	0	0.06	0.94	0
3	0	0.12	0	0.88

*k*NNhte

	0	1	2	3
0	0.98	0	0.02	0
1	0	0.71	0.19	0.10
2	0	0.06	0.94	0
3	0	0.08	0.02	0.90

DThte

	0	1	2	3
0	1.00	0	0	0
1	0	0.86	0.04	0.10
2	0	0.12	0.88	0
3	0	0.08	0	0.92

SVMhte

	0	1	2	3
0	1.0	0	0	0
1	0	0.89	0.06	0.05
2	0	0.12	0.88	0
3	0	0.09	0	0.91

NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.92	0.04	0.04
2	0	0	**1.00**	0
3	0	0.09	0	0.91

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.05	0.06
2	0	0	**1.00**	0
3	0	0.05	0	0.95

HTEdf

	0	1	2	3
0	1.0	0	0	0
1	0	0.91	0.04	0.05
2	0	0	**1.00**	0
3	0	0.05	0	0.95

35-65%

NBE

	0	1	2	3
0	1.00	0	0	0
1	0	0.69	0.19	0.12
2	0	0	**1.00**	0
3	0	0.16	0	0.84

*k*NNE

	0	1	2	3
0	1.00	0	0	0
1	0.01	0.83	0.09	0.07
2	0	0.14	0.86	0
3	0.01	0.13	0.01	0.85

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.73	0.12	0.15
2	0	0	**1.00**	0
3	0	0.22	0	0.78

RF

	0	1	2	3
0	1.00	0	0	0
1	0	0.71	0.15	0.14
2	0	0	**1.00**	0
3	0	0.24	0	0.76

SVME

	0	1	2	3
0	1.00	0	0	0
1	0	0.91	0.01	0.08
2	0	0.32	0.68	0
3	0	0.08	0	0.92

NNE

	0	1	2	3
0	1.00	0	0	0
1	0	0.92	0.04	0.04
2	0	0.05	0.95	0
3	0	0.05	0	0.95

NBhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.70	0.19	0.11
2	0	0	**1.00**	0
3	0	0.16	0	0.84

*k*NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.79	0.12	0.09
2	0	0.09	0.91	0
3	0	0.09	0.01	0.90

DThte

	0	1	2	3
0	1.00	0	0	0
1	0	0.77	0.12	0.11
2	0	0	**1.00**	0
3	0	0.15	0	0.85

SVMhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.90	0.03	0.07
2	0	0	**1.00**	0
3	0	0.06	0	0.94

NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.04	0.07
2	0	0.05	0.95	0
3	0	0.06	0	0.94

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.08	0.03
2	0	0	**1.00**	0
3	0	0.06	0	0.94

HTEdf

	0	1	2	3
0	1.00	0	0	0
1	0	0.87	0.08	0.05
2	0	0	**1.00**	0
3	0	0.04	0	0.96

40-60%

NBE

	0	1	2	3
0	1.00	0	0	0
1	0	0.66	0.10	0.24
2	0	0.06	**0.94**	0
3	0	0.12	0	0.88

*k*NNE

	0	1	2	3
0	1.00	0	0	0
1	0.01	0.76	0.16	0.07
2	0	0.06	**0.94**	0
3	0.03	0.15	0.02	0.80

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.76	0.09	0.15
2	0	0.06	**0.94**	0
3	0	0.16	0	0.84

RF

	0	1	2	3
0	1.00	0	0	0
1	0	0.66	0.14	0.20
2	0	0.06	**0.94**	0
3	0	0.13	0.02	0.85

SVME

	0	1	2	3
0	1.00	0	0	0
1	0	0.91	0.01	0.08
2	0	0.65	0.35	0
3	0	0.07	0	0.93

NNE

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.04	0.07
2	0	0.06	**0.94**	0
3	0	0.09	0	0.91

NBhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.66	0.12	0.22
2	0	0.06	**0.94**	0
3	0	0.12	0	0.88

*k*NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.71	0.19	0.10
2	0	0.06	**0.94**	0
3	0	0.09	0.02	0.89

DThte

	0	1	2	3
0	1.00	0	0	0
1	0	0.80	0.09	0.11
2	0	0.06	**0.94**	0
3	0	0.12	0	0.88

SVMhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.87	0.06	0.07
2	0	0.18	0.82	0
3	0	0.07	0	0.93

NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.88	0.05	0.07
2	0	0.12	0.88	0
3	0	0.11	0	0.89

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.85	0.06	0.09
2	0	0.06	**0.94**	0
3	0	0.06	0	0.94

HTEdf

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.04	0.07
2	0	0.06	**0.94**	0
3	0	0.06	0	0.94

45-55%

NBE

	0	1	2	3
0	1.00	0	0	0
1	0	0.68	0.19	0.13
2	0	0	**1.00**	0
3	0	0.18	0	0.82

*k*NNE

	0	1	2	3
0	1.00	0	0	0
1	0.03	0.82	0.08	0.07
2	0	0.14	0.86	0
3	0.04	0.12	0.01	0.83

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.84	0.05	0.11
2	0	0.05	0.95	0
3	0	0.20	0	0.80

RF

	0	1	2	3
0	1.00	0	0	0
1	0	0.72	0.17	0.11
2	0	0	**1.00**	0
3	0	0.20	0.01	0.79

SVME

	0	1	2	3
0	1.00	0	0	0
1	0	0.92	0	0.08
2	0	0.32	0.68	0
3	0	0.05	0	0.95

NNE

	0	1	2	3
0	1.00	0	0	0
1	0	0.92	0.04	0.04
2	0	0.05	0.95	0
3	0	0.05	0	0.95

NBhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.68	0.19	0.13
2	0	0	**1.00**	0
3	0	0.16	0	0.84

*k*NNhte

	0	1	2	3
0	1.00	0	0	0
1	0.03	0.76	0.12	0.09
2	0	0.05	0.95	0
3	0.01	0.07	0.02	0.90

DThte

	0	1	2	3
0	1.00	0	0	0
1	0	0.80	0.09	0.11
2	0	0	**1.00**	0
3	0	0.14	0	0.86

SVMhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.03	0.08
2	0	0.05	0.95	0
3	0	0.06	0	0.94

NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.89	0.05	0.06
2	0	0.05	0.95	0
3	0	0.06	0	0.94

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.88	0.06	0.06
2	0	0	**1.00**	0
3	0	0.04	0	0.96

HTEdf

	0	1	2	3
0	1.00	0	0	0
1	0	0.90	0.05	0.05
2	0	0	**1.00**	0
3	0	0.04	0	0.96

50-50%

NBE

	0	1	2	3
0	1.00	0	0	0
1	0	0.67	0.10	0.23
2	0	0.06	0.94	0
3	0	0.13	0	0.87

*k*NNE

	0	1	2	3
0	1.00	0	0	0
1	0.03	0.75	0.15	0.07
2	0	0.12	0.88	0
3	0.04	0.16	0.02	0.78

DTE

	0	1	2	3
0	1.00	0	0	0
1	0	0.72	0.07	0.21
2	0	0	0.94	0.06
3	0	0.22	0	0.78

RF

	0	1	2	3
0	1.00	0	0	0
1	0	0.67	0.24	0.09
2	0	0.06	0.94	0
3	0	0.23	0.03	0.74

SVME

	0	1	2	3
0	1.00	0	0	0
1	0	0.92	0.01	0.07
2	0	0.47	0.53	0
3	0	0.06	0	0.94

NNE

	0	1	2	3
0	1.00	0	0	0
1	0	0.87	0.07	0.06
2	0	0	**1.00**	0
3	0	0.08	0	0.92

NBhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.65	0.15	0.20
2	0	0.06	0.94	0
3	0	0.14	0	0.86

*k*NNhte

	0	1	2	3
0	1.00	0	0	0
1	0.01	0.72	0.18	0.09
2	0	0.06	0.94	0
3	0.02	0.10	0.01	0.87

DThte

	0	1	2	3
0	1.00	0	0	0
1	0	0.77	0.09	0.14
2	0	0	**1.00**	0
3	0	0.09	0	0.91

SVMhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.87	0.07	0.06
2	0	0.12	0.88	0
3	0	0.06	0	0.94

NNhte

	0	1	2	3
0	1.00	0	0	0
1	0	0.86	0.08	0.06
2	0	0	**1.00**	0
3	0	0.11	0	0.89

HTEsm

	0	1	2	3
0	1.00	0	0	0
1	0	0.81	0.07	0.12
2	0	0	**1.00**	0
3	0	0.06	0	0.94

HTEdf

	0	1	2	3
0	1.00	0	0	0
1	0	0.84	0.05	0.11
2	0	0	**1.00**	0
3	0	0.03	0	0.97

Bank Marketing Dataset

Table A.17: Ensemble Performance on Skewed Class Distributions for Bank Marketing Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.556	0.556	0.556	0.556	0.556	0.556	0.556	0.556	0.556
	Training accuracy	0.924	0.892	0.838	0.801	0.730	0.729	0.729	0.760	0.783
	GF	5.842	4.111	2.741	2.231	1.644	1.638	1.638	1.850	2.046
	F1-Score	0.42	0.47	0.39	0.48	0.21	0.41	0.24	0.66	0.86
*k*NNE	Testing accuracy	0.897	0.898	0.897	0.898	0.897	0.898	0.898	0.897	0.897
	Training accuracy	0.916	0.881	0.855	0.821	0.803	0.783	0.783	0.768	0.761
	GF	1.226	0.857	0.710	0.570	0.523	0.470	0.470	0.444	0.431
	F1-Score	0.36	0.44	0.50	0.51	0.55	0.58	0.61	0.64	0.67
DTE	Testing accuracy	0.886	0.884	0.886	0.886	0.885	0.886	0.886	0.887	0.886
	Training accuracy	0.981	0.967	0.961	0.956	0.946	0.941	0.941	0.938	0.943
	GF	6.000	3.515	2.923	2.591	2.130	1.932	1.932	1.823	2.000
	F1-Score	0.42	0.76	0.76	0.83	0.85	0.86	0.86	0.86	0.86
RF	Testing accuracy	0.899	0.898	0.900	0.897	0.899	0.899	0.899	0.899	0.897
	Training accuracy	0.983	0.974	0.970	0.967	0.964	0.963	0.963	0.966	0.970
	GF	5.941	3.923	3.333	3.121	2.806	2.730	2.730	2.971	3.433
	F1-Score	0.75	0.78	0.84	0.87	0.87	0.87	0.86	0.87	0.86
SVME	Testing accuracy	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894
	Training accuracy	0.972	0.978	0.978	0.982	0.976	0.976	0.976	0.963	0.960
	GF	3.786	4.818	4.818	5.889	4.417	4.417	4.417	2.865	2.650
	F1-Score	0.84	0.84	0.84	0.84	0.84	0.84	0.83	0.84	0.84
NNE	Testing accuracy	0.892	0.893	0.892	0.890	0.891	0.892	0.892	0.891	0.889
	Training accuracy	0.988	0.980	0.978	0.976	0.973	0.972	0.972	0.966	0.961
	GF	9.000	5.350	4.909	4.583	4.037	3.857	3.857	3.206	2.846
	F1-Score	0.74	0.82	0.86	0.88	0.89	0.89	0.89	0.88	0.89

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.831	0.831	0.831	0.831	0.831	0.831	0.831	0.831	0.831
	Training accuracy	0.978	0.959	0.902	0.873	0.836	0.815	0.815	0.851	0.872
	GF	7.682	4.122	1.724	1.331	1.030	0.914	0.914	1.134	1.320
	F1-Score	0.05	0.21	0.49	0.55	0.55	0.64	0.66	0.78	0.87
*k*NNhte	Testing accuracy	0.897	0.897	0.897	0.897	0.897	0.897	0.897	0.897	0.897
	Training accuracy	0.908	0.869	0.838	0.803	0.784	0.763	0.763	0.749	0.746
	GF	1.120	0.786	0.636	0.523	0.477	0.435	0.435	0.410	0.406
	F1-Score	0.20	0.32	0.39	0.45	0.50	0.55	0.59	0.65	0.67
DThte	Testing accuracy	0.895	0.894	0.895	0.896	0.895	0.896	0.896	0.897	0.894
	Training accuracy	0.988	0.976	0.972	0.967	0.959	0.958	0.958	0.955	0.960
	GF	8.750	4.417	3.750	3.152	2.561	2.476	2.476	2.289	2.650
	F1-Score	0.44	0.77	0.82	0.84	0.86	0.87	0.87	0.87	0.87
SVMhte	Testing accuracy	0.903	0.902	0.903	0.902	0.903	0.902	0.902	0.903	0.903
	Training accuracy	0.980	0.968	0.953	0.930	0.924	0.910	0.910	0.920	0.935
	GF	4.850	3.062	2.064	1.400	1.276	1.089	1.089	1.213	1.492
	F1-Score	0.84	0.84	0.84	0.84	0.85	0.85	0.86	0.88	0.88
NNhte	Testing accuracy	0.899	0.896	0.900	0.899	0.900	0.899	0.899	0.899	0.898
	Training accuracy	0.979	0.968	0.968	0.964	0.963	0.962	0.962	0.963	0.958
	GF	4.810	3.250	3.125	2.806	2.703	2.658	2.658	2.730	2.429
	F1-Score	0.73	0.76	0.83	0.85	0.86	0.88	0.88	0.89	0.89
HTEsm	Testing accuracy	0.896	0.899	0.898	0.897	0.897	0.896	0.896	0.897	0.896
	Training accuracy	0.980	0.969	0.963	0.960	0.947	0.951	0.951	0.954	0.966
	GF	5.200	3.258	2.757	2.575	1.943	2.122	2.122	2.239	3.059
	F1-Score	0.60	0.76	0.81	0.84	0.84	0.85	0.85	0.88	0.88
HTEdf	Testing accuracy	0.902	0.902	0.900	0.902	0.900	0.901	0.901	0.899	0.901
	Training accuracy	0.983	0.974	0.967	0.964	0.957	0.959	0.959	0.961	0.970
	GF	5.765	3.769	3.030	2.722	2.326	2.415	2.415	2.590	3.300
	F1-Score	0.70	0.81	0.83	0.86	0.86	0.88	0.88	0.89	0.89

Table A.18: Confusion Matrices of Ensembles on Skewed Class Distributions for Bank Marketing Dataset

10-90%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.30	0.70	0.24	0.76	0.30	0.70	0.67	0.33	0.97	0.03	0.66	0.34	0.02	0.98
1	0.21	0.79	0.11	0.89	0.18	0.82	0.08	0.92	0.97	0.03	0.11	0.89	0.11	0.89

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.11	0.89	0.30	0.70	0.82	0.18	0.63	0.37	0.47	0.53	0.53	0.47
1	0.03	**0.97**	0.05	0.95	0.27	0.73	0.06	0.94	0.05	0.95	0.03	**0.97**

15-85%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.34	0.66	0.31	0.69	0.71	0.29	0.73	0.27	0.97	0.03	0.77	0.23	0.12	0.88
1	0.16	0.84	0.18	0.82	0.33	0.67	0.24	0.76	0.96	0.04	0.15	0.85	0.15	0.85

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	.20	0.80	0.71	0.29	0.83	0.17	0.68	0.32	0.70	0.30	0.69	0.31
1	0.08	0.92	0.26	0.74	0.37	0.63	0.08	0.92	0.08	0.92	0.06	**0.94**

20-80%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.27	0.73	0.36	0.64	0.70	0.30	0.82	0.18	0.97	0.03	0.86	0.14	0.37	0.63
1	0.22	0.78	0.16	0.84	0.28	0.72	0.31	0.69	0.97	0.03	0.31	0.69	0.23	0.77

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.27	0.73	0.79	0.21	0.85	0.15	0.78	0.22	0.77	0.23	0.76	0.24
1	0.12	0.88	0.20	0.80	0.33	0.67	0.14	0.86	0.12	0.88	0.10	**90**

25-75%

	NBE 0	NBE 1	*k*NNE 0	*k*NNE 1	DTE 0	DTE 1	RF 0	RF 1	SVME 0	SVME 1	NNE 0	NNE 1	NBhte 0	NBhte 1
0	0.35	0.65	0.38	0.62	0.83	0.17	0.89	0.11	0.97	0.03	0.91	0.09	0.42	0.58
1	0.18	0.82	0.23	0.77	0.37	0.63	0.47	0.53	0.96	0.04	0.42	0.58	0.23	0.77

	*k*NNhte 0	*k*NNhte 1	DThte 0	DThte 1	SVMhte 0	SVMhte 1	NNhte 0	NNhte 1	HTEsm 0	HTEsm 1	HTEdf 0	HTEdf 1
0	0.32	0.68	0.83	0.17	0.84	0.16	0.81	0.19	0.85	0.15	0.81	0.19
1	0.12	0.88	0.27	0.73	0.35	0.65	0.16	0.84	0.14	0.86	0.11	**0.89**

30-70%

	NBE 0	NBE 1	*k*NNE 0	*k*NNE 1	DTE 0	DTE 1	RF 0	RF 1	SVME 0	SVME 1	NNE 0	NNE 1	NBhte 0	NBhte 1
0	0.12	0.88	0.43	0.57	0.86	0.14	0.92	0.08	0.97	0.03	0.93	0.07	0.44	0.56
1	0.24	0.76	0.25	0.75	0.43	0.57	0.56	0.44	0.97	0.03	0.48	0.52	0.31	0.69

	*k*NNhte 0	*k*NNhte 1	DThte 0	DThte 1	SVMhte 0	SVMhte 1	NNhte 0	NNhte 1	HTEsm 0	HTEsm 1	HTEdf 0	HTEdf 1
0	0.37	0.63	0.86	0.14	0.86	0.14	0.84	0.16	0.85	0.15	0.86	0.14
1	0.20	0.80	0.31	0.69	0.41	0.59	0.20	0.80	0.20	0.80	0.16	**0.84**

35-65%

	NBE 0	NBE 1	*k*NNE 0	*k*NNE 1	DTE 0	DTE 1	RF 0	RF 1	SVME 0	SVME 1	NNE 0	NNE 1	NBhte 0	NBhte 1
0	0.29	0.71	0.46	0.54	0.89	0.11	0.95	0.05	0.97	0.03	0.93	0.07	0.55	0.45
1	0.27	0.73	0.24	0.76	0.48	0.52	0.72	0.28	0.96	0.04	0.51	0.49	0.33	0.67

	*k*NNhte 0	*k*NNhte 1	DThte 0	DThte 1	SVMhte 0	SVMhte 1	NNhte 0	NNhte 1	HTEsm 0	HTEsm 1	HTEdf 0	HTEdf 1
0	0.43	0.57	0.89	0.11	0.88	0.12	0.89	0.11	0.89	0.11	0.89	0.11
1	0.18	0.82	0.46	0.54	0.51	0.49	0.35	0.65	0.20	0.80	0.17	**0.83**

40-60%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.14	0.86	0.50	0.50	0.87	0.13	0.97	0.03	0.97	0.03	0.94	0.06	0.58	0.42
1	0.27	0.73	0.30	0.70	0.37	0.63	0.83	0.17	0.98	0.02	0.55	0.45	0.39	0.61

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.47	0.53	0.89	0.11	0.90	0.10	0.90	0.10	0.88	0.12	0.91	0.09
1	0.26	0.74	0.37	0.63	0.57	0.43	0.37	0.63	0.25	0.75	0.22	**0.78**

45-55%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.55	0.45	0.54	0.46	0.88	0.12	0.97	0.03	0.97	0.03	0.92	0.08	0.72	0.28
1	0.28	0.72	0.31	0.69	0.47	0.53	0.79	0.21	0.97	0.03	0.46	0.54	0.25	0.75

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.55	0.45	0.89	0.11	0.93	0.07	0.91	0.09	0.91	0.09	0.91	0.09
1	0.32	0.68	0.37	0.63	0.55	0.45	0.32	0.68	0.30	0.70	0.24	**0.76**

50-50%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.86	0.14	0.59	0.41	0.91	0.09	0.98	0.02	0.97	0.03	0.94	0.06	0.89	0.11
1	0.34	0.66	0.36	0.64	0.66	0.34	0.89	0.11	0.96	0.04	0.50	0.50	0.40	0.60

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.59	0.41	0.93	0.07	0.94	0.06	0.92	0.08	0.94	0.06	0.93	0.07
1	0.34	0.66	0.60	0.40	0.60	0.40	0.37	0.63	0.32	0.68	0.29	**0.71**

Censor Income Dataset

Table A.19: Ensemble Performance on Skewed Class Distributions in Censor Income Dataset

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBE	Testing accuracy	0.546	0.521	0.546	0.521	0.530	0.602	0.530	0.602	0.397
	Training accuracy	0.915	0.911	0.887	0.889	0.857	0.859	0.815	0.823	0.707
	GF	5.341	5.382	4.018	4.315	3.287	2.823	2.541	2.249	2.058
	F1-Score	0.68	0.70	0.68	0.69	0.65	0.69	0.63	0.69	0.55
*k*NNE	Testing accuracy	0.825	0.809	0.825	0.809	0.804	0.805	0.804	0.805	0.792
	Training accuracy	0.921	0.912	0.879	0.880	0.855	0.855	0.838	0.836	0.811
	GF	2.215	2.170	1.446	1.592	1.352	1.345	1.210	1.189	1.101
	F1-Score	0.60	0.67	0.69	0.70	0.72	0.72	0.72	0.72	0.71
DTE	Testing accuracy	0.794	0.797	0.795	0.796	0.781	0.784	0.780	0.784	0.743
	Training accuracy	0.967	0.940	0.906	0.901	0.865	0.865	0.828	0.836	0.847
	GF	6.242	3.383	2.181	2.061	1.622	1.600	1.279	1.317	1.680
	F1-Score	0.47	0.60	0.60	0.69	0.70	0.74	0.72	0.76	0.76
RF	Testing accuracy	0.806	0.802	0.810	0.805	0.804	0.799	0.795	0.801	0.790
	Training accuracy	0.978	0.952	0.925	0.915	0.885	0.879	0.852	0.854	0.872
	GF	8.818	4.125	2.533	2.294	1.704	1.661	1.385	1.363	1.641
	F1-Score	0.47	0.62	0.67	0.73	0.73	0.75	0.76	0.76	0.76
SVME	Testing accuracy	0.808	0.811	0.809	0.811	0.809	0.789	0.809	0.789	0.789
	Training accuracy	0.933	0.916	0.887	0.882	0.859	0.862	0.840	0.848	0.839
	GF	2.866	2.250	1.690	1.602	1.355	1.529	1.194	1.388	1.311
	F1-Score	0.53	0.71	0.76	0.75	0.76	0.77	0.77	0.77	0.76
NNE	Testing accuracy	0.814	0.801	0.815	0.801	0.797	0.799	0.796	0.799	0.783
	Training accuracy	0.965	0.937	0.905	0.894	0.869	0.863	0.838	0.842	0.814
	GF	5.314	3.159	1.947	1.877	1.550	1.467	1.259	1.272	1.167
	F1-Score	0.61	0.69	0.72	0.74	0.75	0.75	0.76	0.76	0.75

Ensemble	Measure	Skewed Classes %								
		10-90	15-85	20-80	25-75	30-70	35-75	40-60	45-55	50-50
NBhte	Testing accuracy	0.707	0.694	0.707	0.694	0.703	0.737	0.703	0.737	0.647
	Training accuracy	0.953	0.950	0.922	0.925	0.902	0.894	0.865	0.860	0.800
	GF	6.234	6.120	3.756	4.080	3.031	2.481	2.200	1.879	1.765
	F1-Score	0.66	0.69	0.71	0.72	0.70	0.73	0.70	0.73	0.69
*k*NNhte	Testing accuracy	0.819	0.808	0.819	0.808	0.806	0.805	0.806	0.805	0.795
	Training accuracy	0.926	0.915	0.880	0.883	0.858	0.856	0.839	0.839	0.794
	GF	2.446	2.259	1.508	1.641	1.366	1.354	1.205	1.211	0.995
	F1-Score	0.60	0.67	0.68	0.70	0.71	0.71	0.72	0.72	0.69
DThte	Testing accuracy	0.810	0.812	0.810	0.813	0.800	0.808	0.803	0.804	0.784
	Training accuracy	0.970	0.949	0.919	0.909	0.875	0.873	0.845	0.851	0.866
	GF	6.333	3.686	2.346	2.055	1.600	1.512	1.271	1.315	1.612
	F1-Score	0.43	0.64	0.65	0.71	0.71	0.74	0.74	0.76	0.76
SVMhte	Testing accuracy	0.834	0.827	0.833	0.827	0.825	0.820	0.825	0.819	0.811
	Training accuracy	0.936	0.925	0.894	0.896	0.871	0.865	0.848	0.852	0.822
	GF	2.594	2.307	1.575	1.663	1.357	1.333	1.151	1.223	1.062
	F1-Score	0.58	0.70	0.71	0.73	0.73	0.74	0.74	0.75	0.76
NNhte	Testing accuracy	0.826	0.810	0.825	0.812	0.807	0.807	0.808	0.807	0.800
	Training accuracy	0.957	0.935	0.900	0.883	0.873	0.868	0.849	0.845	0.824
	GF	4.047	2.923	1.750	1.607	1.520	1.462	1.272	1.245	1.136
	F1-Score	0.61	0.70	0.74	0.75	0.73	0.76	0.76	0.76	0.76
HTEsm	Testing accuracy	0.818	0.803	0.818	0.802	0.794	0.803	0.794	0.801	0.787
	Training accuracy	0.953	0.936	0.905	0.904	0.880	0.876	0.857	0.856	0.839
	GF	3.872	3.078	1.916	2.062	1.717	1.589	1.441	1.382	1.323
	F1-Score	0.59	0.69	0.72	0.75	0.74	0.75	0.75	0.76	0.77
HTEdf	Testing accuracy	0.827	0.813	0.828	0.814	0.808	0.808	0.808	0.808	0.801
	Training accuracy	0.958	0.941	0.912	0.909	0.886	0.885	0.866	0.865	0.849
	GF	4.119	3.169	1.955	2.044	1.684	1.669	1.433	1.422	1.318
	F1-Score	0.57	0.71	0.74	0.75	0.75	0.77	0.77	0.77	0.77

Table A.20: Confusion Matrices of Ensembles on Skewed Class Distributions for Censor Income Dataset

10-90%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.58	0.42	0.46	0.54	0.31	0.69	0.31	0.69	0.38	0.62	0.48	0.52	0.55	0.45
1	0.11	0.89	0.04	0.96	0.02	0.98	0.01	0.99	0.03	0.97	0.04	0.96	0.10	0.90

	NBhte		*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.55	0.45	0.46	0.54	0.27	0.73	0.43	0.57	0.48	0.52	0.44	0.56	0.42	0.58
1	0.10	0.90	0.03	0.97	0.02	0.98	0.02	0.98	0.04	0.96	0	**1.00**	0	**1.00**

15-85%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.61	0.39	0.56	0.44	0.47	0.53	0.49	0.51	0.62	0.38	0.59	0.41	0.60	0.40
1	0.12	0.88	0.08	0.92	0.06	0.94	0.07	0.93	0.12	0.88	0.11	0.89	0.11	0.89

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.55	0.45	0.52	0.48	0.60	0.40	0.61	0.39	0.58	0.42	0.59	0.41
1	0.07	0.93	0.07	0.93	0.08	0.92	0.10	0.90	0.05	0.95	0.04	**0.96**

20-80%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.58	0.42	0.58	0.42	0.47	0.53	0.56	0.44	0.70	0.30	0.63	0.37	0.63	0.37
1	0.11	0.89	0.08	0.92	0.05	0.95	0.05	0.95	0.12	0.88	0.09	0.91	0.11	0.89

	NBhte		*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.63	0.37	0.56	0.44	0.52	0.48	0.61	0.39	0.65	0.35	0.62	0.38	0.60	0.40
1	0.11	0.89	0.06	0.94	0.04	0.96	0.06	0.94	0.08	0.92	0.04	0.96	0.03	**0.97**

25-75%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.60	0.40	0.61	0.39	0.60	0.40	0.64	0.36	0.72	0.28	0.68	0.32	0.64	0.36
1	0.12	0.88	0.10	0.90	0.12	0.88	0.10	0.90	0.24	0.76	0.17	0.83	0.11	0.89

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.60	0.40	0.62	0.38	0.65	0.35	0.68	0.32	0.68	0.32	0.67	0.33
1	0.09	0.91	0.10	0.90	0.11	0.89	0.11	0.89	0.08	0.92	0.05	**0.95**

30-70%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.53	0.47	0.63	0.37	0.60	0.40	0.66	0.34	0.73	0.27	0.69	0.31	0.62	0.38
1	0.12	0.88	0.11	0.89	0.09	0.91	0.13	0.87	0.21	0.79	0.12	0.88	0.10	0.90

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.62	0.38	0.62	0.38	0.67	0.33	0.65	0.35	0.66	0.34	0.68	0.32
1	0.10	0.90	0.10	0.90	0.15	0.85	0.10	0.90	0.09	0.91	0.07	**0.93**

35-65%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.60	0.40	0.64	0.36	0.67	0.33	0.72	0.28	0.76	0.24	0.71	0.29	0.67	0.33
1	0.16	0.84	0.13	0.87	0.13	0.87	0.22	0.78	0.28	0.72	0.22	0.78	0.14	0.86

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.62	0.38	0.68	0.32	0.68	0.32	0.74	0.26	0.71	0.29	0.72	0.28
1	0.11	0.89	0.15	0.85	0.14	0.86	0.23	0.77	0.12	0.88	0.10	**0.90**

40-60%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.50	0.50	0.64	0.36	0.66	0.34	0.73	0.27	0.76	0.24	0.71	0.29	0.64	0.36
1	0.13	0.87	0.12	0.88	0.18	0.82	0.20	0.80	0.27	0.73	0.15	0.85	0.12	0.88

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.64	0.36	0.68	0.32	0.69	0.31	0.70	0.30	0.69	0.31	0.70	0.30
1	0.12	0.88	0.15	0.85	0.17	0.83	0.12	0.88	0.12	0.88	0.10	**0.90**

45-55%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.60	0.40	0.65	0.35	0.71	0.29	0.76	0.24	0.78	0.22	0.77	0.23	0.66	0.34
1	0.15	0.85	0.14	0.86	0.16	0.84	0.26	0.74	0.31	0.69	0.26	0.74	0.13	0.87

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.65	0.35	0.72	0.28	0.71	0.29	0.77	0.23	0.72	0.28	0.74	0.26
1	0.13	0.87	0.18	0.82	0.17	0.83	0.25	0.75	0.13	0.87	0.12	**0.88**

50-50%

	NBE		*k*NNE		DTE		RF		SVME		NNE		NBhte	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.40	0.60	0.64	0.36	0.80	0.20	0.89	0.11	0.82	0.18	0.84	0.16	0.60	0.40
1	0.15	0.85	0.16	0.84	0.36	0.64	0.58	0.42	0.43	0.57	0.53	0.47	0.14	0.86

	*k*NNhte		DThte		SVMhte		NNhte		HTEsm		HTEdf	
	0	1	0	1	0	1	0	1	0	1	0	1
0	0.61	0.39	0.83	0.17	0.72	0.28	0.80	0.20	0.75	0.25	0.73	0.27
1	0.15	0.85	0.43	0.57	0.18	0.82	0.41	0.59	0.14	0.86	0.13	**0.87**

Appendix B

Ensemble Performance on Outlier Ratios for Classification Problems

The results of the ensembles over the different datasets in the number of outliers study for classification problems are provided in this appendix. The results consist of training and testing accuracy, GF, and F1-score of the ensembles over the classification datasets.

Sonar Dataset

Table B.1: Ensemble Performance on the Number of Outliers for Sonar Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.747	0.747	0.747	0.747	0.747
	Training Accuracy	0.719	0.712	0.712	0.706	0.756
	GF	0.900	0.878	0.878	0.861	1.037
	F1-score	0.760	0.790	0.760	0.840	0.790
*k*NNE	Testing Accuracy	0.586	0.586	0.586	0.586	0.586
	Training Accuracy	0.774	0.766	0.751	0.743	0.754
	GF	1.832	1.769	1.663	1.611	1.683
	F1-score	0.860	0.860	0.860	0.860	0.810
DTE	Testing Accuracy	0.635	0.614	0.616	0.615	0.639
	Training Accuracy	0.731	0.738	0.708	0.737	0.737
	GF	1.357	1.473	1.315	1.464	1.373
	F1-score	0.810	0.720	0.810	0.790	0.740
RF	Testing Accuracy	0.720	0.690	0.695	0.702	0.722
	Training Accuracy	0.798	0.779	0.797	0.779	0.784
	GF	1.386	1.403	1.502	1.348	1.287
	F1-score	0.810	0.740	0.830	0.880	0.720
SVME	Testing Accuracy	0.729	0.719	0.724	0.725	0.719
	Training Accuracy	0.856	0.846	0.833	0.843	0.837
	GF	1.882	1.825	1.653	1.752	1.724
	F1-score	0.860	0.880	0.880	0.840	0.810
NNE	Testing Accuracy	0.764	0.764	0.769	0.759	0.764
	Training Accuracy	0.855	0.848	0.829	0.836	0.831
	GF	1.628	1.553	1.351	1.470	1.396
	F1-score	0.900	0.910	0.910	0.860	0.910

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.762	0.762	0.762	0.762	0.762
	Training Accuracy	0.724	0.710	0.711	0.707	0.754
	GF	0.862	0.821	0.824	0.812	0.967
	F1-score	0.760	0.790	0.790	0.840	0.790
*k*NNhte	Testing Accuracy	0.655	0.655	0.655	0.655	0.655
	Training Accuracy	0.806	0.808	0.770	0.771	0.782
	GF	1.778	1.797	1.500	1.507	1.583
	F1-score	0.900	0.900	0.900	0.900	0.840
DThte	Testing Accuracy	0.653	0.672	0.633	0.702	0.637
	Training Accuracy	0.786	0.801	0.799	0.773	0.764
	GF	1.621	1.648	1.826	1.313	1.538
	F1-score	0.840	0.810	0.840	0.760	0.760
SVMhte	Testing Accuracy	0.705	0.700	0.716	0.686	0.697
	Training Accuracy	0.772	0.775	0.760	0.763	0.775
	GF	1.294	1.333	1.183	1.325	1.347
	F1-score	0.810	0.810	0.810	0.790	0.760
NNhte	Testing Accuracy	0.768	0.764	0.754	0.759	0.763
	Training Accuracy	0.825	0.821	0.810	0.806	0.816
	GF	1.326	1.318	1.295	1.242	1.288
	F1-score	0.830	0.880	0.880	0.880	0.810
HTEsm	Testing Accuracy	0.776	0.776	0.783	0.788	0.778
	Training Accuracy	0.857	0.846	0.835	0.833	0.852
	GF	1.566	1.455	1.315	1.269	1.500
	F1-score	0.950	0.910	0.920	0.880	0.870
HTEdf	Testing Accuracy	0.796	0.810	0.783	0.782	0.786
	Training Accuracy	0.852	0.853	0.832	0.842	0.845
	GF	1.378	1.293	1.292	1.380	1.381
	F1-score	0.950	0.930	0.930	0.930	0.910

Breast Cancer Dataset

Table B.2: Ensemble Performance on the Number of Outliers for Breast Cancer Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.517	0.517	0.517	0.517	0.517
	Training Accuracy	0.611	0.687	0.675	0.651	0.619
	GF	1.242	1.543	1.486	1.384	1.268
	F1-score	0.510	0.630	0.560	0.530	0.490
*k*NNE	Testing Accuracy	0.666	0.666	0.666	0.666	0.666
	Training Accuracy	0.798	0.797	0.802	0.797	0.806
	GF	1.653	1.645	1.687	1.645	1.722
	F1-score	0.590	0.560	0.580	0.550	0.590
DTE	Testing Accuracy	0.573	0.557	0.572	0.566	0.584
	Training Accuracy	0.718	0.748	0.739	0.734	0.758
	GF	1.514	1.758	1.640	1.632	1.719
	F1-score	0.640	0.620	0.660	0.670	0.610
RF	Testing Accuracy	0.619	0.609	0.627	0.611	0.658
	Training Accuracy	0.779	0.789	0.782	0.791	0.789
	GF	1.724	1.853	1.711	1.861	1.621
	F1-score	0.650	0.670	0.620	0.580	0.620
SVME	Testing Accuracy	0.617	0.627	0.627	0.637	0.606
	Training Accuracy	0.791	0.799	0.794	0.804	0.810
	GF	1.833	1.856	1.811	1.852	2.074
	F1-score	0.510	0.510	0.510	0.510	0.510
NNE	Testing Accuracy	0.556	0.555	0.560	0.555	0.554
	Training Accuracy	0.790	0.787	0.794	0.792	0.804
	GF	2.114	2.089	2.136	2.139	2.276
	F1-score	0.600	0.630	0.630	0.590	0.600

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.518	0.518	0.518	0.518	0.518
	Training Accuracy	0.656	0.709	0.691	0.686	0.683
	GF	1.401	1.656	1.560	1.535	1.521
	F1-score	0.590	0.650	0.590	0.580	0.580
*k*NNhte	Testing Accuracy	0.635	0.635	0.635	0.635	0.635
	Training Accuracy	0.806	0.818	0.820	0.816	0.815
	GF	1.881	2.005	2.028	1.984	1.973
	F1-score	0.720	0.690	0.690	0.690	0.690
DThte	Testing Accuracy	0.569	0.598	0.574	0.589	0.605
	Training Accuracy	0.743	0.748	0.743	0.750	0.775
	GF	1.677	1.595	1.658	1.644	1.756
	F1-score	0.690	0.660	0.660	0.710	0.660
SVMhte	Testing Accuracy	0.621	0.634	0.628	0.623	0.627
	Training Accuracy	0.795	0.795	0.797	0.805	0.810
	GF	1.849	1.785	1.833	1.933	1.963
	F1-score	0.620	0.630	0.650	0.620	0.630
NNhte	Testing Accuracy	0.567	0.571	0.584	0.567	0.568
	Training Accuracy	0.780	0.799	0.795	0.798	0.804
	GF	1.968	2.134	2.029	2.144	2.204
	F1-score	0.660	0.630	0.600	0.620	0.570
HTEsm	Testing Accuracy	0.627	0.630	0.630	0.630	0.633
	Training Accuracy	0.814	0.809	0.801	0.811	0.815
	GF	2.005	1.937	1.859	1.958	1.984
	F1-score	0.700	0.680	0.700	0.630	0.730
HTEdf	Testing Accuracy	0.633	0.633	0.637	0.633	0.633
	Training Accuracy	0.780	0.781	0.784	0.788	0.792
	GF	1.668	1.676	1.681	1.731	1.764
	F1-score	0.720	0.740	0.720	0.720	0.730

Indian Liver Dataset

Table B.3: Ensemble Performance on the Number of Outliers for Indian Liver Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.665	0.665	0.665	0.665	0.665
	Training Accuracy	0.683	0.683	0.682	0.682	0.683
	GF	1.057	1.057	1.053	1.053	1.057
	F1-score	0.560	0.570	0.570	0.580	0.580
*k*NNE	Testing Accuracy	0.730	0.730	0.730	0.730	0.730
	Training Accuracy	0.669	0.666	0.658	0.662	0.660
	GF	0.816	0.808	0.789	0.799	0.794
	F1-score	0.570	0.570	0.570	0.570	0.570
DTE	Testing Accuracy	0.750	0.748	0.738	0.750	0.745
	Training Accuracy	0.727	0.720	0.707	0.716	0.721
	GF	0.916	0.900	0.894	0.880	0.914
	F1-score	0.660	0.680	0.690	0.650	0.650
RF	Testing Accuracy	0.733	0.734	0.736	0.730	0.719
	Training Accuracy	0.773	0.771	0.769	0.767	0.770
	GF	1.176	1.162	1.143	1.159	1.222
	F1-score	0.700	0.790	0.750	0.720	0.720
SVME	Testing Accuracy	0.744	0.746	0.746	0.753	0.748
	Training Accuracy	0.698	0.695	0.688	0.682	0.686
	GF	0.848	0.833	0.814	0.777	0.803
	F1-score	0.680	0.680	0.660	0.660	0.660
NNE	Testing Accuracy	0.730	0.733	0.725	0.727	0.737
	Training Accuracy	0.727	0.724	0.723	0.719	0.715
	GF	0.989	0.967	0.993	0.972	0.923
	F1-score	0.640	0.640	0.660	0.680	0.690

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.668	0.668	0.668	0.668	0.668
	Training Accuracy	0.685	0.688	0.686	0.683	0.684
	GF	1.054	1.064	1.057	1.047	1.051
	F1-score	0.560	0.560	0.570	0.570	0.570
*k*NNhte	Testing Accuracy	0.727	0.727	0.727	0.727	0.727
	Training Accuracy	0.719	0.709	0.704	0.703	0.706
	GF	0.972	0.938	0.922	0.919	0.929
	F1-score	0.630	0.630	0.630	0.630	0.630
DThte	Testing Accuracy	0.747	0.746	0.728	0.744	0.740
	Training Accuracy	0.738	0.740	0.732	0.734	0.727
	GF	0.966	0.977	1.015	0.962	0.952
	F1-score	0.670	0.660	0.640	0.640	0.680
SVMhte	Testing Accuracy	0.743	0.743	0.741	0.747	0.743
	Training Accuracy	0.675	0.671	0.667	0.668	0.662
	GF	0.791	0.781	0.778	0.762	0.760
	F1-score	0.690	0.670	0.640	0.620	0.620
NNhte	Testing Accuracy	0.762	0.762	0.752	0.747	0.764
	Training Accuracy	0.766	0.764	0.753	0.759	0.762
	GF	1.017	1.008	1.004	1.050	0.992
	F1-score	0.690	0.680	0.740	0.700	0.670
HTEsm	Testing Accuracy	0.766	0.761	0.755	0.759	0.766
	Training Accuracy	0.719	0.714	0.721	0.711	0.711
	GF	0.833	0.836	0.878	0.834	0.810
	F1-score	0.670	0.670	0.660	0.670	0.630
HTEdf	Testing Accuracy	0.768	0.764	0.761	0.765	0.769
	Training Accuracy	0.756	0.754	0.749	0.746	0.742
	GF	0.951	0.959	0.952	0.925	0.895
	F1-score	0.730	0.700	0.700	0.700	0.670

Credit Approval Dataset

Table B.4: Ensemble Performance on the Number of Outliers for Credit Approval Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.535	0.535	0.535	0.535	0.535
	Training Accuracy	0.612	0.614	0.623	0.622	0.616
	GF	1.198	1.205	1.233	1.230	1.211
	F1-score	0.470	0.470	0.470	0.480	0.490
*k*NNE	Testing Accuracy	0.785	0.785	0.785	0.785	0.785
	Training Accuracy	0.863	0.868	0.864	0.869	0.860
	GF	1.569	1.629	1.581	1.641	1.536
	F1-score	0.800	0.800	0.810	0.810	0.800
DTE	Testing Accuracy	0.763	0.751	0.763	0.757	0.766
	Training Accuracy	0.864	0.864	0.873	0.869	0.868
	GF	1.743	1.831	1.866	1.855	1.773
	F1-score	0.790	0.800	0.800	0.800	0.800
RF	Testing Accuracy	0.795	0.778	0.794	0.794	0.794
	Training Accuracy	0.887	0.892	0.881	0.891	0.892
	GF	1.814	2.056	1.731	1.890	1.907
	F1-score	0.820	0.830	0.830	0.830	0.820
SVME	Testing Accuracy	0.775	0.778	0.777	0.777	0.778
	Training Accuracy	0.881	0.885	0.881	0.883	0.881
	GF	1.891	1.930	1.874	1.906	1.866
	F1-score	0.850	0.860	0.860	0.860	0.850
NNE	Testing Accuracy	0.668	0.677	0.666	0.672	0.665
	Training Accuracy	0.860	0.863	0.858	0.853	0.851
	GF	2.371	2.358	2.352	2.231	2.248
	F1-score	0.740	0.740	0.760	0.780	0.700

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.651	0.651	0.651	0.651	0.651
	Training Accuracy	0.709	0.710	0.710	0.704	0.719
	GF	1.199	1.203	1.203	1.179	1.242
	F1-score	0.620	0.620	0.630	0.620	0.630
*k*NNhte	Testing Accuracy	0.788	0.788	0.788	0.788	0.788
	Training Accuracy	0.875	0.878	0.874	0.879	0.876
	GF	1.696	1.738	1.683	1.752	1.710
	F1-score	0.850	0.850	0.860	0.860	0.840
DThte	Testing Accuracy	0.759	0.761	0.764	0.758	0.758
	Training Accuracy	0.880	0.879	0.881	0.882	0.875
	GF	2.008	1.975	1.983	2.051	1.936
	F1-score	0.800	0.800	0.820	0.830	0.810
SVMhte	Testing Accuracy	0.821	0.807	0.812	0.819	0.822
	Training Accuracy	0.878	0.877	0.875	0.880	0.879
	GF	1.467	1.569	1.504	1.508	1.471
	F1-score	0.830	0.820	0.830	0.830	0.830
NNhte	Testing Accuracy	0.761	0.769	0.772	0.766	0.766
	Training Accuracy	0.891	0.891	0.886	0.885	0.883
	GF	2.193	2.119	2.000	2.035	2.000
	F1-score	0.780	0.780	0.800	0.800	0.770
HTEsm	Testing Accuracy	0.771	0.774	0.773	0.771	0.774
	Training Accuracy	0.889	0.892	0.890	0.888	0.889
	GF	2.063	2.093	2.064	2.045	2.036
	F1-score	0.800	0.800	0.830	0.800	0.800
HTEdf	Testing Accuracy	0.796	0.797	0.796	0.803	0.801
	Training Accuracy	0.904	0.901	0.902	0.899	0.901
	GF	2.125	2.051	2.082	1.950	2.010
	F1-score	0.830	0.820	0.830	0.830	0.840

Red Wine Dataset

Table B.5: Ensemble Performance on the Number of Outliers for Red Wine Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.471	0.471	0.471	0.471	0.471
	Training Accuracy	0.532	0.537	0.549	0.557	0.567
	GF	1.130	1.143	1.173	1.194	1.222
	F1-score	0.370	0.390	0.380	0.390	0.390
*k*NNE	Testing Accuracy	0.501	0.501	0.501	0.501	0.501
	Training Accuracy	0.784	0.784	0.785	0.783	0.785
	GF	2.310	2.310	2.321	2.300	2.321
	F1-score	0.470	0.470	0.460	0.460	0.470
DTE	Testing Accuracy	0.484	0.485	0.486	0.481	0.483
	Training Accuracy	0.764	0.765	0.765	0.761	0.757
	GF	2.186	2.191	2.187	2.172	2.128
	F1-score	0.480	0.550	0.520	0.540	0.490
RF	Testing Accuracy	0.534	0.548	0.536	0.522	0.544
	Training Accuracy	0.858	0.853	0.855	0.856	0.853
	GF	3.282	3.075	3.200	3.319	3.102
	F1-score	0.610	0.610	0.630	0.620	0.620
SVME	Testing Accuracy	0.531	0.538	0.526	0.528	0.532
	Training Accuracy	0.585	0.583	0.567	0.555	0.557
	GF	1.130	1.108	1.095	1.061	1.056
	F1-score	0.470	0.470	0.460	0.460	0.440
NNE	Testing Accuracy	0.540	0.554	0.549	0.542	0.541
	Training Accuracy	0.829	0.833	0.829	0.829	0.830
	GF	2.690	2.671	2.637	2.678	2.700
	F1-score	0.580	0.590	0.530	0.570	0.580

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.489	0.489	0.489	0.489	0.489
	Training Accuracy	0.529	0.536	0.547	0.555	0.566
	GF	1.085	1.101	1.128	1.148	1.177
	F1-score	0.370	0.380	0.390	0.390	0.380
*k*NNhte	Testing Accuracy	0.531	0.531	0.531	0.531	0.531
	Training Accuracy	0.835	0.837	0.840	0.838	0.840
	GF	2.842	2.877	2.931	2.895	2.931
	F1-score	0.570	0.570	0.560	0.560	0.570
DThte	Testing Accuracy	0.531	0.532	0.524	0.524	0.520
	Training Accuracy	0.848	0.849	0.850	0.849	0.852
	GF	3.086	3.099	3.173	3.152	3.243
	F1-score	0.530	0.570	0.540	0.540	0.550
SVMhte	Testing Accuracy	0.541	0.538	0.538	0.531	0.536
	Training Accuracy	0.828	0.828	0.826	0.825	0.823
	GF	2.669	2.686	2.655	2.680	2.621
	F1-score	0.500	0.500	0.520	0.510	0.530
NNhte	Testing Accuracy	0.564	0.564	0.564	0.564	0.558
	Training Accuracy	0.848	0.852	0.850	0.847	0.846
	GF	2.868	2.946	2.907	2.850	2.870
	F1-score	0.590	0.560	0.550	0.580	0.620
HTEsm	Testing Accuracy	0.549	0.549	0.548	0.549	0.549
	Training Accuracy	0.676	0.679	0.681	0.680	0.681
	GF	1.392	1.405	1.417	1.409	1.414
	F1-score	0.580	0.580	0.540	0.580	0.570
HTEdf	Testing Accuracy	0.552	0.559	0.556	0.556	0.552
	Training Accuracy	0.845	0.844	0.841	0.842	0.837
	GF	2.890	2.827	2.792	2.810	2.748
	F1-score	0.620	0.610	0.630	0.630	0.630

Car Evaluation Dataset

Table B.6: Ensemble Performance on the Number of Outliers for Car Evaluation Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.809	0.809	0.809	0.809	0.809
	Training Accuracy	0.839	0.838	0.836	0.835	0.836
	GF	1.186	1.179	1.165	1.158	1.165
	F1-score	0.820	0.820	0.820	0.820	0.820
*k*NNE	Testing Accuracy	0.830	0.830	0.830	0.830	0.830
	Training Accuracy	0.932	0.931	0.930	0.927	0.925
	GF	2.500	2.464	2.429	2.329	2.267
	F1-score	0.770	0.770	0.780	0.770	0.760
DTE	Testing Accuracy	0.919	0.924	0.924	0.921	0.922
	Training Accuracy	0.949	0.949	0.948	0.949	0.948
	GF	1.588	1.490	1.462	1.549	1.500
	F1-score	0.960	0.960	0.960	0.960	0.960
RF	Testing Accuracy	0.886	0.880	0.884	0.882	0.881
	Training Accuracy	0.970	0.969	0.969	0.967	0.968
	GF	3.800	3.871	3.742	3.576	3.719
	F1-score	0.920	0.940	0.940	0.930	0.910
SVME	Testing Accuracy	0.869	0.871	0.872	0.871	0.872
	Training Accuracy	0.982	0.982	0.981	0.980	0.979
	GF	7.278	7.167	6.737	6.450	6.095
	F1-score	0.900	0.900	0.900	0.900	0.890
NNE	Testing Accuracy	0.946	0.948	0.952	0.948	0.951
	Training Accuracy	0.976	0.976	0.975	0.975	0.975
	GF	2.250	2.167	1.920	2.080	1.960
	F1-score	0.980	0.970	0.980	0.970	0.980

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.809	0.809	0.809	0.809	0.809
	Training Accuracy	0.839	0.838	0.836	0.835	0.836
	GF	1.186	1.179	1.165	1.158	1.165
	F1-score	0.820	0.820	0.820	0.820	0.820
*k*NNhte	Testing Accuracy	0.844	0.844	0.844	0.844	0.844
	Training Accuracy	0.921	0.921	0.918	0.917	0.916
	GF	1.975	1.975	1.902	1.880	1.857
	F1-score	0.790	0.780	0.790	0.790	0.780
DThte	Testing Accuracy	0.934	0.935	0.933	0.934	0.933
	Training Accuracy	0.977	0.977	0.976	0.975	0.975
	GF	2.870	2.826	2.792	2.640	2.680
	F1-score	0.960	0.960	0.960	0.960	0.960
SVMhte	Testing Accuracy	0.903	0.903	0.901	0.906	0.904
	Training Accuracy	0.963	0.962	0.962	0.962	0.959
	GF	2.622	2.553	2.605	2.474	2.341
	F1-score	0.970	0.960	0.960	0.960	0.960
NNhte	Testing Accuracy	0.947	0.947	0.948	0.949	0.951
	Training Accuracy	0.984	0.982	0.981	0.982	0.982
	GF	3.312	2.944	2.737	2.833	2.722
	F1-score	0.990	0.990	0.990	0.990	0.990
HTEsm	Testing Accuracy	0.951	0.952	0.949	0.951	0.950
	Training Accuracy	0.976	0.975	0.974	0.974	0.974
	GF	2.042	1.920	1.962	1.885	1.923
	F1-score	0.970	0.970	0.980	0.980	0.970
HTEdf	Testing Accuracy	0.951	0.956	0.954	0.953	0.958
	Training Accuracy	0.986	0.985	0.985	0.984	0.984
	GF	3.500	2.933	3.067	2.938	2.625
	F1-score	1.000	0.990	0.990	1.000	1.000

White Wine Dataset

Table B.7: Ensemble Performance on the Number of Outliers for White Wine Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.427	0.427	0.427	0.427	0.427
	Training Accuracy	0.448	0.436	0.427	0.427	0.424
	GF	1.038	1.016	1.000	1.000	0.995
	F1-score	0.330	0.310	0.310	0.310	0.320
*k*NNE	Testing Accuracy	0.473	0.473	0.473	0.473	0.473
	Training Accuracy	0.709	0.705	0.705	0.703	0.695
	GF	1.811	1.786	1.786	1.774	1.728
	F1-score	0.440	0.440	0.440	0.440	0.430
DTE	Testing Accuracy	0.477	0.478	0.477	0.476	0.477
	Training Accuracy	0.710	0.710	0.713	0.706	0.705
	GF	1.803	1.800	1.822	1.782	1.773
	F1-score	0.410	0.410	0.440	0.420	0.460
RF	Testing Accuracy	0.527	0.526	0.519	0.512	0.522
	Training Accuracy	0.793	0.789	0.791	0.791	0.788
	GF	2.285	2.246	2.301	2.335	2.255
	F1-score	0.520	0.510	0.550	0.500	0.520
SVME	Testing Accuracy	0.499	0.502	0.499	0.500	0.501
	Training Accuracy	0.506	0.455	0.467	0.462	0.458
	GF	1.014	0.914	0.940	0.929	0.921
	F1-score	0.370	0.340	0.360	0.350	0.340
NNE	Testing Accuracy	0.515	0.518	0.514	0.512	0.517
	Training Accuracy	0.786	0.781	0.781	0.778	0.777
	GF	2.266	2.201	2.219	2.198	2.166
	F1-score	0.460	0.530	0.500	0.520	0.520

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.449	0.449	0.449	0.449	0.449
	Training Accuracy	0.458	0.448	0.441	0.439	0.439
	GF	1.017	0.998	0.986	0.982	0.982
	F1-score	0.330	0.320	0.310	0.300	0.320
*k*NNhte	Testing Accuracy	0.503	0.503	0.503	0.503	0.503
	Training Accuracy	0.767	0.764	0.764	0.766	0.761
	GF	2.133	2.106	2.106	2.124	2.079
	F1-score	0.500	0.500	0.500	0.500	0.490
DThte	Testing Accuracy	0.519	0.518	0.517	0.517	0.521
	Training Accuracy	0.792	0.791	0.791	0.787	0.787
	GF	2.313	2.306	2.311	2.268	2.249
	F1-score	0.540	0.500	0.520	0.510	0.510
SVMhte	Testing Accuracy	0.505	0.504	0.504	0.504	0.506
	Training Accuracy	0.558	0.555	0.552	0.553	0.546
	GF	1.120	1.115	1.107	1.110	1.088
	F1-score	0.400	0.400	0.400	0.390	0.400
NNhte	Testing Accuracy	0.532	0.533	0.532	0.531	0.535
	Training Accuracy	0.782	0.776	0.776	0.774	0.768
	GF	2.147	2.085	2.089	2.075	2.004
	F1-score	0.500	0.500	0.570	0.480	0.480
HTEsm	Testing Accuracy	0.518	0.522	0.520	0.518	0.526
	Training Accuracy	0.763	0.757	0.755	0.753	0.748
	GF	2.034	1.967	1.959	1.951	1.881
	F1-score	0.490	0.470	0.460	0.450	0.480
HTEdf	Testing Accuracy	0.532	0.535	0.534	0.532	0.536
	Training Accuracy	0.775	0.771	0.771	0.768	0.765
	GF	2.080	2.031	2.035	2.017	1.974
	F1-score	0.510	0.500	0.500	0.480	0.480

Nursery Dataset

Table B.8: Ensemble Performance on the Number of Outliers for Nursery Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.856	0.856	0.814	0.814	0.836
	Training Accuracy	0.873	0.872	0.870	0.871	0.877
	GF	1.134	1.125	1.431	1.442	1.333
	F1-score	0.860	0.860	0.830	0.830	0.850
*k*NNE	Testing Accuracy	0.822	0.822	0.824	0.824	0.818
	Training Accuracy	0.913	0.911	0.910	0.908	0.904
	GF	2.046	2.000	1.956	1.913	1.896
	F1-score	0.890	0.900	0.870	0.890	0.880
DTE	Testing Accuracy	0.890	0.888	0.897	0.899	0.921
	Training Accuracy	0.958	0.959	0.959	0.957	0.960
	GF	2.619	2.732	2.512	2.349	1.975
	F1-score	0.950	0.950	0.940	0.940	0.950
RF	Testing Accuracy	0.893	0.893	0.890	0.896	0.914
	Training Accuracy	0.963	0.963	0.964	0.965	0.962
	GF	2.892	2.892	3.056	2.971	2.263
	F1-score	0.950	0.950	0.940	0.940	0.960
SVME	Testing Accuracy	0.890	0.891	0.870	0.872	0.871
	Training Accuracy	0.970	0.969	0.967	0.968	0.969
	GF	3.667	3.516	3.939	4.000	4.161
	F1-score	0.940	0.940	0.940	0.940	0.960
NNE	Testing Accuracy	0.931	0.931	0.930	0.927	0.944
	Training Accuracy	0.984	0.985	0.984	0.984	0.979
	GF	4.312	4.600	4.375	4.562	2.667
	F1-score	0.970	0.980	0.980	0.980	0.980

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.857	0.857	0.829	0.829	0.846
	Training Accuracy	0.887	0.886	0.888	0.889	0.894
	GF	1.265	1.254	1.527	1.541	1.453
	F1-score	0.870	0.860	0.850	0.860	0.870
*k*NNhte	Testing Accuracy	0.846	0.846	0.840	0.840	0.850
	Training Accuracy	0.920	0.917	0.917	0.914	0.916
	GF	1.925	1.855	1.928	1.860	1.786
	F1-score	0.900	0.900	0.870	0.880	0.900
DThte	Testing Accuracy	0.900	0.899	0.902	0.904	0.919
	Training Accuracy	0.977	0.976	0.975	0.974	0.972
	GF	4.348	4.208	3.920	3.692	2.893
	F1-score	0.970	0.970	0.970	0.960	0.970
SVMhte	Testing Accuracy	0.916	0.916	0.910	0.909	0.925
	Training Accuracy	0.979	0.977	0.976	0.975	0.976
	GF	4.000	3.652	3.750	3.640	3.125
	F1-score	0.980	0.970	0.960	0.960	0.970
NNhte	Testing Accuracy	0.934	0.933	0.934	0.933	0.948
	Training Accuracy	0.991	0.991	0.988	0.988	0.986
	GF	7.333	7.444	5.500	5.583	3.714
	F1-score	0.980	0.980	0.990	0.990	0.990
HTEsm	Testing Accuracy	0.953	0.955	0.948	0.947	0.950
	Training Accuracy	0.998	0.998	0.994	0.995	0.994
	GF	23.500	22.500	8.667	10.600	8.333
	F1-score	1.000	1.000	1.000	1.000	1.000
HTEdf	Testing Accuracy	0.961	0.961	0.949	0.947	0.956
	Training Accuracy	0.999	0.999	0.995	0.996	0.996
	GF	39.000	39.000	10.200	13.250	11.000
	F1-score	1.000	1.000	1.000	1.000	1.000

Bank Marketing Dataset

Table B.9: Ensemble Performance on the Number of Outliers for Bank Marketing Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.556	0.556	0.556	0.556	0.556
	Training Accuracy	0.800	0.798	0.800	0.799	0.795
	GF	2.220	2.198	2.220	2.209	2.166
	F1-score	0.790	0.790	0.790	0.790	0.790
*k*NNE	Testing Accuracy	0.897	0.897	0.897	0.897	0.897
	Training Accuracy	0.825	0.830	0.828	0.828	0.827
	GF	0.589	0.606	0.599	0.599	0.595
	F1-score	0.730	0.730	0.730	0.730	0.730
DTE	Testing Accuracy	0.887	0.886	0.885	0.885	0.886
	Training Accuracy	0.936	0.935	0.941	0.936	0.936
	GF	1.766	1.754	1.949	1.797	1.781
	F1-score	0.890	0.890	0.880	0.890	0.890
RF	Testing Accuracy	0.901	0.894	0.899	0.900	0.896
	Training Accuracy	0.950	0.951	0.951	0.953	0.951
	GF	1.980	2.163	2.061	2.128	2.122
	F1-score	0.890	0.890	0.880	0.890	0.890
SVME	Testing Accuracy	0.894	0.894	0.894	0.894	0.894
	Training Accuracy	0.974	0.973	0.973	0.973	0.973
	GF	4.077	3.926	3.926	3.926	3.926
	F1-score	0.840	0.840	0.840	0.840	0.840
NNE	Testing Accuracy	0.890	0.891	0.890	0.892	0.891
	Training Accuracy	0.972	0.971	0.971	0.970	0.971
	GF	3.929	3.759	3.793	3.600	3.759
	F1-score	0.890	0.890	0.890	0.890	0.890

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.831	0.831	0.831	0.831	0.831
	Training Accuracy	0.828	0.827	0.828	0.825	0.823
	GF	0.983	0.977	0.983	0.966	0.955
	F1-score	0.840	0.840	0.840	0.840	0.840
*k*NNhte	Testing Accuracy	0.893	0.893	0.893	0.893	0.893
	Training Accuracy	0.858	0.859	0.860	0.858	0.860
	GF	0.754	0.759	0.764	0.754	0.764
	F1-score	0.780	0.780	0.780	0.780	0.770
DThte	Testing Accuracy	0.895	0.894	0.895	0.896	0.893
	Training Accuracy	0.952	0.952	0.953	0.953	0.951
	GF	2.187	2.208	2.234	2.213	2.184
	F1-score	0.900	0.890	0.890	0.890	0.890
SVMhte	Testing Accuracy	0.902	0.903	0.903	0.903	0.903
	Training Accuracy	0.938	0.940	0.933	0.918	0.920
	GF	1.581	1.617	1.448	1.183	1.213
	F1-score	0.890	0.890	0.890	0.880	0.880
NNhte	Testing Accuracy	0.895	0.900	0.898	0.898	0.898
	Training Accuracy	0.968	0.965	0.966	0.967	0.966
	GF	3.281	2.857	3.000	3.091	3.000
	F1-score	0.900	0.900	0.890	0.890	0.890
HTEsm	Testing Accuracy	0.898	0.899	0.897	0.898	0.898
	Training Accuracy	0.961	0.963	0.963	0.962	0.962
	GF	2.615	2.730	2.784	2.684	2.684
	F1-score	0.890	0.890	0.890	0.890	0.890
HTEdf	Testing Accuracy	0.901	0.900	0.898	0.901	0.899
	Training Accuracy	0.954	0.954	0.956	0.954	0.956
	GF	2.152	2.174	2.318	2.152	2.295
	F1-score	0.900	0.900	0.900	0.900	0.900

Censor Income Dataset

Table B.10: Ensemble Performance on the Number of Outliers for Censor Income Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBE	Testing Accuracy	0.505	0.505	0.505	0.505	0.505
	Training Accuracy	0.808	0.808	0.807	0.805	0.808
	GF	2.578	2.578	2.565	2.538	2.578
	F1-score	0.750	0.750	0.750	0.750	0.750
*k*NNE	Testing Accuracy	0.805	0.805	0.805	0.805	0.805
	Training Accuracy	0.860	0.862	0.862	0.862	0.862
	GF	1.393	1.413	1.413	1.413	1.413
	F1-score	0.800	0.800	0.800	0.800	0.800
DTE	Testing Accuracy	0.805	0.802	0.804	0.800	0.802
	Training Accuracy	0.873	0.874	0.872	0.872	0.873
	GF	1.535	1.571	1.531	1.562	1.559
	F1-score	0.820	0.820	0.820	0.820	0.820
RF	Testing Accuracy	0.806	0.806	0.810	0.804	0.807
	Training Accuracy	0.872	0.872	0.869	0.871	0.870
	GF	1.516	1.516	1.450	1.519	1.485
	F1-score	0.800	0.800	0.800	0.810	0.800
SVME	Testing Accuracy	0.805	0.805	0.805	0.805	0.805
	Training Accuracy	0.869	0.870	0.870	0.870	0.867
	GF	1.489	1.500	1.500	1.500	1.466
	F1-score	0.810	0.810	0.810	0.800	0.810
NNE	Testing Accuracy	0.802	0.804	0.803	0.805	0.802
	Training Accuracy	0.871	0.871	0.870	0.871	0.870
	GF	1.535	1.519	1.515	1.512	1.523
	F1-score	0.810	0.820	0.810	0.810	0.810

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
NBhte	Testing Accuracy	0.726	0.726	0.726	0.726	0.726
	Training Accuracy	0.838	0.838	0.837	0.836	0.840
	GF	1.691	1.691	1.681	1.671	1.712
	F1-score	0.780	0.780	0.780	0.780	0.780
*k*NNhte	Testing Accuracy	0.811	0.811	0.811	0.811	0.811
	Training Accuracy	0.855	0.856	0.855	0.856	0.857
	GF	1.303	1.312	1.303	1.312	1.322
	F1-score	0.800	0.800	0.800	0.800	0.800
DThte	Testing Accuracy	0.807	0.806	0.807	0.804	0.806
	Training Accuracy	0.840	0.837	0.838	0.835	0.836
	GF	1.206	1.190	1.191	1.188	1.183
	F1-score	0.790	0.790	0.780	0.800	0.790
SVMhte	Testing Accuracy	0.829	0.829	0.828	0.828	0.828
	Training Accuracy	0.869	0.859	0.857	0.860	0.861
	GF	1.305	1.213	1.203	1.229	1.237
	F1-score	0.820	0.820	0.820	0.820	0.820
NNhte	Testing Accuracy	0.805	0.806	0.807	0.804	0.806
	Training Accuracy	0.873	0.874	0.876	0.873	0.874
	GF	1.535	1.540	1.556	1.543	1.540
	F1-score	0.830	0.820	0.830	0.830	0.830
HTEsm	Testing Accuracy	0.815	0.812	0.810	0.812	0.813
	Training Accuracy	0.878	0.874	0.868	0.874	0.875
	GF	1.516	1.492	1.439	1.492	1.496
	F1-score	0.840	0.830	0.830	0.830	0.830
HTEdf	Testing Accuracy	0.813	0.814	0.815	0.813	0.815
	Training Accuracy	0.861	0.861	0.861	0.862	0.865
	GF	1.345	1.338	1.331	1.355	1.370
	F1-score	0.840	0.830	0.830	0.830	0.830

Appendix C

Ensemble Performance on Outlier Severities for Classification Problems

The results of the ensembles over the different datasets in the severity of outliers study for classification problems are provided in this appendix. The results consist of the training and testing accuracy, GF, and F1-score of the ensembles over the classification datasets.

Sonar Dataset

Table C.1: Ensemble Performance on the Severity of Outliers for Sonar Dataset

Ensemble	**Measure**	**Severity of Outliers (σ)**				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.747	0.747	0.747	0.747	0.747
	Training Accuracy	0.746	0.752	0.735	0.708	0.720
	GF	0.996	1.020	0.955	0.866	0.904
	F1-score	0.860	0.840	0.860	0.810	0.760
*k*NNE	Testing Accuracy	0.586	0.586	0.586	0.586	0.586
	Training Accuracy	0.759	0.788	0.761	0.782	0.777
	GF	1.718	1.953	1.732	1.899	1.857
	F1-score	0.790	0.790	0.790	0.810	0.860
DTE	Testing Accuracy	0.618	0.618	0.619	0.625	0.638
	Training Accuracy	0.785	0.773	0.760	0.762	0.745
	GF	1.777	1.683	1.588	1.576	1.420
	F1-score	0.790	0.770	0.690	0.770	0.690
RF	Testing Accuracy	0.727	0.718	0.712	0.719	0.741
	Training Accuracy	0.811	0.809	0.778	0.794	0.784
	GF	1.444	1.476	1.297	1.364	1.199
	F1-score	0.770	0.770	0.840	0.770	0.880
SVME	Testing Accuracy	0.690	0.681	0.695	0.686	0.710
	Training Accuracy	0.758	0.785	0.768	0.763	0.768
	GF	1.281	1.484	1.315	1.325	1.250
	F1-score	0.690	0.710	0.710	0.740	0.760
NNE	Testing Accuracy	0.764	0.758	0.760	0.773	0.773
	Training Accuracy	0.859	0.856	0.831	0.863	0.863
	GF	1.674	1.681	1.420	1.657	1.657
	F1-score	0.840	0.860	0.920	0.880	0.860

Ensemble	**Measure**	**Severity of Outliers σ**				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.762	0.762	0.762	0.762	0.762
	Training Accuracy	0.748	0.753	0.736	0.710	0.717
	GF	0.944	0.964	0.902	0.821	0.841
	F1-score	0.860	0.840	0.840	0.790	0.760
*k*NNhte	Testing Accuracy	0.655	0.655	0.655	0.655	0.655
	Training Accuracy	0.799	0.802	0.777	0.818	0.814
	GF	1.716	1.742	1.547	1.896	1.855
	F1-score	0.880	0.860	0.860	0.880	0.910
DThte	Testing Accuracy	0.648	0.686	0.678	0.696	0.676
	Training Accuracy	0.825	0.803	0.785	0.799	0.779
	GF	2.011	1.594	1.498	1.512	1.466
	F1-score	0.810	0.840	0.740	0.790	0.770
SVMhte	Testing Accuracy	0.728	0.725	0.730	0.725	0.724
	Training Accuracy	0.854	0.851	0.812	0.852	0.854
	GF	1.863	1.846	1.436	1.858	1.890
	F1-score	0.810	0.810	0.790	0.840	0.860
NNhte	Testing Accuracy	0.768	0.755	0.768	0.774	0.755
	Training Accuracy	0.851	0.840	0.812	0.819	0.832
	GF	1.557	1.531	1.234	1.249	1.458
	F1-score	0.880	0.880	0.910	0.930	0.900
HTEsm	Testing Accuracy	0.795	0.811	0.778	0.769	0.779
	Training Accuracy	0.865	0.842	0.820	0.852	0.868
	GF	1.519	1.196	1.233	1.561	1.674
	F1-score	0.910	0.910	0.910	0.950	0.930
HTEdf	Testing Accuracy	0.798	0.791	0.794	0.776	0.771
	Training Accuracy	0.874	0.851	0.824	0.853	0.849
	GF	1.603	1.403	1.170	1.524	1.517
	F1-score	0.920	0.930	0.930	0.930	0.940

Breast Cancer Dataset

Table C.2: Ensemble Performance on the Severity of Outliers for Breast Cancer Dataset

Ensemble	**Measure**	**Severity of Outliers (σ)**				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.517	0.517	0.517	0.517	0.517
	Training Accuracy	0.635	0.671	0.683	0.663	0.622
	GF	1.323	1.468	1.524	1.433	1.278
	F1-score	0.560	0.580	0.590	0.560	0.520
*k*NNE	Testing Accuracy	0.666	0.666	0.666	0.666	0.666
	Training Accuracy	0.798	0.801	0.794	0.794	0.797
	GF	1.653	1.678	1.621	1.621	1.645
	F1-score	0.550	0.560	0.550	0.580	0.580
DTE	Testing Accuracy	0.552	0.542	0.563	0.578	0.565
	Training Accuracy	0.740	0.748	0.723	0.733	0.753
	GF	1.723	1.817	1.578	1.581	1.761
	F1-score	0.640	0.570	0.660	0.690	0.640
RF	Testing Accuracy	0.645	0.627	0.623	0.654	0.652
	Training Accuracy	0.775	0.784	0.774	0.782	0.799
	GF	1.578	1.727	1.668	1.587	1.731
	F1-score	0.670	0.590	0.690	0.590	0.590
SVME	Testing Accuracy	0.628	0.638	0.623	0.627	0.625
	Training Accuracy	0.801	0.803	0.790	0.802	0.806
	GF	1.869	1.838	1.795	1.884	1.933
	F1-score	0.510	0.510	0.510	0.510	0.510
NNE	Testing Accuracy	0.564	0.543	0.561	0.561	0.560
	Training Accuracy	0.781	0.795	0.769	0.794	0.792
	GF	1.991	2.229	1.900	2.131	2.115
	F1-score	0.620	0.650	0.680	0.590	0.720

Ensemble	Measure	Severity of Outliers σ				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.518	0.518	0.518	0.518	0.518
	Training Accuracy	0.673	0.692	0.697	0.691	0.677
	GF	1.474	1.565	1.591	1.560	1.492
	F1-score	0.560	0.600	0.590	0.580	0.590
*k*NNhte	Testing Accuracy	0.635	0.635	0.635	0.635	0.635
	Training Accuracy	0.808	0.817	0.804	0.818	0.812
	GF	1.901	1.995	1.862	2.005	1.941
	F1-score	0.690	0.690	0.690	0.720	0.720
DThte	Testing Accuracy	0.592	0.575	0.581	0.595	0.596
	Training Accuracy	0.746	0.754	0.742	0.744	0.761
	GF	1.606	1.728	1.624	1.582	1.690
	F1-score	0.660	0.660	0.680	0.690	0.710
SVMhte	Testing Accuracy	0.633	0.621	0.630	0.630	0.621
	Training Accuracy	0.787	0.809	0.793	0.798	0.806
	GF	1.723	1.984	1.787	1.832	1.954
	F1-score	0.630	0.620	0.650	0.630	0.620
NNhte	Testing Accuracy	0.575	0.567	0.578	0.569	0.566
	Training Accuracy	0.782	0.792	0.781	0.801	0.795
	GF	1.950	2.082	1.927	2.166	2.117
	F1-score	0.620	0.650	0.660	0.580	0.650
HTEsm	Testing Accuracy	0.637	0.627	0.630	0.637	0.630
	Training Accuracy	0.813	0.816	0.797	0.818	0.822
	GF	1.941	2.027	1.823	1.995	2.079
	F1-score	0.680	0.680	0.680	0.690	0.720
HTEdf	Testing Accuracy	0.633	0.633	0.633	0.633	0.637
	Training Accuracy	0.780	0.787	0.786	0.785	0.797
	GF	1.668	1.723	1.715	1.707	1.788
	F1-score	0.700	0.700	0.700	0.710	0.720

Indian Liver Dataset

Table C.3: Ensemble Performance on the Severity of Outliers for Indian Liver Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.665	0.665	0.665	0.665	0.665
	Training Accuracy	0.652	0.659	0.665	0.657	0.672
	GF	0.963	0.982	1.000	0.977	1.021
	F1-score	0.680	0.690	0.670	0.640	0.630
*k*NNE	Testing Accuracy	0.727	0.727	0.727	0.727	0.727
	Training Accuracy	0.680	0.681	0.676	0.679	0.685
	GF	0.853	0.856	0.843	0.850	0.867
	F1-score	0.580	0.570	0.590	0.590	0.610
DTE	Testing Accuracy	0.733	0.742	0.738	0.738	0.733
	Training Accuracy	0.690	0.706	0.693	0.695	0.696
	GF	0.861	0.878	0.853	0.859	0.878
	F1-score	0.670	0.630	0.670	0.610	0.640
RF	Testing Accuracy	0.721	0.732	0.721	0.734	0.726
	Training Accuracy	0.726	0.742	0.737	0.735	0.732
	GF	1.018	1.039	1.061	1.004	1.022
	F1-score	0.730	0.710	0.660	0.750	0.730
SVME	Testing Accuracy	0.746	0.750	0.743	0.745	0.745
	Training Accuracy	0.652	0.648	0.662	0.666	0.667
	GF	0.730	0.710	0.760	0.763	0.766
	F1-score	0.590	0.590	0.640	0.630	0.630
NNE	Testing Accuracy	0.732	0.720	0.730	0.738	0.727
	Training Accuracy	0.662	0.674	0.670	0.678	0.683
	GF	0.793	0.859	0.818	0.814	0.861
	F1-score	0.600	0.590	0.570	0.590	0.610

Ensemble	Measure	Severity of Outliers σ				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.668	0.668	0.668	0.668	0.668
	Training Accuracy	0.667	0.671	0.672	0.668	0.678
	GF	0.997	1.009	1.012	1.000	1.031
	F1-score	0.680	0.690	0.650	0.630	0.620
*k*NNhte	Testing Accuracy	0.730	0.730	0.730	0.730	0.730
	Training Accuracy	0.646	0.642	0.647	0.639	0.650
	GF	0.763	0.754	0.765	0.748	0.771
	F1-score	0.480	0.480	0.510	0.490	0.510
DThte	Testing Accuracy	0.731	0.748	0.745	0.740	0.753
	Training Accuracy	0.682	0.693	0.672	0.690	0.709
	GF	0.846	0.821	0.777	0.839	0.849
	F1-score	0.660	0.670	0.690	0.620	0.650
SVMhte	Testing Accuracy	0.747	0.747	0.745	0.749	0.746
	Training Accuracy	0.661	0.664	0.667	0.662	0.667
	GF	0.746	0.753	0.766	0.743	0.763
	F1-score	0.510	0.590	0.610	0.580	0.590
NNhte	Testing Accuracy	0.759	0.753	0.764	0.752	0.767
	Training Accuracy	0.668	0.672	0.679	0.681	0.681
	GF	0.726	0.753	0.735	0.777	0.730
	F1-score	0.660	0.660	0.680	0.680	0.670
HTEsm	Testing Accuracy	0.760	0.771	0.756	0.760	0.758
	Training Accuracy	0.722	0.728	0.716	0.724	0.745
	GF	0.863	0.842	0.859	0.870	0.949
	F1-score	0.660	0.620	0.660	0.670	0.660
HTEdf	Testing Accuracy	0.771	0.757	0.770	0.757	0.764
	Training Accuracy	0.707	0.716	0.714	0.722	0.722
	GF	0.782	0.856	0.804	0.874	0.849
	F1-score	0.670	0.670	0.680	0.670	0.670

Credit Approval Dataset

Table C.4: Ensemble Performance on the Severity of Outliers for Credit Approval Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.535	0.535	0.535	0.535	0.535
	Training Accuracy	0.609	0.629	0.638	0.632	0.632
	GF	1.189	1.253	1.285	1.264	1.264
	F1-score	0.470	0.480	0.500	0.470	0.470
kNNE	Testing Accuracy	0.788	0.788	0.788	0.788	0.788
	Training Accuracy	0.876	0.875	0.870	0.874	0.872
	GF	1.710	1.696	1.631	1.683	1.656
	F1-score	0.810	0.800	0.800	0.800	0.800
DTE	Testing Accuracy	0.760	0.758	0.755	0.757	0.766
	Training Accuracy	0.866	0.873	0.860	0.859	0.864
	GF	1.791	1.906	1.750	1.723	1.721
	F1-score	0.790	0.790	0.790	0.790	0.790
RF	Testing Accuracy	0.803	0.800	0.797	0.798	0.788
	Training Accuracy	0.890	0.889	0.881	0.892	0.889
	GF	1.791	1.802	1.706	1.870	1.910
	F1-score	0.800	0.810	0.820	0.800	0.830
SVME	Testing Accuracy	0.778	0.775	0.783	0.778	0.777
	Training Accuracy	0.877	0.875	0.878	0.879	0.884
	GF	1.805	1.800	1.779	1.835	1.922
	F1-score	0.840	0.850	0.850	0.840	0.850
NNE	Testing Accuracy	0.670	0.667	0.677	0.681	0.675
	Training Accuracy	0.859	0.866	0.846	0.858	0.848
	GF	2.340	2.485	2.097	2.246	2.138
	F1-score	0.780	0.780	0.780	0.800	0.770

Ensemble	Measure	Severity of Outliers σ				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.651	0.651	0.651	0.651	0.651
	Training Accuracy	0.707	0.717	0.706	0.716	0.706
	GF	1.191	1.233	1.187	1.229	1.187
	F1-score	0.620	0.620	0.640	0.640	0.630
*k*NNhte	Testing Accuracy	0.808	0.808	0.808	0.808	0.808
	Training Accuracy	0.866	0.865	0.863	0.865	0.861
	GF	1.433	1.422	1.401	1.422	1.381
	F1-score	0.850	0.850	0.850	0.860	0.850
DThte	Testing Accuracy	0.762	0.765	0.751	0.765	0.764
	Training Accuracy	0.881	0.879	0.874	0.878	0.878
	GF	2.000	1.942	1.976	1.926	1.934
	F1-score	0.810	0.810	0.810	0.810	0.800
SVMhte	Testing Accuracy	0.818	0.819	0.815	0.819	0.821
	Training Accuracy	0.877	0.882	0.879	0.874	0.876
	GF	1.480	1.534	1.529	1.437	1.444
	F1-score	0.820	0.830	0.830	0.830	0.830
NNhte	Testing Accuracy	0.775	0.766	0.767	0.775	0.756
	Training Accuracy	0.891	0.893	0.876	0.883	0.882
	GF	2.064	2.187	1.879	1.923	2.068
	F1-score	0.740	0.750	0.730	0.730	0.720
HTEsm	Testing Accuracy	0.777	0.771	0.767	0.774	0.768
	Training Accuracy	0.888	0.895	0.884	0.886	0.885
	GF	1.991	2.181	2.009	1.982	2.017
	F1-score	0.850	0.850	0.850	0.850	0.850
HTEdf	Testing Accuracy	0.793	0.796	0.806	0.801	0.803
	Training Accuracy	0.905	0.907	0.898	0.900	0.901
	GF	2.179	2.194	1.902	1.990	1.990
	F1-score	0.870	0.870	0.870	0.860	0.860

Red Wine Dataset

Table C.5: Ensemble Performance on the Severity of Outliers for Red Wine Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.471	0.471	0.471	0.471	0.471
	Training Accuracy	0.529	0.536	0.553	0.553	0.560
	GF	1.123	1.140	1.183	1.183	1.202
	F1-score	0.380	0.360	0.380	0.390	0.380
*k*NNE	Testing Accuracy	0.501	0.501	0.501	0.501	0.501
	Training Accuracy	0.784	0.785	0.786	0.783	0.783
	GF	2.310	2.321	2.332	2.300	2.300
	F1-score	0.470	0.470	0.460	0.460	0.470
DTE	Testing Accuracy	0.480	0.473	0.481	0.489	0.472
	Training Accuracy	0.762	0.759	0.767	0.767	0.760
	GF	2.185	2.187	2.227	2.193	2.200
	F1-score	0.490	0.520	0.530	0.540	0.520
RF	Testing Accuracy	0.556	0.526	0.534	0.546	0.536
	Training Accuracy	0.854	0.857	0.850	0.852	0.851
	GF	3.041	3.315	3.107	3.068	3.114
	F1-score	0.580	0.620	0.590	0.600	0.570
SVME	Testing Accuracy	0.532	0.531	0.526	0.532	0.535
	Training Accuracy	0.583	0.579	0.569	0.560	0.555
	GF	1.122	1.114	1.100	1.064	1.045
	F1-score	0.430	0.450	0.490	0.410	0.440
NNE	Testing Accuracy	0.534	0.540	0.536	0.544	0.539
	Training Accuracy	0.826	0.824	0.827	0.824	0.821
	GF	2.678	2.614	2.682	2.591	2.575
	F1-score	0.550	0.540	0.560	0.540	0.560

Ensemble	Measure	Severity of Outliers σ				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.489	0.489	0.489	0.489	0.489
	Training Accuracy	0.527	0.537	0.550	0.551	0.559
	GF	1.080	1.104	1.136	1.138	1.159
	F1-score	0.380	0.360	0.380	0.390	0.380
*k*NNhte	Testing Accuracy	0.531	0.531	0.531	0.531	0.531
	Training Accuracy	0.838	0.837	0.839	0.840	0.836
	GF	2.895	2.877	2.913	2.931	2.860
	F1-score	0.570	0.570	0.560	0.560	0.570
DThte	Testing Accuracy	0.515	0.539	0.521	0.523	0.532
	Training Accuracy	0.848	0.850	0.849	0.853	0.852
	GF	3.191	3.073	3.172	3.245	3.162
	F1-score	0.640	0.620	0.610	0.640	0.610
SVMhte	Testing Accuracy	0.549	0.552	0.551	0.550	0.548
	Training Accuracy	0.677	0.678	0.680	0.680	0.679
	GF	1.396	1.391	1.403	1.406	1.408
	F1-score	0.490	0.500	0.510	0.510	0.510
NNhte	Testing Accuracy	0.558	0.546	0.548	0.550	0.552
	Training Accuracy	0.826	0.830	0.832	0.830	0.842
	GF	2.540	2.671	2.690	2.647	2.835
	F1-score	0.540	0.540	0.530	0.560	0.540
HTEsm	Testing Accuracy	0.549	0.558	0.554	0.556	0.556
	Training Accuracy	0.852	0.848	0.848	0.848	0.830
	GF	3.047	2.908	2.934	2.921	2.612
	F1-score	0.590	0.580	0.540	0.560	0.560
HTEdf	Testing Accuracy	0.552	0.551	0.559	0.556	0.562
	Training Accuracy	0.842	0.842	0.843	0.845	0.843
	GF	2.835	2.842	2.809	2.865	2.790
	F1-score	0.620	0.610	0.600	0.600	0.600

Car Evaluation Dataset

Table C.6: Ensemble Performance on the Severity of Outliers for Car Evaluation Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.809	0.809	0.809	0.809	0.809
	Training Accuracy	0.945	0.945	0.945	0.945	0.945
	GF	3.473	3.473	3.473	3.473	3.473
	F1-score	0.820	0.820	0.820	0.820	0.820
kNNE	Testing Accuracy	0.844	0.844	0.844	0.844	0.844
	Training Accuracy	0.964	0.964	0.964	0.964	0.964
	GF	4.333	4.333	4.333	4.333	4.333
	F1-score	0.790	0.780	0.790	0.790	0.780
DTE	Testing Accuracy	0.925	0.923	0.924	0.924	0.924
	Training Accuracy	0.973	0.973	0.973	0.973	0.973
	GF	2.778	2.852	2.815	2.815	2.815
	F1-score	0.960	0.960	0.960	0.960	0.960
RF	Testing Accuracy	0.879	0.884	0.879	0.884	0.883
	Training Accuracy	0.971	0.972	0.972	0.972	0.971
	GF	4.172	4.143	4.321	4.143	4.034
	F1-score	0.920	0.940	0.940	0.930	0.910
SVME	Testing Accuracy	0.873	0.872	0.872	0.872	0.870
	Training Accuracy	0.972	0.972	0.972	0.972	0.972
	GF	4.536	4.571	4.571	4.571	4.643
	F1-score	0.900	0.900	0.900	0.900	0.890
NNE	Testing Accuracy	0.947	0.949	0.947	0.946	0.950
	Training Accuracy	0.969	0.964	0.969	0.967	0.967
	GF	1.710	1.417	1.710	1.636	1.515
	F1-score	1.000	0.990	0.990	1.000	1.000

Ensemble	**Measure**	**Severity of Outliers σ**				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.809	0.809	0.809	0.809	0.809
	Training Accuracy	0.945	0.945	0.945	0.945	0.945
	GF	3.473	3.473	3.473	3.473	3.473
	F1-score	0.820	0.820	0.820	0.820	0.820
*k*NNhte	Testing Accuracy	0.830	0.830	0.830	0.830	0.830
	Training Accuracy	0.969	0.969	0.969	0.969	0.969
	GF	5.484	5.484	5.484	5.484	5.484
	F1-score	0.770	0.770	0.780	0.770	0.760
DThte	Testing Accuracy	0.935	0.936	0.932	0.932	0.934
	Training Accuracy	0.974	0.974	0.974	0.974	0.974
	GF	2.500	2.462	2.615	2.615	2.538
	F1-score	0.960	0.960	0.960	0.960	0.960
SVMhte	Testing Accuracy	0.904	0.904	0.902	0.902	0.905
	Training Accuracy	0.964	0.964	0.964	0.964	0.964
	GF	2.667	2.667	2.722	2.722	2.639
	F1-score	0.970	0.960	0.960	0.960	0.960
NNhte	Testing Accuracy	0.954	0.955	0.955	0.955	0.955
	Training Accuracy	0.973	0.970	0.969	0.971	0.973
	GF	1.704	1.500	1.452	1.552	1.667
	F1-score	0.990	0.990	0.990	0.990	0.990
HTEsm	Testing Accuracy	0.948	0.949	0.949	0.948	0.951
	Training Accuracy	0.972	0.969	0.974	0.974	0.974
	GF	1.857	1.645	1.962	2.000	1.885
	F1-score	0.990	0.990	0.990	0.990	0.990
HTEdf	Testing Accuracy	0.948	0.949	0.949	0.948	0.951
	Training Accuracy	0.972	0.969	0.974	0.974	0.974
	GF	1.857	1.645	1.962	2.000	1.885
	F1-score	0.990	0.990	0.990	0.990	0.992

White Wine Dataset

Table C.7: Ensemble Performance on the Severity of Outliers for White Wine Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.427	0.427	0.427	0.427	0.427
	Training Accuracy	0.448	0.436	0.427	0.427	0.424
	GF	1.038	1.016	1.000	1.000	0.995
	F1-score	0.330	0.310	0.310	0.310	0.310
*k*NNE	Testing Accuracy	0.473	0.473	0.473	0.473	0.473
	Training Accuracy	0.709	0.705	0.705	0.703	0.695
	GF	1.811	1.786	1.786	1.774	1.728
	F1-score	0.440	0.440	0.440	0.440	0.430
DTE	Testing Accuracy	0.477	0.478	0.477	0.476	0.477
	Training Accuracy	0.710	0.710	0.713	0.706	0.705
	GF	1.803	1.800	1.822	1.782	1.773
	F1-score	0.410	0.410	0.440	0.410	0.450
RF	Testing Accuracy	0.527	0.526	0.519	0.512	0.522
	Training Accuracy	0.793	0.789	0.791	0.791	0.788
	GF	2.285	2.246	2.301	2.335	2.255
	F1-score	0.520	0.510	0.520	0.480	0.500
SVME	Testing Accuracy	0.499	0.502	0.499	0.500	0.501
	Training Accuracy	0.506	0.455	0.467	0.462	0.458
	GF	1.014	0.914	0.940	0.929	0.921
	F1-score	0.370	0.340	0.360	0.340	0.330
NNE	Testing Accuracy	0.515	0.518	0.514	0.512	0.517
	Training Accuracy	0.786	0.781	0.781	0.778	0.777
	GF	2.266	2.201	2.219	2.198	2.166
	F1-score	0.500	0.530	0.530	0.480	0.470

Ensemble	**Measure**	**Severity of Outliers σ**				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.449	0.449	0.449	0.449	0.449
	Training Accuracy	0.458	0.448	0.441	0.439	0.439
	GF	1.017	0.998	0.986	0.982	0.982
	F1-score	0.330	0.320	0.310	0.300	0.320
*k*NNhte	Testing Accuracy	0.503	0.503	0.503	0.503	0.503
	Training Accuracy	0.767	0.764	0.764	0.766	0.761
	GF	2.133	2.106	2.106	2.124	2.079
	F1-score	0.500	0.500	0.500	0.490	0.480
DThte	Testing Accuracy	0.519	0.518	0.517	0.517	0.521
	Training Accuracy	0.792	0.791	0.791	0.787	0.787
	GF	2.313	2.306	2.311	2.268	2.249
	F1-score	0.540	0.500	0.520	0.510	0.510
SVMhte	Testing Accuracy	0.505	0.504	0.504	0.504	0.506
	Training Accuracy	0.558	0.555	0.552	0.553	0.546
	GF	1.120	1.115	1.107	1.110	1.088
	F1-score	0.400	0.400	0.400	0.370	0.380
NNhte	Testing Accuracy	0.532	0.533	0.532	0.527	0.535
	Training Accuracy	0.782	0.776	0.776	0.768	0.768
	GF	2.147	2.085	2.089	2.039	2.004
	F1-score	0.460	0.530	0.500	0.500	0.500
HTEsm	Testing Accuracy	0.518	0.522	0.520	0.518	0.526
	Training Accuracy	0.763	0.757	0.755	0.753	0.748
	GF	2.034	1.967	1.959	1.951	1.881
	F1-score	0.510	0.510	0.510	0.500	0.500
HTEdf	Testing Accuracy	0.532	0.535	0.534	0.531	0.536
	Training Accuracy	0.775	0.771	0.771	0.774	0.765
	GF	2.080	2.031	2.035	2.075	1.974
	F1-score	0.470	0.520	0.510	0.510	0.510

Nursery Dataset

Table C.8: Ensemble Performance on the Severity of Outliers for Nursery Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.854	0.854	0.854	0.854	0.854
	Training Accuracy	0.877	0.879	0.877	0.874	0.876
	GF	1.187	1.207	1.187	1.159	1.177
	F1-score	0.840	0.840	0.840	0.840	0.840
kNNE	Testing Accuracy	0.810	0.810	0.810	0.810	0.810
	Training Accuracy	0.907	0.905	0.904	0.901	0.902
	GF	2.043	2.000	1.979	1.919	1.939
	F1-score	0.900	0.900	0.900	0.890	0.890
DTE	Testing Accuracy	0.900	0.902	0.902	0.901	0.901
	Training Accuracy	0.955	0.956	0.956	0.954	0.952
	GF	2.222	2.227	2.227	2.152	2.062
	F1-score	0.940	0.940	0.940	0.940	0.940
RF	Testing Accuracy	0.901	0.892	0.897	0.899	0.893
	Training Accuracy	0.965	0.964	0.964	0.963	0.963
	GF	2.829	3.000	2.861	2.730	2.892
	F1-score	0.950	0.960	0.940	0.950	0.950
SVME	Testing Accuracy	0.869	0.869	0.869	0.869	0.868
	Training Accuracy	0.972	0.973	0.971	0.970	0.970
	GF	4.679	4.852	4.517	4.367	4.400
	F1-score	0.950	0.960	0.960	0.950	0.950
NNE	Testing Accuracy	0.952	0.955	0.951	0.952	0.955
	Training Accuracy	0.997	0.997	0.997	0.997	0.997
	GF	16.000	15.000	16.333	16.000	15.000
	F1-score	1.000	1.000	1.000	1.000	1.000

Ensemble	Measure	Severity of Outliers σ				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.859	0.859	0.859	0.859	0.859
	Training Accuracy	0.891	0.893	0.891	0.888	0.890
	GF	1.294	1.318	1.294	1.259	1.282
	F1-score	0.860	0.860	0.850	0.860	0.860
*k*NNhte	Testing Accuracy	0.821	0.821	0.821	0.821	0.821
	Training Accuracy	0.909	0.909	0.907	0.908	0.907
	GF	1.967	1.967	1.925	1.946	1.925
	F1-score	0.890	0.890	0.890	0.870	0.880
DThte	Testing Accuracy	0.914	0.914	0.915	0.915	0.911
	Training Accuracy	0.973	0.975	0.977	0.975	0.972
	GF	3.185	3.440	3.696	3.400	3.179
	F1-score	0.970	0.970	0.970	0.970	0.970
SVMhte	Testing Accuracy	0.915	0.914	0.914	0.915	0.916
	Training Accuracy	0.979	0.979	0.974	0.973	0.975
	GF	4.048	4.095	3.308	3.148	3.360
	F1-score	0.970	0.970	0.970	0.970	0.970
NNhte	Testing Accuracy	0.958	0.957	0.958	0.960	0.958
	Training Accuracy	0.998	0.998	0.998	0.998	0.998
	GF	21.000	21.500	21.000	20.000	21.000
	F1-score	1.000	1.000	1.000	1.000	1.000
HTEsm	Testing Accuracy	0.939	0.939	0.939	0.940	0.941
	Training Accuracy	0.979	0.979	0.980	0.978	0.978
	GF	2.905	2.905	3.050	2.727	2.682
	F1-score	0.980	0.980	0.980	0.980	0.980
HTEdf	Testing Accuracy	0.943	0.940	0.941	0.941	0.942
	Training Accuracy	0.985	0.985	0.986	0.985	0.985
	GF	3.800	4.000	4.214	3.933	3.867
	F1-score	0.990	0.990	0.990	0.990	0.990

Bank Marketing Dataset

Table C.9: Ensemble Performance on the Severity of Outliers for Bank Marketing Dataset

Ensemble	**Measure**	**Severity of Outliers (σ)**				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.556	0.556	0.556	0.556	0.556
	Training Accuracy	0.798	0.798	0.799	0.799	0.797
	GF	2.198	2.198	2.209	2.209	2.187
	F1-score	0.780	0.790	0.790	0.790	0.790
*k*NNE	Testing Accuracy	0.893	0.893	0.893	0.893	0.893
	Training Accuracy	0.858	0.858	0.860	0.859	0.859
	GF	0.754	0.754	0.764	0.759	0.759
	F1-score	0.730	0.730	0.730	0.730	0.730
DTE	Testing Accuracy	0.887	0.887	0.886	0.886	0.887
	Training Accuracy	0.938	0.937	0.937	0.939	0.937
	GF	1.823	1.794	1.810	1.869	1.794
	F1-score	0.890	0.890	0.880	0.880	0.880
RF	Testing Accuracy	0.900	0.897	0.902	0.896	0.899
	Training Accuracy	0.951	0.950	0.951	0.952	0.952
	GF	2.041	2.060	2.000	2.167	2.104
	F1-score	0.890	0.890	0.890	0.890	0.890
SVME	Testing Accuracy	0.894	0.894	0.894	0.894	0.894
	Training Accuracy	0.974	0.974	0.973	0.973	0.973
	GF	4.077	4.077	3.926	3.926	3.926
	F1-score	0.840	0.840	0.840	0.840	0.840
NNE	Testing Accuracy	0.891	0.890	0.891	0.888	0.891
	Training Accuracy	0.969	0.972	0.972	0.972	0.971
	GF	3.516	3.929	3.893	4.000	3.759
	F1-score	0.890	0.890	0.890	0.890	0.890

Ensemble	**Measure**	**Severity of Outliers σ**				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.831	0.831	0.831	0.831	0.831
	Training Accuracy	0.828	0.826	0.824	0.828	0.823
	GF	0.983	0.971	0.960	0.983	0.955
	F1-score	0.840	0.840	0.840	0.840	0.840
*k*NNhte	Testing Accuracy	0.897	0.897	0.897	0.897	0.897
	Training Accuracy	0.827	0.827	0.826	0.825	0.827
	GF	0.595	0.595	0.592	0.589	0.595
	F1-score	0.780	0.780	0.780	0.780	0.780
DThte	Testing Accuracy	0.893	0.896	0.895	0.895	0.893
	Training Accuracy	0.952	0.952	0.950	0.954	0.952
	GF	2.229	2.167	2.100	2.283	2.229
	F1-score	0.890	0.890	0.890	0.890	0.890
SVMhte	Testing Accuracy	0.903	0.903	0.903	0.903	0.903
	Training Accuracy	0.938	0.933	0.924	0.920	0.919
	GF	1.565	1.448	1.276	1.213	1.198
	F1-score	0.880	0.880	0.880	0.880	0.880
NNhte	Testing Accuracy	0.897	0.899	0.898	0.898	0.896
	Training Accuracy	0.962	0.963	0.963	0.963	0.965
	GF	2.711	2.730	2.757	2.757	2.971
	F1-score	0.890	0.890	0.890	0.890	0.890
HTEsm	Testing Accuracy	0.899	0.900	0.897	0.899	0.897
	Training Accuracy	0.955	0.956	0.956	0.957	0.955
	GF	2.244	2.273	2.341	2.349	2.289
	F1-score	0.890	0.890	0.890	0.890	0.890
HTEdf	Testing Accuracy	0.897	0.898	0.899	0.899	0.899
	Training Accuracy	0.966	0.967	0.967	0.967	0.967
	GF	3.029	3.091	3.061	3.061	3.061
	F1-score	0.890	0.890	0.890	0.890	0.890

Censor Income Dataset

Table C.10: Ensemble Performance on the Severity of Outliers for Censor Income Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
NBE	Testing Accuracy	0.591	0.591	0.591	0.591	0.591
	Training Accuracy	0.777	0.779	0.772	0.768	0.773
	GF	1.834	1.851	1.794	1.763	1.802
	F1-score	0.680	0.690	0.680	0.670	0.680
*k*NNE	Testing Accuracy	0.813	0.813	0.813	0.813	0.813
	Training Accuracy	0.862	0.860	0.860	0.859	0.859
	GF	1.355	1.336	1.336	1.326	1.326
	F1-score	0.810	0.800	0.800	0.800	0.800
DTE	Testing Accuracy	0.811	0.811	0.810	0.812	0.811
	Training Accuracy	0.851	0.849	0.852	0.845	0.843
	GF	1.268	1.252	1.284	1.213	1.204
	F1-score	0.790	0.790	0.790	0.790	0.790
RF	Testing Accuracy	0.811	0.813	0.814	0.815	0.812
	Training Accuracy	0.879	0.877	0.876	0.873	0.875
	GF	1.562	1.520	1.500	1.457	1.504
	F1-score	0.800	0.810	0.820	0.810	0.810
SVME	Testing Accuracy	0.808	0.808	0.808	0.808	0.808
	Training Accuracy	0.872	0.869	0.871	0.869	0.869
	GF	1.500	1.466	1.488	1.466	1.466
	F1-score	0.810	0.810	0.810	0.810	0.810
NNE	Testing Accuracy	0.815	0.817	0.816	0.815	0.814
	Training Accuracy	0.878	0.877	0.877	0.874	0.878
	GF	1.516	1.488	1.496	1.468	1.525
	F1-score	0.830	0.830	0.830	0.820	0.820

Ensemble	Measure	Severity of Outliers σ				
		2.0	2.5	3.0	3.5	4.0
NBhte	Testing Accuracy	0.733	0.733	0.733	0.733	0.733
	Training Accuracy	0.829	0.830	0.830	0.828	0.831
	GF	1.561	1.571	1.571	1.552	1.580
	F1-score	0.750	0.760	0.760	0.760	0.760
*k*NNhte	Testing Accuracy	0.823	0.823	0.823	0.823	0.823
	Training Accuracy	0.856	0.854	0.855	0.854	0.856
	GF	1.229	1.212	1.221	1.212	1.229
	F1-score	0.810	0.800	0.810	0.800	0.800
DThte	Testing Accuracy	0.823	0.823	0.826	0.827	0.819
	Training Accuracy	0.869	0.869	0.868	0.881	0.864
	GF	1.351	1.351	1.318	1.454	1.331
	F1-score	0.810	0.810	0.820	0.820	0.820
SVMhte	Testing Accuracy	0.830	0.829	0.829	0.829	0.829
	Training Accuracy	0.868	0.860	0.860	0.859	0.862
	GF	1.288	1.221	1.221	1.213	1.239
	F1-score	0.810	0.810	0.810	0.810	0.810
NNhte	Testing Accuracy	0.817	0.818	0.816	0.819	0.819
	Training Accuracy	0.882	0.880	0.872	0.878	0.881
	GF	1.551	1.517	1.438	1.484	1.521
	F1-score	0.810	0.820	0.810	0.820	0.810
HTEsm	Testing Accuracy	0.824	0.824	0.825	0.827	0.821
	Training Accuracy	0.878	0.876	0.876	0.874	0.876
	GF	1.443	1.419	1.411	1.373	1.444
	F1-score	0.830	0.830	0.830	0.820	0.820
HTEdf	Testing Accuracy	0.831	0.830	0.829	0.828	0.827
	Training Accuracy	0.864	0.879	0.878	0.877	0.881
	GF	1.243	1.405	1.402	1.398	1.454
	F1-score	0.830	0.830	0.830	0.830	0.830

Appendix D

Ensemble Performance on Bagged Subsets for Classification Problems

The results of the ensembles over the bagged subsets of the training dataset for classification problems are provided in this appendix. Plots of the training and testing accuracies for each classification dataset are first presented. Then the results of training and testing accuracy, GF, and F1-score of the ensembles over the classification datasets are provided.

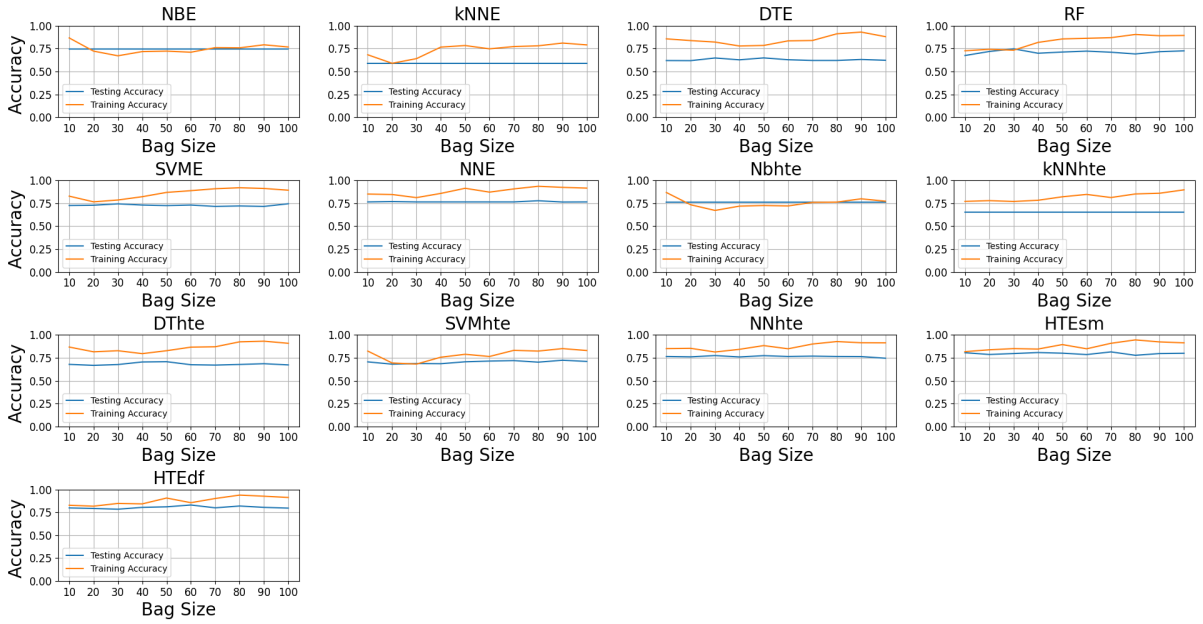

Figure D.1: Ensemble Performance on Bagged Subsets of the Sonar Dataset

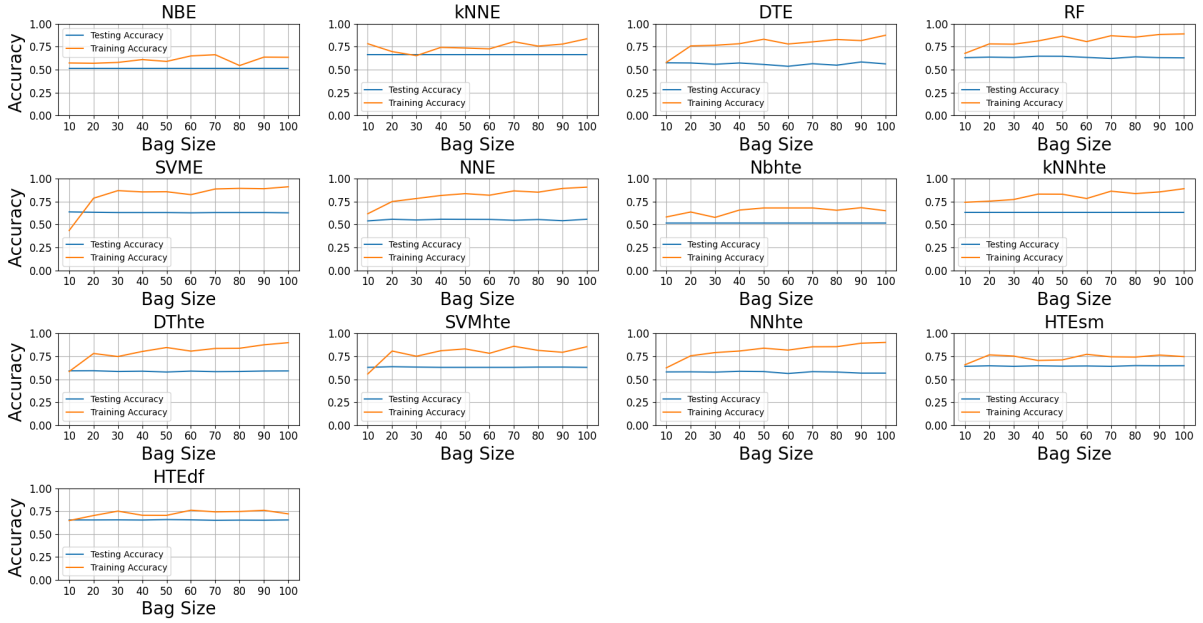

Figure D.2: Ensemble Performance on Bagged Subsets of the Breast Cancer Dataset

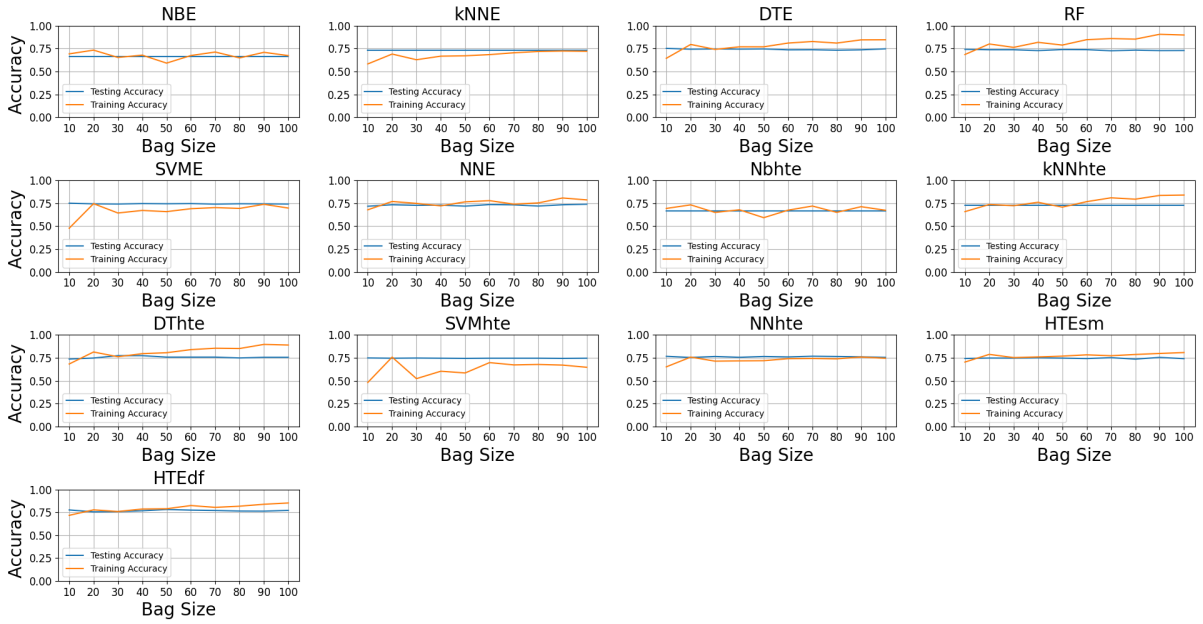

Figure D.3: Ensemble Performance on Bagged Subsets of the Indian Liver Dataset

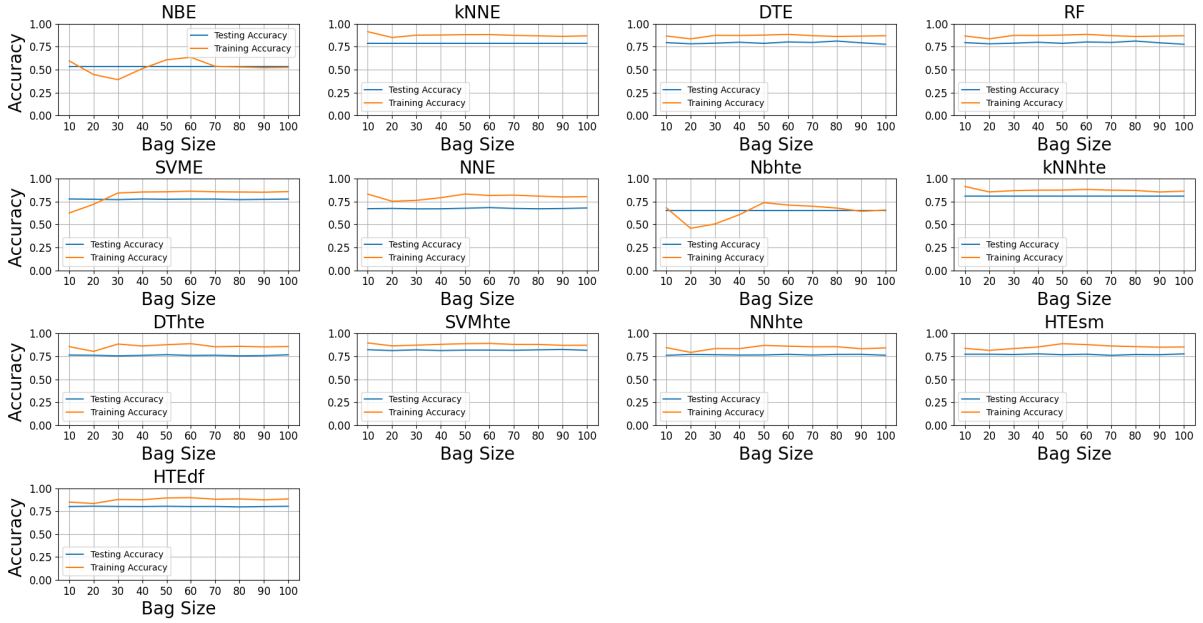

Figure D.4: Ensemble Performance on Bagged Subsets of the Credit Approval Dataset

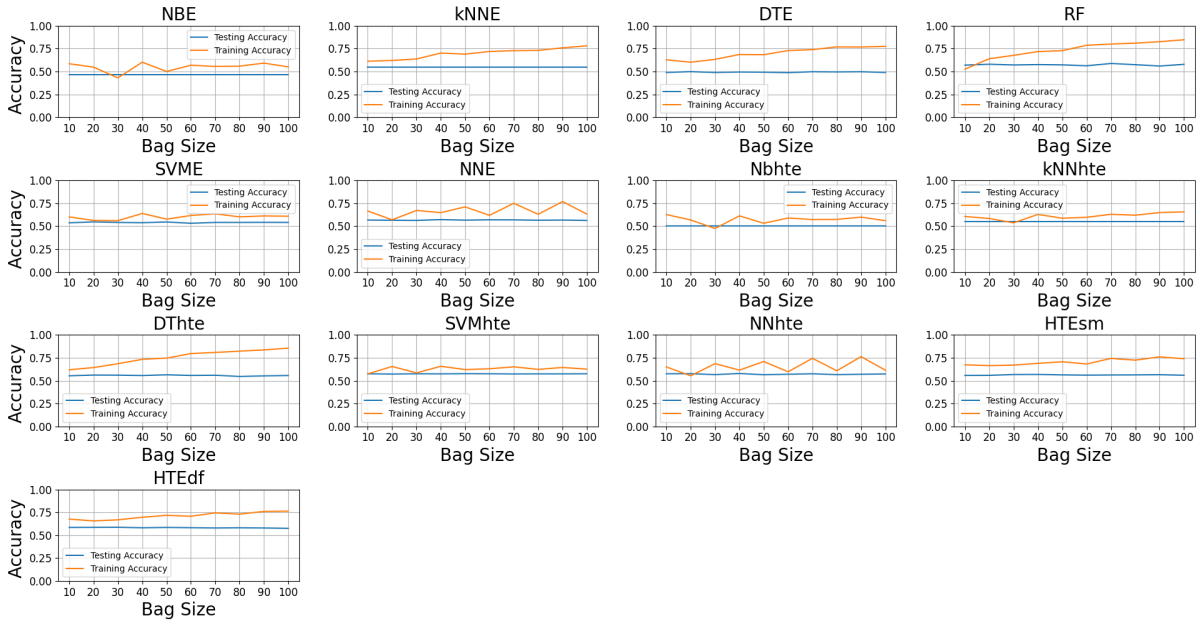

Figure D.5: Ensemble Performance on Bagged Subsets of the Red Wine Dataset

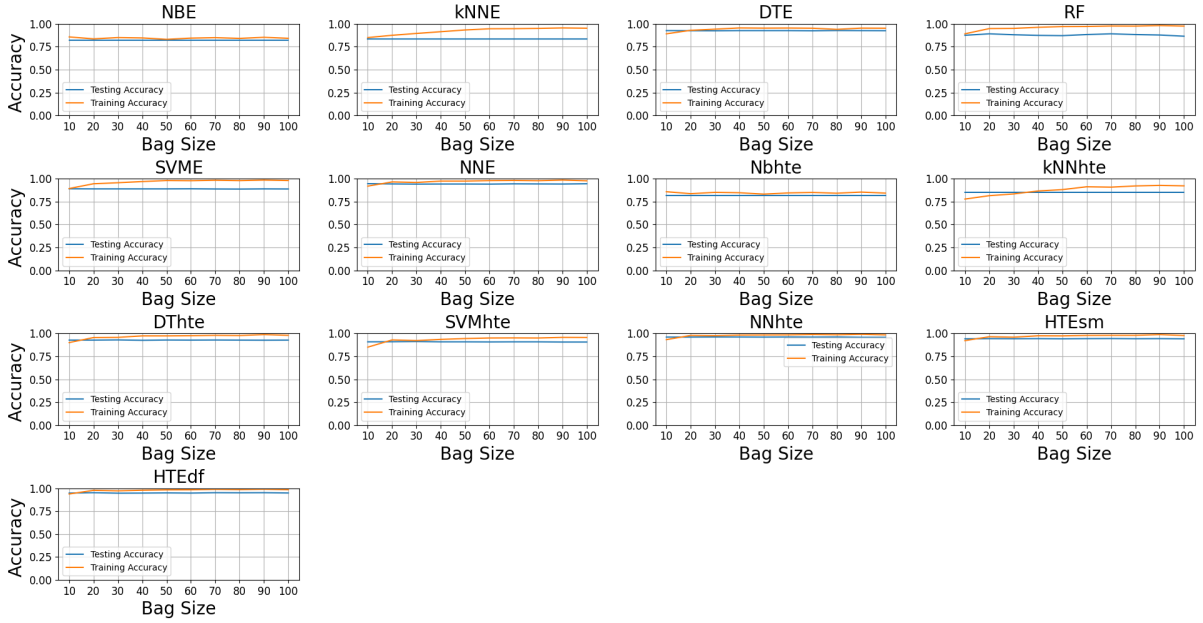

Figure D.6: Ensemble Performance on Bagged Subsets of the Car Evaluation Dataset

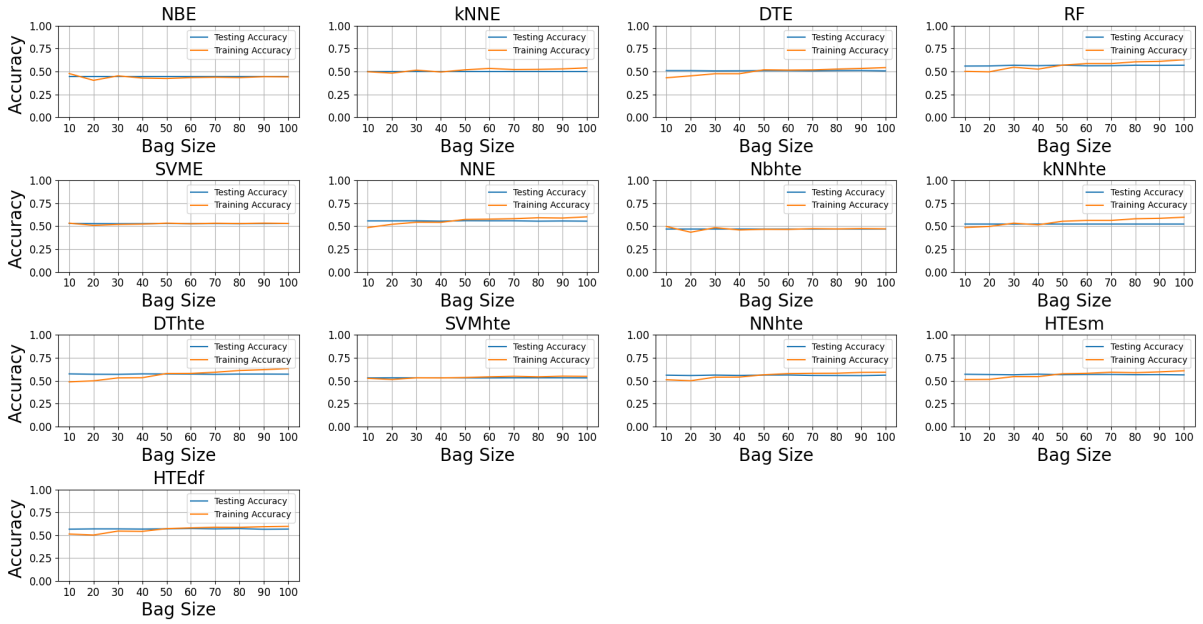

Figure D.7: Ensemble Performance on Bagged Subsets of the White Wine Dataset

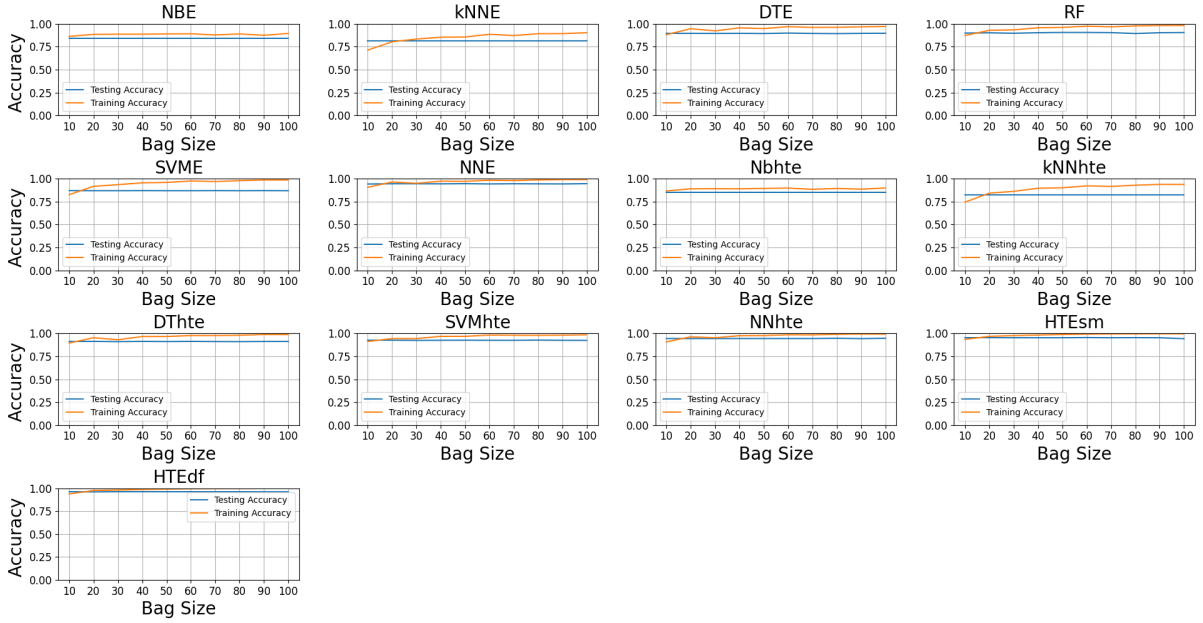

Figure D.8: Ensemble Performance on Bagged Subsets of the Nursery Dataset

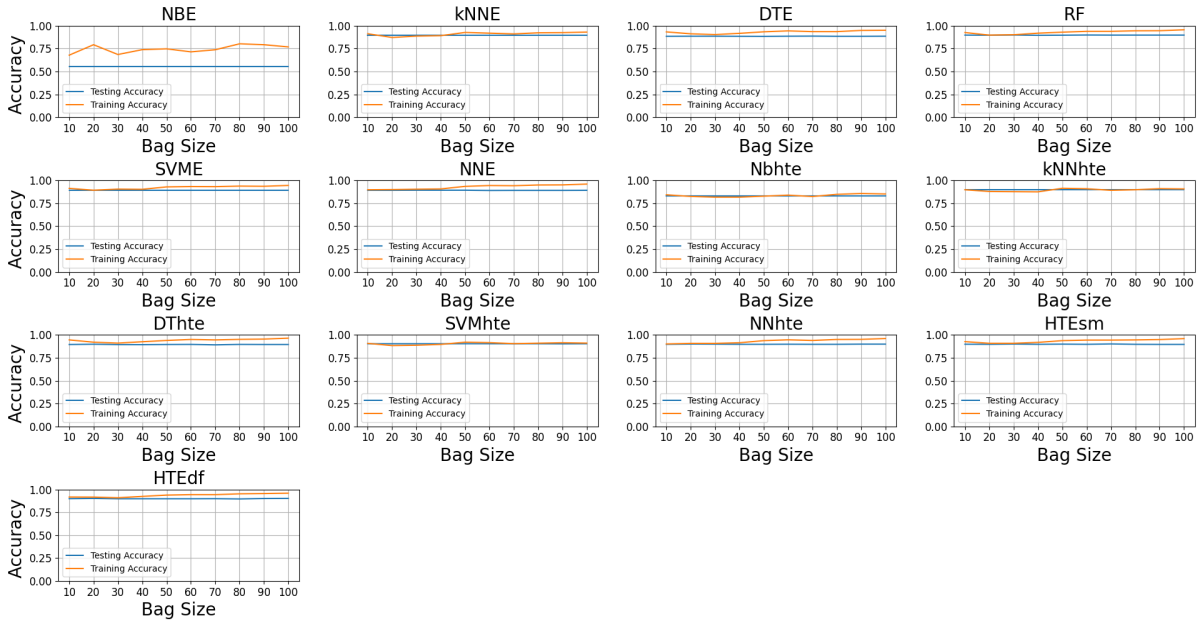

Figure D.9: Ensemble Performance on Bagged Subsets of the Bank Marketing Dataset

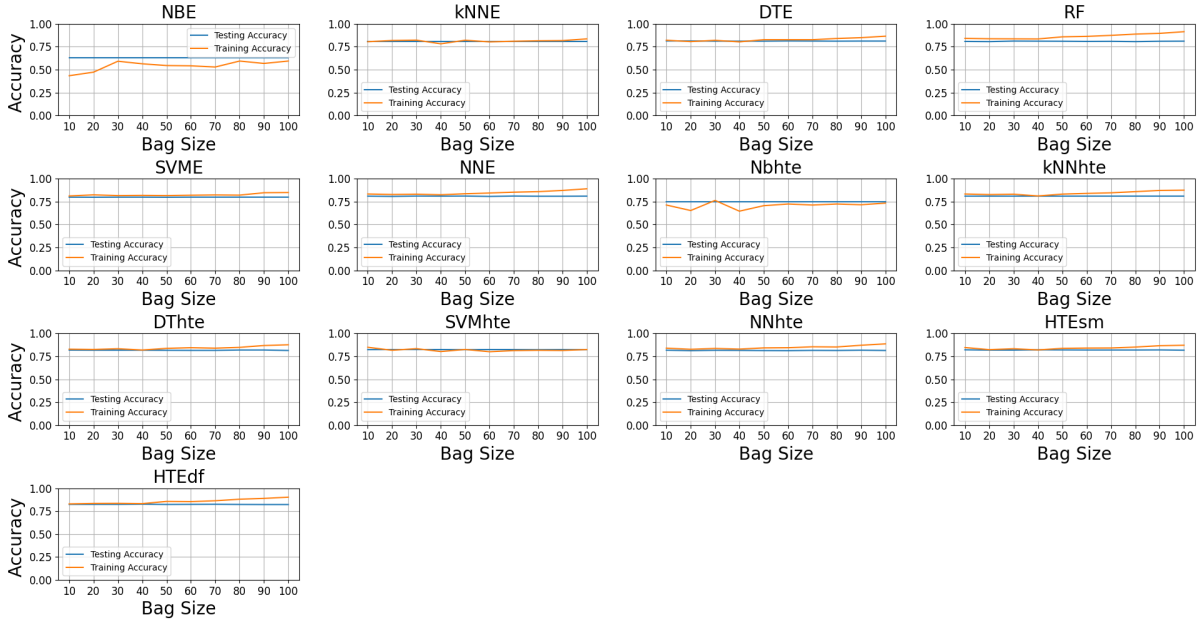

Figure D.10: Ensemble Performance on Bagged Subsets of the Censor Income Dataset

Sonar Dataset

Table D.1: Ensemble Performance on Bagged Subsets of the Sonar Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.747	0.747	0.747	0.747	0.747	0.747	0.747	0.747	0.747	0.747
	Training accuracy	0.867	0.723	0.672	0.718	0.723	0.711	0.761	0.759	0.792	0.768
	GF	1.902	0.913	0.771	0.897	0.913	0.875	1.059	1.050	1.216	1.091
	F1-Score	0.43	0.69	0.76	0.74	0.72	0.79	0.79	0.67	0.74	0.81
*k*NNE	Testing accuracy	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.586
	Training accuracy	0.683	0.590	0.641	0.767	0.784	0.748	0.773	0.781	0.811	0.791
	GF	1.306	1.010	1.153	1.777	1.917	1.643	1.824	1.890	2.190	1.981
	F1-Score	0.65	0.69	0.63	0.64	0.63	0.74	0.74	0.79	0.86	0.86
DTE	Testing accuracy	0.620	0.619	0.648	0.628	0.649	0.629	0.621	0.621	0.632	0.623
	Training accuracy	0.857	0.838	0.822	0.779	0.785	0.835	0.839	0.912	0.931	0.881
	GF	2.657	2.352	1.978	1.683	1.633	2.248	2.354	4.307	5.333	3.168
	F1-Score	0.60	0.69	0.72	0.74	0.86	0.62	0.69	0.69	0.81	0.67
RF	Testing accuracy	0.675	0.719	0.748	0.700	0.713	0.723	0.711	0.692	0.717	0.726
	Training accuracy	0.727	0.743	0.733	0.818	0.856	0.863	0.870	0.905	0.891	0.894
	GF	1.190	1.093	0.944	1.648	1.993	2.022	2.223	3.242	2.596	2.585
	F1-Score	0.55	0.69	0.83	0.81	0.78	0.67	0.74	0.72	0.76	0.74
SVME	Testing accuracy	0.725	0.728	0.743	0.730	0.724	0.730	0.715	0.720	0.715	0.744
	Training accuracy	0.827	0.765	0.785	0.821	0.868	0.887	0.908	0.919	0.911	0.892
	GF	1.590	1.157	1.195	1.508	2.091	2.389	3.098	3.457	3.202	2.370
	F1-Score	0.41	0.65	0.81	0.74	0.81	0.74	0.71	0.81	0.86	0.86
NNE	Testing accuracy	0.764	0.768	0.764	0.764	0.764	0.764	0.764	0.777	0.763	0.764
	Training accuracy	0.850	0.845	0.811	0.857	0.913	0.871	0.906	0.935	0.923	0.915
	GF	1.573	1.497	1.249	1.650	2.713	1.829	2.511	3.431	3.078	2.776
	F1-Score	0.45	0.77	0.88	0.81	0.86	0.88	0.81	0.79	0.86	0.87

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.762	0.762	0.762	0.762	0.762	0.762	0.762	0.762	0.762	0.762
	Training accuracy	0.867	0.733	0.671	0.718	0.725	0.720	0.759	0.762	0.798	0.772
	GF	1.789	0.891	0.723	0.844	0.865	0.850	0.988	1.000	1.178	1.044
	F1-Score	0.43	0.69	0.76	0.74	0.77	0.79	0.79	0.65	0.74	0.79
*k*NNhte	Testing accuracy	0.655	0.655	0.655	0.655	0.655	0.655	0.655	0.655	0.655	0.655
	Training accuracy	0.770	0.778	0.769	0.782	0.820	0.846	0.812	0.851	0.859	0.896
	GF	1.500	1.554	1.494	1.583	1.917	2.240	1.835	2.315	2.447	3.317
	F1-Score	0.62	0.71	0.73	0.74	0.83	0.81	0.74	0.77	0.87	0.87
DThte	Testing accuracy	0.678	0.667	0.676	0.704	0.707	0.674	0.670	0.677	0.685	0.672
	Training accuracy	0.867	0.815	0.827	0.795	0.827	0.866	0.870	0.924	0.931	0.908
	GF	2.421	1.800	1.873	1.444	1.694	2.433	2.538	4.250	4.565	3.565
	F1-Score	0.59	0.67	0.77	0.83	0.81	0.67	0.72	0.76	0.81	0.76
SVMhte	Testing accuracy	0.705	0.680	0.687	0.685	0.706	0.714	0.719	0.702	0.723	0.710
	Training accuracy	0.823	0.693	0.681	0.756	0.788	0.764	0.831	0.823	0.850	0.829
	GF	1.667	1.042	0.981	1.291	1.387	1.212	1.663	1.684	1.847	1.696
	F1-Score	0.47	0.67	0.81	0.69	0.79	0.72	0.69	0.79	0.81	0.79
NNhte	Testing accuracy	0.764	0.760	0.774	0.759	0.773	0.764	0.768	0.764	0.763	0.745
	Training accuracy	0.850	0.853	0.813	0.842	0.883	0.848	0.900	0.927	0.914	0.913
	GF	1.573	1.633	1.209	1.525	1.940	1.553	2.320	3.233	2.756	2.931
	F1-Score	0.55	0.74	0.90	0.79	0.83	0.83	0.79	0.86	0.86	0.86
HTEsm	Testing accuracy	0.805	0.786	0.796	0.807	0.800	0.785	0.814	0.777	0.796	0.799
	Training accuracy	0.817	0.837	0.850	0.845	0.894	0.848	0.908	0.945	0.923	0.913
	GF	1.066	1.313	1.360	1.245	1.887	1.414	2.022	4.055	2.649	2.310
	F1-Score	0.50	0.74	0.88	0.84	0.88	0.81	0.77	0.86	0.86	0.86
HTEdf	Testing accuracy	0.798	0.792	0.784	0.804	0.810	0.830	0.799	0.819	0.804	0.796
	Training accuracy	0.827	0.817	0.847	0.842	0.906	0.855	0.901	0.938	0.926	0.912
	GF	1.168	1.137	1.412	1.241	2.021	1.172	2.030	2.919	2.649	2.318
	F1-Score	0.53	0.74	0.88	0.84	0.88	0.91	0.82	0.88	0.88	0.88

Breast Cancer Dataset

Table D.2: Ensemble Performance on Bagged Subsets of the Breast Cancer Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.517	0.517	0.517	0.517	0.517	0.517	0.517	0.517	0.517	0.517
	Training accuracy	0.573	0.570	0.579	0.610	0.590	0.650	0.663	0.545	0.637	0.635
	GF	1.131	1.123	1.147	1.238	1.178	1.380	1.433	1.062	1.331	1.323
	F1-Score	0.59	0.49	0.52	0.23	0.40	0.52	0.55	0.43	0.51	0.43
*k*NNE	Testing accuracy	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666
	Training accuracy	0.783	0.697	0.653	0.743	0.736	0.727	0.805	0.757	0.779	0.837
	GF	1.539	1.102	0.963	1.300	1.265	1.223	1.713	1.374	1.511	2.049
	F1-Score	0.62	0.50	0.59	0.64	0.52	0.57	0.61	0.56	0.58	0.57
DTE	Testing accuracy	0.575	0.573	0.559	0.573	0.557	0.537	0.565	0.549	0.584	0.563
	Training accuracy	0.580	0.759	0.766	0.783	0.832	0.780	0.803	0.829	0.817	0.875
	GF	1.012	1.772	1.885	1.968	2.637	2.105	2.208	2.637	2.273	3.496
	F1-Score	0.58	0.64	0.60	0.67	0.62	0.66	0.61	0.66	0.65	0.57
RF	Testing accuracy	0.631	0.637	0.633	0.647	0.646	0.634	0.622	0.640	0.631	0.629
	Training accuracy	0.678	0.781	0.778	0.813	0.865	0.806	0.870	0.855	0.884	0.890
	GF	1.146	1.658	1.653	1.888	2.622	1.887	2.908	2.483	3.181	3.373
	F1-Score	0.56	0.67	0.62	0.57	0.64	0.58	0.61	0.60	0.62	0.61
SVME	Testing accuracy	0.637	0.633	0.630	0.630	0.630	0.627	0.630	0.630	0.630	0.627
	Training accuracy	0.435	0.786	0.869	0.855	0.857	0.825	0.887	0.894	0.890	0.911
	GF	0.642	1.715	2.824	2.552	2.587	2.131	3.274	3.491	3.364	4.191
	F1-Score	0.50	0.47	0.50	0.52	0.50	0.53	0.50	0.53	0.53	0.53
NNE	Testing accuracy	0.539	0.557	0.549	0.557	0.556	0.555	0.546	0.554	0.541	0.557
	Training accuracy	0.617	0.750	0.783	0.816	0.836	0.819	0.866	0.852	0.893	0.907
	GF	1.204	1.772	2.078	2.408	2.707	2.459	3.388	3.014	4.290	4.763
	F1-Score	0.58	0.56	0.65	0.69	0.64	0.55	0.64	0.66	0.56	0.63

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.518	0.518	0.518	0.518	0.518	0.518	0.518	0.518	0.518	0.518
	Training accuracy	0.582	0.636	0.577	0.657	0.680	0.680	0.680	0.655	0.683	0.650
	GF	1.153	1.324	1.139	1.405	1.506	1.506	1.506	1.397	1.521	1.377
	F1-Score	0.59	0.49	0.52	0.36	0.51	0.65	0.65	0.68	0.66	0.66
*k*NNhte	Testing accuracy	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635
	Training accuracy	0.742	0.754	0.772	0.831	0.830	0.783	0.863	0.837	0.855	0.890
	GF	1.415	1.484	1.601	2.160	2.147	1.682	2.664	2.239	2.517	3.318
	F1-Score	0.65	0.57	0.62	0.60	0.56	0.56	0.59	0.62	0.57	0.58
DThte	Testing accuracy	0.592	0.593	0.585	0.588	0.579	0.589	0.583	0.585	0.590	0.591
	Training accuracy	0.585	0.781	0.748	0.803	0.846	0.807	0.836	0.838	0.876	0.899
	GF	0.983	1.858	1.647	2.091	2.734	2.130	2.543	2.562	3.306	4.050
	F1-Score	0.61	0.67	0.60	0.65	0.62	0.66	0.62	0.58	0.67	0.61
SVMhte	Testing accuracy	0.630	0.637	0.633	0.630	0.630	0.630	0.630	0.633	0.633	0.630
	Training accuracy	0.558	0.808	0.751	0.811	0.831	0.783	0.860	0.815	0.794	0.855
	GF	0.837	1.891	1.474	1.958	2.189	1.705	2.643	1.984	1.782	2.552
	F1-Score	0.53	0.50	0.59	0.50	0.50	0.59	0.50	0.59	0.62	0.53
NNhte	Testing accuracy	0.580	0.581	0.578	0.587	0.584	0.563	0.583	0.579	0.567	0.567
	Training accuracy	0.625	0.756	0.791	0.808	0.839	0.818	0.854	0.855	0.893	0.902
	GF	1.120	1.717	2.019	2.151	2.584	2.401	2.856	2.903	4.047	4.418
	F1-Score	0.57	0.51	0.71	0.65	0.57	0.59	0.64	0.65	0.62	0.67
HTEsm	Testing accuracy	0.641	0.647	0.641	0.647	0.643	0.645	0.641	0.649	0.647	0.648
	Training accuracy	0.658	0.766	0.753	0.705	0.711	0.772	0.745	0.742	0.764	0.747
	GF	1.050	1.509	1.453	1.197	1.235	1.557	1.408	1.360	1.496	1.391
	F1-Score	0.60	0.61	0.67	0.65	0.69	0.67	0.65	0.69	0.65	0.65
HTEdf	Testing accuracy	0.654	0.654	0.655	0.653	0.658	0.655	0.650	0.652	0.651	0.654
	Training accuracy	0.647	0.702	0.750	0.705	0.704	0.760	0.742	0.746	0.759	0.720
	GF	0.980	1.161	1.380	1.176	1.155	1.438	1.357	1.370	1.448	1.236
	F1-Score	0.64	0.66	0.67	0.68	0.65	0.67	0.66	0.70	0.66	0.66

Indian Liver Dataset

Table D.3: Ensemble Performance on Bagged Subsets of the Indian Liver Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.665
	Training accuracy	0.693	0.734	0.654	0.679	0.592	0.675	0.713	0.649	0.710	0.674
	GF	1.091	1.259	0.968	1.044	0.821	1.031	1.167	0.954	1.155	1.028
	F1-Score	0.61	0.51	0.55	0.51	0.51	0.55	0.51	0.55	0.56	0.55
*k*NNE	Testing accuracy	0.730	0.730	0.730	0.730	0.730	0.730	0.730	0.730	0.730	0.730
	Training accuracy	0.583	0.691	0.629	0.668	0.672	0.685	0.705	0.719	0.724	0.721
	GF	0.647	0.874	0.728	0.813	0.823	0.857	0.915	0.961	0.978	0.968
	F1-Score	0.52	0.63	0.63	0.61	0.65	0.61	0.61	0.65	0.60	0.63
DTE	Testing accuracy	0.753	0.744	0.747	0.745	0.747	0.737	0.738	0.733	0.737	0.748
	Training accuracy	0.645	0.796	0.742	0.770	0.770	0.811	0.828	0.811	0.846	0.847
	GF	0.696	1.255	0.981	1.109	1.100	1.392	1.523	1.413	1.708	1.647
	F1-Score	0.72	0.69	0.65	0.66	0.61	0.67	0.68	0.73	0.63	0.74
RF	Testing accuracy	0.743	0.738	0.738	0.729	0.740	0.739	0.727	0.734	0.729	0.730
	Training accuracy	0.686	0.801	0.764	0.819	0.789	0.848	0.860	0.854	0.907	0.899
	GF	0.818	1.317	1.110	1.497	1.232	1.717	1.950	1.822	2.914	2.673
	F1-Score	0.67	0.73	0.68	0.73	0.72	0.71	0.72	0.73	0.73	0.80
SVME	Testing accuracy	0.750	0.743	0.741	0.746	0.744	0.746	0.740	0.744	0.743	0.741
	Training accuracy	0.477	0.745	0.643	0.671	0.658	0.690	0.702	0.693	0.738	0.697
	GF	0.478	1.008	0.725	0.772	0.749	0.819	0.872	0.834	0.981	0.855
	F1-Score	0.43	0.65	0.65	0.65	0.66	0.66	0.71	0.65	0.70	0.73
NNE	Testing accuracy	0.717	0.733	0.727	0.729	0.718	0.735	0.732	0.719	0.733	0.739
	Training accuracy	0.678	0.769	0.748	0.721	0.765	0.778	0.740	0.753	0.807	0.786
	GF	0.879	1.156	1.083	0.971	1.200	1.194	1.031	1.138	1.383	1.220
	F1-Score	0.59	0.69	0.67	0.66	0.69	0.72	0.66	0.67	0.70	0.66

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.668	0.668	0.668	0.668	0.668	0.668	0.668	0.668	0.668	0.668
	Training accuracy	0.693	0.732	0.649	0.678	0.592	0.674	0.719	0.650	0.712	0.673
	GF	1.081	1.239	0.946	1.031	0.814	1.018	1.181	0.949	1.153	1.015
	F1-Score	0.63	0.54	0.55	0.51	0.51	0.55	0.51	0.55	0.56	0.55
*k*NNhte	Testing accuracy	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727
	Training accuracy	0.658	0.736	0.723	0.760	0.707	0.767	0.809	0.794	0.835	0.839
	GF	0.798	1.034	0.986	1.138	0.932	1.172	1.429	1.325	1.655	1.696
	F1-Score	0.54	0.60	0.65	0.59	0.60	0.60	0.69	0.65	0.62	0.63
DThte	Testing accuracy	0.736	0.747	0.773	0.773	0.757	0.757	0.757	0.748	0.755	0.755
	Training accuracy	0.682	0.813	0.761	0.795	0.805	0.839	0.854	0.851	0.896	0.889
	GF	0.830	1.353	0.950	1.107	1.246	1.509	1.664	1.691	2.356	2.207
	F1-Score	0.69	0.69	0.65	0.69	0.73	0.69	0.69	0.75	0.67	0.72
SVMhte	Testing accuracy	0.748	0.745	0.747	0.745	0.743	0.745	0.745	0.745	0.743	0.745
	Training accuracy	0.480	0.757	0.522	0.603	0.584	0.697	0.672	0.677	0.670	0.646
	GF	0.485	1.049	0.529	0.642	0.618	0.842	0.777	0.789	0.779	0.720
	F1-Score	0.63	0.65	0.40	0.71	0.66	0.61	0.68	0.67	0.67	0.68
NNhte	Testing accuracy	0.766	0.753	0.764	0.755	0.764	0.759	0.767	0.764	0.760	0.754
	Training accuracy	0.651	0.757	0.713	0.716	0.718	0.740	0.742	0.738	0.758	0.744
	GF	0.670	1.016	0.822	0.863	0.837	0.927	0.903	0.901	0.992	0.961
	F1-Score	0.51	0.65	0.74	0.64	0.75	0.74	0.67	0.69	0.65	0.65
HTEsm	Testing accuracy	0.742	0.747	0.746	0.749	0.746	0.741	0.751	0.735	0.754	0.740
	Training accuracy	0.703	0.787	0.752	0.759	0.768	0.782	0.772	0.786	0.797	0.807
	GF	0.869	1.188	1.024	1.041	1.095	1.188	1.092	1.238	1.212	1.347
	F1-Score	0.62	0.65	0.67	0.65	0.68	0.66	0.68	0.71	0.67	0.69
HTEdf	Testing accuracy	0.776	0.755	0.758	0.767	0.781	0.774	0.770	0.764	0.763	0.771
	Training accuracy	0.717	0.777	0.759	0.786	0.789	0.824	0.804	0.817	0.838	0.852
	GF	0.792	1.099	1.004	1.089	1.038	1.284	1.173	1.290	1.463	1.547
	F1-Score	0.63	0.65	0.68	0.67	0.70	0.68	0.68	0.73	0.72	0.73

Credit Approval Dataset

Table D.4: Ensemble Performance on Bagged Subsets of the Credit Approval Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535
	Training accuracy	0.596	0.447	0.391	0.511	0.608	0.635	0.536	0.530	0.521	0.525
	GF	1.151	0.841	0.764	0.951	1.186	1.274	1.002	0.989	0.971	0.979
	F1-Score	0.53	0.57	0.48	0.42	0.55	0.62	0.47	0.44	0.45	0.47
*k*NNE	Testing accuracy	0.788	0.788	0.788	0.788	0.788	0.788	0.788	0.788	0.788	0.788
	Training accuracy	0.914	0.851	0.876	0.878	0.882	0.883	0.874	0.869	0.863	0.869
	GF	2.465	1.423	1.710	1.738	1.797	1.812	1.683	1.618	1.547	1.618
	F1-Score	0.80	0.84	0.83	0.83	0.83	0.83	0.81	0.81	0.82	0.82
DTE	Testing accuracy	0.758	0.763	0.761	0.757	0.761	0.757	0.757	0.763	0.760	0.757
	Training accuracy	0.797	0.787	0.874	0.855	0.875	0.888	0.839	0.838	0.826	0.843
	GF	1.192	1.113	1.897	1.676	1.912	2.170	1.509	1.463	1.379	1.548
	F1-Score	0.78	0.84	0.75	0.80	0.80	0.83	0.81	0.79	0.79	0.80
RF	Testing accuracy	0.795	0.782	0.789	0.799	0.787	0.802	0.797	0.812	0.793	0.777
	Training accuracy	0.867	0.836	0.874	0.873	0.877	0.885	0.871	0.862	0.866	0.870
	GF	1.541	1.329	1.675	1.583	1.732	1.722	1.574	1.362	1.545	1.715
	F1-Score	0.75	0.80	0.81	0.84	0.82	0.83	0.84	0.83	0.82	0.83
SVME	Testing accuracy	0.778	0.775	0.771	0.778	0.775	0.777	0.777	0.771	0.774	0.777
	Training accuracy	0.625	0.720	0.843	0.854	0.856	0.863	0.856	0.854	0.852	0.858
	GF	0.592	0.804	1.459	1.521	1.562	1.628	1.549	1.568	1.527	1.570
	F1-Score	0.63	0.72	0.78	0.80	0.83	0.85	0.84	0.83	0.82	0.82
NNE	Testing accuracy	0.672	0.675	0.670	0.671	0.677	0.684	0.675	0.671	0.674	0.680
	Training accuracy	0.831	0.753	0.763	0.792	0.832	0.816	0.820	0.809	0.800	0.803
	GF	1.941	1.316	1.392	1.582	1.923	1.717	1.806	1.723	1.630	1.624
	F1-Score	0.72	0.72	0.76	0.78	0.78	0.73	0.77	0.71	0.74	0.73

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651
	Training accuracy	0.681	0.458	0.506	0.607	0.738	0.712	0.699	0.679	0.644	0.657
	GF	1.094	0.644	0.706	0.888	1.332	1.212	1.159	1.087	0.980	1.017
	F1-Score	0.62	0.58	0.61	0.55	0.68	0.64	0.66	0.63	0.62	0.64
*k*NNhte	Testing accuracy	0.808	0.808	0.808	0.808	0.808	0.808	0.808	0.808	0.808	0.808
	Training accuracy	0.914	0.855	0.869	0.874	0.875	0.882	0.874	0.871	0.854	0.863
	GF	2.233	1.324	1.466	1.524	1.536	1.627	1.524	1.488	1.315	1.401
	F1-Score	0.80	0.84	0.83	0.83	0.84	0.84	0.82	0.82	0.82	0.82
DThte	Testing accuracy	0.764	0.762	0.755	0.761	0.768	0.760	0.762	0.755	0.758	0.767
	Training accuracy	0.856	0.804	0.884	0.863	0.877	0.888	0.854	0.859	0.853	0.857
	GF	1.639	1.214	2.112	1.745	1.886	2.143	1.630	1.738	1.646	1.629
	F1-Score	0.79	0.84	0.82	0.82	0.80	0.84	0.83	0.82	0.82	0.81
SVMhte	Testing accuracy	0.822	0.813	0.821	0.813	0.818	0.818	0.816	0.821	0.825	0.816
	Training accuracy	0.896	0.864	0.872	0.881	0.888	0.891	0.880	0.880	0.870	0.871
	GF	1.712	1.375	1.398	1.571	1.625	1.670	1.533	1.492	1.346	1.426
	F1-Score	0.78	0.84	0.82	0.83	0.82	0.83	0.85	0.84	0.83	0.84
NNhte	Testing accuracy	0.761	0.771	0.768	0.764	0.765	0.772	0.764	0.771	0.772	0.762
	Training accuracy	0.845	0.793	0.835	0.834	0.869	0.861	0.854	0.855	0.833	0.842
	GF	1.542	1.106	1.406	1.422	1.794	1.640	1.616	1.579	1.365	1.506
	F1-Score	0.74	0.80	0.82	0.78	0.80	0.75	0.82	0.79	0.75	0.80
HTEsm	Testing accuracy	0.773	0.773	0.770	0.777	0.768	0.773	0.761	0.770	0.768	0.777
	Training accuracy	0.837	0.815	0.835	0.852	0.888	0.878	0.863	0.856	0.850	0.852
	GF	1.393	1.227	1.394	1.507	2.071	1.861	1.745	1.597	1.547	1.507
	F1-Score	0.81	0.81	0.82	0.80	0.82	0.84	0.83	0.82	0.82	0.81
HTEdf	Testing accuracy	0.800	0.804	0.801	0.800	0.803	0.800	0.801	0.796	0.800	0.803
	Training accuracy	0.849	0.833	0.877	0.874	0.894	0.897	0.880	0.883	0.873	0.883
	GF	1.325	1.174	1.618	1.587	1.858	1.942	1.658	1.744	1.575	1.684
	F1-Score	0.81	0.85	0.84	0.84	0.82	0.86	0.85	0.84	0.83	0.84

Red Wine Dataset

Table D.5: Ensemble Performance on Bagged Subsets of the Red Wine Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.468	0.468	0.468	0.468	0.468	0.468	0.468	0.468	0.468	0.468
	Training accuracy	0.585	0.548	0.431	0.602	0.501	0.569	0.556	0.559	0.592	0.551
	GF	1.282	1.177	0.935	1.337	1.066	1.234	1.198	1.206	1.304	1.185
	F1-Score	0.46	0.55	0.47	0.49	0.52	0.52	0.54	0.47	0.52	0.55
*k*NNE	Testing accuracy	0.551	0.551	0.551	0.551	0.551	0.551	0.551	0.551	0.551	0.551
	Training accuracy	0.613	0.621	0.637	0.702	0.690	0.719	0.728	0.731	0.759	0.781
	GF	1.160	1.185	1.237	1.507	1.448	1.598	1.651	1.669	1.863	2.050
	F1-Score	0.48	0.54	0.54	0.52	0.52	0.58	0.56	0.54	0.56	0.58
DTE	Testing accuracy	0.490	0.499	0.490	0.495	0.493	0.488	0.498	0.496	0.498	0.490
	Training accuracy	0.628	0.602	0.633	0.686	0.684	0.729	0.739	0.769	0.768	0.775
	GF	1.371	1.259	1.390	1.608	1.604	1.889	1.923	2.182	2.164	2.267
	F1-Score	0.49	0.49	0.50	0.52	0.50	0.55	0.58	0.50	0.55	0.55
RF	Testing accuracy	0.570	0.580	0.572	0.576	0.573	0.563	0.588	0.575	0.560	0.579
	Training accuracy	0.526	0.640	0.677	0.718	0.728	0.787	0.800	0.810	0.826	0.847
	GF	0.907	1.167	1.325	1.504	1.570	2.052	2.060	2.237	2.529	2.752
	F1-Score	0.49	0.54	0.58	0.56	0.57	0.59	0.60	0.59	0.58	0.58
SVME	Testing accuracy	0.536	0.546	0.540	0.537	0.545	0.531	0.540	0.540	0.541	0.540
	Training accuracy	0.600	0.562	0.559	0.638	0.576	0.616	0.634	0.601	0.611	0.608
	GF	1.160	1.037	1.043	1.279	1.073	1.221	1.257	1.153	1.180	1.173
	F1-Score	0.50	0.44	0.49	0.52	0.54	0.52	0.53	0.51	0.51	0.56
NNE	Testing accuracy	0.566	0.564	0.562	0.571	0.565	0.569	0.568	0.564	0.566	0.562
	Training accuracy	0.665	0.569	0.671	0.647	0.710	0.618	0.749	0.629	0.768	0.631
	GF	1.296	1.012	1.331	1.215	1.500	1.128	1.721	1.175	1.871	1.187
	F1-Score	0.52	0.55	0.54	0.55	0.56	0.58	0.57	0.57	0.55	0.59

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.503	0.503	0.503	0.503	0.503	0.503	0.503	0.503	0.503	0.503
	Training accuracy	0.626	0.567	0.474	0.612	0.531	0.588	0.572	0.573	0.598	0.559
	GF	1.329	1.148	0.945	1.281	1.060	1.206	1.161	1.164	1.236	1.127
	F1-Score	0.47	0.55	0.50	0.49	0.53	0.50	0.53	0.49	0.51	0.55
*k*NNhte	Testing accuracy	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552	0.552
	Training accuracy	0.605	0.583	0.536	0.628	0.586	0.596	0.629	0.619	0.648	0.655
	GF	1.134	1.074	0.966	1.204	1.082	1.109	1.208	1.176	1.273	1.299
	F1-Score	0.50	0.52	0.54	0.51	0.52	0.51	0.54	0.52	0.51	0.52
DThte	Testing accuracy	0.553	0.561	0.560	0.556	0.564	0.557	0.559	0.546	0.552	0.556
	Training accuracy	0.619	0.643	0.685	0.733	0.746	0.796	0.808	0.822	0.836	0.855
	GF	1.173	1.230	1.397	1.663	1.717	2.172	2.297	2.551	2.732	3.062
	F1-Score	0.51	0.51	0.57	0.52	0.56	0.58	0.58	0.57	0.57	0.55
SVMhte	Testing accuracy	0.575	0.572	0.575	0.574	0.576	0.575	0.573	0.574	0.574	0.575
	Training accuracy	0.574	0.655	0.585	0.657	0.621	0.630	0.651	0.623	0.644	0.626
	GF	0.998	1.241	1.024	1.242	1.119	1.149	1.223	1.130	1.197	1.136
	F1-Score	0.50	0.53	0.56	0.51	0.52	0.53	0.55	0.52	0.51	0.57
NNhte	Testing accuracy	0.575	0.576	0.566	0.578	0.565	0.570	0.575	0.566	0.570	0.573
	Training accuracy	0.650	0.553	0.686	0.615	0.709	0.597	0.744	0.608	0.762	0.614
	GF	1.214	0.949	1.382	1.096	1.495	1.067	1.660	1.107	1.807	1.106
	F1-Score	0.53	0.54	0.55	0.54	0.54	0.55	0.56	0.60	0.52	0.56
HTEsm	Testing accuracy	0.557	0.558	0.567	0.568	0.563	0.560	0.562	0.563	0.565	0.559
	Training accuracy	0.673	0.664	0.670	0.689	0.706	0.682	0.743	0.724	0.760	0.739
	GF	1.355	1.315	1.312	1.389	1.486	1.384	1.704	1.583	1.813	1.690
	F1-Score	0.53	0.57	0.56	0.55	0.55	0.58	0.59	0.55	0.55	0.58
HTEdf	Testing accuracy	0.585	0.586	0.587	0.581	0.585	0.582	0.579	0.581	0.579	0.575
	Training accuracy	0.676	0.656	0.668	0.696	0.717	0.707	0.744	0.730	0.760	0.763
	GF	1.281	1.203	1.244	1.378	1.466	1.427	1.645	1.552	1.754	1.793
	F1-Score	0.53	0.56	0.57	0.55	0.56	0.58	0.58	0.56	0.56	0.60

Car Evaluation Dataset

Table D.6: Ensemble Performance on Bagged Subsets of the Car Evaluation Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.819	0.819	0.819	0.819	0.819	0.819	0.819	0.819	0.819	0.819
	Training accuracy	0.857	0.836	0.850	0.846	0.831	0.844	0.849	0.841	0.853	0.842
	GF	1.266	1.104	1.207	1.175	1.071	1.160	1.199	1.138	1.231	1.146
	F1-Score	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
*k*NNE	Testing accuracy	0.834	0.834	0.834	0.834	0.834	0.834	0.834	0.834	0.834	0.834
	Training accuracy	0.850	0.875	0.895	0.914	0.933	0.946	0.947	0.950	0.955	0.952
	GF	1.107	1.328	1.581	1.930	2.478	3.074	3.132	3.320	3.689	3.458
	F1-Score	0.65	0.66	0.69	0.75	0.74	0.76	0.76	0.79	0.81	0.79
DTE	Testing accuracy	0.926	0.925	0.924	0.926	0.926	0.926	0.924	0.927	0.926	0.925
	Training accuracy	0.891	0.929	0.942	0.955	0.953	0.954	0.952	0.940	0.953	0.952
	GF	0.679	1.056	1.310	1.644	1.574	1.609	1.583	1.217	1.574	1.562
	F1-Score	0.90	0.86	0.93	0.93	0.92	0.93	0.95	0.91	0.93	0.93
RF	Testing accuracy	0.875	0.891	0.881	0.874	0.871	0.883	0.891	0.883	0.878	0.865
	Training accuracy	0.891	0.948	0.950	0.962	0.970	0.971	0.976	0.975	0.981	0.975
	GF	1.147	2.096	2.380	3.316	4.300	4.034	4.542	4.680	6.421	5.400
	F1-Score	0.84	0.88	0.90	0.92	0.90	0.93	0.94	0.91	0.90	0.90
SVME	Testing accuracy	0.889	0.888	0.888	0.888	0.888	0.889	0.887	0.886	0.888	0.887
	Training accuracy	0.892	0.944	0.955	0.968	0.979	0.977	0.982	0.978	0.984	0.979
	GF	1.028	2.000	2.489	3.500	5.333	4.826	6.278	5.182	7.000	5.381
	F1-Score	0.80	0.88	0.87	0.90	0.88	0.87	0.90	0.88	0.87	0.89
NNE	Testing accuracy	0.945	0.942	0.940	0.941	0.941	0.940	0.943	0.942	0.941	0.944
	Training accuracy	0.918	0.964	0.958	0.973	0.973	0.978	0.980	0.977	0.984	0.976
	GF	0.671	1.611	1.429	2.185	2.185	2.727	2.850	2.522	3.688	2.333
	F1-Score	0.90	0.94	0.93	0.93	0.94	0.94	0.95	0.93	0.95	0.95

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.819	0.819	0.819	0.819	0.819	0.819	0.819	0.819	0.819	0.819
	Training accuracy	0.857	0.836	0.850	0.846	0.831	0.844	0.849	0.841	0.853	0.842
	GF	1.266	1.104	1.207	1.175	1.071	1.160	1.199	1.138	1.231	1.146
	F1-Score	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
*k*NNhte	Testing accuracy	0.852	0.852	0.852	0.852	0.852	0.852	0.852	0.852	0.852	0.852
	Training accuracy	0.777	0.814	0.833	0.865	0.880	0.911	0.907	0.920	0.926	0.921
	GF	0.664	0.796	0.886	1.096	1.233	1.663	1.591	1.850	2.000	1.873
	F1-Score	0.69	0.74	0.75	0.77	0.79	0.80	0.80	0.80	0.81	0.80
DThte	Testing accuracy	0.927	0.927	0.929	0.925	0.928	0.927	0.928	0.927	0.926	0.927
	Training accuracy	0.900	0.955	0.957	0.973	0.973	0.975	0.979	0.977	0.987	0.979
	GF	0.730	1.622	1.651	2.778	2.667	2.920	3.429	3.174	5.692	3.476
	F1-Score	0.86	0.88	0.94	0.94	0.91	0.95	0.93	0.93	0.95	0.94
SVMhte	Testing accuracy	0.908	0.909	0.911	0.908	0.908	0.907	0.908	0.908	0.906	0.906
	Training accuracy	0.850	0.929	0.922	0.935	0.944	0.950	0.951	0.950	0.957	0.956
	GF	0.613	1.282	1.141	1.415	1.643	1.860	1.878	1.840	2.186	2.136
	F1-Score	0.82	0.90	0.89	0.90	0.92	0.93	0.93	0.94	0.93	0.95
NNhte	Testing accuracy	0.960	0.960	0.961	0.960	0.960	0.961	0.960	0.961	0.958	0.958
	Training accuracy	0.932	0.978	0.975	0.982	0.983	0.984	0.988	0.985	0.990	0.984
	GF	0.588	1.818	1.560	2.167	2.353	2.438	3.333	2.600	4.200	2.625
	F1-Score	0.93	0.95	0.98	0.97	0.97	0.98	0.98	0.97	0.98	0.98
HTEsm	Testing accuracy	0.941	0.943	0.942	0.943	0.941	0.943	0.944	0.942	0.943	0.941
	Training accuracy	0.921	0.964	0.959	0.974	0.973	0.978	0.979	0.979	0.987	0.977
	GF	0.747	1.583	1.415	2.192	2.185	2.591	2.667	2.762	4.385	2.565
	F1-Score	0.90	0.93	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96
HTEdf	Testing accuracy	0.947	0.951	0.946	0.947	0.949	0.947	0.951	0.950	0.951	0.948
	Training accuracy	0.937	0.977	0.971	0.978	0.982	0.983	0.988	0.985	0.989	0.984
	GF	0.841	2.130	1.862	2.409	2.833	3.118	4.083	3.333	4.455	3.250
	F1-Score	0.92	0.95	0.98	0.97	0.97	0.98	0.98	0.97	0.97	0.97

White Wine Dataset

Table D.7: Ensemble Performance on Bagged Subsets of the White Wine Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.448	0.448	0.448	0.448	0.448	0.448	0.448	0.448	0.448	0.448
	Training accuracy	0.478	0.404	0.453	0.429	0.424	0.434	0.438	0.434	0.444	0.442
	GF	1.057	0.926	1.009	0.967	0.958	0.975	0.982	0.975	0.993	0.989
	F1-Score	0.44	0.38	0.41	0.39	0.40	0.41	0.42	0.40	0.41	0.41
*k*NNE	Testing accuracy	0.503	0.503	0.503	0.503	0.503	0.503	0.503	0.503	0.503	0.503
	Training accuracy	0.498	0.483	0.515	0.495	0.519	0.534	0.522	0.524	0.529	0.540
	GF	0.990	0.961	1.025	0.984	1.033	1.067	1.040	1.044	1.055	1.080
	F1-Score	0.47	0.49	0.49	0.50	0.51	0.51	0.53	0.53	0.54	0.54
DTE	Testing accuracy	0.510	0.510	0.507	0.508	0.510	0.510	0.508	0.510	0.511	0.507
	Training accuracy	0.432	0.453	0.476	0.476	0.519	0.516	0.518	0.527	0.534	0.543
	GF	0.863	0.896	0.941	0.939	1.019	1.012	1.021	1.036	1.049	1.079
	F1-Score	0.45	0.47	0.48	0.51	0.50	0.54	0.54	0.55	0.55	0.54
RF	Testing accuracy	0.560	0.561	0.569	0.564	0.570	0.563	0.564	0.569	0.568	0.569
	Training accuracy	0.502	0.497	0.547	0.526	0.570	0.587	0.587	0.606	0.611	0.629
	GF	0.884	0.873	0.951	0.920	1.00	1.058	1.056	1.094	1.111	1.162
	F1-Score	0.49	0.51	0.54	0.57	0.57	0.59	0.61	0.62	0.62	0.62
SVME	Testing accuracy	0.529	0.527	0.526	0.527	0.529	0.527	0.528	0.527	0.528	0.529
	Training accuracy	0.532	0.508	0.518	0.522	0.532	0.526	0.531	0.528	0.532	0.529
	GF	1.006	0.961	0.983	0.990	1.006	0.998	1.006	1.002	1.009	1.000
	F1-Score	0.45	0.49	0.45	0.48	0.48	0.51	0.49	0.51	0.49	0.50
NNE	Testing accuracy	0.557	0.557	0.558	0.554	0.559	0.558	0.558	0.554	0.556	0.554
	Training accuracy	0.484	0.520	0.542	0.541	0.573	0.576	0.581	0.591	0.588	0.601
	GF	0.859	0.923	0.965	0.972	1.033	1.042	1.055	1.090	1.078	1.118
	F1-Score	0.51	0.51	0.54	0.56	0.57	0.58	0.59	0.59	0.59	0.60

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.470	0.470	0.470	0.470	0.470	0.470	0.470	0.470	0.470	0.47
	Training accuracy	0.496	0.433	0.483	0.458	0.465	0.464	0.472	0.469	0.473	0.47
	GF	1.052	0.935	1.025	0.978	0.991	0.989	1.004	0.998	1.006	1.00
	F1-Score	0.47	0.44	0.45	0.44	0.46	0.46	0.46	0.45	0.45	0.46
*k*NNhte	Testing accuracy	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520	0.520
	Training accuracy	0.485	0.496	0.532	0.514	0.553	0.562	0.562	0.580	0.585	0.597
	GF	0.932	0.952	1.026	0.988	1.074	1.096	1.096	1.143	1.157	1.191
	F1-Score	0.48	0.51	0.53	0.54	0.55	0.57	0.58	0.6	0.60	0.60
DThte	Testing accuracy	0.574	0.570	0.569	0.574	0.573	0.572	0.569	0.572	0.572	0.571
	Training accuracy	0.487	0.498	0.530	0.532	0.578	0.579	0.592	0.611	0.620	0.632
	GF	0.830	0.857	0.917	0.910	1.012	1.017	1.056	1.100	1.126	1.166
	F1-Score	0.51	0.50	0.55	0.57	0.57	0.60	0.61	0.63	0.63	0.63
SVMhte	Testing accuracy	0.530	0.531	0.531	0.531	0.530	0.530	0.531	0.531	0.531	0.530
	Training accuracy	0.525	0.513	0.531	0.530	0.534	0.542	0.548	0.542	0.549	0.547
	GF	0.989	0.963	1.000	0.998	1.009	1.026	1.038	1.024	1.040	1.038
	F1-Score	0.46	0.50	0.49	0.50	0.49	0.51	0.51	0.52	0.51	0.51
NNhte	Testing accuracy	0.560	0.556	0.561	0.557	0.561	0.562	0.557	0.556	0.555	0.560
	Training accuracy	0.510	0.499	0.538	0.539	0.564	0.577	0.580	0.581	0.590	0.592
	GF	0.898	0.886	0.950	0.961	1.007	1.035	1.055	1.060	1.085	1.078
	F1-Score	0.51	0.51	0.53	0.54	0.55	0.57	0.57	0.57	0.56	0.57
HTEsm	Testing accuracy	0.570	0.567	0.564	0.571	0.565	0.568	0.568	0.565	0.566	0.563
	Training accuracy	0.512	0.514	0.544	0.544	0.575	0.580	0.592	0.587	0.596	0.608
	GF	0.881	0.891	0.956	0.941	1.024	1.029	1.059	1.053	1.074	1.115
	F1-Score	0.51	0.51	0.54	0.55	0.55	0.59	0.58	0.59	0.58	0.60
HTEdf	Testing accuracy	0.566	0.569	0.569	0.567	0.570	0.573	0.569	0.572	0.565	0.567
	Training accuracy	0.513	0.502	0.545	0.542	0.572	0.580	0.587	0.586	0.593	0.597
	GF	0.891	0.865	0.947	0.945	1.005	1.017	1.044	1.034	1.069	1.074
	F1-Score	0.52	0.52	0.55	0.55	0.56	0.59	0.59	0.60	0.59	0.59

Nursery Dataset

Table D.8: Ensemble Performance on Bagged Subsets of the Nursery Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842
	Training accuracy	0.863	0.885	0.887	0.887	0.890	0.891	0.879	0.890	0.875	0.895
	GF	1.153	1.374	1.398	1.398	1.436	1.450	1.306	1.436	1.264	1.505
	F1-Score	0.83	0.87	0.85	0.87	0.87	0.85	0.87	0.86	0.86	0.86
*k*NNE	Testing accuracy	0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816
	Training accuracy	0.713	0.804	0.834	0.854	0.856	0.886	0.872	0.892	0.893	0.903
	GF	0.641	0.939	1.108	1.260	1.278	1.614	1.438	1.704	1.720	1.897
	F1-Score	0.76	0.74	0.74	0.79	0.82	0.80	0.81	0.80	0.82	0.83
DTE	Testing accuracy	0.896	0.896	0.895	0.896	0.894	0.899	0.895	0.893	0.896	0.897
	Training accuracy	0.881	0.947	0.923	0.956	0.948	0.969	0.961	0.962	0.967	0.971
	GF	0.874	1.962	1.364	2.364	2.038	3.258	2.692	2.816	3.152	3.552
	F1-Score	0.89	0.93	0.92	0.92	0.94	0.94	0.94	0.95	0.94	0.94
RF	Testing accuracy	0.899	0.902	0.897	0.903	0.906	0.906	0.904	0.894	0.903	0.905
	Training accuracy	0.872	0.929	0.934	0.957	0.961	0.974	0.967	0.977	0.981	0.982
	GF	0.789	1.380	1.561	2.256	2.410	3.615	2.909	4.609	5.105	5.278
	F1-Score	0.87	0.91	0.89	0.92	0.94	0.94	0.94	0.93	0.95	0.94
SVME	Testing accuracy	0.869	0.868	0.868	0.869	0.868	0.868	0.869	0.868	0.869	0.868
	Training accuracy	0.825	0.916	0.934	0.954	0.958	0.974	0.968	0.978	0.985	0.985
	GF	0.749	1.571	2.000	2.848	3.143	5.077	4.094	6.000	8.733	8.800
	F1-Score	0.83	0.85	0.88	0.91	0.92	0.92	0.92	0.93	0.94	0.94
NNE	Testing accuracy	0.941	0.944	0.944	0.943	0.945	0.942	0.944	0.943	0.942	0.945
	Training accuracy	0.905	0.963	0.949	0.972	0.970	0.982	0.978	0.986	0.990	0.990
	GF	0.621	1.514	1.098	2.036	1.833	3.222	2.545	4.071	5.800	5.500
	F1-Score	0.92	0.95	0.94	0.96	0.97	0.97	0.96	0.97	0.98	0.98

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.847	0.847	0.847	0.847	0.847	0.847	0.847	0.847	0.847	0.847
	Training accuracy	0.866	0.889	0.890	0.889	0.893	0.897	0.883	0.893	0.885	0.898
	GF	1.142	1.378	1.391	1.378	1.430	1.485	1.308	1.430	1.330	1.500
	F1-Score	0.84	0.87	0.86	0.87	0.87	0.86	0.87	0.87	0.87	0.86
*k*NNhte	Testing accuracy	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820
	Training accuracy	0.743	0.841	0.861	0.895	0.900	0.921	0.915	0.928	0.937	0.936
	GF	0.700	1.132	1.295	1.714	1.800	2.278	2.118	2.500	2.857	2.813
	F1-Score	0.79	0.80	0.78	0.80	0.83	0.82	0.82	0.82	0.84	0.82
DThte	Testing accuracy	0.912	0.913	0.910	0.913	0.911	0.913	0.911	0.91	0.912	0.912
	Training accuracy	0.895	0.953	0.931	0.966	0.966	0.977	0.977	0.98	0.988	0.987
	GF	0.838	1.851	1.304	2.559	2.618	3.783	3.870	4.50	7.333	6.769
	F1-Score	0.87	0.93	0.92	0.95	0.96	0.95	0.95	0.98	0.96	0.96
SVMhte	Testing accuracy	0.926	0.926	0.924	0.925	0.926	0.925	0.925	0.927	0.925	0.924
	Training accuracy	0.912	0.944	0.944	0.968	0.968	0.982	0.980	0.979	0.981	0.985
	GF	0.841	1.321	1.357	2.344	2.312	4.167	3.750	3.476	3.947	5.067
	F1-Score	0.86	0.93	0.90	0.94	0.95	0.95	0.96	0.96	0.96	0.96
NNhte	Testing accuracy	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.947	0.943	0.947
	Training accuracy	0.907	0.963	0.953	0.975	0.977	0.985	0.984	0.990	0.993	0.992
	GF	0.602	1.514	1.191	2.240	2.435	3.733	3.500	5.300	8.143	6.625
	F1-Score	0.92	0.95	0.94	0.97	0.97	0.97	0.97	0.98	0.98	0.98
HTEsm	Testing accuracy	0.954	0.954	0.953	0.953	0.953	0.955	0.953	0.954	0.953	0.942
	Training accuracy	0.935	0.968	0.976	0.982	0.988	0.992	0.994	0.995	0.996	0.996
	GF	0.708	1.438	1.958	2.611	3.917	5.625	7.833	9.200	11.750	14.500
	F1-Score	0.93	0.95	0.97	0.98	0.98	0.99	0.99	0.99	1.00	1.00
HTEdf	Testing accuracy	0.962	0.960	0.962	0.962	0.962	0.962	0.962	0.962	0.961	0.961
	Training accuracy	0.937	0.975	0.980	0.986	0.990	0.994	0.995	0.996	0.997	0.998
	GF	0.603	1.600	1.900	2.714	3.800	6.333	7.600	9.500	13.000	19.500
	F1-Score	0.94	0.96	0.97	0.98	0.99	0.99	0.99	1.00	1.00	1.00

Bank Marketing Dataset

Table D.9: Ensemble Performance on Bagged Subsets of the Bank Marketing Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.556	0.556	0.556	0.556	0.556	0.556	0.556	0.556	0.556	0.556
	Training accuracy	0.679	0.793	0.686	0.739	0.748	0.715	0.738	0.803	0.793	0.769
	GF	1.383	2.145	1.414	1.701	1.762	1.558	1.695	2.254	2.145	1.922
	F1-Score	0.79	0.87	0.76	0.78	0.79	0.79	0.78	0.81	0.80	0.81
*k*NNE	Testing accuracy	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.893
	Training accuracy	0.913	0.872	0.887	0.892	0.927	0.919	0.911	0.923	0.925	0.931
	GF	1.230	0.836	0.947	0.991	1.466	1.321	1.202	1.390	1.427	1.551
	F1-Score	0.87	0.85	0.87	0.86	0.85	0.87	0.86	0.86	0.86	0.86
DTE	Testing accuracy	0.885	0.886	0.886	0.885	0.884	0.886	0.887	0.885	0.885	0.886
	Training accuracy	0.933	0.912	0.904	0.917	0.934	0.944	0.936	0.936	0.949	0.951
	GF	1.716	1.295	1.188	1.386	1.758	2.036	1.766	1.797	2.255	2.327
	F1-Score	0.87	0.88	0.88	0.88	0.89	0.89	0.89	0.88	0.89	0.89
RF	Testing accuracy	0.898	0.897	0.899	0.896	0.897	0.899	0.898	0.898	0.898	0.898
	Training accuracy	0.926	0.898	0.902	0.919	0.930	0.939	0.939	0.945	0.946	0.956
	GF	1.378	1.010	1.031	1.284	1.471	1.656	1.672	1.855	1.889	2.318
	F1-Score	0.89	0.88	0.89	0.88	0.89	0.89	0.88	0.89	0.88	0.89
SVME	Testing accuracy	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894
	Training accuracy	0.912	0.892	0.904	0.902	0.928	0.932	0.931	0.937	0.935	0.944
	GF	1.205	0.981	1.104	1.082	1.472	1.559	1.536	1.683	1.631	1.893
	F1-Score	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84
NNE	Testing accuracy	0.891	0.891	0.892	0.892	0.892	0.889	0.890	0.890	0.890	0.891
	Training accuracy	0.897	0.899	0.903	0.906	0.934	0.943	0.941	0.949	0.950	0.959
	GF	1.058	1.079	1.113	1.149	1.636	1.947	1.864	2.157	2.200	2.659
	F1-Score	0.88	0.88	0.88	0.89	0.88	0.89	0.88	0.89	0.89	0.88

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.831	0.831	0.831	0.831	0.831	0.831	0.831	0.831	0.831	0.831
	Training accuracy	0.842	0.824	0.816	0.816	0.827	0.838	0.823	0.847	0.856	0.851
	GF	1.070	0.960	0.918	0.918	0.977	1.043	0.955	1.105	1.174	1.134
	F1-Score	0.84	0.87	0.83	0.84	0.85	0.85	0.84	0.85	0.85	0.86
*k*NNhte	Testing accuracy	0.897	0.897	0.897	0.897	0.897	0.897	0.897	0.897	0.897	0.897
	Training accuracy	0.897	0.878	0.876	0.874	0.913	0.908	0.891	0.897	0.909	0.906
	GF	1.000	0.844	0.831	0.817	1.184	1.120	0.945	1.000	1.132	1.096
	F1-Score	0.86	0.87	0.87	0.87	0.86	0.87	0.87	0.86	0.87	0.87
DThte	Testing accuracy	0.895	0.898	0.894	0.893	0.895	0.896	0.891	0.896	0.895	0.895
	Training accuracy	0.945	0.920	0.910	0.925	0.939	0.950	0.945	0.951	0.954	0.964
	GF	1.909	1.275	1.178	1.427	1.721	2.080	1.982	2.122	2.283	2.917
	F1-Score	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.90	0.90
SVMhte	Testing accuracy	0.903	0.903	0.903	0.903	0.903	0.903	0.903	0.903	0.902	0.903
	Training accuracy	0.907	0.883	0.887	0.896	0.920	0.915	0.904	0.909	0.914	0.910
	GF	1.043	0.829	0.858	0.933	1.213	1.141	1.010	1.066	1.140	1.078
	F1-Score	0.87	0.87	0.86	0.88	0.85	0.87	0.87	0.88	0.87	0.88
NNhte	Testing accuracy	0.897	0.900	0.899	0.897	0.897	0.898	0.897	0.897	0.899	0.899
	Training accuracy	0.901	0.907	0.907	0.914	0.937	0.946	0.939	0.950	0.951	0.960
	GF	1.040	1.075	1.086	1.198	1.635	1.889	1.689	2.060	2.061	2.525
	F1-Score	0.89	0.88	0.88	0.88	0.88	0.89	0.88	0.89	0.89	0.90
HTEsm	Testing accuracy	0.898	0.897	0.900	0.897	0.899	0.897	0.901	0.897	0.896	0.896
	Training accuracy	0.926	0.908	0.908	0.918	0.937	0.943	0.943	0.945	0.949	0.959
	GF	1.378	1.120	1.087	1.256	1.603	1.807	1.737	1.873	2.039	2.537
	F1-Score	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.90	0.90
HTEdf	Testing accuracy	0.899	0.902	0.898	0.898	0.898	0.898	0.899	0.896	0.901	0.902
	Training accuracy	0.917	0.916	0.909	0.924	0.938	0.943	0.943	0.952	0.955	0.959
	GF	1.217	1.167	1.121	1.342	1.645	1.789	1.772	2.167	2.200	2.390
	F1-Score	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.90	0.90	0.90

Censor Income Dataset

Table D.10: Ensemble Performance on Bagged Subsets of the Censor Income Dataset

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.627	0.627	0.627	0.627	0.627	0.627	0.627	0.627	0.627	0.627
	Training accuracy	0.434	0.473	0.592	0.564	0.545	0.542	0.529	0.594	0.568	0.594
	GF	0.659	0.708	0.914	0.856	0.820	0.814	0.792	0.919	0.863	0.919
	F1-Score	0.79	0.87	0.76	0.78	0.79	0.79	0.78	0.81	0.80	0.81
*k*NNE	Testing accuracy	0.805	0.805	0.805	0.805	0.805	0.805	0.805	0.805	0.805	0.805
	Training accuracy	0.805	0.818	0.822	0.782	0.821	0.803	0.810	0.815	0.817	0.836
	GF	1.000	1.071	1.096	0.894	1.089	0.990	1.026	1.054	1.066	1.189
	F1-Score	0.86	0.87	0.87	0.87	0.86	0.87	0.87	0.86	0.87	0.87
DTE	Testing accuracy	0.813	0.813	0.811	0.811	0.811	0.813	0.812	0.811	0.812	0.812
	Training accuracy	0.821	0.806	0.820	0.803	0.827	0.827	0.827	0.840	0.849	0.865
	GF	1.045	0.964	1.050	0.959	1.092	1.081	1.087	1.181	1.245	1.393
	F1-Score	0.87	0.87	0.88	0.88	0.89	0.89	0.89	0.88	0.89	0.89
RF	Testing accuracy	0.808	0.806	0.812	0.811	0.810	0.808	0.809	0.806	0.810	0.811
	Training accuracy	0.841	0.837	0.836	0.835	0.858	0.863	0.874	0.888	0.896	0.914
	GF	1.208	1.190	1.146	1.145	1.338	1.401	1.516	1.732	1.827	2.198
	F1-Score	0.89	0.88	0.88	0.88	0.89	0.89	0.88	0.89	0.88	0.89
SVME	Testing accuracy	0.798	0.797	0.798	0.798	0.797	0.798	0.798	0.798	0.798	0.798
	Training accuracy	0.811	0.823	0.815	0.817	0.815	0.819	0.822	0.820	0.846	0.848
	GF	1.069	1.147	1.092	1.104	1.097	1.116	1.135	1.122	1.312	1.329
	F1-Score	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84
NNE	Testing accuracy	0.809	0.806	0.810	0.809	0.810	0.806	0.810	0.808	0.808	0.809
	Training accuracy	0.832	0.827	0.830	0.824	0.835	0.843	0.852	0.858	0.871	0.889
	GF	1.137	1.121	1.118	1.085	1.152	1.236	1.284	1.352	1.488	1.721
	F1-Score	0.88	0.88	0.88	0.89	0.88	0.89	0.88	0.89	0.89	0.88

Ensemble	Measure	Bagged Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.745	0.745	0.745	0.745	0.745	0.745	0.745	0.745	0.745	0.745
	Training accuracy	0.712	0.652	0.762	0.645	0.705	0.723	0.712	0.723	0.715	0.733
	GF	0.885	0.733	1.071	0.718	0.864	0.921	0.885	0.921	0.895	0.955
	F1-Score	0.84	0.87	0.83	0.84	0.85	0.85	0.84	0.85	0.85	0.86
*k*NNhte	Testing accuracy	0.809	0.809	0.809	0.809	0.809	0.809	0.809	0.809	0.809	0.809
	Training accuracy	0.832	0.826	0.830	0.811	0.831	0.839	0.845	0.858	0.871	0.874
	GF	1.137	1.098	1.124	1.011	1.130	1.186	1.232	1.345	1.481	1.516
	F1-Score	0.87	0.85	0.87	0.86	0.85	0.87	0.86	0.86	0.86	0.86
DThte	Testing accuracy	0.818	0.817	0.817	0.817	0.816	0.815	0.815	0.819	0.819	0.814
	Training accuracy	0.828	0.825	0.833	0.819	0.837	0.844	0.839	0.848	0.868	0.876
	GF	1.058	1.046	1.096	1.011	1.129	1.186	1.149	1.191	1.371	1.500
	F1-Score	0.88	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89
SVMhte	Testing accuracy	0.823	0.824	0.824	0.824	0.823	0.824	0.824	0.823	0.824	0.823
	Training accuracy	0.849	0.816	0.834	0.802	0.825	0.800	0.813	0.816	0.814	0.824
	GF	1.172	0.957	1.060	0.889	1.011	0.880	0.941	0.962	0.946	1.006
	F1-Score	0.87	0.88	0.86	0.88	0.85	0.87	0.87	0.88	0.87	0.88
NNhte	Testing accuracy	0.817	0.812	0.816	0.815	0.813	0.812	0.815	0.814	0.817	0.814
	Training accuracy	0.839	0.827	0.836	0.829	0.842	0.844	0.854	0.852	0.871	0.886
	GF	1.137	1.087	1.122	1.082	1.184	1.205	1.267	1.257	1.419	1.632
	F1-Score	0.89	0.88	0.88	0.88	0.89	0.89	0.88	0.89	0.89	0.89
HTEsm	Testing accuracy	0.821	0.818	0.818	0.821	0.820	0.819	0.819	0.819	0.820	0.817
	Training accuracy	0.846	0.823	0.833	0.820	0.837	0.840	0.841	0.851	0.866	0.871
	GF	1.162	1.028	1.090	0.994	1.104	1.131	1.138	1.215	1.343	1.419
	F1-Score	0.89	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89
HTEdf	Testing accuracy	0.824	0.824	0.824	0.826	0.823	0.824	0.825	0.823	0.822	0.822
	Training accuracy	0.829	0.834	0.835	0.832	0.856	0.854	0.863	0.880	0.889	0.902
	GF	1.029	1.060	1.067	1.036	1.229	1.205	1.277	1.475	1.604	1.816
	F1-Score	0.89	0.88	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89

Appendix E

Ensemble Performance on Feature Subsets for Classification Problems

The results of the ensembles over the feature subsets of the training dataset for classification problems are provided in this appendix. Plots of the training and testing accuracies for each classification dataset are first presented. Then the results of training and testing accuracy, GF, and F1-score of the ensembles over the classification datasets are provided.

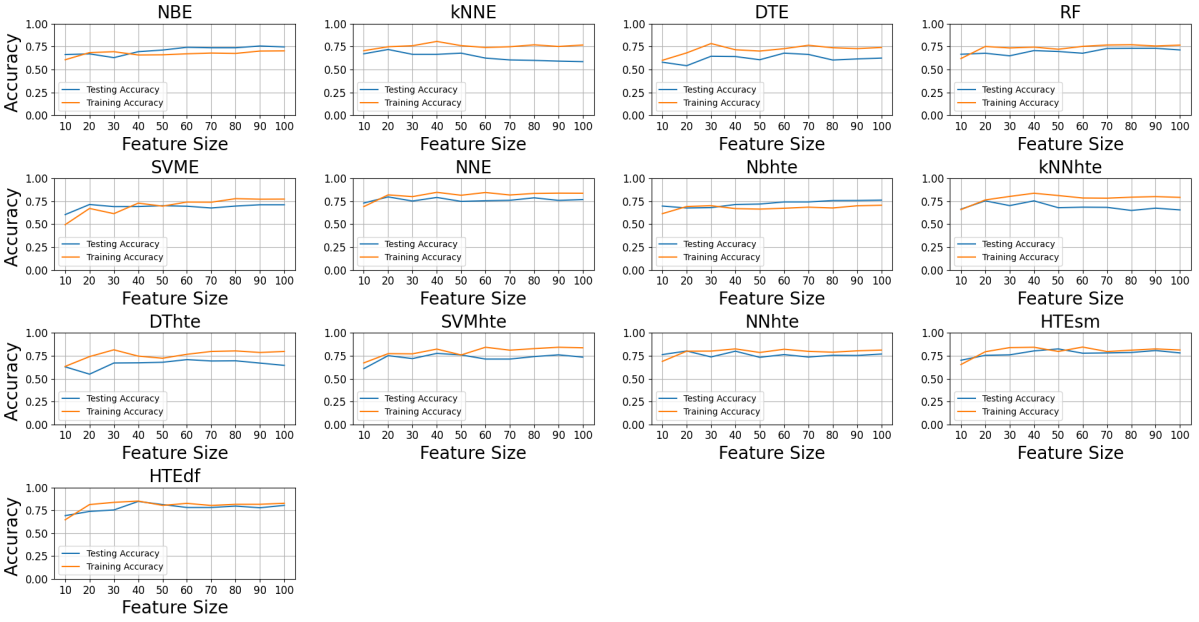

Figure E.1: Ensemble Performance on Feature Subsets of the Sonar Dataset

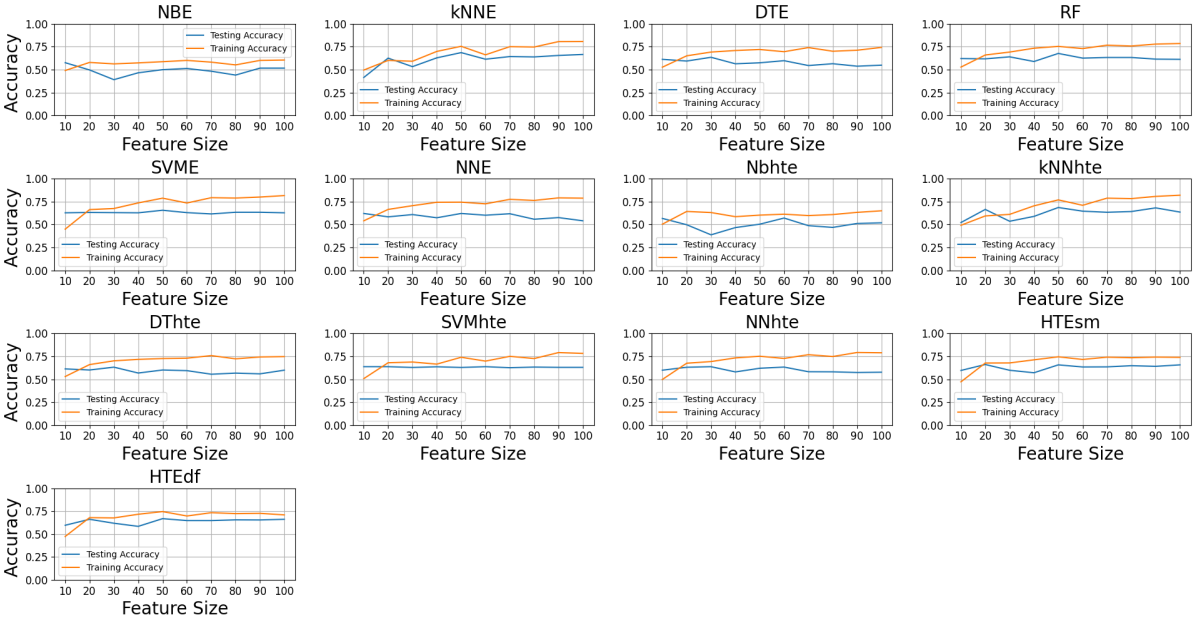

Figure E.2: Ensemble Performance on Feature Subsets of the Breast Cancer Dataset

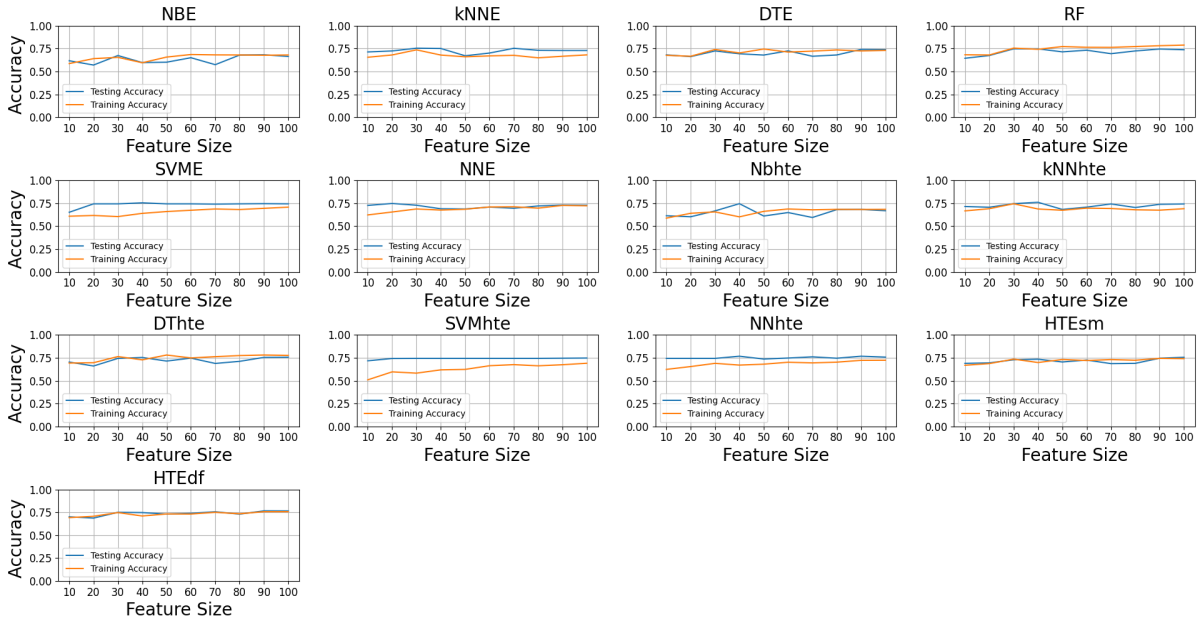

Figure E.3: Ensemble Performance on Feature Subsets of the Indian Liver Dataset

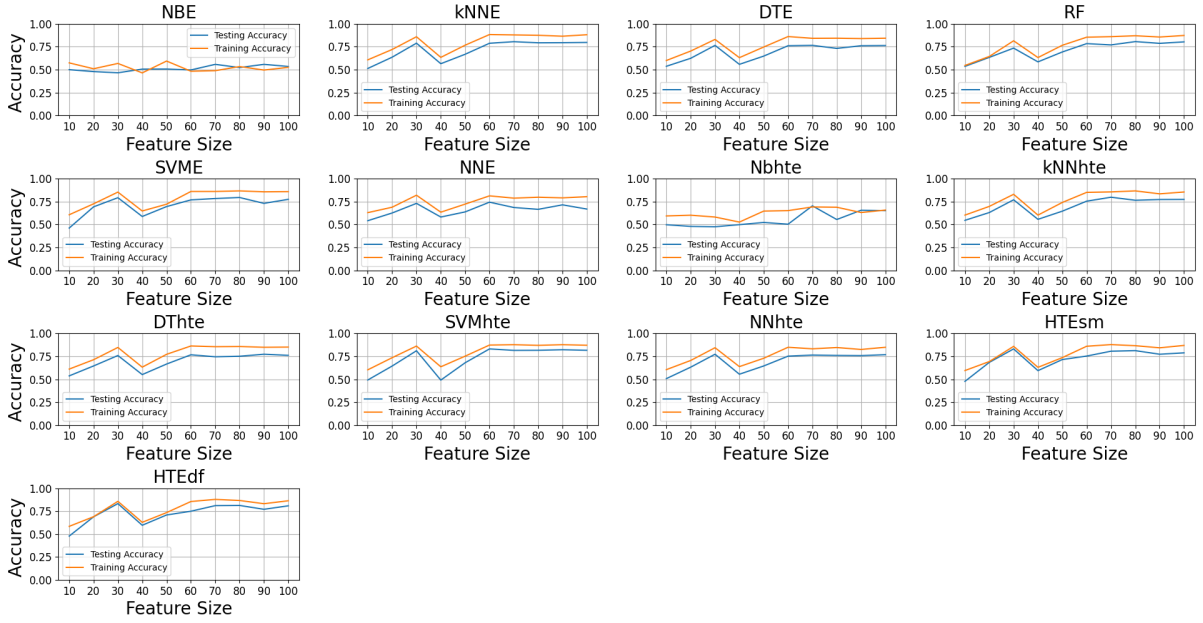

Figure E.4: Ensemble Performance on Feature Subsets of the Credit Approval Dataset

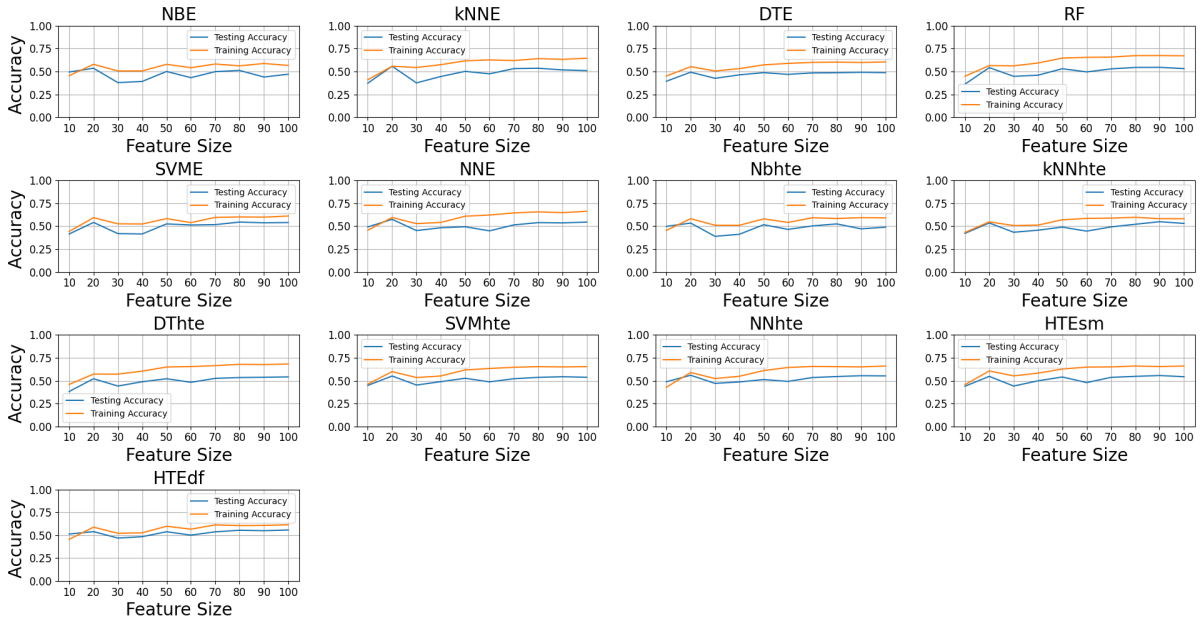

Figure E.5: Ensemble Performance on Feature Subsets of the Red Wine Dataset

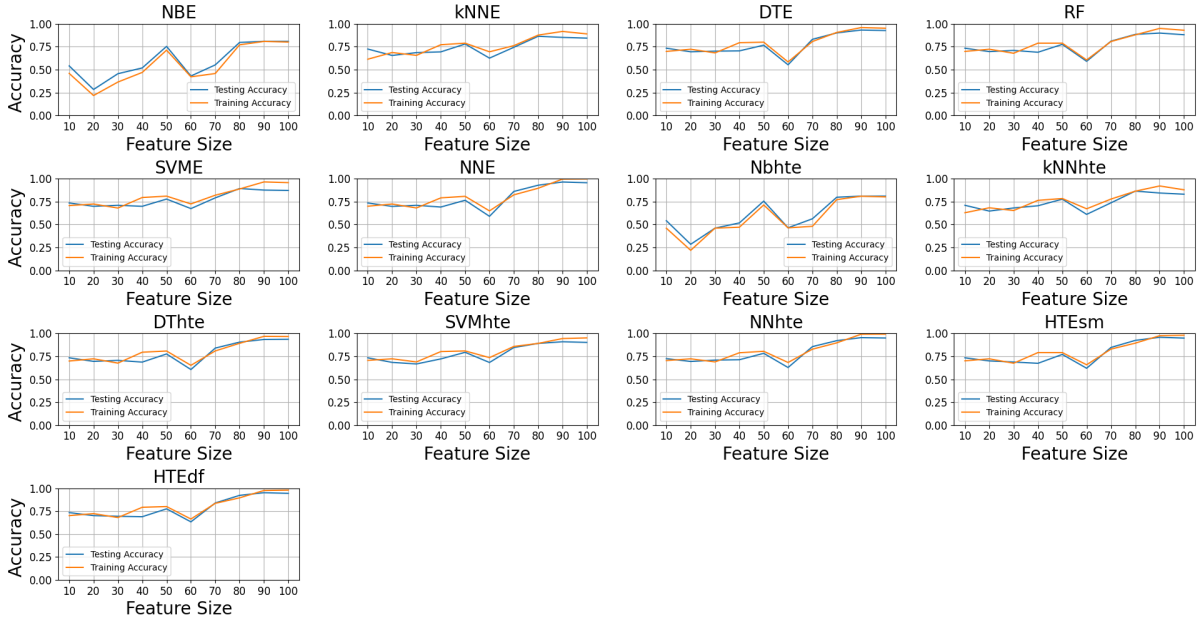

Figure E.6: Ensemble Performance on Feature Subsets of the Car Evaluation Dataset

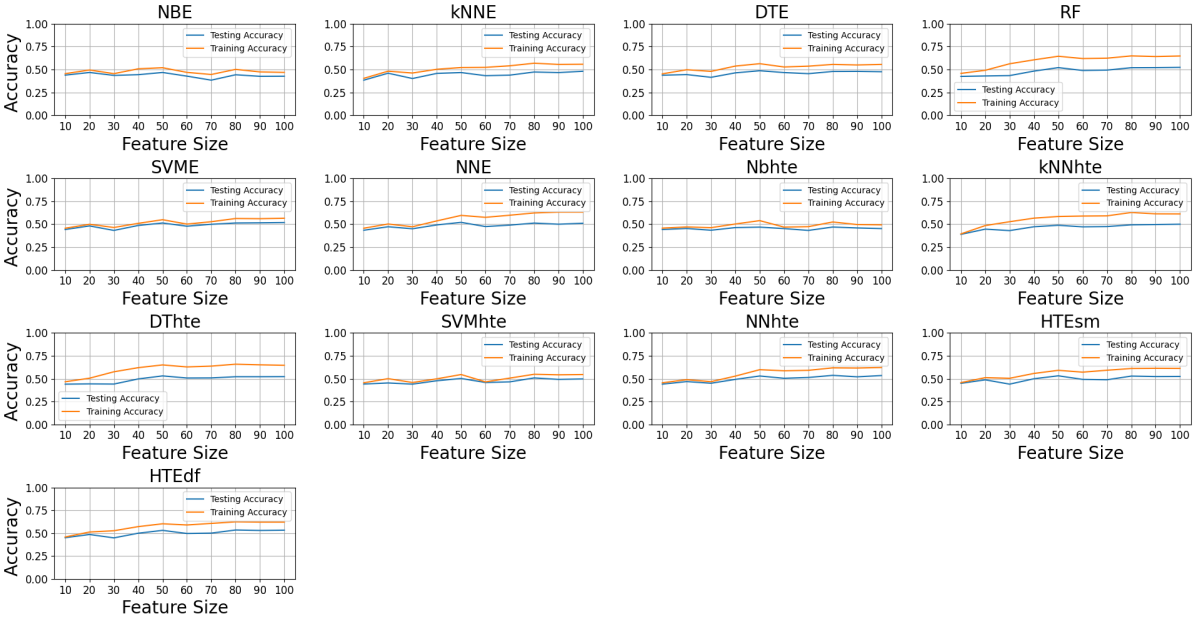

Figure E.7: Ensemble Performance on Feature Subsets of the White Wine Dataset

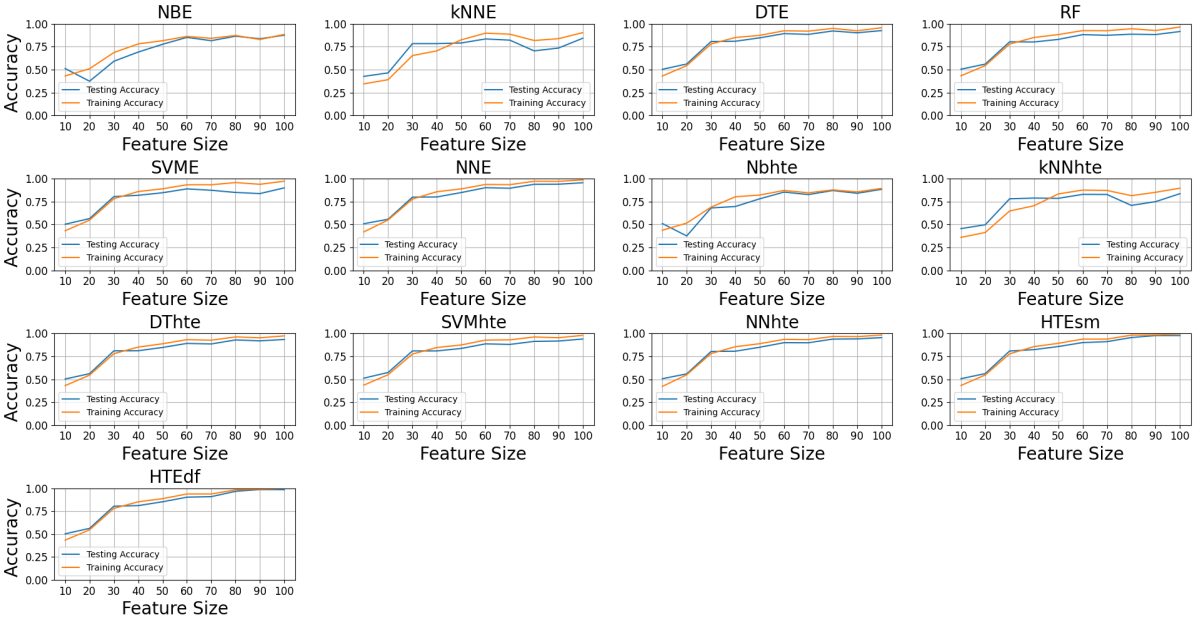

Figure E.8: Ensemble Performance on Feature Subsets of the Nursery Dataset

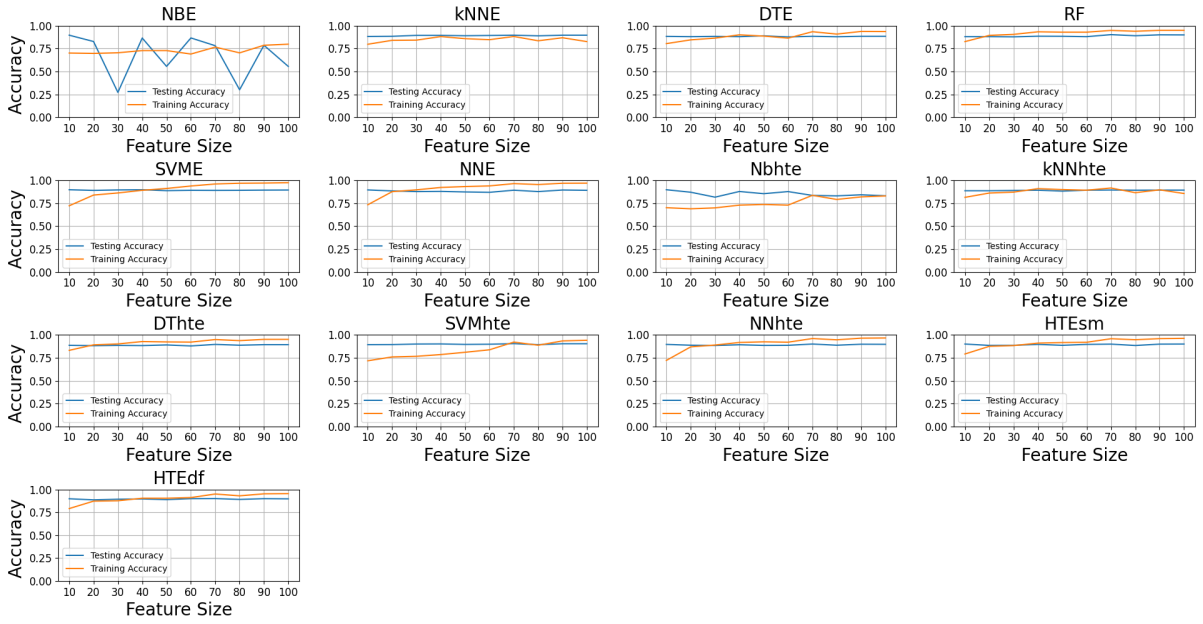

Figure E.9: Ensemble Performance on Feature Subsets of the Bank Marketing Dataset

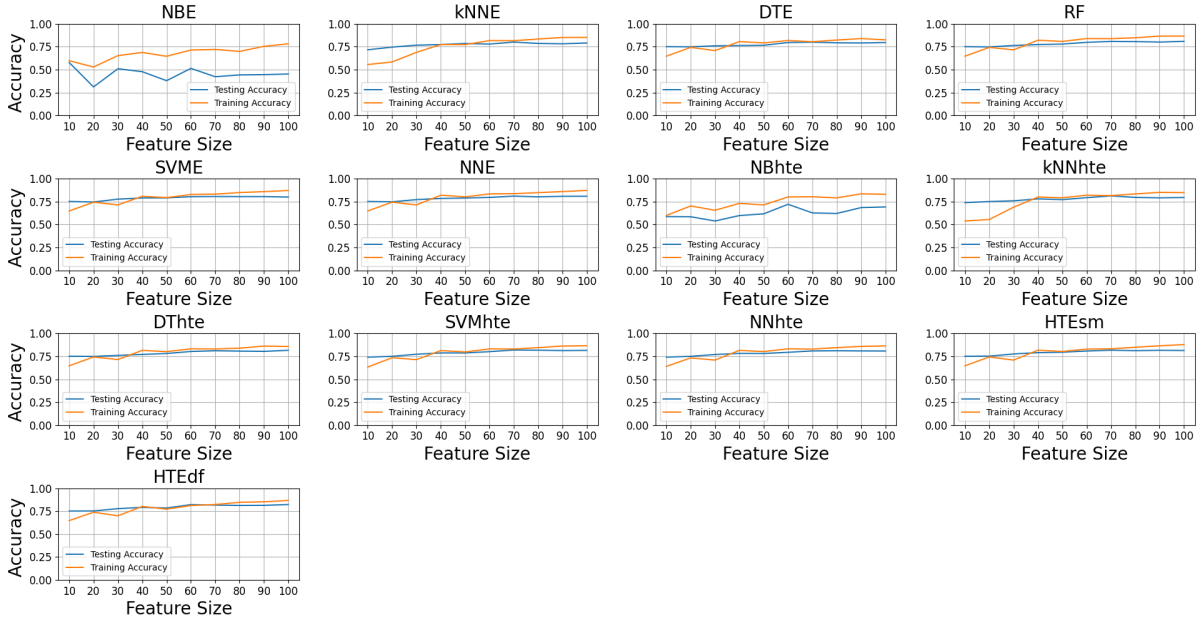

Figure E.10: Ensemble Performance on Feature Subsets of the Censor Income Dataset

Sonar Dataset

Table E.1: Ensemble Performance on Feature Subsets of the Sonar Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.664	0.671	0.630	0.695	0.713	0.743	0.738	0.738	0.757	0.747
	Training accuracy	0.607	0.685	0.695	0.659	0.661	0.672	0.680	0.676	0.702	0.704
	GF	0.855	1.044	1.213	0.894	0.847	0.784	0.819	0.809	0.815	0.855
	F1-Score	0.81	0.69	0.62	0.76	0.74	0.79	0.72	0.79	0.74	0.74
*k*NNE	Testing accuracy	0.672	0.719	0.666	0.666	0.679	0.625	0.605	0.600	0.592	0.586
	Training accuracy	0.706	0.749	0.760	0.807	0.761	0.741	0.749	0.770	0.752	0.768
	GF	1.116	1.120	1.392	1.731	1.343	1.448	1.574	1.739	1.645	1.784
	F1-Score	0.76	0.83	0.88	0.79	0.88	0.81	0.79	0.81	0.86	0.86
DTE	Testing accuracy	0.580	0.542	0.645	0.642	0.607	0.679	0.665	0.604	0.616	0.625
	Training accuracy	0.601	0.682	0.784	0.716	0.702	0.728	0.765	0.738	0.729	0.741
	GF	1.053	1.440	1.644	1.261	1.319	1.180	1.426	1.511	1.417	1.448
	F1-Score	0.74	0.74	0.67	0.79	0.81	0.69	0.79	0.79	0.77	0.65
RF	Testing accuracy	0.667	0.678	0.650	0.707	0.697	0.678	0.730	0.732	0.732	0.714
	Training accuracy	0.620	0.752	0.736	0.745	0.722	0.753	0.768	0.771	0.757	0.767
	GF	0.876	1.298	1.326	1.149	1.090	1.304	1.164	1.170	1.103	1.227
	F1-Score	0.78	0.83	0.83	0.79	0.86	0.81	0.69	0.84	0.88	0.83
SVME	Testing accuracy	0.604	0.714	0.691	0.692	0.701	0.695	0.676	0.697	0.711	0.711
	Training accuracy	0.494	0.671	0.614	0.729	0.696	0.741	0.739	0.778	0.772	0.773
	GF	0.783	0.869	0.801	1.137	0.984	1.178	1.241	1.365	1.268	1.273
	F1-Score	0.43	0.71	0.63	0.76	0.86	0.67	0.72	0.74	0.74	0.74
NNE	Testing accuracy	0.730	0.797	0.752	0.791	0.748	0.755	0.760	0.787	0.759	0.768
	Training accuracy	0.691	0.819	0.800	0.847	0.815	0.845	0.818	0.835	0.838	0.837
	GF	0.874	1.122	1.240	1.366	1.362	1.581	1.319	1.291	1.488	1.423
	F1-Score	0.45	0.77	0.88	0.81	0.86	0.88	0.81	0.79	0.86	0.88

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.697	0.676	0.680	0.714	0.719	0.742	0.742	0.757	0.758	0.762
	Training accuracy	0.614	0.692	0.701	0.669	0.664	0.673	0.685	0.676	0.700	0.706
	GF	0.785	1.052	1.070	0.864	0.836	0.789	0.819	0.750	0.807	0.810
	F1-Score	0.79	0.69	0.62	0.76	0.74	0.77	0.72	0.79	0.74	0.74
*k*NNhte	Testing accuracy	0.663	0.754	0.702	0.754	0.679	0.685	0.683	0.649	0.674	0.655
	Training accuracy	0.658	0.764	0.802	0.837	0.812	0.785	0.783	0.794	0.800	0.792
	GF	0.985	1.042	1.505	1.509	1.707	1.465	1.461	1.704	1.630	1.659
	F1-Score	0.77	0.90	0.86	0.81	0.93	0.88	0.90	0.90	0.92	0.90
DThte	Testing accuracy	0.629	0.549	0.671	0.673	0.679	0.708	0.693	0.695	0.670	0.645
	Training accuracy	0.634	0.741	0.815	0.746	0.723	0.766	0.797	0.803	0.786	0.797
	GF	1.014	1.741	1.778	1.287	1.159	1.248	1.512	1.548	1.542	1.749
	F1-Score	0.72	0.81	0.71	0.84	0.86	0.72	0.81	0.74	0.77	0.74
SVMhte	Testing accuracy	0.609	0.751	0.719	0.776	0.759	0.714	0.714	0.741	0.760	0.735
	Training accuracy	0.672	0.773	0.771	0.823	0.759	0.843	0.811	0.828	0.843	0.837
	GF	1.192	1.097	1.227	1.266	1.000	1.822	1.513	1.506	1.529	1.626
	F1-Score	0.74	0.86	0.88	0.84	0.88	0.84	0.79	0.81	0.84	0.86
NNhte	Testing accuracy	0.763	0.802	0.737	0.800	0.733	0.763	0.736	0.755	0.753	0.769
	Training accuracy	0.689	0.800	0.801	0.825	0.786	0.821	0.798	0.789	0.804	0.812
	GF	0.762	0.990	1.322	1.143	1.248	1.324	1.307	1.161	1.260	1.229
	F1-Score	0.67	0.88	0.86	0.93	0.90	0.88	0.86	0.86	0.88	0.86
HTEsm	Testing accuracy	0.701	0.755	0.760	0.803	0.825	0.778	0.781	0.786	0.807	0.781
	Training accuracy	0.654	0.794	0.839	0.843	0.797	0.845	0.796	0.811	0.825	0.813
	GF	0.864	1.189	1.491	1.255	0.862	1.432	1.074	1.132	1.103	1.171
	F1-Score	0.77	0.86	0.96	0.88	0.91	0.95	0.88	0.86	0.88	0.93
HTEdf	Testing accuracy	0.692	0.738	0.754	0.847	0.813	0.781	0.781	0.796	0.778	0.804
	Training accuracy	0.647	0.813	0.837	0.851	0.804	0.826	0.803	0.816	0.816	0.827
	GF	0.873	1.401	1.509	1.027	0.954	1.259	1.112	1.109	1.207	1.133
	F1-Score	0.77	0.88	0.88	0.91	0.98	0.91	0.88	0.91	0.93	0.91

Breast Cancer Dataset

Table E.2: Ensemble Performance on Feature Subsets of the Breast Cancer Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.576	0.497	0.391	0.466	0.500	0.513	0.483	0.441	0.517	0.517
	Training accuracy	0.491	0.579	0.563	0.574	0.587	0.601	0.582	0.552	0.601	0.605
	GF	0.833	1.195	1.394	1.254	1.211	1.221	1.237	1.248	1.211	1.223
	F1-Score	0.57	0.53	0.59	0.45	0.44	0.52	0.45	0.36	0.50	0.51
*k*NNE	Testing accuracy	0.415	0.626	0.533	0.629	0.686	0.614	0.643	0.639	0.655	0.666
	Training accuracy	0.496	0.601	0.592	0.699	0.755	0.662	0.751	0.747	0.806	0.807
	GF	1.161	0.937	1.145	1.233	1.282	1.142	1.434	1.427	1.778	1.731
	F1-Score	0.56	0.62	0.60	0.62	0.65	0.59	0.56	0.59	0.55	0.60
DTE	Testing accuracy	0.613	0.594	0.635	0.564	0.575	0.598	0.545	0.565	0.538	0.549
	Training accuracy	0.527	0.651	0.692	0.709	0.720	0.696	0.741	0.701	0.712	0.743
	GF	0.818	1.163	1.185	1.498	1.518	1.322	1.757	1.455	1.604	1.755
	F1-Score	0.53	0.62	0.49	0.62	0.54	0.60	0.65	0.62	0.58	0.62
RF	Testing accuracy	0.621	0.618	0.640	0.589	0.677	0.626	0.633	0.633	0.615	0.613
	Training accuracy	0.528	0.660	0.692	0.734	0.754	0.730	0.767	0.758	0.779	0.785
	GF	0.803	1.124	1.169	1.545	1.313	1.385	1.575	1.517	1.742	1.800
	F1-Score	0.56	0.57	0.58	0.70	0.58	0.53	0.62	0.56	0.57	0.64
SVME	Testing accuracy	0.627	0.631	0.629	0.627	0.655	0.629	0.615	0.633	0.633	0.627
	Training accuracy	0.449	0.662	0.674	0.735	0.787	0.734	0.792	0.788	0.799	0.815
	GF	0.677	1.092	1.138	1.408	1.620	1.395	1.851	1.731	1.826	2.016
	F1-Score	0.49	0.63	0.61	0.62	0.63	0.51	0.61	0.53	0.50	0.51
NNE	Testing accuracy	0.620	0.583	0.608	0.573	0.620	0.601	0.617	0.557	0.575	0.540
	Training accuracy	0.540	0.664	0.705	0.742	0.743	0.725	0.774	0.762	0.790	0.787
	GF	0.826	1.241	1.329	1.655	1.479	1.451	1.695	1.861	2.024	2.160
	F1-Score	0.53	0.62	0.55	0.66	0.65	0.57	0.59	0.59	0.60	0.68

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.566	0.497	0.387	0.466	0.503	0.570	0.487	0.468	0.511	0.518
	Training accuracy	0.503	0.642	0.630	0.585	0.602	0.612	0.597	0.608	0.632	0.649
	GF	0.873	1.405	1.657	1.287	1.249	1.108	1.273	1.357	1.329	1.373
	F1-Score	0.53	0.67	0.58	0.44	0.48	0.51	0.45	0.47	0.59	0.59
*k*NNhte	Testing accuracy	0.521	0.664	0.535	0.588	0.685	0.645	0.633	0.641	0.681	0.635
	Training accuracy	0.491	0.593	0.610	0.703	0.768	0.709	0.787	0.782	0.806	0.819
	GF	0.941	0.826	1.192	1.387	1.358	1.220	1.723	1.647	1.644	2.017
	F1-Score	0.29	0.53	0.50	0.70	0.64	0.60	0.62	0.65	0.58	0.62
DThte	Testing accuracy	0.613	0.601	0.632	0.568	0.601	0.594	0.555	0.567	0.559	0.599
	Training accuracy	0.530	0.660	0.701	0.717	0.726	0.730	0.758	0.723	0.743	0.747
	GF	0.823	1.174	1.231	1.527	1.456	1.504	1.839	1.563	1.716	1.585
	F1-Score	0.53	0.66	0.52	0.62	0.59	0.62	0.66	0.63	0.57	0.68
SVMhte	Testing accuracy	0.637	0.637	0.629	0.636	0.629	0.637	0.626	0.633	0.630	0.630
	Training accuracy	0.507	0.680	0.688	0.665	0.740	0.698	0.750	0.726	0.791	0.782
	GF	0.736	1.134	1.189	1.087	1.427	1.202	1.496	1.339	1.770	1.697
	F1-Score	0.49	0.63	0.63	0.58	0.62	0.65	0.62	0.62	0.65	0.65
NNhte	Testing accuracy	0.599	0.631	0.637	0.580	0.620	0.633	0.582	0.581	0.574	0.577
	Training accuracy	0.498	0.675	0.693	0.733	0.751	0.727	0.768	0.748	0.792	0.789
	GF	0.799	1.135	1.182	1.573	1.526	1.344	1.802	1.663	2.048	2.005
	F1-Score	0.53	0.66	0.58	0.69	0.62	0.56	0.60	0.61	0.61	0.64
HTEsm	Testing accuracy	0.596	0.661	0.598	0.571	0.657	0.634	0.635	0.648	0.641	0.657
	Training accuracy	0.472	0.677	0.678	0.712	0.745	0.716	0.741	0.735	0.742	0.739
	GF	0.765	1.050	1.248	1.490	1.345	1.289	1.409	1.328	1.391	1.314
	F1-Score	0.59	0.61	0.61	0.68	0.65	0.64	0.66	0.64	0.64	0.71
HTEdf	Testing accuracy	0.596	0.661	0.618	0.584	0.668	0.647	0.647	0.655	0.654	0.661
	Training accuracy	0.474	0.679	0.676	0.716	0.745	0.697	0.733	0.723	0.726	0.709
	GF	0.768	1.056	1.179	1.465	1.302	1.165	1.322	1.245	1.263	1.165
	F1-Score	0.62	0.62	0.62	0.69	0.67	0.66	0.67	0.66	0.66	0.72

Indian Liver Dataset

Table E.3: Ensemble Performance on Feature Subsets of the Indian Liver Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.618	0.571	0.675	0.598	0.602	0.651	0.575	0.680	0.683	0.665
	Training accuracy	0.587	0.641	0.655	0.596	0.658	0.686	0.682	0.681	0.678	0.681
	GF	0.925	1.195	0.942	0.995	1.164	1.111	1.336	1.003	0.984	1.050
	F1-Score	0.36	0.44	0.52	0.37	0.47	0.54	0.49	0.55	0.55	0.55
*k*NNE	Testing accuracy	0.714	0.725	0.755	0.752	0.672	0.702	0.754	0.732	0.730	0.730
	Training accuracy	0.656	0.681	0.737	0.680	0.661	0.671	0.677	0.650	0.667	0.682
	GF	0.831	0.862	0.932	0.775	0.968	0.906	0.762	0.766	0.811	0.849
	F1-Score	0.66	0.70	0.70	0.64	0.70	0.59	0.62	0.62	0.56	0.57
DTE	Testing accuracy	0.682	0.663	0.725	0.694	0.680	0.726	0.667	0.681	0.742	0.740
	Training accuracy	0.677	0.667	0.741	0.703	0.746	0.713	0.723	0.736	0.725	0.731
	GF	0.985	1.012	1.062	1.030	1.260	0.955	1.202	1.208	0.938	0.967
	F1-Score	0.64	0.66	0.71	0.71	0.68	0.61	0.69	0.71	0.65	0.65
RF	Testing accuracy	0.645	0.675	0.748	0.748	0.715	0.733	0.695	0.725	0.746	0.738
	Training accuracy	0.683	0.681	0.756	0.743	0.773	0.765	0.764	0.773	0.782	0.789
	GF	1.120	1.019	1.033	0.981	1.256	1.136	1.292	1.211	1.165	1.242
	F1-Score	0.64	0.63	0.74	0.65	0.69	0.68	0.76	0.74	0.71	0.68
SVME	Testing accuracy	0.651	0.743	0.743	0.754	0.743	0.743	0.740	0.743	0.745	0.743
	Training accuracy	0.608	0.616	0.604	0.639	0.659	0.673	0.687	0.681	0.694	0.707
	GF	0.890	0.669	0.649	0.681	0.754	0.786	0.831	0.806	0.833	0.877
	F1-Score	0.40	0.62	0.61	0.66	0.67	0.67	0.66	0.69	0.66	0.69
NNE	Testing accuracy	0.726	0.747	0.728	0.689	0.687	0.708	0.695	0.720	0.730	0.728
	Training accuracy	0.622	0.654	0.687	0.675	0.684	0.709	0.711	0.696	0.726	0.722
	GF	0.725	0.731	0.869	0.957	0.991	1.003	1.055	0.921	0.985	0.978
	F1-Score	0.49	0.53	0.67	0.72	0.68	0.62	0.64	0.63	0.71	0.66

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.613	0.602	0.666	0.745	0.610	0.648	0.594	0.680	0.682	0.668
	Training accuracy	0.587	0.640	0.655	0.601	0.661	0.687	0.678	0.683	0.682	0.683
	GF	0.937	1.106	0.968	0.639	1.150	1.125	1.261	1.009	1.000	1.047
	F1-Score	0.36	0.45	0.52	0.40	0.47	0.55	0.49	0.55	0.55	0.55
*k*NNhte	Testing accuracy	0.714	0.706	0.745	0.760	0.682	0.707	0.742	0.703	0.738	0.742
	Training accuracy	0.666	0.690	0.744	0.687	0.673	0.696	0.692	0.678	0.674	0.690
	GF	0.856	0.948	0.996	0.767	0.972	0.964	0.838	0.922	0.804	0.832
	F1-Score	0.68	0.70	0.71	0.59	0.69	0.62	0.62	0.62	0.58	0.58
DThte	Testing accuracy	0.703	0.660	0.743	0.753	0.714	0.746	0.688	0.711	0.754	0.756
	Training accuracy	0.692	0.696	0.763	0.727	0.780	0.748	0.762	0.774	0.780	0.776
	GF	0.964	1.118	1.084	0.905	1.300	1.008	1.311	1.279	1.118	1.089
	F1-Score	0.62	0.67	0.79	0.70	0.67	0.63	0.73	0.71	0.72	0.67
SVMhte	Testing accuracy	0.717	0.742	0.743	0.743	0.743	0.743	0.743	0.743	0.745	0.747
	Training accuracy	0.509	0.596	0.582	0.618	0.623	0.663	0.675	0.662	0.674	0.690
	GF	0.576	0.639	0.615	0.673	0.682	0.763	0.791	0.760	0.782	0.816
	F1-Score	0.27	0.69	0.59	0.63	0.63	0.67	0.63	0.70	0.66	0.66
NNhte	Testing accuracy	0.743	0.743	0.743	0.767	0.736	0.747	0.760	0.745	0.767	0.757
	Training accuracy	0.623	0.654	0.689	0.670	0.680	0.701	0.694	0.702	0.722	0.723
	GF	0.682	0.743	0.826	0.706	0.825	0.846	0.784	0.856	0.838	0.877
	F1-Score	0.46	0.53	0.66	0.73	0.72	0.60	0.70	0.71	0.67	0.66
HTEsm	Testing accuracy	0.689	0.694	0.728	0.734	0.704	0.725	0.686	0.689	0.745	0.755
	Training accuracy	0.668	0.686	0.735	0.698	0.731	0.720	0.730	0.723	0.743	0.740
	GF	0.937	0.975	1.026	0.881	1.100	0.982	1.163	1.123	0.992	0.942
	F1-Score	0.61	0.68	0.69	0.67	0.68	0.68	0.73	0.70	0.72	0.69
HTEdf	Testing accuracy	0.701	0.687	0.750	0.747	0.731	0.739	0.756	0.730	0.766	0.765
	Training accuracy	0.692	0.707	0.747	0.710	0.732	0.731	0.750	0.734	0.757	0.757
	GF	0.971	1.068	0.988	0.872	1.004	0.970	0.976	1.015	0.963	0.967
	F1-Score	0.63	0.70	0.70	0.69	0.70	0.70	0.75	0.73	0.73	0.70

Credit Approval Dataset

Table E.4: Ensemble Performance on Feature Subsets of the Credit Approval Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.500	0.479	0.466	0.506	0.507	0.497	0.558	0.523	0.558	0.535
	Training accuracy	0.574	0.510	0.568	0.465	0.594	0.483	0.489	0.534	0.496	0.525
	GF	1.174	1.063	1.236	0.923	1.214	0.973	0.865	1.024	0.877	0.979
	F1-Score	0.35	0.45	0.40	0.45	0.48	0.41	0.44	0.41	0.46	0.47
*k*NNE	Testing accuracy	0.477	0.636	0.789	0.565	0.668	0.788	0.805	0.793	0.794	0.797
	Training accuracy	0.595	0.720	0.859	0.634	0.767	0.883	0.879	0.875	0.865	0.881
	GF	1.291	1.300	1.496	1.189	1.425	1.812	1.612	1.656	1.526	1.706
	F1-Score	0.53	0.68	0.80	0.59	0.72	0.81	0.82	0.85	0.78	0.82
DTE	Testing accuracy	0.537	0.624	0.765	0.558	0.648	0.761	0.765	0.732	0.761	0.763
	Training accuracy	0.600	0.703	0.830	0.631	0.747	0.861	0.842	0.843	0.839	0.843
	GF	1.157	1.266	1.382	1.198	1.391	1.719	1.487	1.707	1.484	1.510
	F1-Score	0.52	0.66	0.81	0.53	0.70	0.80	0.79	0.82	0.80	0.81
RF	Testing accuracy	0.536	0.634	0.735	0.585	0.693	0.784	0.770	0.806	0.787	0.803
	Training accuracy	0.546	0.644	0.815	0.632	0.766	0.854	0.860	0.870	0.856	0.873
	GF	1.022	1.028	1.432	1.128	1.312	1.479	1.643	1.492	1.479	1.551
	F1-Score	0.54	0.69	0.81	0.56	0.74	0.84	0.82	0.83	0.78	0.82
SVME	Testing accuracy	0.461	0.692	0.792	0.587	0.695	0.768	0.783	0.794	0.730	0.774
	Training accuracy	0.606	0.725	0.853	0.647	0.721	0.860	0.860	0.866	0.856	0.858
	GF	1.368	1.120	1.415	1.170	1.093	1.657	1.550	1.537	1.875	1.592
	F1-Score	0.55	0.70	0.81	0.59	0.69	0.85	0.84	0.83	0.82	0.85
NNE	Testing accuracy	0.541	0.625	0.730	0.581	0.637	0.743	0.685	0.665	0.714	0.668
	Training accuracy	0.629	0.688	0.819	0.634	0.722	0.812	0.787	0.797	0.790	0.803
	GF	1.237	1.202	1.492	1.145	1.306	1.367	1.479	1.650	1.362	1.685
	F1-Score	0.53	0.68	0.78	0.56	0.68	0.71	0.72	0.72	0.72	0.72

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.497	0.479	0.475	0.498	0.522	0.503	0.703	0.554	0.655	0.651
	Training accuracy	0.593	0.601	0.581	0.526	0.646	0.651	0.691	0.688	0.630	0.657
	GF	1.236	1.306	1.253	1.059	1.350	1.424	0.961	1.429	0.932	1.017
	F1-Score	0.39	0.45	0.44	0.48	0.51	0.67	0.66	0.60	0.61	0.64
*k*NNhte	Testing accuracy	0.478	0.630	0.769	0.556	0.645	0.754	0.797	0.764	0.772	0.773
	Training accuracy	0.584	0.697	0.829	0.602	0.740	0.850	0.855	0.866	0.834	0.854
	GF	1.255	1.221	1.351	1.116	1.365	1.640	1.400	1.761	1.373	1.555
	F1-Score	0.52	0.70	0.80	0.59	0.72	0.82	0.85	0.85	0.80	0.82
DThte	Testing accuracy	0.537	0.645	0.759	0.551	0.666	0.767	0.745	0.751	0.773	0.761
	Training accuracy	0.611	0.713	0.847	0.632	0.772	0.863	0.855	0.857	0.849	0.851
	GF	1.190	1.237	1.575	1.220	1.465	1.701	1.759	1.741	1.503	1.604
	F1-Score	0.55	0.68	0.79	0.57	0.73	0.82	0.81	0.83	0.83	0.82
SVMhte	Testing accuracy	0.492	0.644	0.811	0.491	0.680	0.831	0.815	0.816	0.822	0.816
	Training accuracy	0.603	0.735	0.862	0.636	0.752	0.873	0.877	0.870	0.877	0.871
	GF	1.280	1.343	1.370	1.398	1.290	1.331	1.504	1.415	1.447	1.426
	F1-Score	0.44	0.72	0.83	0.51	0.74	0.85	0.84	0.85	0.85	0.84
NNhte	Testing accuracy	0.507	0.632	0.773	0.555	0.645	0.752	0.764	0.760	0.758	0.768
	Training accuracy	0.605	0.705	0.844	0.638	0.729	0.849	0.832	0.846	0.825	0.849
	GF	1.248	1.247	1.455	1.229	1.310	1.642	1.405	1.558	1.383	1.536
	F1-Score	0.59	0.69	0.78	0.58	0.71	0.79	0.80	0.77	0.79	0.75
HTEsm	Testing accuracy	0.513	0.684	0.831	0.594	0.715	0.753	0.806	0.812	0.773	0.788
	Training accuracy	0.607	0.693	0.859	0.630	0.736	0.860	0.878	0.866	0.843	0.869
	GF	1.239	1.029	1.199	1.097	1.080	1.764	1.590	1.403	1.446	1.618
	F1-Score	0.50	0.68	0.83	0.60	0.74	0.85	0.82	0.80	0.84	0.83
HTEdf	Testing accuracy	0.545	0.688	0.832	0.596	0.708	0.749	0.810	0.812	0.770	0.808
	Training accuracy	0.602	0.689	0.857	0.628	0.734	0.855	0.878	0.866	0.831	0.863
	GF	1.143	1.003	1.175	1.086	1.098	1.731	1.557	1.403	1.361	1.401
	F1-Score	0.54	0.70	0.84	0.62	0.75	0.86	0.86	0.86	0.86	0.87

Red Wine Dataset

Table E.5: Ensemble Performance on Feature Subsets of the Red Wine Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.495	0.536	0.379	0.392	0.501	0.433	0.499	0.512	0.440	0.471
	Training accuracy	0.455	0.579	0.506	0.507	0.579	0.542	0.582	0.562	0.588	0.567
	GF	0.927	1.102	1.257	1.233	1.185	1.238	1.199	1.114	1.359	1.222
	F1-Score	0.44	0.50	0.42	0.43	0.48	0.48	0.49	0.52	0.52	0.51
*k*NNE	Testing accuracy	0.373	0.558	0.376	0.446	0.502	0.475	0.532	0.536	0.518	0.510
	Training accuracy	0.410	0.558	0.544	0.575	0.617	0.627	0.620	0.641	0.633	0.645
	GF	1.063	1.000	1.368	1.304	1.300	1.408	1.232	1.292	1.313	1.380
	F1-Score	0.41	0.49	0.47	0.47	0.52	0.54	0.58	0.53	0.52	0.52
DTE	Testing accuracy	0.394	0.493	0.426	0.464	0.488	0.469	0.485	0.487	0.492	0.488
	Training accuracy	0.451	0.553	0.506	0.532	0.573	0.589	0.600	0.603	0.599	0.605
	GF	1.104	1.134	1.162	1.145	1.199	1.292	1.287	1.292	1.267	1.296
	F1-Score	0.43	0.52	0.50	0.49	0.54	0.55	0.56	0.57	0.50	0.53
RF	Testing accuracy	0.362	0.543	0.448	0.460	0.531	0.496	0.529	0.545	0.546	0.532
	Training accuracy	0.449	0.565	0.562	0.593	0.647	0.655	0.658	0.674	0.675	0.672
	GF	1.158	1.051	1.260	1.327	1.329	1.461	1.377	1.396	1.397	1.427
	F1-Score	0.43	0.48	0.54	0.54	0.56	0.59	0.60	0.59	0.59	0.63
SVME	Testing accuracy	0.414	0.540	0.418	0.414	0.524	0.512	0.516	0.544	0.536	0.539
	Training accuracy	0.442	0.592	0.525	0.523	0.582	0.538	0.595	0.600	0.598	0.610
	GF	1.050	1.127	1.225	1.229	1.139	1.056	1.195	1.140	1.154	1.182
	F1-Score	0.44	0.50	0.41	0.44	0.53	0.49	0.52	0.53	0.51	0.54
NNE	Testing accuracy	0.491	0.574	0.451	0.482	0.493	0.448	0.513	0.538	0.535	0.544
	Training accuracy	0.455	0.594	0.527	0.540	0.608	0.621	0.644	0.655	0.647	0.662
	GF	0.934	1.049	1.161	1.126	1.293	1.456	1.368	1.339	1.317	1.349
	F1-Score	0.44	0.50	0.48	0.48	0.54	0.57	0.59	0.58	0.58	0.63

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.497	0.533	0.388	0.411	0.515	0.464	0.502	0.524	0.470	0.488
	Training accuracy	0.453	0.581	0.509	0.509	0.579	0.540	0.591	0.583	0.592	0.590
	GF	0.920	1.115	1.246	1.200	1.152	1.165	1.218	1.141	1.299	1.249
	F1-Score	0.46	0.49	0.42	0.42	0.46	0.49	0.50	0.53	0.52	0.52
*k*NNhte	Testing accuracy	0.421	0.534	0.433	0.455	0.489	0.445	0.491	0.520	0.548	0.529
	Training accuracy	0.431	0.546	0.506	0.511	0.569	0.584	0.587	0.596	0.581	0.580
	GF	1.018	1.026	1.148	1.115	1.186	1.334	1.232	1.188	1.079	1.121
	F1-Score	0.40	0.51	0.52	0.53	0.57	0.59	0.62	0.59	0.58	0.62
DThte	Testing accuracy	0.382	0.521	0.441	0.488	0.521	0.482	0.525	0.534	0.537	0.541
	Training accuracy	0.459	0.572	0.571	0.604	0.649	0.654	0.664	0.678	0.676	0.682
	GF	1.142	1.119	1.303	1.293	1.365	1.497	1.414	1.447	1.429	1.443
	F1-Score	0.43	0.52	0.55	0.55	0.58	0.59	0.61	0.60	0.58	0.59
SVMhte	Testing accuracy	0.448	0.551	0.452	0.489	0.526	0.487	0.521	0.536	0.543	0.537
	Training accuracy	0.462	0.599	0.534	0.551	0.617	0.633	0.646	0.653	0.650	0.653
	GF	1.026	1.120	1.176	1.138	1.238	1.398	1.353	1.337	1.306	1.334
	F1-Score	0.42	0.50	0.42	0.44	0.52	0.47	0.52	0.53	0.52	0.53
NNhte	Testing accuracy	0.488	0.559	0.470	0.486	0.512	0.492	0.534	0.545	0.554	0.552
	Training accuracy	0.428	0.589	0.523	0.548	0.610	0.644	0.655	0.653	0.650	0.660
	GF	0.895	1.073	1.111	1.137	1.251	1.427	1.351	1.311	1.274	1.318
	F1-Score	0.43	0.49	0.42	0.46	0.53	0.58	0.57	0.61	0.61	0.60
HTEsm	Testing accuracy	0.439	0.547	0.441	0.498	0.540	0.479	0.536	0.547	0.556	0.543
	Training accuracy	0.456	0.606	0.552	0.582	0.627	0.648	0.650	0.660	0.654	0.660
	GF	1.031	1.150	1.248	1.201	1.233	1.480	1.326	1.332	1.283	1.344
	F1-Score	0.41	0.51	0.48	0.48	0.53	0.58	0.59	0.58	0.58	0.59
HTEdf	Testing accuracy	0.513	0.539	0.469	0.484	0.538	0.501	0.537	0.554	0.549	0.557
	Training accuracy	0.455	0.588	0.521	0.526	0.598	0.566	0.613	0.605	0.607	0.614
	GF	0.894	1.119	1.109	1.089	1.149	1.150	1.196	1.129	1.148	1.148
	F1-Score	0.42	0.54	0.51	0.50	0.56	0.60	0.57	0.59	0.59	0.59

Car Evaluation Dataset

Table E.6: Ensemble Performance on Feature Subsets of the Car Evaluation Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.541	0.284	0.456	0.519	0.754	0.431	0.552	0.797	0.809	0.809
	Training accuracy	0.459	0.218	0.365	0.470	0.713	0.422	0.458	0.771	0.808	0.801
	GF	0.848	0.916	0.857	0.908	0.857	0.984	0.827	0.886	0.995	0.960
	F1-Score	0.62	0.34	0.54	0.57	0.79	0.50	0.60	0.81	0.82	0.82
*k*NNE	Testing accuracy	0.725	0.655	0.686	0.694	0.780	0.626	0.744	0.864	0.852	0.844
	Training accuracy	0.614	0.687	0.656	0.772	0.789	0.695	0.762	0.877	0.917	0.890
	GF	0.712	1.102	0.913	1.342	1.043	1.226	1.076	1.106	1.783	1.418
	F1-Score	0.70	0.61	0.64	0.67	0.77	0.59	0.72	0.83	0.91	0.85
DTE	Testing accuracy	0.734	0.695	0.702	0.705	0.767	0.553	0.831	0.901	0.932	0.927
	Training accuracy	0.699	0.723	0.683	0.794	0.800	0.584	0.807	0.906	0.959	0.952
	GF	0.884	1.101	0.940	1.432	1.165	1.075	0.876	1.053	1.659	1.521
	F1-Score	0.55	0.55	0.61	0.68	0.77	0.60	0.79	0.88	0.94	0.97
RF	Testing accuracy	0.734	0.697	0.710	0.690	0.776	0.591	0.811	0.885	0.899	0.881
	Training accuracy	0.699	0.723	0.679	0.789	0.789	0.606	0.807	0.880	0.950	0.930
	GF	0.884	1.094	0.903	1.469	1.062	1.038	0.979	0.958	2.020	1.700
	F1-Score	0.55	0.55	0.60	0.67	0.76	0.56	0.77	0.86	0.94	0.91
SVME	Testing accuracy	0.734	0.698	0.709	0.698	0.777	0.673	0.790	0.892	0.875	0.871
	Training accuracy	0.706	0.723	0.679	0.794	0.809	0.724	0.819	0.887	0.964	0.956
	GF	0.905	1.090	0.907	1.466	1.168	1.185	1.160	0.956	3.472	2.932
	F1-Score	0.55	0.55	0.55	0.70	0.75	0.55	0.78	0.86	0.96	0.94
NNE	Testing accuracy	0.734	0.697	0.709	0.690	0.764	0.589	0.860	0.928	0.963	0.955
	Training accuracy	0.699	0.723	0.680	0.790	0.807	0.648	0.823	0.895	0.995	0.997
	GF	0.884	1.094	0.909	1.476	1.223	1.168	0.791	0.686	7.400	15.000
	F1-Score	0.55	0.55	0.59	0.70	0.77	0.60	0.80	0.89	0.97	0.98

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.541	0.284	0.460	0.517	0.755	0.464	0.562	0.797	0.809	0.809
	Training accuracy	0.459	0.218	0.460	0.470	0.713	0.463	0.480	0.771	0.808	0.801
	GF	0.848	0.916	1.000	0.911	0.854	0.998	0.842	0.886	0.995	0.960
	F1-Score	0.62	0.34	0.58	0.57	0.79	0.50	0.61	0.81	0.82	0.82
*k*NNhte	Testing accuracy	0.709	0.646	0.680	0.704	0.776	0.610	0.736	0.864	0.843	0.830
	Training accuracy	0.628	0.682	0.652	0.763	0.783	0.671	0.777	0.863	0.920	0.878
	GF	0.782	1.113	0.920	1.249	1.032	1.185	1.184	0.993	1.963	1.393
	F1-Score	0.56	0.63	0.60	0.67	0.74	0.58	0.72	0.83	0.91	0.83
DThte	Testing accuracy	0.734	0.695	0.706	0.687	0.776	0.606	0.840	0.905	0.934	0.936
	Training accuracy	0.699	0.723	0.676	0.795	0.808	0.652	0.809	0.892	0.968	0.966
	GF	0.884	1.101	0.907	1.527	1.167	1.132	0.838	0.880	2.062	1.882
	F1-Score	0.55	0.55	0.57	0.68	0.76	0.60	0.79	0.89	0.97	0.94
SVMhte	Testing accuracy	0.734	0.684	0.667	0.722	0.794	0.683	0.846	0.891	0.909	0.902
	Training accuracy	0.706	0.723	0.689	0.802	0.809	0.736	0.858	0.891	0.944	0.951
	GF	0.905	1.141	1.071	1.404	1.079	1.201	1.085	1.000	1.625	2.000
	F1-Score	0.55	0.55	0.55	0.70	0.75	0.63	0.80	0.87	0.93	0.94
NNhte	Testing accuracy	0.725	0.694	0.709	0.713	0.783	0.628	0.856	0.921	0.954	0.950
	Training accuracy	0.704	0.723	0.689	0.789	0.805	0.683	0.828	0.897	0.991	0.992
	GF	0.929	1.105	0.936	1.360	1.113	1.174	0.837	0.767	5.111	6.250
	F1-Score	0.55	0.55	0.55	0.71	0.77	0.59	0.83	0.90	0.98	0.99
HTEsm	Testing accuracy	0.734	0.700	0.688	0.674	0.771	0.620	0.847	0.924	0.958	0.949
	Training accuracy	0.701	0.723	0.675	0.791	0.791	0.658	0.829	0.894	0.975	0.980
	GF	0.890	1.083	0.960	1.560	1.096	1.111	0.895	0.717	1.680	2.550
	F1-Score	0.70	0.56	0.60	0.69	0.77	0.61	0.82	0.90	0.98	0.99
HTEdf	Testing accuracy	0.734	0.701	0.694	0.689	0.775	0.633	0.839	0.922	0.951	0.944
	Training accuracy	0.701	0.723	0.680	0.792	0.800	0.664	0.835	0.895	0.975	0.979
	GF	0.890	1.079	0.956	1.495	1.125	1.092	0.976	0.743	1.960	2.667
	F1-Score	0.70	0.57	0.60	0.71	0.78	0.62	0.83	0.90	0.98	0.99

White Wine Dataset

Table E.7: Ensemble Performance on Feature Subsets of the White Wine Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.439	0.468	0.436	0.444	0.468	0.428	0.384	0.443	0.426	0.427
	Training accuracy	0.455	0.495	0.455	0.508	0.520	0.469	0.446	0.502	0.474	0.469
	GF	1.029	1.053	1.035	1.130	1.108	1.077	1.112	1.118	1.091	1.079
	F1-Score	0.44	0.38	0.41	0.39	0.40	0.41	0.42	0.40	0.41	0.41
*k*NNE	Testing accuracy	0.384	0.460	0.402	0.458	0.467	0.433	0.439	0.473	0.467	0.481
	Training accuracy	0.404	0.482	0.462	0.503	0.522	0.524	0.541	0.569	0.556	0.558
	GF	1.034	1.042	1.112	1.091	1.115	1.191	1.222	1.223	1.200	1.174
	F1-Score	0.47	0.49	0.49	0.50	0.51	0.51	0.53	0.53	0.54	0.54
DTE	Testing accuracy	0.438	0.445	0.415	0.464	0.487	0.467	0.455	0.479	0.480	0.476
	Training accuracy	0.453	0.498	0.480	0.538	0.564	0.528	0.537	0.556	0.550	0.556
	GF	1.027	1.106	1.125	1.160	1.177	1.129	1.177	1.173	1.156	1.180
	F1-Score	0.45	0.47	0.48	0.51	0.50	0.54	0.54	0.55	0.55	0.54
RF	Testing accuracy	0.425	0.430	0.434	0.483	0.521	0.490	0.494	0.520	0.521	0.524
	Training accuracy	0.458	0.492	0.563	0.606	0.645	0.620	0.624	0.649	0.642	0.648
	GF	1.061	1.122	1.295	1.312	1.349	1.342	1.346	1.368	1.338	1.352
	F1-Score	0.49	0.52	0.54	0.57	0.58	0.59	0.60	0.62	0.62	0.64
SVME	Testing accuracy	0.441	0.480	0.431	0.485	0.512	0.477	0.498	0.512	0.514	0.517
	Training accuracy	0.455	0.498	0.463	0.508	0.548	0.500	0.526	0.561	0.559	0.564
	GF	1.026	1.036	1.060	1.047	1.080	1.046	1.059	1.112	1.102	1.108
	F1-Score	0.46	0.50	0.49	0.50	0.49	0.51	0.51	0.52	0.51	0.51
NNE	Testing accuracy	0.433	0.471	0.449	0.491	0.519	0.473	0.489	0.511	0.499	0.509
	Training accuracy	0.455	0.501	0.471	0.535	0.595	0.574	0.597	0.622	0.631	0.631
	GF	1.040	1.060	1.042	1.095	1.188	1.237	1.268	1.294	1.358	1.331
	F1-Score	0.51	0.51	0.54	0.56	0.57	0.58	0.59	0.59	0.59	0.60

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.441	0.452	0.433	0.462	0.467	0.451	0.431	0.468	0.458	0.451
	Training accuracy	0.456	0.469	0.461	0.501	0.538	0.467	0.472	0.523	0.495	0.492
	GF	1.028	1.032	1.052	1.078	1.154	1.030	1.078	1.115	1.073	1.081
	F1-Score	0.47	0.44	0.45	0.44	0.46	0.46	0.46	0.45	0.45	0.46
*k*NNhte	Testing accuracy	0.388	0.445	0.429	0.471	0.487	0.470	0.473	0.492	0.495	0.499
	Training accuracy	0.393	0.484	0.526	0.565	0.583	0.588	0.590	0.627	0.612	0.611
	GF	1.008	1.076	1.205	1.216	1.230	1.286	1.285	1.362	1.302	1.288
	F1-Score	0.48	0.51	0.53	0.54	0.55	0.57	0.58	0.60	0.60	0.60
DThte	Testing accuracy	0.440	0.443	0.441	0.497	0.530	0.507	0.508	0.521	0.521	0.522
	Training accuracy	0.466	0.505	0.575	0.620	0.650	0.628	0.637	0.658	0.651	0.646
	GF	1.049	1.125	1.315	1.324	1.343	1.325	1.355	1.401	1.372	1.350
	F1-Score	0.51	0.50	0.55	0.57	0.57	0.60	0.60	0.64	0.63	0.63
SVMhte	Testing accuracy	0.441	0.453	0.440	0.477	0.502	0.458	0.465	0.508	0.492	0.497
	Training accuracy	0.455	0.500	0.458	0.498	0.545	0.465	0.506	0.548	0.542	0.545
	GF	1.026	1.094	1.033	1.042	1.095	1.013	1.083	1.088	1.109	1.105
	F1-Score	0.45	0.49	0.45	0.48	0.48	0.51	0.49	0.51	0.49	0.50
NNhte	Testing accuracy	0.440	0.468	0.450	0.492	0.529	0.504	0.513	0.536	0.520	0.534
	Training accuracy	0.455	0.489	0.467	0.528	0.598	0.586	0.591	0.618	0.616	0.622
	GF	1.028	1.041	1.032	1.076	1.172	1.198	1.191	1.215	1.250	1.233
	F1-Score	0.52	0.51	0.50	0.57	0.54	0.55	0.49	0.58	0.58	0.59
HTEsm	Testing accuracy	0.449	0.487	0.440	0.499	0.531	0.492	0.488	0.528	0.523	0.524
	Training accuracy	0.457	0.511	0.504	0.557	0.591	0.571	0.592	0.611	0.613	0.612
	GF	1.015	1.049	1.129	1.131	1.147	1.184	1.255	1.213	1.233	1.227
	F1-Score	0.51	0.52	0.53	0.54	0.55	0.57	0.57	0.57	0.56	0.57
HTEdf	Testing accuracy	0.451	0.486	0.449	0.500	0.531	0.497	0.501	0.535	0.530	0.533
	Training accuracy	0.459	0.513	0.527	0.572	0.603	0.590	0.607	0.624	0.621	0.621
	GF	1.015	1.055	1.165	1.168	1.181	1.227	1.270	1.237	1.240	1.232
	F1-Score	0.51	0.53	0.53	0.55	0.55	0.59	0.58	0.59	0.58	0.60

Nursery Dataset

Table E.8: Ensemble Performance on Feature Subsets of the Nursery Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.512	0.374	0.592	0.692	0.777	0.852	0.816	0.865	0.837	0.876
	Training accuracy	0.432	0.510	0.687	0.781	0.816	0.863	0.842	0.873	0.829	0.885
	GF	0.859	1.278	1.304	1.406	1.212	1.080	1.165	1.063	0.953	1.078
	F1-Score	0.38	0.42	0.61	0.73	0.76	0.81	0.79	0.82	0.80	0.83
*k*NNE	Testing accuracy	0.427	0.465	0.784	0.784	0.791	0.835	0.822	0.705	0.736	0.843
	Training accuracy	0.345	0.391	0.654	0.705	0.825	0.898	0.887	0.818	0.838	0.904
	GF	0.875	0.878	0.624	0.732	1.194	1.618	1.575	1.621	1.630	1.635
	F1-Score	0.42	0.52	0.76	0.80	0.79	0.84	0.83	0.75	0.82	0.89
DTE	Testing accuracy	0.504	0.562	0.807	0.810	0.847	0.893	0.885	0.922	0.902	0.926
	Training accuracy	0.431	0.544	0.780	0.851	0.874	0.924	0.920	0.950	0.924	0.957
	GF	0.872	0.961	0.877	1.275	1.214	1.408	1.438	1.560	1.289	1.721
	F1-Score	0.36	0.52	0.76	0.79	0.81	0.88	0.87	0.94	0.92	0.95
RF	Testing accuracy	0.505	0.561	0.804	0.802	0.830	0.882	0.875	0.887	0.883	0.916
	Training accuracy	0.433	0.545	0.779	0.851	0.883	0.926	0.925	0.946	0.927	0.965
	GF	0.873	0.965	0.887	1.329	1.453	1.595	1.667	2.093	1.603	2.400
	F1-Score	0.36	0.52	0.76	0.79	0.81	0.88	0.87	0.91	0.89	0.94
SVME	Testing accuracy	0.503	0.564	0.804	0.818	0.846	0.888	0.873	0.849	0.837	0.899
	Training accuracy	0.432	0.547	0.781	0.859	0.889	0.933	0.933	0.957	0.938	0.972
	GF	0.875	0.962	0.895	1.291	1.387	1.672	1.896	3.512	2.629	3.607
	F1-Score	0.36	0.53	0.75	0.80	0.82	0.89	0.87	0.92	0.90	0.94
NNE	Testing accuracy	0.508	0.557	0.798	0.800	0.848	0.901	0.895	0.938	0.939	0.955
	Training accuracy	0.419	0.549	0.777	0.857	0.888	0.936	0.934	0.971	0.970	0.986
	GF	0.847	0.982	0.906	1.399	1.357	1.547	1.591	2.138	2.033	3.214
	F1-Score	0.36	0.50	0.75	0.80	0.83	0.89	0.88	0.95	0.95	0.97

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.510	0.375	0.680	0.696	0.780	0.853	0.826	0.870	0.840	0.883
	Training accuracy	0.437	0.515	0.689	0.802	0.821	0.871	0.845	0.877	0.855	0.893
	GF	0.870	1.289	1.029	1.535	1.229	1.140	1.123	1.057	1.103	1.093
	F1-Score	0.38	0.48	0.63	0.74	0.76	0.82	0.80	0.83	0.81	0.84
*k*NNhte	Testing accuracy	0.455	0.497	0.780	0.788	0.785	0.828	0.827	0.708	0.749	0.836
	Training accuracy	0.359	0.413	0.647	0.704	0.833	0.875	0.871	0.814	0.851	0.896
	GF	0.850	0.857	0.623	0.716	1.287	1.376	1.341	1.570	1.685	1.577
	F1-Score	0.44	0.54	0.78	0.81	0.80	0.86	0.84	0.77	0.79	0.89
DThte	Testing accuracy	0.503	0.561	0.810	0.811	0.847	0.891	0.885	0.929	0.919	0.934
	Training accuracy	0.431	0.545	0.779	0.851	0.887	0.932	0.926	0.962	0.952	0.973
	GF	0.873	0.965	0.860	1.268	1.354	1.603	1.554	1.868	1.687	2.444
	F1-Score	0.38	0.51	0.74	0.80	0.82	0.88	0.87	0.94	0.94	0.97
SVMhte	Testing accuracy	0.512	0.574	0.809	0.809	0.836	0.886	0.880	0.913	0.917	0.939
	Training accuracy	0.436	0.550	0.775	0.846	0.874	0.927	0.930	0.961	0.953	0.979
	GF	0.865	0.947	0.849	1.240	1.302	1.562	1.714	2.231	1.766	2.905
	F1-Score	0.38	0.52	0.76	0.79	0.82	0.89	0.89	0.93	0.93	0.95
NNhte	Testing accuracy	0.506	0.559	0.803	0.806	0.849	0.899	0.898	0.938	0.940	0.954
	Training accuracy	0.422	0.549	0.780	0.855	0.888	0.935	0.932	0.967	0.965	0.982
	GF	0.855	0.978	0.895	1.338	1.348	1.554	1.500	1.879	1.714	2.556
	F1-Score	0.39	0.50	0.75	0.80	0.84	0.89	0.88	0.95	0.95	0.98
HTEsm	Testing accuracy	0.506	0.562	0.808	0.823	0.857	0.900	0.910	0.955	0.977	0.977
	Training accuracy	0.432	0.547	0.777	0.856	0.892	0.938	0.938	0.981	0.985	0.995
	GF	0.870	0.967	0.861	1.229	1.324	1.613	1.452	2.368	1.533	4.600
	F1-Score	0.38	0.53	0.76	0.80	0.83	0.90	0.89	0.97	0.99	1.00
HTEdf	Testing accuracy	0.503	0.562	0.804	0.810	0.852	0.902	0.908	0.966	0.985	0.985
	Training accuracy	0.434	0.546	0.780	0.852	0.886	0.937	0.937	0.982	0.987	0.997
	GF	0.878	0.965	0.891	1.284	1.298	1.556	1.460	1.889	1.154	5.000
	F1-Score	0.38	0.50	0.75	0.80	0.82	0.90	0.89	0.97	0.99	1.00

Bank Marketing Dataset

Table E.9: Ensemble Performance on Feature Subsets of the Bank Marketing Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.897	0.828	0.272	0.866	0.556	0.867	0.782	0.301	0.787	0.556
	Training accuracy	0.702	0.698	0.705	0.729	0.729	0.691	0.767	0.704	0.787	0.799
	GF	0.346	0.570	2.468	0.494	1.638	0.430	0.936	2.361	1.000	2.209
	F1-Score	0.79	0.69	0.83	0.86	0.86	0.61	0.70	0.63	0.78	0.78
*k*NNE	Testing accuracy	0.883	0.885	0.895	0.895	0.891	0.894	0.897	0.890	0.897	0.897
	Training accuracy	0.798	0.841	0.843	0.882	0.859	0.848	0.883	0.836	0.869	0.827
	GF	0.579	0.723	0.669	0.890	0.773	0.697	0.880	0.671	0.786	0.595
	F1-Score	0.87	0.86	0.85	0.85	0.82	0.79	0.81	0.74	0.79	0.73
DTE	Testing accuracy	0.884	0.881	0.884	0.881	0.890	0.878	0.885	0.880	0.885	0.885
	Training accuracy	0.805	0.846	0.865	0.901	0.887	0.865	0.935	0.909	0.938	0.937
	GF	0.595	0.773	0.859	1.202	0.973	0.904	1.769	1.319	1.855	1.825
	F1-Score	0.86	0.79	0.87	0.87	0.86	0.85	0.88	0.87	0.88	0.89
RF	Testing accuracy	0.881	0.881	0.879	0.886	0.885	0.881	0.903	0.891	0.901	0.900
	Training accuracy	0.827	0.895	0.906	0.934	0.930	0.930	0.949	0.940	0.950	0.950
	GF	0.688	1.133	1.287	1.727	1.643	1.700	1.902	1.817	1.980	2.000
	F1-Score	0.86	0.86	0.86	0.87	0.87	0.87	0.90	0.86	0.90	0.90
SVME	Testing accuracy	0.897	0.890	0.895	0.897	0.887	0.891	0.890	0.891	0.893	0.894
	Training accuracy	0.722	0.839	0.862	0.891	0.911	0.938	0.960	0.968	0.970	0.974
	GF	0.371	0.683	0.761	0.945	1.270	1.758	2.750	3.406	3.567	4.077
	F1-Score	0.77	0.83	0.84	0.86	0.84	0.84	0.85	0.84	0.85	0.84
NNE	Testing accuracy	0.895	0.884	0.877	0.879	0.873	0.869	0.892	0.877	0.894	0.891
	Training accuracy	0.733	0.876	0.896	0.922	0.932	0.939	0.964	0.954	0.968	0.969
	GF	0.393	0.935	1.183	1.551	1.868	2.148	3.000	2.674	3.312	3.516
	F1-Score	0.79	0.86	0.84	0.86	0.85	0.84	0.89	0.86	0.89	0.89

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.897	0.870	0.816	0.878	0.854	0.877	0.835	0.830	0.842	0.831
	Training accuracy	0.701	0.689	0.699	0.729	0.736	0.729	0.836	0.791	0.819	0.829
	GF	0.344	0.418	0.611	0.450	0.553	0.454	1.006	0.813	0.873	0.988
	F1-Score	0.79	0.68	0.84	0.86	0.86	0.75	0.83	0.80	0.84	0.84
*k*NNhte	Testing accuracy	0.886	0.886	0.889	0.892	0.882	0.892	0.894	0.892	0.893	0.893
	Training accuracy	0.814	0.861	0.871	0.909	0.899	0.892	0.916	0.865	0.896	0.856
	GF	0.613	0.820	0.860	1.187	1.168	1.000	1.262	0.800	1.029	0.743
	F1-Score	0.84	0.86	0.86	0.85	0.84	0.82	0.84	0.78	0.82	0.78
DThte	Testing accuracy	0.886	0.882	0.885	0.883	0.890	0.879	0.896	0.887	0.893	0.893
	Training accuracy	0.831	0.891	0.901	0.927	0.923	0.921	0.949	0.937	0.951	0.951
	GF	0.675	1.083	1.162	1.603	1.429	1.532	2.039	1.794	2.184	2.184
	F1-Score	0.86	0.86	0.88	0.87	0.87	0.87	0.89	0.88	0.89	0.89
SVMhte	Testing accuracy	0.893	0.894	0.900	0.901	0.896	0.898	0.904	0.893	0.903	0.903
	Training accuracy	0.717	0.759	0.766	0.785	0.810	0.837	0.921	0.887	0.934	0.941
	GF	0.378	0.440	0.427	0.460	0.547	0.626	1.215	0.947	1.470	1.644
	F1-Score	0.77	0.83	0.85	0.86	0.86	0.85	0.89	0.87	0.89	0.89
NNhte	Testing accuracy	0.896	0.887	0.884	0.892	0.885	0.886	0.900	0.887	0.898	0.897
	Training accuracy	0.721	0.869	0.889	0.917	0.924	0.920	0.960	0.946	0.964	0.966
	GF	0.373	0.863	1.045	1.301	1.513	1.425	2.500	2.093	2.833	3.029
	F1-Score	0.79	0.87	0.87	0.87	0.86	0.86	0.89	0.86	0.89	0.89
HTEsm	Testing accuracy	0.900	0.885	0.886	0.896	0.886	0.896	0.899	0.884	0.899	0.900
	Training accuracy	0.791	0.875	0.883	0.910	0.916	0.919	0.958	0.947	0.959	0.962
	GF	0.478	0.920	0.974	1.156	1.357	1.284	2.405	2.189	2.463	2.632
	F1-Score	0.86	0.85	0.87	0.87	0.86	0.86	0.89	0.88	0.89	0.89
HTEdf	Testing accuracy	0.898	0.886	0.893	0.896	0.889	0.899	0.900	0.891	0.899	0.897
	Training accuracy	0.791	0.872	0.876	0.904	0.904	0.911	0.950	0.930	0.952	0.954
	GF	0.488	0.891	0.863	1.083	1.156	1.135	2.000	1.557	2.104	2.239
	F1-Score	0.86	0.86	0.86	0.87	0.88	0.89	0.89	0.89	0.89	0.89

Censor Income Dataset

Table E.10: Ensemble Performance on Feature Subsets of the Censor Income Dataset

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBE	Testing accuracy	0.578	0.312	0.510	0.478	0.380	0.514	0.423	0.443	0.446	0.453
	Training accuracy	0.598	0.529	0.653	0.688	0.646	0.714	0.721	0.699	0.754	0.782
	GF	1.050	1.461	1.412	1.673	1.751	1.699	2.068	1.850	2.252	2.509
	F1-Score	0.42	0.26	0.56	0.57	0.52	0.62	0.64	0.62	0.68	0.73
*k*NNE	Testing accuracy	0.717	0.746	0.767	0.774	0.785	0.779	0.801	0.786	0.782	0.791
	Training accuracy	0.556	0.584	0.689	0.774	0.773	0.817	0.816	0.834	0.851	0.852
	GF	0.637	0.611	0.749	1.000	0.947	1.208	1.082	1.289	1.463	1.412
	F1-Score	0.61	0.75	0.71	0.75	0.75	0.76	0.76	0.78	0.79	0.79
DTE	Testing accuracy	0.751	0.749	0.760	0.763	0.766	0.796	0.800	0.793	0.791	0.796
	Training accuracy	0.647	0.743	0.708	0.806	0.792	0.818	0.805	0.823	0.839	0.825
	GF	0.705	0.977	0.822	1.222	1.125	1.121	1.026	1.169	1.298	1.166
	F1-Score	0.61	0.75	0.68	0.75	0.75	0.76	0.77	0.77	0.78	0.77
RF	Testing accuracy	0.751	0.748	0.764	0.774	0.779	0.798	0.808	0.806	0.801	0.809
	Training accuracy	0.647	0.743	0.716	0.821	0.808	0.839	0.838	0.848	0.866	0.866
	GF	0.705	0.981	0.831	1.263	1.151	1.255	1.185	1.276	1.485	1.425
	F1-Score	0.62	0.75	0.69	0.76	0.75	0.77	0.77	0.79	0.80	0.80
SVME	Testing accuracy	0.751	0.745	0.776	0.789	0.790	0.803	0.805	0.804	0.804	0.799
	Training accuracy	0.647	0.743	0.713	0.807	0.793	0.827	0.831	0.848	0.858	0.870
	GF	0.705	0.992	0.780	1.093	1.014	1.139	1.154	1.289	1.380	1.546
	F1-Score	0.62	0.75	0.70	0.76	0.75	0.78	0.77	0.80	0.79	0.79
NNE	Testing accuracy	0.751	0.747	0.771	0.784	0.788	0.795	0.809	0.801	0.807	0.808
	Training accuracy	0.647	0.743	0.713	0.819	0.801	0.833	0.836	0.847	0.859	0.871
	GF	0.705	0.984	0.798	1.193	1.065	1.228	1.165	1.301	1.369	1.488
	F1-Score	0.62	0.75	0.70	0.77	0.76	0.78	0.78	0.80	0.81	0.81

Ensemble	Measure	Feature Subsets %									
		10	20	30	40	50	60	70	80	90	100
NBhte	Testing accuracy	0.586	0.584	0.538	0.597	0.616	0.72	0.626	0.62	0.684	0.691
	Training accuracy	0.598	0.702	0.655	0.730	0.713	0.80	0.802	0.79	0.834	0.829
	GF	1.030	1.396	1.339	1.493	1.338	1.40	1.889	1.81	1.904	1.807
	F1-Score	0.45	0.60	0.57	0.64	0.64	0.76	0.75	0.75	0.78	0.79
*k*NNhte	Testing accuracy	0.738	0.750	0.758	0.779	0.771	0.792	0.812	0.795	0.790	0.794
	Training accuracy	0.538	0.554	0.688	0.799	0.791	0.819	0.815	0.833	0.850	0.847
	GF	0.567	0.561	0.776	1.100	1.096	1.149	1.016	1.228	1.400	1.346
	F1-Score	0.61	0.74	0.72	0.77	0.77	0.76	0.78	0.78	0.80	0.79
DThte	Testing accuracy	0.751	0.749	0.760	0.771	0.781	0.803	0.811	0.807	0.804	0.816
	Training accuracy	0.647	0.743	0.715	0.815	0.801	0.832	0.830	0.839	0.862	0.857
	GF	0.705	0.977	0.842	1.238	1.101	1.173	1.112	1.199	1.420	1.287
	F1-Score	0.61	0.75	0.68	0.75	0.76	0.78	0.77	0.77	0.81	0.79
SVMhte	Testing accuracy	0.741	0.750	0.773	0.787	0.787	0.801	0.819	0.817	0.813	0.815
	Training accuracy	0.633	0.735	0.714	0.813	0.798	0.831	0.830	0.845	0.862	0.866
	GF	0.706	0.943	0.794	1.139	1.054	1.178	1.065	1.181	1.355	1.381
	F1-Score	0.63	0.74	0.69	0.76	0.77	0.78	0.78	0.80	0.80	0.81
NNhte	Testing accuracy	0.741	0.750	0.770	0.782	0.781	0.794	0.809	0.811	0.809	0.808
	Training accuracy	0.640	0.733	0.709	0.815	0.802	0.832	0.829	0.845	0.858	0.864
	GF	0.719	0.936	0.790	1.178	1.106	1.226	1.117	1.219	1.345	1.412
	F1-Score	0.62	0.75	0.68	0.77	0.77	0.78	0.78	0.80	0.81	0.82
HTEsm	Testing accuracy	0.751	0.752	0.776	0.790	0.795	0.808	0.817	0.812	0.816	0.814
	Training accuracy	0.647	0.743	0.709	0.817	0.803	0.829	0.832	0.849	0.865	0.878
	GF	0.705	0.965	0.770	1.148	1.041	1.123	1.089	1.245	1.363	1.525
	F1-Score	0.60	0.75	0.73	0.76	0.77	0.78	0.78	0.80	0.81	0.82
HTEdf	Testing accuracy	0.751	0.752	0.777	0.791	0.784	0.821	0.816	0.812	0.813	0.822
	Training accuracy	0.647	0.738	0.699	0.801	0.772	0.811	0.822	0.846	0.852	0.866
	GF	0.705	0.947	0.741	1.050	0.947	0.947	1.034	1.221	1.264	1.328
	F1-Score	0.60	0.76	0.75	0.78	0.78	0.79	0.79	0.81	0.82	0.83

Appendix F

Ensemble Performance on Outlier Ratios for Regression Problems

The results of the ensembles over the different datasets in the number of outliers study for regression problems are provided in this appendix. The results consist of testing and training RMSE, and GF of the ensembles over the regression datasets.

Yacht Hydrodynamics Dataset

Table F.1: Ensemble Performance on the Number of Outliers for Yacht Hydrodynamics Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	9.01409	8.94113	9.02351	9.08400	9.00838
	Training RMSE	8.13776	7.96610	7.48927	7.79383	7.40027
	GF	1.22697	1.25978	1.45168	1.35848	1.48183
DTE	Testing RMSE	0.53723	0.52319	0.89180	0.70809	0.70481
	Training RMSE	0.04188	0.04170	0.04223	0.04210	0.04233
	GF	164.52740	157.37931	445.87201	282.85671	277.27188
RF	Testing RMSE	0.53925	0.54388	0.45945	0.49779	0.47186
	Training RMSE	0.46636	0.47561	0.41751	0.49743	0.51396
	GF	1.33700	1.30769	1.21097	1.00144	0.84287
SVRE	Testing RMSE	7.94635	8.01688	8.18180	8.15826	8.17077
	Training RMSE	11.02098	10.67599	10.39554	10.61917	10.30177
	GF	0.51987	0.56389	0.61945	0.59022	0.62908
NNE	Testing RMSE	3.51475	3.52582	3.77698	3.63915	3.86070
	Training RMSE	3.78378	3.69470	3.60550	3.69917	3.76867
	GF	0.86285	0.91067	1.09738	0.96781	1.04943

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	7.52081	7.58630	7.89268	7.63206	7.82731
	Training RMSE	6.80552	6.66952	6.33908	6.53327	6.25547
	GF	1.22126	1.29381	1.55023	1.36465	1.56569
DThte	Testing RMSE	0.63106	0.57262	0.69637	0.70525	0.56433
	Training RMSE	0.42913	0.43574	0.65471	0.45238	0.44788
	GF	2.16253	1.72690	1.13131	2.43041	1.58759
SVRhte	Testing RMSE	6.91455	7.02424	7.21253	7.21008	7.21562
	Training RMSE	9.23611	8.95115	8.75508	8.97002	8.71709
	GF	0.56047	0.61580	0.67866	0.64609	0.68518
NNhte	Testing RMSE	2.95474	2.90999	3.16792	3.1206	3.00638
	Training RMSE	3.03224	2.96496	3.03656	3.0699	2.99082
	GF	0.94954	0.96327	1.08839	1.03330	1.01043
HTEsm	Testing RMSE	0.91143	0.82604	0.98462	1.08479	0.80258
	Training RMSE	0.84678	0.78759	0.85949	0.90555	0.73603
	GF	1.15853	1.10002	1.31235	1.43505	1.18901
HTEdf	Testing RMSE	0.81582	0.77358	0.79181	0.91544	0.85483
	Training RMSE	0.87596	0.73911	0.79346	0.79336	0.71066
	GF	0.86740	1.09543	0.99586	1.33145	1.44687

Residential Building Dataset

Table F.2: Ensemble Performance on the Number of Outliers for Residential Building Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	805.90286	806.32629	807.51937	810.40833	816.67088
	Training RMSE	638.13987	637.54068	630.95983	618.33322	621.54650
	GF	1.59490	1.59958	1.63796	1.71776	1.72642
DTE	Testing RMSE	435.69867	327.53881	294.22788	455.19911	453.03353
	Training RMSE	7.93289	3.40179	3.42496	3.44828	3.31527
	GF	3016.54570	9270.66579	7380.02143	17425.99008	18673.37537
RF	Testing RMSE	265.69165	269.49343	236.92167	390.42718	263.52308
	Training RMSE	85.73902	113.95746	94.20497	102.10363	95.70077
	GF	9.60282	5.59256	6.32502	14.62170	7.58240
SVRE	Testing RMSE	751.61102	750.12141	749.00883	740.42989	739.48209
	Training RMSE	607.19904	599.65068	597.53082	578.71110	580.89459
	GF	1.53223	1.56483	1.57128	1.63698	1.62054
NNE	Testing RMSE	375.73858	381.98441	368.64363	369.27518	369.95761
	Training RMSE	268.74985	262.12374	269.56925	264.67078	259.43792
	GF	1.95468	2.12363	1.87013	1.94665	2.03347

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	754.09193	752.58073	752.01289	751.54431	760.77719
	Training RMSE	539.82891	531.62523	528.72066	518.11217	519.35478
	GF	1.95136	2.00399	2.02301	2.10408	2.14579
DThte	Testing RMSE	342.59631	259.06064	251.74033	265.87418	311.68387
	Training RMSE	60.31612	56.48676	49.26577	53.98279	46.63759
	GF	32.26254	21.03341	26.11049	24.25725	44.66388
SVRhte	Testing RMSE	743.32101	740.35744	740.36876	736.24081	736.01579
	Training RMSE	601.55238	594.38418	591.53018	574.19508	576.32307
	GF	1.52688	1.55149	1.56654	1.64407	1.63096
NNhte	Testing RMSE	331.43936	360.33552	344.22434	340.83468	351.60316
	Training RMSE	230.35095	225.83486	224.09040	225.09383	231.23452
	GF	2.07028	2.54585	2.35959	2.29277	2.31207
HTEsm	Testing RMSE	227.58750	192.42969	165.65231	175.72638	162.21155
	Training RMSE	142.75832	117.42994	85.39754	96.58745	85.83037
	GF	2.54152	2.68526	3.76274	3.31003	3.57175
HTEdf	Testing RMSE	177.44122	183.89753	171.85951	174.21503	177.49872
	Training RMSE	97.46849	106.77283	99.78769	93.73375	99.12949
	GF	3.31421	2.96640	2.96615	3.45445	3.20616

Student Performance Dataset

Table F.3: Ensemble Performance on the Number of Outliers for Student Performance Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	3.73956	3.73956	3.75274	3.73482	3.74139
	Training RMSE	3.24642	3.24451	3.25070	3.24366	3.28737
	GF	1.32688	1.32844	1.33273	1.32577	1.29529
DTE	Testing RMSE	2.40236	2.42569	2.01999	2.34206	2.46027
	Training RMSE	0.03774	0.04894	0.04185	0.03868	0.04276
	GF	4051.47797	2456.95935	2330.21867	3666.24172	3310.29639
RF	Testing RMSE	1.99705	1.89703	1.89923	2.10075	1.89773
	Training RMSE	0.60216	0.62442	0.64714	0.68035	0.55435
	GF	10.99909	9.22982	8.61310	9.53432	11.71947
SVRE	Testing RMSE	3.15151	3.16892	3.15195	3.16100	3.17373
	Training RMSE	1.91669	1.92447	1.93118	1.94011	1.93267
	GF	2.70355	2.71145	2.66386	2.65459	2.69663
NNE	Testing RMSE	2.70033	2.66994	2.58334	2.65461	2.58026
	Training RMSE	0.48970	0.47300	0.49297	0.54970	0.60721
	GF	30.40708	31.86201	27.46148	23.32140	18.05730

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	3.72344	3.72291	3.73111	3.73258	3.72032
	Training RMSE	2.78158	2.78226	2.79576	2.78340	2.83432
	GF	1.79186	1.79049	1.78104	1.79832	1.72291
DThte	Testing RMSE	2.13591	1.87891	1.90825	2.00497	1.86465
	Training RMSE	0.49990	0.54955	0.47787	0.52404	0.58103
	GF	18.25594	11.68936	15.94604	14.63797	10.29903
SVRhte	Testing RMSE	3.10108	3.11413	3.09857	3.10506	3.11165
	Training RMSE	2.07905	2.08653	2.09253	2.10222	2.09963
	GF	2.22482	2.22753	2.19269	2.18164	2.19632
NNhte	Testing RMSE	2.64135	2.64484	2.52247	2.51962	2.56548
	Training RMSE	0.53671	0.46791	0.46707	0.51647	0.45509
	GF	24.21976	31.94977	29.16674	23.80019	31.77881
HTEsm	Testing RMSE	2.43722	2.36529	2.45291	2.36980	2.39773
	Training RMSE	1.15005	1.13048	1.19046	1.15851	1.17406
	GF	4.49111	4.37764	4.24554	4.18429	4.17085
HTEdf	Testing RMSE	2.41203	2.36750	2.30534	2.33523	2.26899
	Training RMSE	1.17370	1.15162	1.09300	1.07567	1.09714
	GF	4.22331	4.22631	4.44869	4.71309	4.27705

Real Estate Dataset

Table F.4: Ensemble Performance on the Number of Outliers for Real Estate Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	8.70595	8.69097	8.70534	8.66053	8.71904
	Training RMSE	7.47411	7.47926	7.56809	7.56377	7.59067
	GF	1.35679	1.35027	1.32312	1.31103	1.31940
DTE	Testing RMSE	7.23313	7.27803	7.18501	6.92563	6.95505
	Training RMSE	3.76827	3.78774	3.80797	3.82718	3.82850
	GF	3.68441	3.69205	3.56013	3.27461	3.30022
RF	Testing RMSE	7.19300	7.55870	7.04157	7.22355	7.03475
	Training RMSE	4.28306	4.42575	4.44096	4.73722	4.02903
	GF	2.82040	2.91689	2.51412	2.32517	3.04858
SVRE	Testing RMSE	10.28262	10.34701	10.32291	10.37663	10.44023
	Training RMSE	11.12286	11.17166	11.17949	11.20637	11.18800
	GF	0.85462	0.85782	0.85263	0.85740	0.87079
NNE	Testing RMSE	8.34432	8.37894	8.28893	8.33161	8.31620
	Training RMSE	8.28841	8.29150	8.36664	8.42288	8.47788
	GF	1.01354	1.02120	0.98151	0.97844	0.96222

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	7.88826	7.85996	7.90083	7.89927	7.89405
	Training RMSE	6.24838	6.27150	6.34438	6.35626	6.35526
	GF	1.59377	1.57072	1.55084	1.54444	1.54288
DThte	Testing RMSE	7.11066	6.94781	6.97323	6.55932	6.77621
	Training RMSE	3.60265	3.63771	3.60417	3.65857	3.76843
	GF	3.89560	3.64787	3.74332	3.21437	3.23335
SVRhte	Testing RMSE	9.18253	9.20184	9.20432	9.21503	9.23549
	Training RMSE	9.95944	10.02045	10.01592	10.07056	10.11024
	GF	0.85007	0.84329	0.84450	0.83731	0.83444
NNhte	Testing RMSE	8.29028	8.35612	8.27151	8.38217	8.28637
	Training RMSE	8.34197	8.34313	8.40912	8.53397	8.50924
	GF	0.98765	1.00312	0.96754	0.96474	0.94830
HTEsm	Testing RMSE	7.43576	7.45590	7.38289	7.34382	7.38525
	Training RMSE	6.08535	6.02306	6.11492	6.18779	6.09705
	GF	1.49307	1.53238	1.45771	1.40856	1.46720
HTEdf	Testing RMSE	7.19234	7.19060	7.17154	7.15525	7.14242
	Training RMSE	6.64597	6.58466	6.71230	6.59332	6.66884
	GF	1.17118	1.19252	1.14152	1.17772	1.14707

Energy Efficiency Dataset

Table F.5: Ensemble Performance on the Number of Outliers for Energy Efficiency Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	1.40676	1.44079	1.39951	1.40741	1.47270
	Training RMSE	1.39885	1.30149	1.20307	1.26811	1.20731
	GF	1.01133	1.22552	1.35323	1.23177	1.48797
DTE	Testing RMSE	1.57038	1.60316	1.58128	1.58497	1.76119
	Training RMSE	0.44623	0.44843	0.45072	0.45300	0.44367
	GF	12.38464	12.78111	12.30841	12.24160	15.75756
RF	Testing RMSE	1.31824	1.29803	1.20718	1.23768	1.49933
	Training RMSE	0.60821	0.58800	0.56906	0.55717	0.57234
	GF	4.69767	4.87316	4.50013	4.93448	6.86258
SVRE	Testing RMSE	3.29608	3.28338	3.29188	3.29205	3.31875
	Training RMSE	3.09388	3.09047	3.10536	3.10974	3.08429
	GF	1.13498	1.12873	1.12374	1.12069	1.15781
NNE	Testing RMSE	2.03066	2.53009	2.03453	2.03599	1.99806
	Training RMSE	1.95067	2.49323	1.92684	1.88497	1.86062
	GF	1.08369	1.02978	1.11490	1.16666	1.15319

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	1.37807	1.36874	1.30634	1.32461	1.36863
	Training RMSE	1.32270	1.27510	1.18748	1.18709	1.10389
	GF	1.08547	1.15228	1.21021	1.24511	1.53716
DThte	Testing RMSE	1.36088	1.25528	1.26292	1.25958	1.38500
	Training RMSE	0.60019	0.60892	0.60755	0.57080	0.59728
	GF	5.14111	4.24975	4.32106	4.86948	5.37699
SVRhte	Testing RMSE	2.33852	2.31122	2.32583	2.31314	2.35189
	Training RMSE	2.17039	2.15771	2.17657	2.16161	2.15105
	GF	1.16093	1.14736	1.14186	1.14511	1.19546
NNhte	Testing RMSE	2.06712	2.72439	2.01293	2.05658	2.02768
	Training RMSE	2.01857	2.62709	1.90672	1.90142	1.86707
	GF	1.04867	1.07545	1.11450	1.16987	1.17945
HTEsm	Testing RMSE	1.17525	1.21673	1.20842	1.20852	1.27638
	Training RMSE	0.94851	0.94573	0.92769	0.90342	0.90569
	GF	1.53527	1.65522	1.69680	1.78947	1.98610
HTEdf	Testing RMSE	1.16163	1.19913	1.16201	1.20734	1.23452
	Training RMSE	1.07520	1.07797	1.06808	1.05073	1.02710
	GF	1.16723	1.23742	1.18362	1.32030	1.44469

Concrete Dataset

Table F.6: Ensemble Performance on the Number of Outliers for Concrete Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	9.81052	9.79300	9.77888	9.78407	9.88952
	Training RMSE	7.38628	7.38719	7.39063	7.35681	7.35962
	GF	1.76414	1.75741	1.75072	1.76872	1.80568
DTE	Testing RMSE	7.21109	6.81774	6.62457	7.53185	7.67483
	Training RMSE	1.74918	2.51357	1.91055	1.86792	2.54515
	GF	16.99552	7.35697	12.02257	16.25867	9.09304
RF	Testing RMSE	5.77574	6.19244	5.92399	5.79660	5.68757
	Training RMSE	2.41377	2.57589	2.52416	2.56357	2.43310
	GF	5.72565	5.77920	5.50800	5.11277	5.46430
SVRE	Testing RMSE	12.25705	12.23746	12.25949	12.25724	12.20848
	Training RMSE	12.06887	12.10954	12.12635	12.12900	12.07327
	GF	1.03143	1.02124	1.02208	1.02126	1.02252
NNE	Testing RMSE	6.98294	7.34882	7.55187	7.06612	6.87920
	Training RMSE	6.69152	6.90367	7.04948	6.50186	6.39132
	GF	1.08900	1.13312	1.14761	1.18110	1.15850

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	9.12790	9.14093	9.13030	9.11538	9.13694
	Training RMSE	6.23007	6.23296	6.24332	6.27789	6.26682
	GF	2.14662	2.15076	2.13865	2.10825	2.12572
DThte	Testing RMSE	6.12696	6.12195	6.03872	6.30868	6.44126
	Training RMSE	2.79240	2.70002	2.64650	2.65793	2.78218
	GF	4.81431	5.14098	5.20650	5.63364	5.36006
SVRhte	Testing RMSE	10.32625	10.32674	10.34020	10.34315	10.29401
	Training RMSE	9.74796	9.78046	9.82311	9.83315	9.78178
	GF	1.12217	1.11483	1.10805	1.10642	1.10747
NNhte	Testing RMSE	6.78315	6.58279	6.51311	7.25605	7.50726
	Training RMSE	6.46397	6.08488	6.10702	6.69278	6.87000
	GF	1.10120	1.17035	1.13741	1.17541	1.19412
HTEsm	Testing RMSE	7.12790	6.7126	6.74383	7.15850	6.75569
	Training RMSE	6.24023	5.8044	5.78678	6.19481	5.68687
	GF	1.30474	1.33742	1.35812	1.33533	1.41122
HTEdf	Testing RMSE	6.39390	6.49002	6.67672	6.65165	6.58799
	Training RMSE	5.22514	5.32801	5.61804	5.66509	5.59447
	GF	1.49739	1.48376	1.41240	1.37862	1.38672

Parkinsons Disease Dataset

Table F.7: Ensemble Performance on the Number of Outliers for Parkinsons Disease Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	0.94547	0.92335	0.95378	0.97415	0.95063
	Training RMSE	0.76136	0.75225	0.72269	0.71362	0.71465
	GF	1.54212	1.50665	1.74175	1.86343	1.76942
DTE	Testing RMSE	0.91169	0.90699	0.88110	0.86819	0.90359
	Training RMSE	0.88832	0.90397	0.85182	0.83796	0.88034
	GF	1.05330	1.00668	1.06992	1.07346	1.05352
RF	Testing RMSE	0.86407	0.95199	0.90170	0.79651	0.92038
	Training RMSE	0.81440	0.91896	0.81968	0.70222	0.80977
	GF	1.12569	1.07318	1.21014	1.28660	1.29186
SVRE	Testing RMSE	3.35919	3.39282	3.33978	3.37904	3.36691
	Training RMSE	3.45897	3.46840	3.47537	3.48200	3.49458
	GF	0.94314	0.95690	0.92349	0.94174	0.92827
NNE	Testing RMSE	1.77933	1.96899	1.77636	2.04187	2.57322
	Training RMSE	1.84572	1.92000	1.64127	1.87935	2.37868
	GF	0.92936	1.05168	1.17139	1.18043	1.17025

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	0.83513	0.82095	0.83406	0.86599	0.86758
	Training RMSE	0.59607	0.59001	0.57221	0.56307	0.56355
	GF	1.96295	1.93605	2.12467	2.36540	2.36998
DThte	Testing RMSE	0.90214	0.81379	0.80120	0.87875	0.92884
	Training RMSE	0.81850	0.80439	0.77407	0.82736	0.85830
	GF	1.21480	1.02352	1.07134	1.12809	1.17113
SVRhte	Testing RMSE	2.50002	2.5230	2.50701	2.50319	2.51821
	Training RMSE	2.54607	2.5534	2.55593	2.55696	2.56528
	GF	0.96415	0.97633	0.96209	0.95839	0.96364
NNhte	Testing RMSE	1.24966	1.81104	1.60300	1.67702	1.52858
	Training RMSE	1.30035	1.73540	1.54437	1.62536	1.33989
	GF	0.92356	1.08907	1.07736	1.06458	1.30147
HTEsm	Testing RMSE	0.55102	0.53724	0.54275	0.52958	0.58950
	Training RMSE	0.44315	0.41724	0.41285	0.40939	0.40984
	GF	1.54608	1.65794	1.72830	1.67334	2.06891
HTEdf	Testing RMSE	0.35981	0.38045	0.38687	0.37073	0.48160
	Training RMSE	0.36697	0.36587	0.34896	0.33115	0.40705
	GF	0.96136	1.08129	1.22908	1.25333	1.39984

Air Quality Dataset

Table F.8: Ensemble Performance on the Number of Outliers for Air Quality Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	18.06296	18.16319	18.38286	18.58582	18.88897
	Training RMSE	14.03762	14.04792	13.93309	13.85070	13.77814
	GF	1.65573	1.67171	1.74073	1.80061	1.87947
DTE	Testing RMSE	18.57556	19.01197	19.05183	19.33378	19.40403
	Training RMSE	11.58164	11.28941	11.53460	11.68317	11.64203
	GF	2.57243	2.83604	2.72815	2.73850	2.77797
RF	Testing RMSE	15.89269	15.88560	15.53334	15.7780	16.60442
	Training RMSE	6.55339	6.74674	6.79115	6.7495	6.60434
	GF	5.88116	5.54395	5.23170	5.46463	6.32105
SVRE	Testing RMSE	27.69508	27.70781	27.79172	27.80594	27.80893
	Training RMSE	27.63729	27.49956	27.39049	27.26740	27.14806
	GF	1.00419	1.01520	1.02951	1.03989	1.04928
NNE	Testing RMSE	17.27135	17.29018	17.70050	17.40294	17.71513
	Training RMSE	16.99002	17.05974	17.44328	16.92508	17.02025
	GF	1.03339	1.02720	1.02971	1.05726	1.08332

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	17.25510	17.34611	17.56419	17.78504	18.18790
	Training RMSE	11.95827	11.96683	11.86080	17.78504	11.73521
	GF	2.08208	2.10110	2.19295	1.00000	2.40206
DThte	Testing RMSE	16.60439	16.36922	16.78832	17.24003	17.30391
	Training RMSE	9.50404	9.51389	9.29158	9.87528	9.44876
	GF	3.05231	2.96033	3.26464	3.04773	3.35381
SVRhte	Testing RMSE	22.30532	22.35318	22.48968	22.53309	22.57695
	Training RMSE	22.13307	22.07445	21.95054	21.89441	21.81579
	GF	1.01563	1.02541	1.04973	1.05919	1.07100
NNhte	Testing RMSE	17.09980	16.77958	16.88041	17.37077	17.29359
	Training RMSE	16.87624	16.61661	16.52482	16.89678	16.73836
	GF	1.02667	1.01971	1.04350	1.05689	1.06744
HTEsm	Testing RMSE	15.43440	15.45223	15.49168	15.74489	15.89219
	Training RMSE	13.11023	12.96054	11.07643	11.02955	10.98338
	GF	1.38599	1.42147	1.95613	2.03781	2.09361
HTEdf	Testing RMSE	15.19142	15.40132	15.32276	15.36462	15.8800
	Training RMSE	13.72678	13.70844	12.41689	12.53212	11.3971
	GF	1.22478	1.26223	1.52282	1.50312	1.94139

Bike Sharing Dataset

Table F.9: Ensemble Performance on the Number of Outliers for Bike Sharing Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	42.29985	43.26563	43.60170	44.75718	49.12784
	Training RMSE	34.77901	34.88726	34.62255	34.79576	34.80163
	GF	1.47926	1.53799	1.58595	1.65452	1.99277
DTE	Testing RMSE	21.39002	21.47770	21.49377	21.30687	21.90507
	Training RMSE	15.26968	15.27003	15.36396	15.14400	14.97995
	GF	1.96229	1.97832	1.95713	1.97951	2.13830
RF	Testing RMSE	17.96188	17.99491	17.91192	18.15218	18.13043
	Training RMSE	7.95093	7.86037	7.60218	7.84663	7.94671
	GF	5.10349	5.24099	5.55147	5.35170	5.20526
SVRE	Testing RMSE	85.82061	85.92702	85.87803	85.97992	86.00327
	Training RMSE	89.71965	89.95967	89.90234	89.51088	89.85034
	GF	0.91497	0.91235	0.91248	0.92266	0.91620
NNE	Testing RMSE	17.81502	18.23934	17.54441	19.47675	18.66296
	Training RMSE	17.65071	17.97331	17.25440	19.08325	17.78623
	GF	1.01871	1.02982	1.03390	1.04167	1.10102

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	38.63689	39.52328	39.94430	40.55688	44.30946
	Training RMSE	28.60214	28.67230	28.44793	28.58426	28.53306
	GF	1.82477	1.90012	1.97155	2.01315	2.41155
DThte	Testing RMSE	20.75278	22.35038	21.40821	21.58477	21.17484
	Training RMSE	14.17345	17.08053	15.22182	15.34748	15.05285
	GF	2.14388	1.71225	1.97801	1.97798	1.97880
SVRhte	Testing RMSE	49.92330	50.18151	50.40702	50.54404	50.77489
	Training RMSE	52.21667	52.27553	52.33697	52.07664	52.21746
	GF	0.91409	0.92149	0.92761	0.94201	0.94551
NNhte	Testing RMSE	17.11003	17.47013	16.72466	16.20576	17.33352
	Training RMSE	16.81573	17.13598	16.23076	15.88041	16.45601
	GF	1.03531	1.03938	1.06178	1.04140	1.10949
HTEsm	Testing RMSE	17.03517	17.52080	17.54401	16.86632	17.96501
	Training RMSE	14.43215	12.77113	12.86548	12.19843	12.89642
	GF	1.39326	1.88213	1.85954	1.91176	1.94051
HTEdf	Testing RMSE	16.53471	17.08001	17.24946	16.58802	16.58749
	Training RMSE	14.55931	14.93604	13.81964	13.40543	13.00954
	GF	1.28977	1.30769	1.55796	1.53118	1.62569

Gas Turbine Dataset

Table F.10: Ensemble Performance on the Number of Outliers for Gas Turbine Dataset

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNE	Testing RMSE	4.43113	4.51663	4.55696	4.71670	4.91183
	Training RMSE	3.42834	3.40296	3.37829	3.32598	3.29765
	GF	1.67056	1.76163	1.81951	2.01111	2.21859
DTE	Testing RMSE	5.82068	5.87249	5.85959	5.91640	6.03200
	Training RMSE	4.37408	4.30660	4.30374	4.15535	4.11893
	GF	1.77081	1.85941	1.85372	2.02722	2.14464
RF	Testing RMSE	4.19360	4.32043	4.37045	4.38629	4.60338
	Training RMSE	1.79348	1.75618	1.76563	1.70153	1.68500
	GF	5.46741	6.05227	6.12707	6.64532	7.46372
SVRE	Testing RMSE	6.65383	6.78696	6.81647	7.01922	7.20270
	Training RMSE	6.58753	6.51383	6.49126	6.37189	6.27259
	GF	1.02023	1.08562	1.10271	1.21350	1.31855
NNE	Testing RMSE	4.97674	4.65347	4.9777	5.04942	5.21828
	Training RMSE	4.75880	4.43193	4.6895	4.61052	4.82738
	GF	1.09369	1.10247	1.12669	1.19945	1.16850

Ensemble	Measure	Number of Outliers %				
		1	2	3	4	5
*k*NNhte	Testing RMSE	4.13956	4.23420	4.27400	4.46163	4.71653
	Training RMSE	2.87695	2.85627	2.83661	2.79370	2.77201
	GF	2.07034	2.19759	2.27024	2.55052	2.89506
DThte	Testing RMSE	4.91382	5.02787	5.01162	5.16773	5.27262
	Training RMSE	3.57797	3.58005	3.50004	3.43606	3.38733
	GF	1.88611	1.97238	2.05026	2.26192	2.42291
SVRhte	Testing RMSE	5.98892	6.02567	6.03791	6.13684	6.28236
	Training RMSE	5.91006	5.87306	5.85670	5.79005	5.74763
	GF	1.02686	1.05265	1.06284	1.12338	1.19472
NNhte	Testing RMSE	4.15273	4.64765	4.61532	4.44180	4.59362
	Training RMSE	3.97677	4.37623	4.28399	4.03603	4.16658
	GF	1.09045	1.12789	1.16066	1.21118	1.21548
HTEsm	Testing RMSE	4.30211	4.29797	4.24537	4.37059	4.60041
	Training RMSE	3.51543	3.43704	3.36862	3.30957	3.53303
	GF	1.49764	1.56371	1.58828	1.74396	1.69550
HTEdf	Testing RMSE	4.07862	4.29184	4.37267	4.29309	4.55657
	Training RMSE	3.57127	3.75352	3.66512	3.44796	3.52297
	GF	1.30431	1.30740	1.42337	1.55030	1.67285

Appendix G

Ensemble Performance on Outlier Severities for Regression Problems

The results of the ensembles over the different datasets in the severity of outliers study for regression problems are provided in this appendix. The results consist of testing and training RMSE, and GF of the ensembles over the regression datasets.

Yacht Hydrodynamics Dataset

Table G.1: Ensemble Performance on the Severity of Outliers for Yacht Hydrodynamics Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	8.38335	7.84940	7.14481	7.05095	7.05095
	Training RMSE	5.67703	6.18965	7.71690	8.09175	8.31455
	GF	2.18069	1.60821	0.85723	0.75929	0.71915
DTE	Testing RMSE	3.98348	1.67341	1.02247	0.99244	0.98808
	Training RMSE	0.01356	0.01690	0.01854	0.01849	0.01379
	GF	86293.65543	9805.34604	3041.33470	2881.11440	5135.96312
RF	Testing RMSE	3.92535	2.13064	0.95973	0.81977	0.79043
	Training RMSE	0.29006	0.43621	0.35036	0.34152	0.57259
	GF	183.13367	23.85772	7.50367	5.76174	1.90566
SVRE	Testing RMSE	9.70830	9.40801	8.78232	8.62483	8.56555
	Training RMSE	7.84219	8.72896	10.65002	11.30465	11.87340
	GF	1.53254	1.16164	0.68001	0.58209	0.52043
NNE	Testing RMSE	5.67079	7.18654	9.18971	9.72014	10.60796
	Training RMSE	0.40075	0.47374	0.61477	0.61652	0.58280
	GF	200.23395	230.12024	223.44635	248.57376	331.29997

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	7.70380	7.16195	6.40384	6.15962	6.13223
	Training RMSE	4.94529	5.36059	6.38951	6.84024	7.00617
	GF	2.42675	1.78499	1.00449	0.81090	0.76608
DThte	Testing RMSE	3.99946	1.92181	0.92332	0.81499	0.81660
	Training RMSE	0.22339	0.29866	0.41701	0.38287	0.42312
	GF	320.52758	41.40602	4.90241	4.53116	3.72479
SVRhte	Testing RMSE	8.88755	8.44558	7.69488	7.57385	7.49372
	Training RMSE	6.86543	7.54310	9.18410	9.84572	10.36701
	GF	1.67582	1.25360	0.70199	0.59175	0.52250
NNhte	Testing RMSE	7.69662	7.01689	9.92877	9.91142	9.30048
	Training RMSE	0.47164	0.48763	0.58140	0.64573	0.57099
	GF	266.30375	207.06733	291.63868	235.59981	265.31401
HTEsm	Testing RMSE	4.51439	3.64901	3.06519	2.92606	2.79791
	Training RMSE	2.69123	3.04765	3.65660	3.93737	4.05557
	GF	2.81381	1.43357	0.70269	0.55227	0.47595
HTEdf	Testing RMSE	4.28117	3.35408	2.67178	2.41778	2.52151
	Training RMSE	2.31203	2.54413	2.95658	3.15297	3.29687
	GF	3.42876	1.73807	0.81662	0.58802	0.58495

Residential Building Dataset

Table G.2: Ensemble Performance on the Severity of Outliers for Residential Building Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	1077.68981	1041.26160	1105.31825	1065.11408	1027.94032
	Training RMSE	456.28497	457.85952	447.34324	502.40804	506.52271
	GF	5.57847	5.17196	6.10510	4.49448	4.11849
DTE	Testing RMSE	487.05435	463.64101	465.73366	415.69096	311.99008
	Training RMSE	10.46792	10.66343	9.88189	8.64423	7.86679
	GF	2164.88060	1890.46999	2221.23947	2312.53656	1572.84502
RF	Testing RMSE	675.72884	509.90993	506.22014	371.30316	333.61223
	Training RMSE	63.99096	45.13511	49.95396	62.31295	66.40235
	GF	111.50842	127.63150	102.69258	35.50592	25.24161
SVRE	Testing RMSE	723.51064	730.33168	731.82325	741.75559	728.38187
	Training RMSE	325.12630	342.84739	341.94765	417.12265	439.90829
	GF	4.95206	4.53773	4.58029	3.16223	2.74154
NNE	Testing RMSE	1178.30820	1162.47113	1177.43282	526.55005	236.57188
	Training RMSE	544.63323	573.88356	562.90999	335.27026	104.20862
	GF	4.68069	4.10314	4.37516	2.46655	5.15370

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	999.99768	978.06577	1060.15576	975.25422	961.90377
	Training RMSE	367.04236	368.48139	360.92965	408.12965	416.55411
	GF	7.42276	7.04539	8.62768	5.71004	5.33237
DThte	Testing RMSE	529.40551	501.65640	492.94317	393.26589	342.17092
	Training RMSE	23.68328	26.67429	30.41415	43.73979	46.51522
	GF	499.68132	353.69331	262.68933	80.83868	54.11231
SVRhte	Testing RMSE	717.34945	721.01191	716.54598	733.72611	745.12117
	Training RMSE	345.24250	360.29950	357.94708	431.19074	449.78295
	GF	4.31731	4.00459	4.00729	2.89554	2.74440
NNhte	Testing RMSE	701.13442	802.60477	1048.05290	474.00173	226.57596
	Training RMSE	336.98313	405.30802	505.12819	317.75254	101.01194
	GF	4.32899	3.92133	4.30490	2.22527	5.03132
HTEsm	Testing RMSE	663.24893	514.02336	802.26091	558.36946	483.15774
	Training RMSE	232.54526	178.32836	291.28893	251.93833	207.33317
	GF	8.13464	8.30854	7.58548	4.91196	5.43051
HTEdf	Testing RMSE	678.44173	618.32571	743.22824	512.72211	448.98755
	Training RMSE	196.72532	201.64930	249.30900	214.21396	174.17227
	GF	11.89336	9.40245	8.88727	5.72886	6.64524

Student Performance Dataset

Table G.3: Ensemble Performance on the Severity of Outliers for Student Performance Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	3.96965	3.76755	3.76496	3.76496	3.76496
	Training RMSE	2.36895	2.68449	2.63408	2.63408	2.63408
	GF	2.80796	1.96967	2.04298	2.04298	2.04298
DTE	Testing RMSE	2.68121	2.83917	2.71715	2.78672	2.89491
	Training RMSE	0.03792	0.02679	0.02521	0.01371	0.01285
	GF	4998.88484	11232.31494	11615.70952	41324.39606	50746.83058
RF	Testing RMSE	2.28371	2.65137	2.31164	2.43693	2.36507
	Training RMSE	0.87162	1.03532	0.88665	0.99756	0.98554
	GF	6.86477	6.55826	6.79723	5.96768	5.75893
SVRE	Testing RMSE	3.35787	3.22840	3.26745	3.26745	3.26745
	Training RMSE	1.67272	1.90643	1.87145	1.87145	1.87145
	GF	4.02979	2.86770	3.04831	3.04831	3.04831
NNE	Testing RMSE	2.56025	2.63498	2.47121	2.49446	2.53195
	Training RMSE	0.25456	0.29213	0.24894	0.28823	0.27862
	GF	101.15730	81.35680	98.54441	74.89640	82.58016

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	3.89027	3.67737	3.69468	3.69468	3.69468
	Training RMSE	2.74013	3.05756	3.01075	3.01075	3.01075
	GF	2.01565	1.44652	1.50593	1.50593	1.50593
DThte	Testing RMSE	2.34553	2.64840	2.54743	2.21758	2.15652
	Training RMSE	0.33279	0.51679	0.42735	0.43709	0.32851
	GF	49.67488	26.26268	35.53417	25.74116	43.09360
SVRhte	Testing RMSE	3.3197	3.20072	3.24523	3.24523	3.24523
	Training RMSE	1.4078	1.65904	1.62596	1.62596	1.62596
	GF	5.56049	3.72206	3.98355	3.98355	3.98355
NNhte	Testing RMSE	2.36791	2.44786	2.49339	2.42544	2.50207
	Training RMSE	0.22332	0.46907	0.27142	0.25003	0.47450
	GF	112.43018	27.23367	84.39420	94.10515	27.80553
HTEsm	Testing RMSE	2.41145	2.31420	2.24694	2.30702	2.36999
	Training RMSE	0.97796	1.07085	1.09856	1.07969	1.10331
	GF	6.08016	4.67028	4.18349	4.56567	4.61423
HTEdf	Testing RMSE	2.3887	2.30543	2.33345	2.19301	2.25575
	Training RMSE	1.0328	0.96428	1.02158	0.99521	1.00810
	GF	5.34922	5.71612	5.21743	4.85573	5.00692

Real Estate Dataset

Table G.4: Ensemble Performance on the Severity of Outliers for Real Estate Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	6.96672	6.70983	6.62052	6.62052	6.85951
	Training RMSE	6.15746	6.13021	6.07381	6.07381	7.29947
	GF	1.28013	1.19804	1.18813	1.18813	0.88309
DTE	Testing RMSE	6.79331	6.95888	6.97584	6.77552	7.44698
	Training RMSE	3.27099	3.31511	3.29997	3.29997	3.36793
	GF	4.31326	4.40640	4.46862	4.21566	4.88918
RF	Testing RMSE	6.15513	5.99390	6.25258	6.02732	6.13248
	Training RMSE	3.77895	3.88097	3.87046	3.95233	4.47813
	GF	2.65297	2.38527	2.60972	2.32564	1.87533
SVRE	Testing RMSE	10.50488	10.09245	9.88062	9.88062	9.82044
	Training RMSE	10.02802	10.12929	10.13345	10.13345	11.06295
	GF	1.09737	0.99274	0.95072	0.95072	0.78799
NNE	Testing RMSE	6.97959	7.05250	7.02721	7.02655	7.10364
	Training RMSE	7.70623	7.57983	7.67734	7.58555	8.51654
	GF	0.82030	0.86570	0.83781	0.85804	0.69572

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	6.59924	6.39150	6.36863	6.36863	6.39730
	Training RMSE	5.41720	5.41872	5.38442	5.38442	6.28196
	GF	1.48401	1.39127	1.39899	1.39899	1.03706
DThte	Testing RMSE	6.10397	6.17568	5.90464	6.17374	6.09071
	Training RMSE	3.36437	3.37607	3.39883	3.38219	3.52141
	GF	3.29168	3.34615	3.01806	3.33197	2.99159
SVRhte	Testing RMSE	8.61685	8.36591	8.22050	8.22050	8.17426
	Training RMSE	9.14632	9.08288	9.00604	9.00604	10.06677
	GF	0.88757	0.84836	0.83316	0.83316	0.65935
NNhte	Testing RMSE	6.93653	6.84324	6.90968	6.95990	7.13421
	Training RMSE	7.54028	7.32671	7.48496	7.49562	8.45136
	GF	0.84627	0.87238	0.85219	0.86217	0.71259
HTEsm	Testing RMSE	6.25160	6.34350	6.18269	6.19147	6.19033
	Training RMSE	5.33535	5.34082	5.24723	5.24629	6.04611
	GF	1.37296	1.41073	1.38834	1.39278	1.04827
HTEdf	Testing RMSE	6.58169	6.18454	5.95629	6.01075	5.9641
	Training RMSE	4.96500	4.95284	4.90619	4.89413	5.6972
	GF	1.75726	1.55922	1.47388	1.50836	1.09589

Energy Efficiency Dataset

Table G.5: Ensemble Performance on the Severity of Outliers for Energy Efficiency Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	1.42806	1.39174	1.40529	1.40019	1.41312
	Training RMSE	1.39864	1.10547	1.20641	1.20030	1.22116
	GF	1.04252	1.58498	1.35687	1.36081	1.33910
DTE	Testing RMSE	1.51933	1.54216	1.39794	1.55801	1.55800
	Training RMSE	0.45076	0.45076	0.45076	0.45076	0.45076
	GF	11.36116	11.70510	9.61822	11.94702	11.94688
RF	Testing RMSE	1.19617	1.23209	1.28467	1.19526	1.32136
	Training RMSE	0.61798	0.59986	0.60462	0.59856	0.59155
	GF	3.74661	4.21878	4.51462	3.98757	4.98944
SVRE	Testing RMSE	3.30104	3.30104	3.30104	3.30104	3.30104
	Training RMSE	3.09299	3.09299	3.09299	3.09299	3.09299
	GF	1.13905	1.13905	1.13905	1.13905	1.13905
NNE	Testing RMSE	2.07685	2.0244	2.03182	2.04253	2.01597
	Training RMSE	1.96971	1.9341	1.95816	1.93358	1.93122
	GF	1.11175	1.09556	1.07665	1.11586	1.08970

Ensemble	**Measure**	**Severity of Outliers (σ)**				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	1.37493	1.31353	1.33753	1.36113	1.36927
	Training RMSE	1.21239	1.23801	1.23524	1.21486	1.21091
	GF	1.28611	1.12572	1.17247	1.25531	1.27866
DThte	Testing RMSE	1.21289	1.25551	1.22171	1.28827	1.26312
	Training RMSE	0.59238	0.59489	0.59057	0.58150	0.59775
	GF	4.19220	4.45420	4.27951	4.90817	4.46523
SVRhte	Testing RMSE	2.33390	2.33390	2.33390	2.33390	2.33390
	Training RMSE	2.15913	2.15913	2.15913	2.15913	2.15913
	GF	1.16844	1.16844	1.16844	1.16844	1.16844
NNhte	Testing RMSE	2.02445	2.03460	2.06360	2.04901	2.01255
	Training RMSE	1.92884	1.94308	1.94725	1.95493	1.91298
	GF	1.10160	1.09643	1.12307	1.09856	1.10681
HTEsm	Testing RMSE	1.11056	1.11056	1.11056	1.11056	1.11056
	Training RMSE	0.94322	0.94322	0.94322	0.94322	0.94322
	GF	1.38629	1.38629	1.38629	1.38629	1.38629
HTEdf	Testing RMSE	1.09623	1.09623	1.09623	1.09623	1.09623
	Training RMSE	1.07436	1.07436	1.07436	1.07436	1.07436
	GF	1.04112	1.04112	1.04112	1.04112	1.04112

Concrete Dataset

Table G.6: Ensemble Performance on the Severity of Outliers for Concrete Dataset

Ensemble	**Measure**	**Severity of Outliers (σ)**				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	9.85978	9.83378	9.81614	9.81614	9.81614
	Training RMSE	7.53180	7.52012	7.43857	7.43857	7.43857
	GF	1.71371	1.70998	1.74142	1.74142	1.74142
DTE	Testing RMSE	7.07547	6.64179	7.31650	7.31996	7.33716
	Training RMSE	1.81659	1.92754	2.18956	2.18956	2.18956
	GF	15.17041	11.87304	11.16592	11.17650	11.22908
RF	Testing RMSE	5.85573	5.57962	5.87157	5.79671	6.04203
	Training RMSE	2.57057	2.60506	2.35973	2.44094	2.49860
	GF	5.18923	4.58749	6.19133	5.63963	5.84751
SVRE	Testing RMSE	12.16523	12.14793	12.13890	12.13890	12.13890
	Training RMSE	12.09689	12.06737	12.02103	12.02103	12.02103
	GF	1.01133	1.01340	1.01971	1.01971	1.01971
NNE	Testing RMSE	13.90033	14.75200	9.79447	8.36763	10.02562
	Training RMSE	7.24825	7.46071	7.48155	6.77458	7.81746
	GF	3.67776	3.90969	1.71387	1.52560	1.64472

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	9.23160	9.25315	9.17459	9.17459	9.17459
	Training RMSE	6.36845	6.36033	6.29068	6.29068	6.29068
	GF	2.10129	2.11651	2.12705	2.12705	2.12705
DThte	Testing RMSE	6.33105	6.37459	6.25706	6.32731	6.42195
	Training RMSE	2.73782	2.74238	2.67389	2.68821	2.76975
	GF	5.34739	5.40319	5.47587	5.54003	5.37592
SVRhte	Testing RMSE	10.59549	10.60058	10.40736	10.40736	10.40736
	Training RMSE	9.58043	9.54474	9.61985	9.61985	9.61985
	GF	1.22313	1.23348	1.17043	1.17043	1.17043
NNhte	Testing RMSE	14.03026	13.42044	7.92141	8.95515	6.60481
	Training RMSE	7.12852	7.08505	6.49364	7.00754	5.85723
	GF	3.87376	3.58796	1.48809	1.63310	1.27156
HTEsm	Testing RMSE	7.24585	6.74992	6.84406	6.77846	6.72493
	Training RMSE	4.92771	4.68290	4.77895	4.74755	4.67982
	GF	2.16216	2.07763	2.05099	2.03856	2.06498
HTEdf	Testing RMSE	6.90757	6.90291	6.21839	6.30245	6.35322
	Training RMSE	4.46656	4.48469	4.11708	4.22471	4.24868
	GF	2.39169	2.36919	2.28127	2.22549	2.23605

Parkinsons Disease Dataset

Table G.7: Ensemble Performance on the Severity of Outliers for Parkinsons Disease Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
kNNE	Testing RMSE	1.00610	0.99359	0.96849	0.95202	1.03465
	Training RMSE	0.68194	0.68721	0.70378	0.71309	0.72911
	GF	2.17667	2.09042	1.89371	1.78240	2.01373
DTE	Testing RMSE	1.08065	1.05848	0.95465	0.91256	0.88499
	Training RMSE	1.02483	1.05757	0.90525	0.86944	0.84451
	GF	1.11189	1.00172	1.11213	1.10166	1.09817
RF	Testing RMSE	0.48932	0.39347	0.40680	0.38329	0.37826
	Training RMSE	0.23235	0.19743	0.19745	0.17498	0.18164
	GF	4.43523	3.97197	4.24491	4.79796	4.33642
SVRE	Testing RMSE	3.37408	3.38413	3.35418	3.37125	3.33845
	Training RMSE	3.52907	3.50957	3.49888	3.49154	3.49230
	GF	0.91409	0.92979	0.91900	0.93229	0.91383
NNE	Testing RMSE	2.68447	1.97284	2.59921	1.96010	1.90395
	Training RMSE	2.08087	1.58957	2.36598	1.78221	1.80734
	GF	1.66428	1.54038	1.20687	1.20959	1.10976

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	0.88384	0.88083	0.88354	0.84517	0.96974
	Training RMSE	0.53263	0.53753	0.55365	0.56258	0.57057
	GF	2.75353	2.68516	2.54675	2.25692	2.88864
DThte	Testing RMSE	0.54759	0.54167	0.55499	0.49339	0.55431
	Training RMSE	0.43123	0.43606	0.42656	0.42701	0.42733
	GF	1.61244	1.54306	1.69286	1.33502	1.68255
SVRhte	Testing RMSE	2.54693	2.57444	2.52577	2.51627	2.54701
	Training RMSE	2.59335	2.57513	2.56518	2.56295	2.56531
	GF	0.96452	0.99947	0.96951	0.96391	0.98578
NNhte	Testing RMSE	1.83540	1.71204	2.21989	1.74936	1.26337
	Training RMSE	1.47136	1.49053	1.94989	1.45282	1.23412
	GF	1.55605	1.31930	1.29611	1.44989	1.04795
HTEsm	Testing RMSE	0.98003	0.90933	1.19712	0.85265	0.88083
	Training RMSE	0.86663	0.82525	1.10878	0.79039	0.77824
	GF	1.27881	1.21412	1.16570	1.16376	1.28104
HTEdf	Testing RMSE	0.86764	0.93673	0.8771	0.95016	0.93548
	Training RMSE	0.77579	0.83000	0.7775	0.91799	0.88158
	GF	1.25080	1.27373	1.27262	1.07132	1.12601

Air Quality Dataset

Table G.8: Ensemble Performance on the Severity of Outliers for Air Quality Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	19.96879	18.89618	18.81328	18.41109	18.49930
	Training RMSE	12.67559	13.16393	13.70278	13.98017	14.14685
	GF	2.48180	2.06052	1.88500	1.73434	1.70998
DTE	Testing RMSE	21.80001	19.62944	19.80078	19.40976	19.06127
	Training RMSE	10.14830	10.84320	11.19249	11.92670	12.14288
	GF	4.61452	3.27718	3.12976	2.64850	2.46411
RF	Testing RMSE	18.11814	17.41050	16.46223	16.66264	16.13618
	Training RMSE	6.02477	6.26907	6.46261	6.68528	6.82203
	GF	9.04371	7.71288	6.48876	6.21224	5.59467
SVRE	Testing RMSE	28.59417	28.46361	28.38742	28.54141	28.59044
	Training RMSE	25.60611	26.54332	27.37580	27.80374	28.10535
	GF	1.24700	1.14993	1.07527	1.05377	1.03482
NNE	Testing RMSE	19.71845	19.43568	19.12897	18.64904	18.81092
	Training RMSE	16.09674	16.27908	16.78254	16.88921	17.61510
	GF	1.50062	1.42541	1.29918	1.21925	1.14038

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	19.43022	18.37249	17.89726	17.61146	17.59477
	Training RMSE	10.81570	11.20130	11.64371	11.88487	12.06055
	GF	3.22735	2.69029	2.36260	2.19585	2.12830
DThte	Testing RMSE	19.07238	18.17248	17.36068	16.85797	16.75181
	Training RMSE	8.07801	8.33126	8.75398	9.16062	9.78064
	GF	5.57443	4.75780	3.93299	3.38657	2.93352
SVRhte	Testing RMSE	22.95817	22.86159	22.81804	22.95700	22.95467
	Training RMSE	20.59446	21.26015	21.94328	22.29737	22.54708
	GF	1.24272	1.15633	1.08132	1.06004	1.03648
NNhte	Testing RMSE	18.82194	18.90116	18.94923	18.54008	18.08490
	Training RMSE	15.36636	15.77933	16.74032	16.65988	16.71065
	GF	1.50033	1.43483	1.28131	1.23845	1.17124
HTEsm	Testing RMSE	17.25672	16.33996	16.24082	15.90768	15.93601
	Training RMSE	10.09008	10.38600	10.84621	11.21791	11.51360
	GF	2.92501	2.47518	2.24213	2.01090	1.91574
HTEdf	Testing RMSE	16.75567	16.02766	15.76371	15.61563	15.33018
	Training RMSE	8.56694	8.85901	9.10510	9.52238	9.56787
	GF	3.82536	3.27318	2.99742	2.68923	2.56723

Bike Sharing Dataset

Table G.9: Ensemble Performance on the Severity of Outliers for Bike Sharing Dataset

Ensemble	**Measure**	**Severity of Outliers (σ)**				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	52.48071	46.61149	42.98166	42.82314	42.82517
	Training RMSE	34.47091	34.78236	34.57036	34.49206	34.49870
	GF	2.31790	1.79584	1.54582	1.54141	1.54097
DTE	Testing RMSE	30.89302	22.66902	21.78735	20.90164	20.56664
	Training RMSE	13.57923	13.30457	14.87266	14.30802	14.10241
	GF	5.17571	2.90311	2.14601	2.13404	2.12687
RF	Testing RMSE	26.89443	19.12063	18.34443	18.19344	18.14037
	Training RMSE	7.94481	7.59783	7.73229	7.88670	7.67069
	GF	11.45930	6.33323	5.62849	5.32156	5.59272
SVRE	Testing RMSE	86.84046	86.01063	86.31709	86.32127	86.33219
	Training RMSE	82.26196	85.45117	88.93576	90.00621	90.04985
	GF	1.11441	1.01314	0.94198	0.91979	0.91914
NNE	Testing RMSE	20.60434	17.56158	17.35812	17.30144	16.97960
	Training RMSE	12.08835	11.87549	11.66342	12.36351	11.87361
	GF	2.90525	2.18688	2.21489	1.95831	2.04498

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	50.49662	43.89881	39.58632	39.37502	39.37534
	Training RMSE	28.43568	28.63009	28.39111	28.32212	28.32566
	GF	3.15353	2.35104	1.94413	1.93281	1.93236
DThte	Testing RMSE	28.74554	22.75687	22.01916	21.34918	21.30328
	Training RMSE	14.46734	14.24531	14.32528	14.89847	14.89844
	GF	3.94788	2.55201	2.36263	2.05343	2.04462
SVRhte	Testing RMSE	52.63703	51.10676	50.76307	50.65350	50.64981
	Training RMSE	50.87582	51.15627	51.71439	52.06904	52.06736
	GF	1.07043	0.99807	0.96355	0.94637	0.94629
NNhte	Testing RMSE	22.01513	17.52969	17.97177	16.89333	16.68307
	Training RMSE	12.23115	10.23633	10.77720	9.93368	10.00393
	GF	3.23972	2.93265	2.78080	2.89208	2.78106
HTEsm	Testing RMSE	18.93371	18.35210	18.62963	18.77078	19.29445
	Training RMSE	17.94761	17.49682	17.94064	18.01195	18.62132
	GF	1.11291	1.10015	1.07828	1.08603	1.07360
HTEdf	Testing RMSE	17.17417	18.05786	16.98915	16.89008	18.06899
	Training RMSE	16.05729	17.29386	16.10228	16.08467	17.28273
	GF	1.14395	1.09031	1.11319	1.10265	1.09306

Gas Turbine Dataset

Table G.10: Ensemble Performance on the Severity of Outliers for Gas Turbine Dataset

Ensemble	Measure	Severity of Outliers (σ)				
		2.0	2.5	3.0	3.5	4.0
*k*NNE	Testing RMSE	5.96655	5.25700	4.91637	4.70868	4.58307
	Training RMSE	3.12869	3.27337	3.30791	3.35919	3.37107
	GF	3.63682	2.57920	2.20893	1.96485	1.84833
DTE	Testing RMSE	7.24747	6.16323	5.93571	5.74889	5.63213
	Training RMSE	3.87462	4.13620	4.19764	4.17750	4.26442
	GF	3.49876	2.22031	1.99956	1.89381	1.74432
RF	Testing RMSE	5.66019	4.96622	4.59384	4.29757	4.20929
	Training RMSE	1.64303	1.68298	1.67765	1.72730	1.73198
	GF	11.86788	8.70756	7.49803	6.19027	5.90653
SVRE	Testing RMSE	7.71420	7.30281	7.08551	6.91786	6.79943
	Training RMSE	5.73676	6.06951	6.24643	6.40502	6.46419
	GF	1.80821	1.44768	1.28670	1.16655	1.10641
NNE	Testing RMSE	5.76858	5.74303	5.46981	5.30234	5.33118
	Training RMSE	4.39148	4.80010	4.44872	4.50632	4.76039
	GF	1.72551	1.43147	1.51173	1.38449	1.25418

Ensemble	**Measure**	**Severity of Outliers (σ)**				
		2.0	2.5	3.0	3.5	4.0
*k*NNhte	Testing RMSE	5.76542	5.03969	4.67621	4.42550	4.29948
	Training RMSE	2.63310	2.75530	2.77904	2.82337	2.83409
	GF	4.79431	3.34556	2.83137	2.45692	2.30147
DThte	Testing RMSE	6.15879	5.47397	5.3025	5.05457	4.98598
	Training RMSE	3.25708	3.36040	3.4726	3.50449	3.59413
	GF	3.57546	2.65353	2.33158	2.08027	1.92448
SVRhte	Testing RMSE	6.74663	6.37385	6.19918	6.09907	6.04138
	Training RMSE	5.29398	5.58465	5.71845	5.82206	5.84037
	GF	1.62409	1.30260	1.17520	1.09742	1.07002
NNhte	Testing RMSE	5.60585	5.08595	5.33598	5.25340	5.02782
	Training RMSE	4.22423	4.17701	4.28420	4.41452	4.16973
	GF	1.76112	1.48256	1.55127	1.41617	1.45393
HTEsm	Testing RMSE	5.13140	4.67598	4.43382	4.53470	4.27738
	Training RMSE	3.10205	3.30846	3.19409	3.55854	3.34138
	GF	2.73636	1.99754	1.92691	1.62388	1.63871
HTEdf	Testing RMSE	5.24853	4.55302	4.49845	4.27547	4.27679
	Training RMSE	2.82534	2.94569	3.04881	3.07904	3.10356
	GF	3.45090	2.38906	2.17704	1.92813	1.89896

Appendix H

Ensemble Performance on Bagged Subsets for Regression Problems

The results of the ensembles over the bagged subsets of the training dataset for regression problems are provided in this appendix. Plots of the testing and training RMSE for each regression dataset are first presented. Then the results of testing RMSE, training RMSE, and GF of the ensembles over the regression datasets are provided.

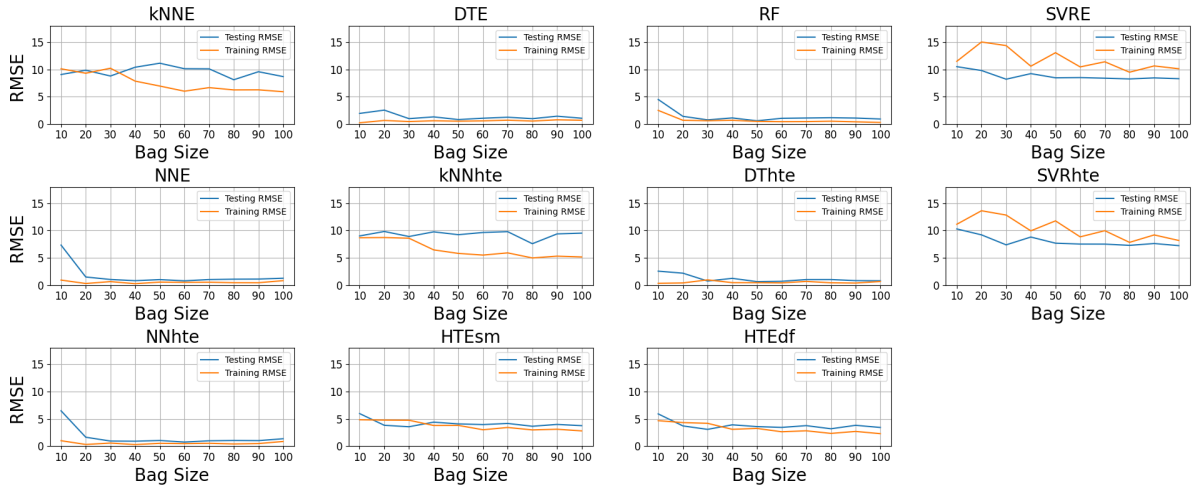

Figure H.1: Ensemble Performance on Bagged Subsets of the Yacht Hydrodynamics Dataset

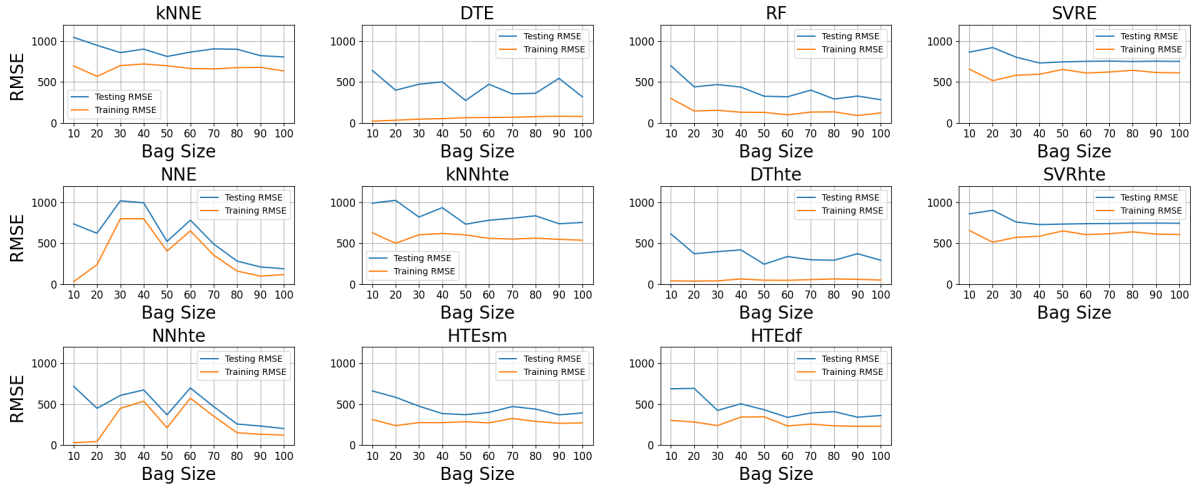

Figure H.2: Ensemble Performance on Bagged Subsets of the Residential Building Dataset

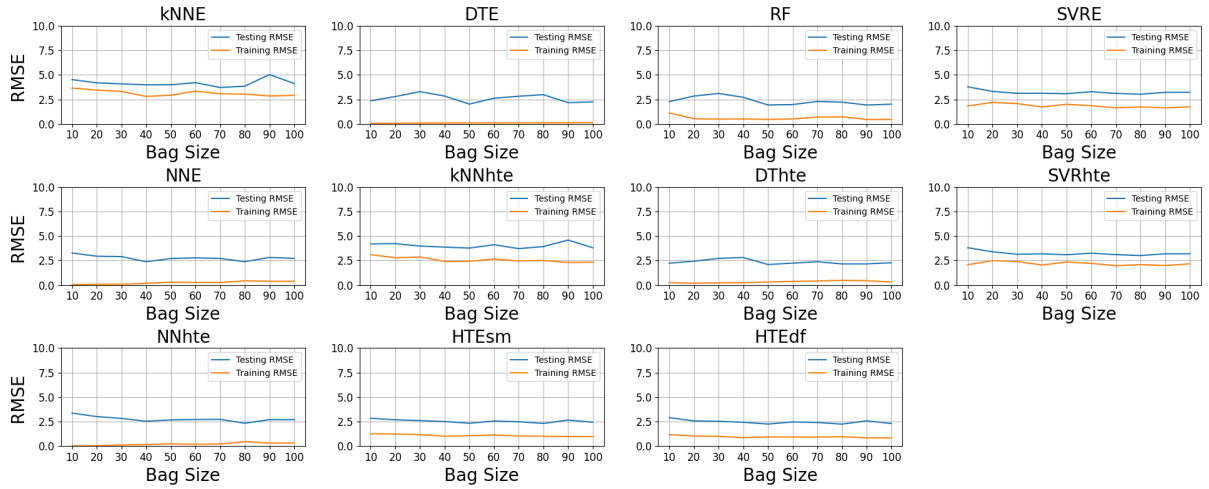

Figure H.3: Ensemble Performance on Bagged Subsets of the Student Performance Dataset

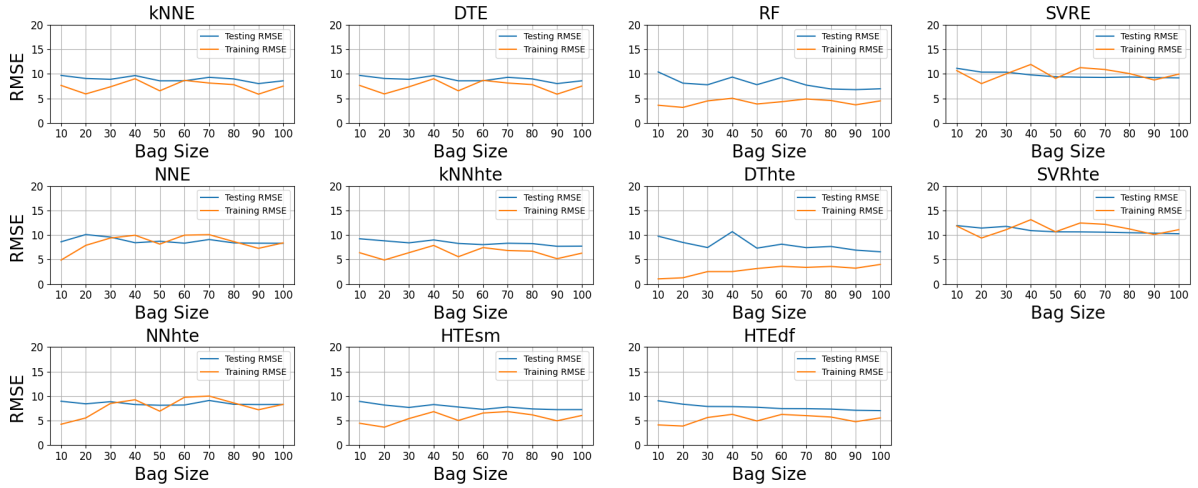

Figure H.4: Ensemble Performance on Bagged Subsets of the Real Estate Dataset

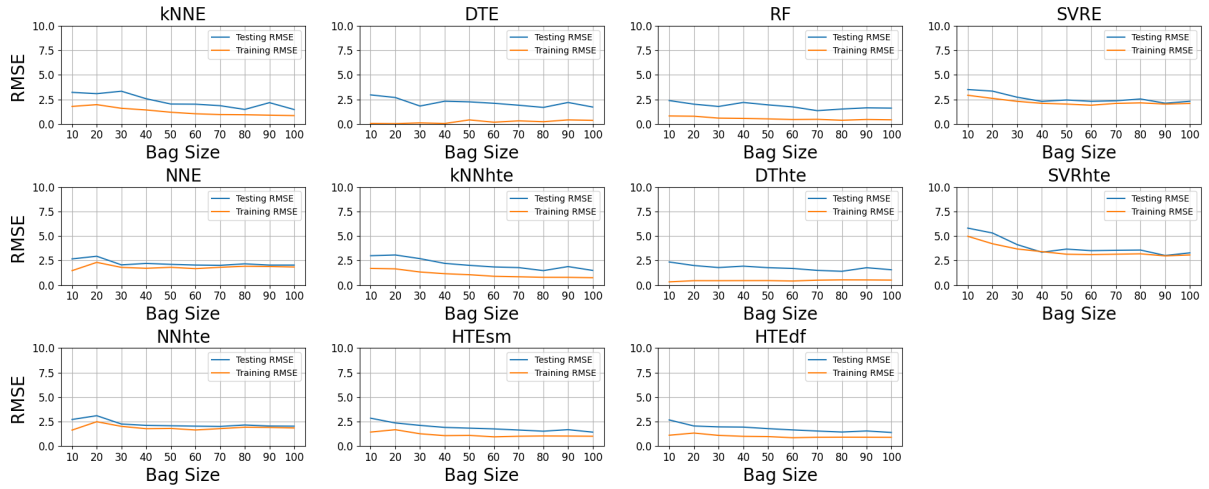

Figure H.5: Ensemble Performance on Bagged Subsets of the Energy Efficiency Dataset

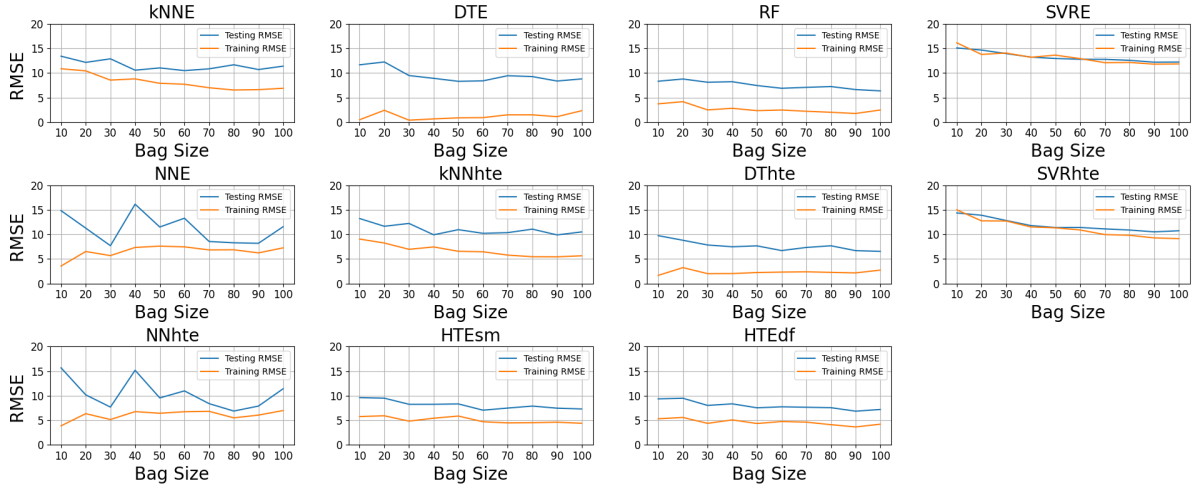

Figure H.6: Ensemble Performance on Bagged Subsets of the Concrete Dataset

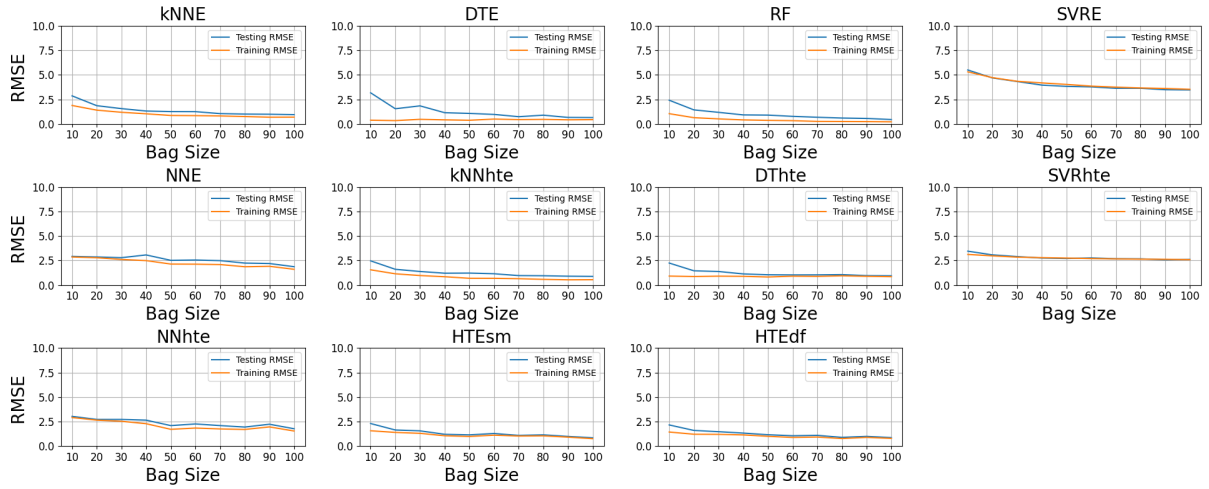

Figure H.7: Ensemble Performance on Bagged Subsets of the Parkinsons Disease Dataset

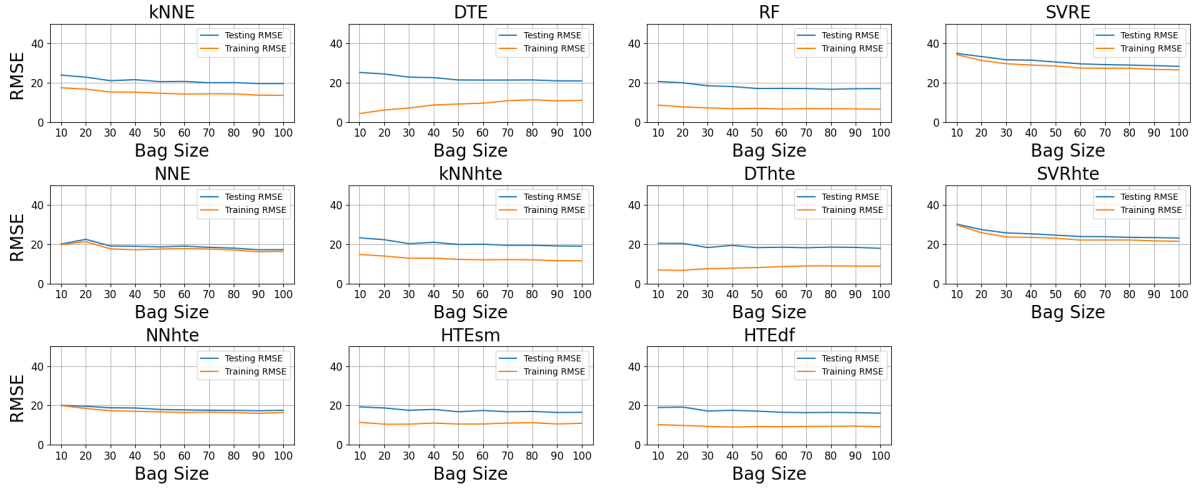

Figure H.8: Ensemble Performance on Bagged Subsets of the Air Quality Dataset

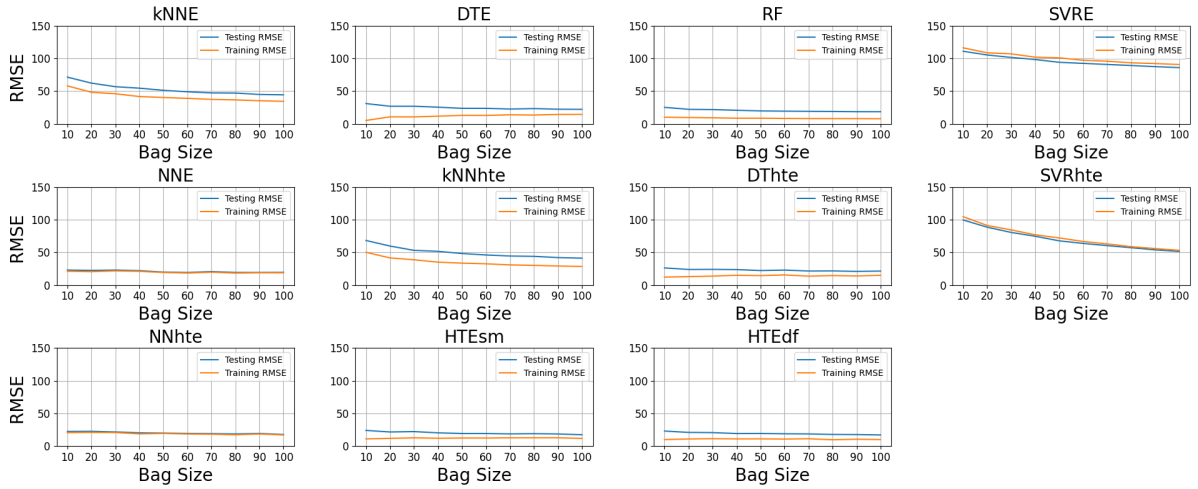

Figure H.9: Ensemble Performance on Bagged Subsets of the Bike Sharing Dataset

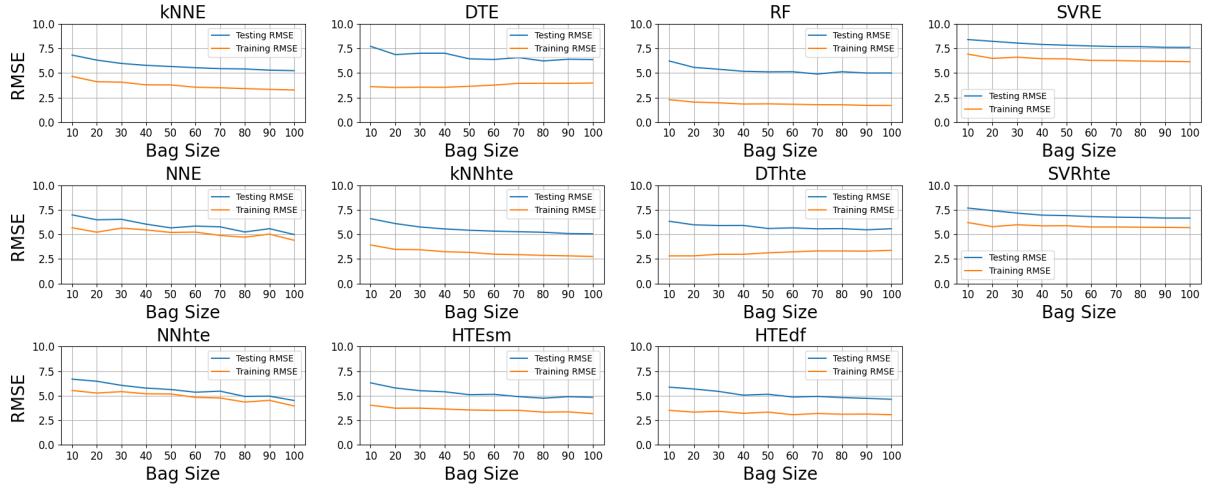

Figure H.10: Ensemble Performance on Bagged Subsets of the Gas Turbine Dataset

Yacht Hydrodynamics Dataset

Table H.1: Ensemble Performance on Bagged Subsets of the Yacht Hydrodynamics Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	9.058	9.843	8.780	10.390	11.128	10.119	10.098	8.095	9.570	8.676
Training RMSE	10.103	9.291	10.209	7.848	6.941	5.996	6.652	6.233	6.246	5.902
GF	0.804	1.122	0.740	1.753	2.571	2.848	2.305	1.686	2.347	2.161
DTE										
Testing RMSE	1.922	2.524	0.961	1.284	0.795	1.039	1.221	0.956	1.410	1.037
Training RMSE	0.204	0.627	0.428	0.575	0.493	0.554	0.679	0.517	0.746	0.666
GF	89.154	16.196	5.042	4.979	2.598	3.512	3.230	3.416	3.575	2.428
RF										
Testing RMSE	4.477	1.384	0.740	1.093	0.565	1.022	1.079	1.132	1.069	0.896
Training RMSE	2.475	0.660	0.579	0.662	0.449	0.409	0.421	0.494	0.380	0.261
GF	3.273	4.400	1.630	2.727	1.587	6.231	6.559	5.252	7.913	11.825
SVRE										
Testing RMSE	10.497	9.792	8.194	9.220	8.452	8.494	8.381	8.241	8.438	8.285
Training RMSE	11.477	15.020	14.363	10.579	13.057	10.450	11.391	9.481	10.631	10.109
GF	0.837	0.425	0.325	0.760	0.419	0.661	0.541	0.755	0.630	0.672
NNE										
Testing RMSE	7.316	1.469	1.012	0.788	0.988	0.767	0.998	1.068	1.088	1.242
Training RMSE	0.900	0.274	0.622	0.241	0.530	0.486	0.510	0.424	0.418	0.797
GF	66.138	28.771	2.650	10.657	3.477	2.488	3.824	6.341	6.766	2.430

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	9.000	9.817	8.895	9.750	9.229	9.635	9.781	7.574	9.372	9.513
Training RMSE	8.668	8.707	8.580	6.443	5.793	5.488	5.890	4.960	5.292	5.144
GF	1.078	1.271	1.075	2.290	2.538	3.082	2.758	2.332	3.136	3.420
DThte										
Testing RMSE	2.539	2.169	0.728	1.219	0.620	0.683	0.995	1.000	0.798	0.774
Training RMSE	0.311	0.377	0.941	0.432	0.427	0.386	0.655	0.407	0.357	0.637
GF	66.560	33.142	0.599	7.942	2.108	3.142	2.307	6.036	4.991	1.477
SVRhte										
Testing RMSE	10.283	9.198	7.364	8.796	7.669	7.508	7.500	7.271	7.606	7.219
Training RMSE	11.139	13.621	12.828	9.936	11.764	8.831	9.962	7.821	9.184	8.163
GF	0.852	0.456	0.330	0.784	0.425	0.723	0.567	0.864	0.686	0.782
NNhte										
Testing RMSE	6.472	1.630	0.929	0.902	1.027	0.745	0.975	1.029	1.005	1.351
Training RMSE	0.991	0.302	0.557	0.283	0.512	0.460	0.519	0.394	0.469	0.849
GF	42.655	29.231	2.789	10.171	4.021	2.627	3.527	6.838	4.597	2.534
HTEsm										
Testing RMSE	5.964	3.817	3.552	4.396	4.074	3.952	4.156	3.642	3.978	3.750
Training RMSE	4.831	4.773	4.744	3.785	3.799	2.999	3.417	2.972	3.095	2.791
GF	1.524	0.640	0.561	1.349	1.150	1.736	1.479	1.502	1.651	1.804
HTEdf										
Testing RMSE	5.909	3.705	3.069	3.905	3.581	3.431	3.756	3.181	3.814	3.420
Training RMSE	4.663	4.315	4.182	3.079	3.244	2.633	2.810	2.332	2.692	2.283
GF	1.606	0.737	0.538	1.608	1.219	1.698	1.787	1.861	2.007	2.244

Residential Building Dataset

Table H.2: Ensemble Performance on Bagged Subsets of the Residential Building Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	1045.241	948.660	859.303	901.854	812.091	865.657	905.063	900.529	822.351	806.101
Training RMSE	696.959	568.959	700.431	720.323	698.990	665.212	660.326	674.975	679.242	635.210
GF	2.249	2.780	1.505	1.568	1.350	1.693	1.879	1.780	1.466	1.610
DTE										
Testing RMSE	641.621	399.555	472.476	501.958	273.969	472.399	354.701	362.479	544.913	319.607
Training RMSE	20.687	32.913	45.774	52.158	63.520	66.122	68.504	76.851	81.602	77.950
GF	961.976	147.372	106.543	92.619	18.603	51.041	26.809	22.247	44.592	16.811
RF										
Testing RMSE	699.136	439.991	468.728	437.645	325.825	319.010	400.686	291.508	327.775	283.167
Training RMSE	299.597	145.142	153.578	130.235	128.550	98.470	132.228	134.745	89.162	122.921
GF	5.446	9.190	9.315	11.292	6.424	10.496	9.182	4.680	13.514	5.307
SVRE										
Testing RMSE	865.425	921.051	803.570	732.199	745.918	752.762	756.200	750.403	755.115	751.679
Training RMSE	656.334	516.319	582.407	594.435	653.798	609.951	622.372	643.079	615.821	611.687
GF	1.739	3.182	1.904	1.517	1.302	1.523	1.476	1.362	1.504	1.510
NNE										
Testing RMSE	734.449	620.055	1017.450	992.686	520.675	781.274	489.289	280.315	208.287	186.759
Training RMSE	28.991	236.930	798.662	797.528	402.912	650.022	352.883	158.590	96.496	114.519
GF	641.807	6.849	1.623	1.549	1.670	1.445	1.923	3.124	4.659	2.660

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	988.505	1023.367	818.560	933.402	731.492	779.652	804.403	834.035	736.880	752.148
Training RMSE	626.134	498.771	601.428	618.233	599.822	558.393	549.920	560.091	546.188	535.444
GF	2.492	4.210	1.852	2.279	1.487	1.949	2.140	2.217	1.820	1.973
DThte										
Testing RMSE	612.774	370.178	395.626	417.501	241.644	335.926	296.209	290.378	370.497	290.026
Training RMSE	38.701	35.335	38.575	61.610	46.604	45.547	53.283	61.838	56.943	48.673
GF	250.697	109.753	105.188	45.920	26.884	54.397	30.905	22.051	42.333	35.506
SVRhte										
Testing RMSE	858.049	900.989	757.879	726.326	732.847	737.608	739.857	743.519	744.834	743.073
Training RMSE	654.333	511.153	570.420	583.866	649.830	603.381	614.933	637.496	610.250	605.883
GF	1.720	3.107	1.765	1.548	1.272	1.494	1.448	1.360	1.490	1.504
NNhte										
Testing RMSE	718.671	449.757	607.722	674.778	370.321	699.025	470.738	258.070	233.600	202.508
Training RMSE	29.455	43.703	449.451	537.592	211.831	573.875	353.718	151.287	132.681	122.524
GF	595.295	105.907	1.828	1.575	3.056	1.484	1.771	2.910	3.100	2.732
HTEsm										
Testing RMSE	662.652	584.795	476.257	385.864	372.384	399.603	471.365	440.497	370.094	393.128
Training RMSE	313.348	238.536	274.703	274.387	285.832	271.542	325.876	290.190	266.021	271.900
GF	4.472	6.010	3.006	1.978	1.697	2.166	2.092	2.304	1.936	2.091
HTEdf										
Testing RMSE	688.482	694.091	424.300	505.054	431.972	339.385	392.940	409.262	341.372	361.498
Training RMSE	302.886	282.261	239.084	343.262	345.835	234.550	256.873	235.730	230.276	231.139
GF	5.167	6.047	3.150	2.165	1.560	2.094	2.340	3.014	2.198	2.446

Student Performance Dataset

Table H.3: Ensemble Performance on Bagged Subsets of the Student Performance Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	4.510	4.182	4.078	3.982	3.984	4.198	3.710	3.842	5.024	4.119
Training RMSE	3.647	3.442	3.304	2.799	2.921	3.342	3.069	3.032	2.845	2.910
GF	1.529	1.476	1.523	2.024	1.861	1.578	1.461	1.606	3.118	2.004
DTE										
Testing RMSE	2.357	2.787	3.285	2.837	2.015	2.618	2.817	2.978	2.162	2.246
Training RMSE	0.053	0.062	0.087	0.091	0.095	0.097	0.100	0.105	0.117	0.126
GF	1986.341	2014.016	1413.108	977.482	449.789	733.659	799.941	808.991	340.468	316.073
RF										
Testing RMSE	2.268	2.836	3.101	2.711	1.930	1.970	2.284	2.216	1.926	2.005
Training RMSE	1.114	0.521	0.486	0.505	0.458	0.504	0.692	0.717	0.454	0.454
GF	4.146	29.638	40.715	28.839	17.781	15.258	10.880	9.562	17.983	19.515
SVRE										
Testing RMSE	3.776	3.307	3.118	3.120	3.074	3.277	3.104	3.022	3.213	3.221
Training RMSE	1.826	2.192	2.065	1.722	1.999	1.850	1.635	1.733	1.634	1.736
GF	4.275	2.275	2.280	3.283	2.365	3.137	3.602	3.041	3.867	3.442
NNE										
Testing RMSE	3.264	2.931	2.894	2.372	2.697	2.768	2.703	2.375	2.811	2.709
Training RMSE	0.045	0.076	0.080	0.175	0.283	0.255	0.252	0.424	0.382	0.376
GF	5153.767	1476.275	1315.048	183.516	91.038	118.094	115.169	31.450	54.124	51.887

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
kNNhte										
Testing RMSE	4.189	4.214	3.965	3.864	3.765	4.101	3.708	3.919	4.584	3.796
Training RMSE	3.085	2.782	2.857	2.393	2.415	2.655	2.445	2.493	2.293	2.325
GF	1.844	2.294	1.926	2.608	2.430	2.386	2.301	2.472	3.997	2.666
DThte										
Testing RMSE	2.221	2.426	2.716	2.801	2.087	2.227	2.370	2.149	2.150	2.267
Training RMSE	0.241	0.181	0.220	0.226	0.309	0.371	0.402	0.473	0.435	0.308
GF	84.619	180.113	152.405	153.666	45.523	35.964	34.743	20.679	24.400	54.168
SVRhte										
Testing RMSE	3.803	3.381	3.133	3.174	3.085	3.242	3.097	3.001	3.178	3.174
Training RMSE	2.066	2.486	2.386	2.039	2.337	2.201	1.967	2.077	1.984	2.160
GF	3.387	1.849	1.724	2.424	1.743	2.169	2.480	2.088	2.566	2.160
NNhte										
Testing RMSE	3.363	3.007	2.822	2.534	2.670	2.713	2.728	2.329	2.700	2.693
Training RMSE	0.038	0.044	0.117	0.152	0.231	0.175	0.212	0.451	0.308	0.317
GF	7867.846	4622.090	581.966	276.811	133.390	240.777	165.332	26.655	76.994	71.992
HTEsm										
Testing RMSE	2.839	2.683	2.602	2.504	2.329	2.559	2.485	2.310	2.656	2.439
Training RMSE	1.257	1.229	1.172	1.005	1.061	1.138	1.035	1.010	0.979	0.966
GF	5.105	4.765	4.926	6.203	4.815	5.063	5.763	5.228	7.362	6.372
HTEdf										
Testing RMSE	2.899	2.570	2.529	2.429	2.247	2.463	2.406	2.240	2.580	2.300
Training RMSE	1.172	1.025	0.996	0.865	0.933	0.927	0.922	0.956	0.834	0.835
GF	6.118	6.287	6.442	7.895	5.803	7.065	6.816	5.488	9.560	7.586

Real Estate Dataset

Table H.4: Ensemble Performance on Bagged Subsets of the Real Estate Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	9.675	9.057	8.880	9.646	8.572	8.588	9.286	8.957	8.009	8.578
Training RMSE	7.636	5.899	7.354	8.992	6.537	8.669	8.123	7.800	5.854	7.488
GF	1.605	2.357	1.458	1.151	1.720	0.981	1.307	1.319	1.872	1.312
DTE										
Testing RMSE	11.047	9.192	8.088	11.901	8.427	10.846	7.848	8.731	7.213	6.725
Training RMSE	0.394	0.653	1.885	1.842	2.738	2.642	2.812	3.017	2.708	3.749
GF	786.414	198.230	18.408	41.760	9.468	16.848	7.788	8.372	7.097	3.219
RF										
Testing RMSE	10.380	8.098	7.768	9.335	7.785	9.240	7.703	6.909	6.776	6.953
Training RMSE	3.596	3.156	4.480	5.021	3.847	4.314	4.873	4.556	3.680	4.485
GF	8.330	6.586	3.006	3.457	4.095	4.588	2.498	2.299	3.391	2.403
SVRE										
Testing RMSE	11.126	10.350	10.341	9.779	9.404	9.310	9.264	9.355	9.238	9.174
Training RMSE	10.651	8.003	9.997	11.912	9.060	11.262	10.879	10.047	8.747	9.943
GF	1.091	1.672	1.070	0.674	1.077	0.683	0.725	0.867	1.115	0.851
NNE										
Testing RMSE	8.604	10.078	9.555	8.409	8.712	8.318	9.068	8.375	8.318	8.301
Training RMSE	4.878	7.875	9.372	9.941	8.131	9.953	10.057	8.642	7.265	8.377
GF	3.111	1.638	1.039	0.715	1.148	0.698	0.813	0.939	1.311	0.982

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	9.217	8.801	8.388	8.983	8.268	8.023	8.310	8.232	7.687	7.715
Training RMSE	6.373	4.890	6.386	7.862	5.561	7.420	6.822	6.688	5.159	6.281
GF	2.092	3.240	1.726	1.305	2.211	1.169	1.484	1.515	2.220	1.509
DThte										
Testing RMSE	9.736	8.469	7.429	10.667	7.307	8.114	7.411	7.652	6.903	6.568
Training RMSE	1.043	1.266	2.529	2.534	3.155	3.606	3.386	3.578	3.224	3.989
GF	87.147	44.754	8.626	17.717	5.364	5.063	4.791	4.574	4.585	2.712
SVRhte										
Testing RMSE	11.903	11.398	11.747	10.881	10.631	10.619	10.559	10.452	10.363	10.240
Training RMSE	11.814	9.368	11.070	13.112	10.644	12.435	12.166	11.213	10.080	11.077
GF	1.015	1.480	1.126	0.689	0.998	0.729	0.753	0.869	1.057	0.855
NNhte										
Testing RMSE	8.941	8.415	8.850	8.272	8.126	8.172	9.096	8.322	8.263	8.289
Training RMSE	4.254	5.536	8.484	9.249	6.916	9.750	9.999	8.592	7.204	8.303
GF	4.417	2.311	1.088	0.800	1.381	0.702	0.828	0.938	1.316	0.997
HTEsm										
Testing RMSE	8.907	8.167	7.676	8.267	7.766	7.285	7.769	7.375	7.228	7.235
Training RMSE	4.452	3.656	5.412	6.820	5.025	6.539	6.834	6.165	4.954	6.034
GF	4.003	4.989	2.012	1.469	2.389	1.241	1.292	1.431	2.129	1.438
HTEdf										
Testing RMSE	9.035	8.337	7.873	7.849	7.718	7.441	7.428	7.345	7.100	7.031
Training RMSE	4.116	3.883	5.630	6.271	4.926	6.254	6.008	5.723	4.776	5.541
GF	4.819	4.610	1.955	1.567	2.454	1.415	1.528	1.647	2.210	1.610

Energy Efficiency Dataset

Table H.5: Ensemble Performance on Bagged Subsets of the Energy Efficiency Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	3.214	3.079	3.329	2.565	2.025	2.009	1.855	1.473	2.163	1.464
Training RMSE	1.781	1.960	1.588	1.420	1.187	1.030	0.951	0.933	0.886	0.844
GF	3.257	2.468	4.396	3.266	2.911	3.802	3.808	2.492	5.957	3.009
DTE										
Testing RMSE	2.953	2.687	1.818	2.309	2.243	2.099	1.901	1.678	2.183	1.717
Training RMSE	0.046	0.023	0.103	0.040	0.406	0.171	0.311	0.221	0.406	0.361
GF	4066.443	13729.487	313.818	3350.429	30.580	149.996	37.329	57.649	28.907	22.623
RF										
Testing RMSE	2.376	2.009	1.775	2.184	1.938	1.733	1.348	1.513	1.636	1.607
Training RMSE	0.808	0.781	0.587	0.558	0.509	0.448	0.467	0.369	0.450	0.416
GF	8.638	6.614	9.139	15.343	14.491	14.942	8.339	16.832	13.222	14.912
SVRE										
Testing RMSE	3.500	3.339	2.718	2.293	2.436	2.298	2.353	2.538	2.106	2.304
Training RMSE	2.915	2.598	2.295	2.094	2.019	1.902	2.087	2.143	2.017	2.082
GF	1.442	1.652	1.403	1.199	1.456	1.459	1.271	1.403	1.089	1.225
NNE										
Testing RMSE	2.660	2.931	2.051	2.198	2.105	2.032	2.004	2.156	2.028	2.029
Training RMSE	1.464	2.306	1.793	1.703	1.801	1.667	1.804	1.911	1.882	1.835
GF	3.300	1.615	1.308	1.666	1.365	1.486	1.235	1.273	1.161	1.222

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	2.984	3.059	2.687	2.203	2.002	1.838	1.773	1.465	1.877	1.485
Training RMSE	1.690	1.645	1.325	1.151	1.048	0.888	0.837	0.784	0.780	0.743
GF	3.117	3.459	4.110	3.664	3.649	4.280	4.485	3.495	5.794	3.999
DThte										
Testing RMSE	2.345	1.986	1.778	1.920	1.766	1.686	1.489	1.397	1.762	1.555
Training RMSE	0.320	0.440	0.438	0.444	0.447	0.407	0.492	0.523	0.517	0.500
GF	53.788	20.348	16.499	18.711	15.630	17.146	9.153	7.147	11.612	9.692
SVRhte										
Testing RMSE	5.796	5.299	4.117	3.340	3.659	3.499	3.537	3.561	3.009	3.276
Training RMSE	4.964	4.203	3.672	3.401	3.137	3.101	3.138	3.177	2.964	3.062
GF	1.363	1.590	1.257	0.965	1.361	1.273	1.271	1.256	1.030	1.145
NNhte										
Testing RMSE	2.714	3.100	2.250	2.113	2.074	2.035	2.008	2.150	2.048	2.033
Training RMSE	1.633	2.486	2.012	1.775	1.801	1.645	1.781	1.928	1.903	1.849
GF	2.762	1.555	1.250	1.418	1.327	1.531	1.271	1.244	1.159	1.209
HTEsm										
Testing RMSE	2.844	2.360	2.111	1.906	1.831	1.753	1.639	1.520	1.677	1.424
Training RMSE	1.431	1.671	1.258	1.067	1.090	0.946	1.007	1.034	1.024	1.007
GF	3.948	1.995	2.816	3.189	2.820	3.434	2.649	2.163	2.679	2.000
HTEdf										
Testing RMSE	2.668	2.052	1.960	1.937	1.782	1.651	1.537	1.435	1.544	1.390
Training RMSE	1.110	1.329	1.093	0.999	0.971	0.855	0.901	0.910	0.906	0.897
GF	5.776	2.382	3.212	3.755	3.368	3.730	2.908	2.486	2.909	2.399

Concrete Dataset

Table H.6: Ensemble Performance on Bagged Subsets of the Concrete Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	13.408	12.150	12.879	10.565	11.026	10.481	10.838	11.663	10.697	11.384
Training RMSE	10.855	10.428	8.546	8.798	7.897	7.705	6.991	6.531	6.605	6.882
GF	1.526	1.357	2.271	1.442	1.949	1.850	2.404	3.189	2.623	2.736
DTE										
Testing RMSE	11.658	12.237	9.473	8.905	8.301	8.394	9.439	9.262	8.355	8.789
Training RMSE	0.479	2.402	0.392	0.650	0.866	0.911	1.482	1.476	1.104	2.314
GF	592.217	25.945	584.855	187.959	91.823	84.864	40.572	39.366	57.249	14.424
RF										
Testing RMSE	8.316	8.768	8.111	8.216	7.443	6.885	7.068	7.231	6.621	6.350
Training RMSE	3.717	4.154	2.481	2.807	2.329	2.456	2.206	1.997	1.754	2.461
GF	5.005	4.455	10.687	8.568	10.211	7.857	10.265	13.115	14.249	6.659
SVRE										
Testing RMSE	15.089	14.658	13.941	13.229	12.937	12.791	12.768	12.571	12.192	12.225
Training RMSE	16.128	13.775	14.057	13.197	13.626	12.922	12.092	12.142	11.788	11.839
GF	0.875	1.132	0.984	1.005	0.902	0.980	1.115	1.072	1.070	1.066
NNE										
Testing RMSE	14.812	11.259	7.686	16.134	11.487	13.276	8.541	8.273	8.175	11.548
Training RMSE	3.555	6.503	5.667	7.327	7.594	7.449	6.825	6.855	6.221	7.247
GF	17.355	2.998	1.840	4.849	2.288	3.177	1.566	1.456	1.727	2.539

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	13.210	11.638	12.215	9.902	10.947	10.205	10.343	11.053	9.878	10.491
Training RMSE	9.034	8.229	6.951	7.428	6.548	6.444	5.769	5.442	5.421	5.631
GF	2.138	2.000	3.089	1.777	2.795	2.508	3.214	4.126	3.320	3.471
DThte										
Testing RMSE	9.728	8.792	7.828	7.468	7.652	6.693	7.326	7.668	6.679	6.530
Training RMSE	1.644	3.223	1.999	2.016	2.223	2.313	2.370	2.256	2.146	2.721
GF	35.006	7.440	15.338	13.725	11.843	8.371	9.553	11.550	9.682	5.757
SVRhte										
Testing RMSE	14.347	13.881	12.812	11.782	11.383	11.391	11.101	10.871	10.501	10.725
Training RMSE	14.984	12.741	12.712	11.499	11.318	10.885	9.946	9.806	9.270	9.124
GF	0.917	1.187	1.016	1.050	1.012	1.095	1.246	1.229	1.283	1.382
NNhte										
Testing RMSE	15.662	10.168	7.683	15.178	9.546	10.974	8.382	6.872	7.886	11.411
Training RMSE	3.870	6.344	5.146	6.757	6.424	6.736	6.815	5.491	6.047	6.977
GF	16.382	2.568	2.229	5.046	2.208	2.654	1.513	1.566	1.701	2.675
HTEsm										
Testing RMSE	9.605	9.493	8.247	8.249	8.324	7.051	7.482	7.882	7.458	7.297
Training RMSE	5.753	5.910	4.823	5.417	5.856	4.692	4.465	4.503	4.605	4.385
GF	2.788	2.580	2.923	2.319	2.021	2.259	2.809	3.064	2.623	2.769
HTEdf										
Testing RMSE	9.352	9.481	8.017	8.342	7.534	7.735	7.646	7.559	6.847	7.196
Training RMSE	5.295	5.572	4.371	5.069	4.340	4.736	4.617	4.100	3.640	4.198
GF	3.119	2.896	3.363	2.709	3.013	2.667	2.743	3.398	3.538	2.938

Parkinsons Disease Dataset

Table H.7: Ensemble Performance on Bagged Subsets of the Parkinsons Disease Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	2.856	1.856	1.556	1.310	1.254	1.247	1.046	1.000	0.984	0.943
Training RMSE	1.876	1.400	1.183	1.027	0.858	0.841	0.809	0.748	0.682	0.688
GF	2.316	1.756	1.730	1.625	2.136	2.200	1.671	1.785	2.082	1.880
DTE										
Testing RMSE	3.165	1.542	1.837	1.140	1.066	0.965	0.733	0.887	0.659	0.645
Training RMSE	0.376	0.334	0.466	0.411	0.369	0.491	0.437	0.460	0.415	0.433
GF	70.822	21.280	15.547	7.682	8.355	3.861	2.820	3.723	2.521	2.215
RF										
Testing RMSE	2.409	1.426	1.176	0.915	0.894	0.768	0.675	0.596	0.553	0.447
Training RMSE	1.040	0.633	0.509	0.405	0.360	0.327	0.253	0.243	0.234	0.211
GF	5.368	5.067	5.343	5.106	6.151	5.530	7.123	6.042	5.595	4.495
SVRE										
Testing RMSE	5.493	4.681	4.316	3.953	3.822	3.770	3.629	3.626	3.483	3.459
Training RMSE	5.302	4.713	4.342	4.175	4.023	3.850	3.744	3.662	3.607	3.526
GF	1.073	0.986	0.988	0.897	0.903	0.959	0.940	0.980	0.932	0.962
NNE										
Testing RMSE	2.916	2.855	2.784	3.064	2.514	2.546	2.482	2.233	2.187	1.865
Training RMSE	2.852	2.780	2.603	2.485	2.134	2.123	2.082	1.861	1.906	1.599
GF	1.046	1.055	1.144	1.521	1.388	1.438	1.421	1.438	1.317	1.361

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	2.459	1.600	1.377	1.202	1.214	1.146	0.958	0.942	0.899	0.879
Training RMSE	1.547	1.140	0.963	0.843	0.683	0.678	0.641	0.576	0.529	0.539
GF	2.529	1.970	2.045	2.036	3.162	2.859	2.232	2.674	2.889	2.658
DThte										
Testing RMSE	2.237	1.447	1.375	1.129	1.043	1.026	1.032	1.058	0.952	0.936
Training RMSE	0.909	0.866	0.902	0.885	0.815	0.898	0.879	0.937	0.884	0.850
GF	6.058	2.794	2.325	1.629	1.638	1.304	1.378	1.276	1.160	1.214
SVRhte										
Testing RMSE	3.447	3.069	2.899	2.745	2.700	2.758	2.678	2.668	2.566	2.606
Training RMSE	3.116	2.969	2.838	2.786	2.745	2.690	2.662	2.636	2.619	2.586
GF	1.224	1.068	1.043	0.971	0.967	1.051	1.012	1.024	0.960	1.016
NNhte										
Testing RMSE	3.031	2.719	2.721	2.648	2.090	2.258	2.091	1.942	2.229	1.769
Training RMSE	2.910	2.647	2.532	2.283	1.711	1.839	1.750	1.698	1.958	1.537
GF	1.084	1.055	1.155	1.345	1.492	1.508	1.428	1.308	1.296	1.325
HTEsm										
Testing RMSE	2.300	1.642	1.555	1.204	1.145	1.286	1.091	1.146	0.987	0.845
Training RMSE	1.564	1.396	1.302	1.054	0.984	1.113	1.026	1.044	0.914	0.757
GF	2.163	1.383	1.427	1.305	1.354	1.336	1.130	1.206	1.167	1.245
HTEdf										
Testing RMSE	2.157	1.597	1.469	1.326	1.159	1.055	1.096	0.890	0.992	0.863
Training RMSE	1.434	1.210	1.199	1.141	0.999	0.884	0.925	0.770	0.889	0.786
GF	2.261	1.744	1.501	1.350	1.347	1.426	1.401	1.335	1.244	1.204

Air Quality Dataset

Table H.8: Ensemble Performance on Bagged Subsets of the Air Quality Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	23.907	22.880	21.065	21.584	20.589	20.705	20.020	20.089	19.597	19.590
Training RMSE	17.476	16.792	15.294	15.238	14.675	14.259	14.383	14.368	13.717	13.573
GF	1.871	1.857	1.897	2.006	1.968	2.108	1.937	1.955	2.041	2.083
DTE										
Testing RMSE	25.264	24.465	22.891	22.613	21.433	21.385	21.397	21.442	20.977	20.957
Training RMSE	4.368	6.159	7.167	8.701	9.201	9.604	10.890	11.349	10.817	11.041
GF	33.456	15.779	10.203	6.754	5.426	4.958	3.860	3.569	3.760	3.603
RF										
Testing RMSE	20.653	19.991	18.492	18.086	17.107	17.152	17.072	16.698	16.974	17.013
Training RMSE	8.638	7.751	7.225	6.823	7.008	6.683	6.871	6.792	6.744	6.577
GF	5.716	6.652	6.551	7.026	5.959	6.587	6.173	6.044	6.335	6.692
SVRE										
Testing RMSE	34.957	33.325	31.709	31.511	30.577	29.639	29.260	29.011	28.712	28.370
Training RMSE	34.326	31.342	29.679	29.030	28.534	27.487	27.425	27.405	26.832	26.582
GF	1.037	1.131	1.142	1.178	1.148	1.163	1.138	1.121	1.145	1.139
NNE										
Testing RMSE	20.010	22.464	19.049	18.943	18.611	19.010	18.366	17.983	17.129	17.197
Training RMSE	19.731	21.278	17.595	17.121	17.499	17.721	17.498	17.032	16.098	16.329
GF	1.028	1.115	1.172	1.224	1.131	1.151	1.102	1.115	1.132	1.109

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	23.230	22.258	20.258	20.933	19.854	19.943	19.421	19.449	19.088	18.974
Training RMSE	14.756	13.942	12.846	12.896	12.241	12.026	12.110	12.010	11.578	11.520
GF	2.478	2.549	2.487	2.635	2.631	2.750	2.572	2.622	2.718	2.713
DThte										
Testing RMSE	20.428	20.393	18.273	19.395	18.228	18.425	18.156	18.494	18.347	17.896
Training RMSE	6.834	6.689	7.531	7.781	8.076	8.567	8.918	8.913	8.837	8.836
GF	8.935	9.294	5.887	6.213	5.094	4.625	4.145	4.306	4.310	4.102
SVRhte										
Testing RMSE	30.176	27.401	25.742	25.268	24.594	23.879	23.809	23.461	23.316	23.089
Training RMSE	29.756	25.840	23.685	23.426	23.050	22.098	22.109	22.105	21.670	21.487
GF	1.028	1.124	1.181	1.163	1.138	1.168	1.160	1.126	1.158	1.155
NNhte										
Testing RMSE	20.090	19.616	18.842	18.740	18.011	17.766	17.605	17.534	17.326	17.566
Training RMSE	19.957	18.504	17.336	17.067	16.773	16.444	16.662	16.416	15.993	16.474
GF	1.013	1.124	1.181	1.206	1.153	1.167	1.116	1.141	1.174	1.137
HTEsm										
Testing RMSE	19.298	18.754	17.561	18.028	16.820	17.466	16.816	16.999	16.452	16.538
Training RMSE	11.432	10.501	10.523	11.052	10.580	10.599	11.069	11.241	10.610	10.941
GF	2.850	3.190	2.785	2.661	2.527	2.715	2.308	2.287	2.404	2.285
HTEdf										
Testing RMSE	19.016	19.210	17.206	17.543	17.160	16.552	16.391	16.509	16.383	16.127
Training RMSE	10.238	9.850	9.369	9.059	9.318	9.258	9.360	9.431	9.516	9.242
GF	3.450	3.803	3.372	3.750	3.392	3.197	3.067	3.064	2.964	3.045

Bike Sharing Dataset

Table H.9: Ensemble Performance on Bagged Subsets of the Bike Sharing Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	71.425	62.305	56.827	54.585	51.315	49.061	47.340	47.081	45.016	44.420
Training RMSE	58.078	48.320	46.046	41.804	40.369	39.006	37.495	36.675	35.379	34.492
GF	1.512	1.663	1.523	1.705	1.616	1.582	1.594	1.648	1.619	1.659
DTE										
Testing RMSE	31.064	26.985	26.972	25.558	23.770	23.718	22.695	23.342	22.465	22.276
Training RMSE	5.258	10.727	10.693	11.778	13.028	12.939	13.896	13.505	14.465	14.560
GF	34.910	6.329	6.362	4.708	3.329	3.360	2.667	2.987	2.412	2.341
RF										
Testing RMSE	25.251	22.270	21.914	20.785	19.822	19.454	19.181	18.965	18.650	18.588
Training RMSE	10.204	9.733	9.307	8.662	8.698	8.297	8.073	8.000	8.023	7.835
GF	6.124	5.235	5.544	5.757	5.193	5.498	5.645	5.620	5.404	5.628
SVRE										
Testing RMSE	111.009	105.343	101.600	98.388	94.158	92.479	90.847	89.243	87.507	86.039
Training RMSE	116.279	108.613	107.030	102.014	100.949	97.298	95.771	93.356	92.187	90.807
GF	0.911	0.941	0.901	0.930	0.870	0.903	0.900	0.914	0.901	0.898
NNE										
Testing RMSE	22.803	22.287	22.790	21.874	19.742	19.105	20.508	19.147	19.139	19.280
Training RMSE	21.020	20.363	21.574	20.860	19.183	18.267	19.588	18.218	18.654	18.624
GF	1.177	1.198	1.116	1.100	1.059	1.094	1.096	1.105	1.053	1.072

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	68.053	59.547	52.938	51.481	48.097	45.921	44.306	43.800	41.895	41.028
Training RMSE	49.971	41.378	38.569	34.916	33.429	32.382	30.719	29.982	29.025	28.378
GF	1.855	2.071	1.884	2.174	2.070	2.011	2.080	2.134	2.083	2.090
DThte										
Testing RMSE	26.061	23.771	23.933	23.585	22.067	22.790	21.348	21.512	20.796	21.285
Training RMSE	12.038	12.824	13.593	14.841	14.333	15.391	13.508	14.394	13.788	14.824
GF	4.687	3.436	3.100	2.525	2.370	2.192	2.497	2.234	2.275	2.062
SVRhte										
Testing RMSE	99.402	88.375	80.354	74.521	67.540	63.542	60.213	57.004	53.710	51.324
Training RMSE	104.419	91.036	84.203	76.760	71.924	66.539	62.808	58.584	55.685	52.983
GF	0.906	0.942	0.911	0.943	0.882	0.912	0.919	0.947	0.930	0.938
NNhte										
Testing RMSE	22.465	22.637	21.665	20.262	19.858	19.341	19.084	18.558	19.334	17.729
Training RMSE	20.597	20.900	20.763	18.738	19.354	18.529	18.198	17.315	18.557	17.003
GF	1.190	1.173	1.089	1.169	1.053	1.090	1.100	1.149	1.085	1.087
HTEsm										
Testing RMSE	23.989	21.608	22.190	20.282	19.267	19.233	18.654	19.005	18.540	17.423
Training RMSE	11.014	11.803	12.671	12.009	12.369	12.337	12.699	12.732	12.757	11.740
GF	4.744	3.352	3.067	2.852	2.426	2.431	2.158	2.228	2.112	2.203
HTEdf										
Testing RMSE	23.023	20.936	20.623	19.237	19.320	18.842	18.632	17.863	17.666	17.052
Training RMSE	9.944	10.913	11.338	11.010	11.061	10.770	11.259	9.705	10.541	10.015
GF	5.361	3.680	3.308	3.053	3.051	3.060	2.739	3.388	2.809	2.899

Gas Turbine Dataset

Table H.10: Ensemble Performance on Bagged Subsets of the Gas Turbine Dataset

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	6.821	6.310	5.967	5.775	5.656	5.535	5.437	5.408	5.284	5.234
Training RMSE	4.649	4.111	4.060	3.791	3.781	3.549	3.492	3.405	3.332	3.262
GF	2.152	2.356	2.161	2.321	2.238	2.432	2.425	2.521	2.515	2.576
DTE										
Testing RMSE	7.708	6.870	7.004	7.010	6.437	6.376	6.561	6.224	6.397	6.367
Training RMSE	3.606	3.527	3.553	3.542	3.641	3.765	3.943	3.946	3.945	3.973
GF	4.570	3.795	3.886	3.917	3.126	2.868	2.769	2.488	2.630	2.568
RF										
Testing RMSE	6.217	5.568	5.378	5.169	5.112	5.125	4.889	5.124	4.992	4.996
Training RMSE	2.283	2.035	1.961	1.840	1.858	1.815	1.768	1.761	1.703	1.688
GF	7.416	7.489	7.523	7.893	7.568	7.976	7.646	8.468	8.589	8.757
SVRE										
Testing RMSE	8.398	8.220	8.043	7.903	7.823	7.746	7.685	7.677	7.608	7.600
Training RMSE	6.913	6.484	6.603	6.448	6.433	6.279	6.262	6.213	6.181	6.147
GF	1.476	1.607	1.484	1.502	1.479	1.522	1.506	1.527	1.515	1.529
NNE										
Testing RMSE	6.980	6.478	6.529	6.035	5.657	5.838	5.761	5.238	5.573	4.983
Training RMSE	5.666	5.218	5.634	5.447	5.193	5.228	4.884	4.714	5.019	4.395
GF	1.518	1.541	1.343	1.228	1.186	1.247	1.391	1.235	1.233	1.285

Measure	Bagged Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	6.592	6.093	5.746	5.548	5.416	5.324	5.265	5.208	5.082	5.048
Training RMSE	3.917	3.464	3.432	3.231	3.165	2.978	2.931	2.857	2.809	2.745
GF	2.832	3.093	2.803	2.949	2.929	3.195	3.226	3.323	3.273	3.383
DThte										
Testing RMSE	6.327	5.966	5.896	5.903	5.594	5.657	5.561	5.580	5.460	5.571
Training RMSE	2.805	2.808	2.972	2.971	3.116	3.218	3.310	3.306	3.291	3.373
GF	5.088	4.513	3.936	3.947	3.222	3.089	2.823	2.849	2.753	2.728
SVRhte										
Testing RMSE	7.672	7.411	7.157	6.953	6.904	6.802	6.746	6.715	6.655	6.652
Training RMSE	6.197	5.777	5.973	5.862	5.876	5.747	5.745	5.724	5.706	5.679
GF	1.533	1.646	1.436	1.407	1.380	1.401	1.379	1.376	1.360	1.372
NNhte										
Testing RMSE	6.682	6.469	6.051	5.772	5.628	5.354	5.460	4.922	4.952	4.512
Training RMSE	5.529	5.266	5.411	5.192	5.173	4.830	4.763	4.351	4.524	3.957
GF	1.461	1.509	1.251	1.236	1.184	1.229	1.314	1.280	1.198	1.300
HTEsm										
Testing RMSE	6.302	5.784	5.507	5.396	5.103	5.138	4.901	4.746	4.898	4.836
Training RMSE	4.032	3.727	3.732	3.644	3.549	3.507	3.508	3.326	3.355	3.167
GF	2.443	2.409	2.177	2.193	2.067	2.146	1.952	2.036	2.131	2.332
HTEdf										
Testing RMSE	5.866	5.682	5.436	5.054	5.147	4.864	4.923	4.811	4.728	4.637
Training RMSE	3.502	3.329	3.414	3.205	3.326	3.064	3.186	3.113	3.133	3.063
GF	2.806	2.913	2.536	2.487	2.394	2.520	2.388	2.389	2.278	2.292

Appendix I

Ensemble Performance on Feature Subsets for Regression Problems

The results of the ensembles over the feature subsets of the training dataset for regression problems are provided in this appendix. Plots of the testing and training RMSE for each regression dataset are first presented. Then the results of testing RMSE, training RMSE, and GF of the ensembles over the regression datasets are provided.

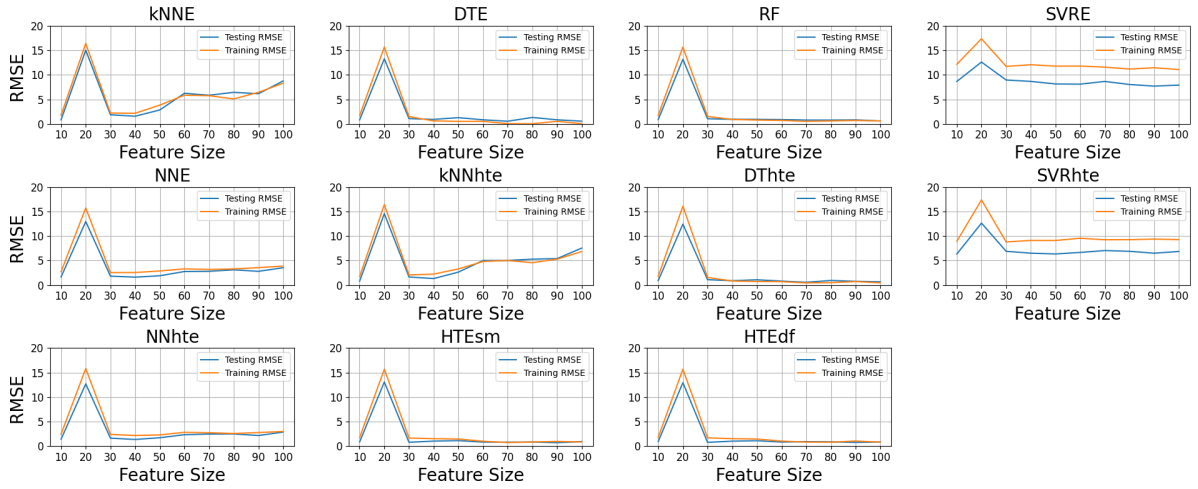

Figure I.1: Ensemble Performance on Feature Subsets of the Yacht Hydrodynamics Dataset

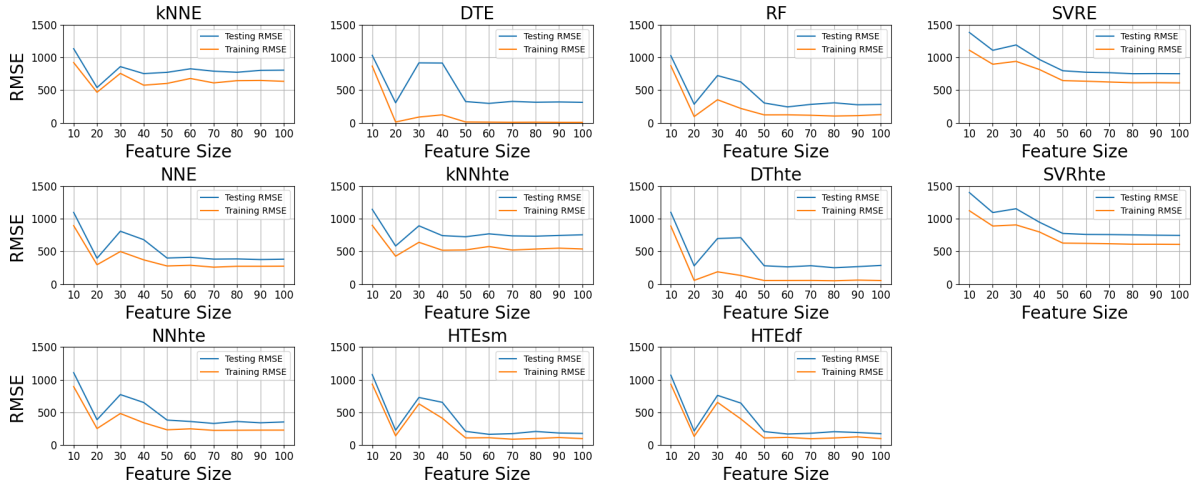

Figure I.2: Ensemble Performance on Feature Subsets of the Residential Building Dataset

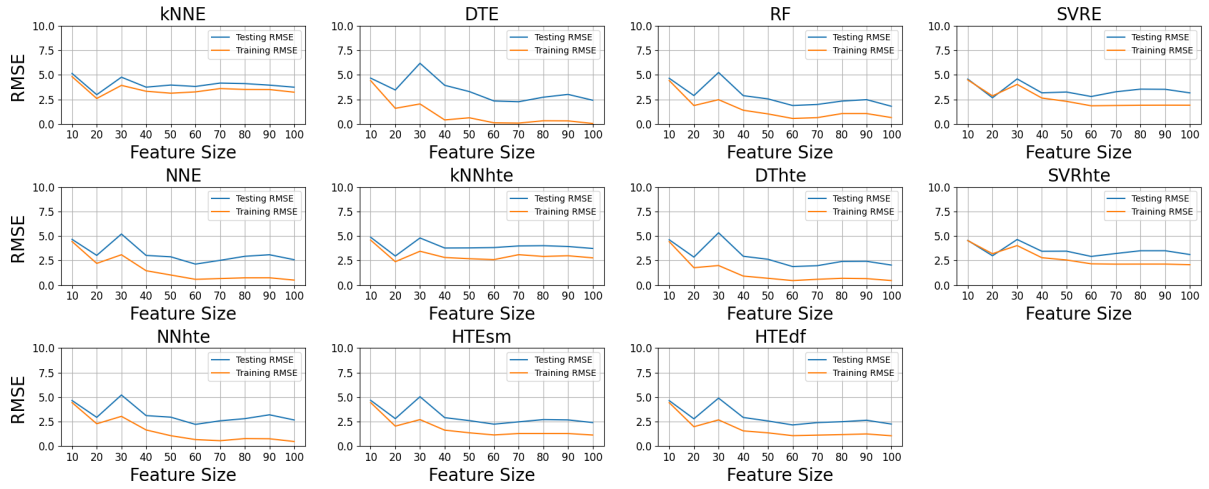

Figure I.3: Ensemble Performance on Feature Subsets of the Student Performance Dataset

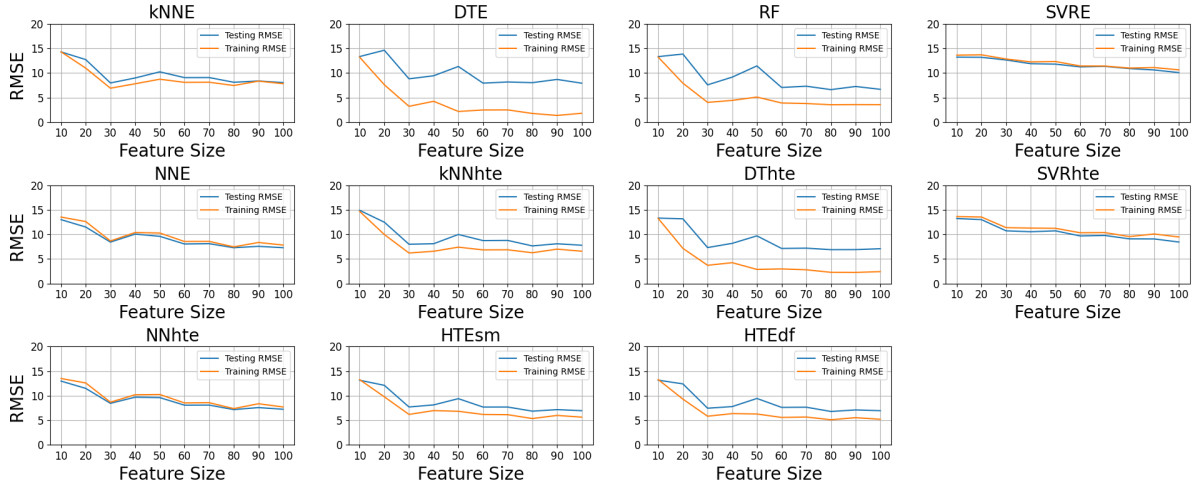

Figure I.4: Ensemble Performance on Feature Subsets of the Real Estate Dataset

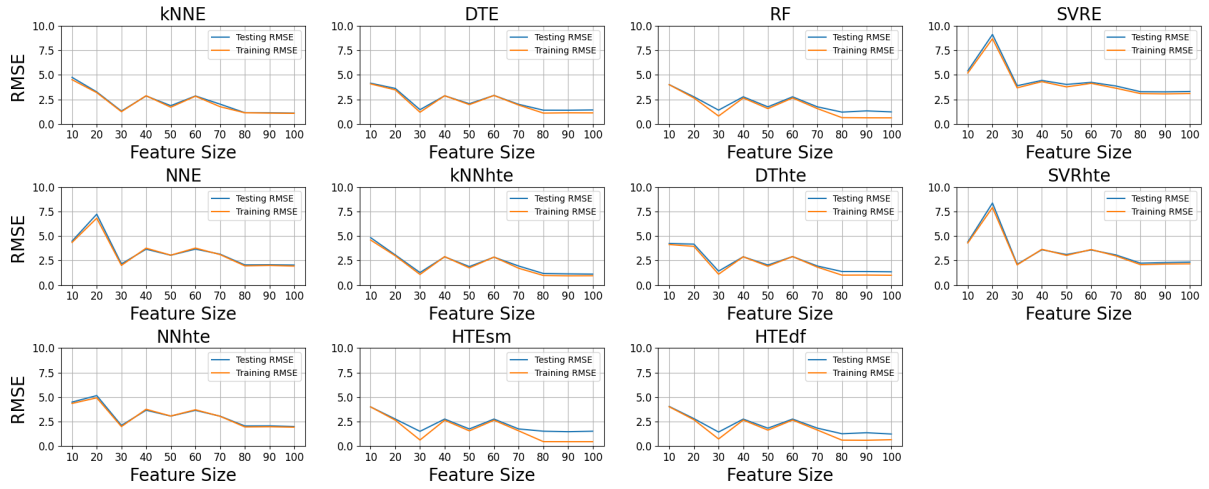

Figure I.5: Ensemble Performance on Feature Subsets of the Energy Efficiency Dataset

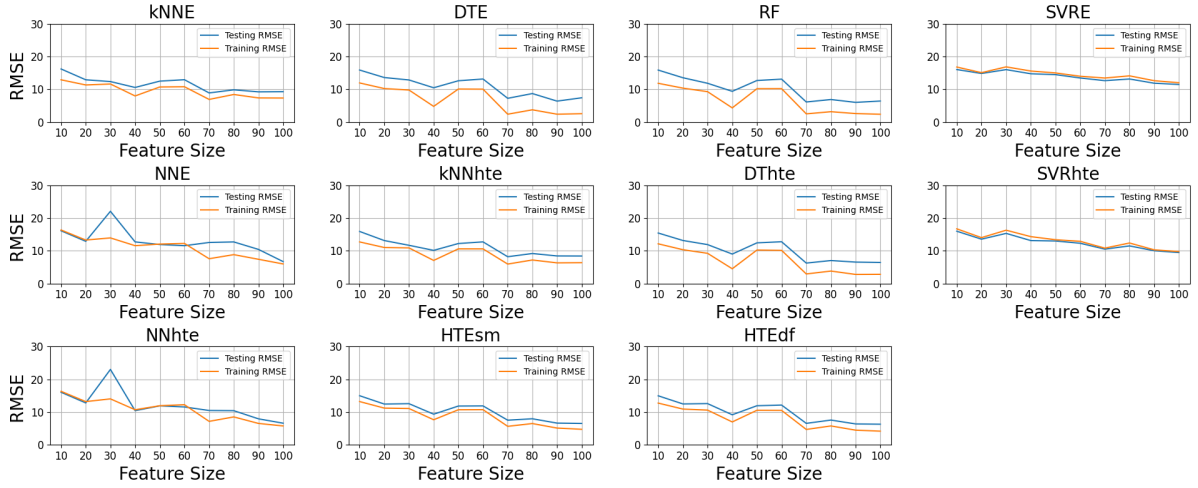

Figure I.6: Ensemble Performance on Feature Subsets of the Concrete Dataset

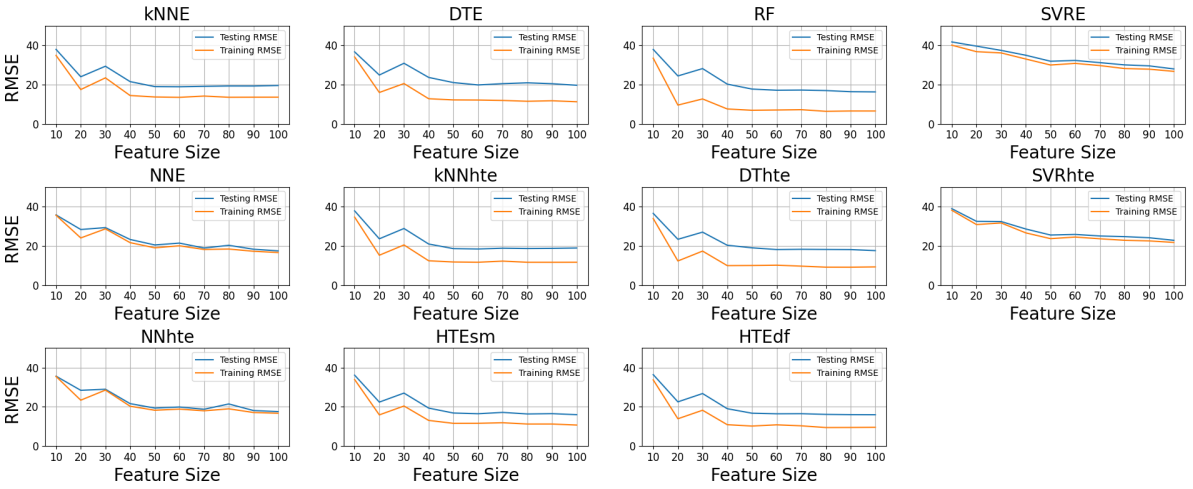

Figure I.7: Ensemble Performance on Feature Subsets of the Air Quality Dataset

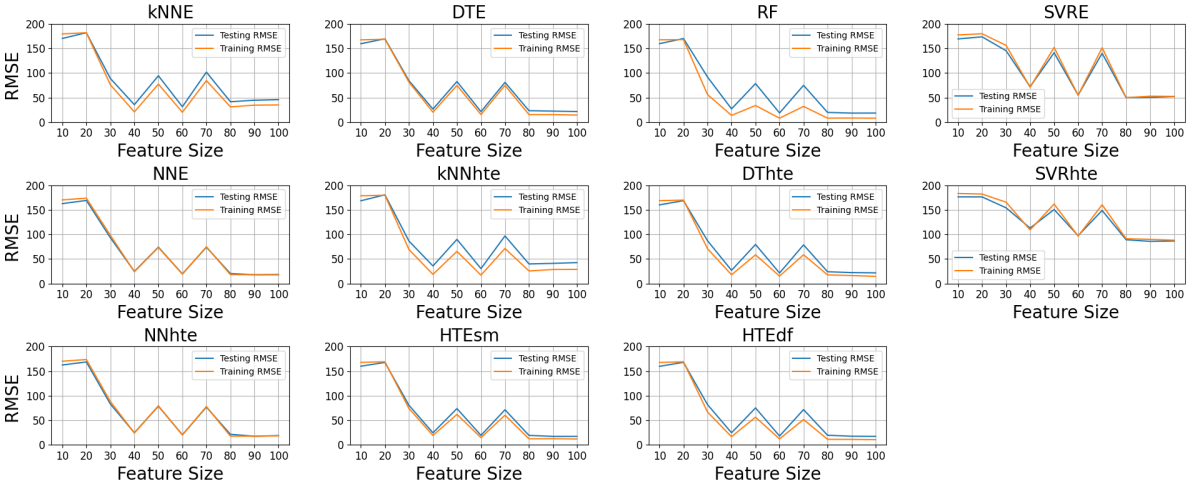

Figure I.8: Ensemble Performance on Feature Subsets of the Bike Sharing Dataset

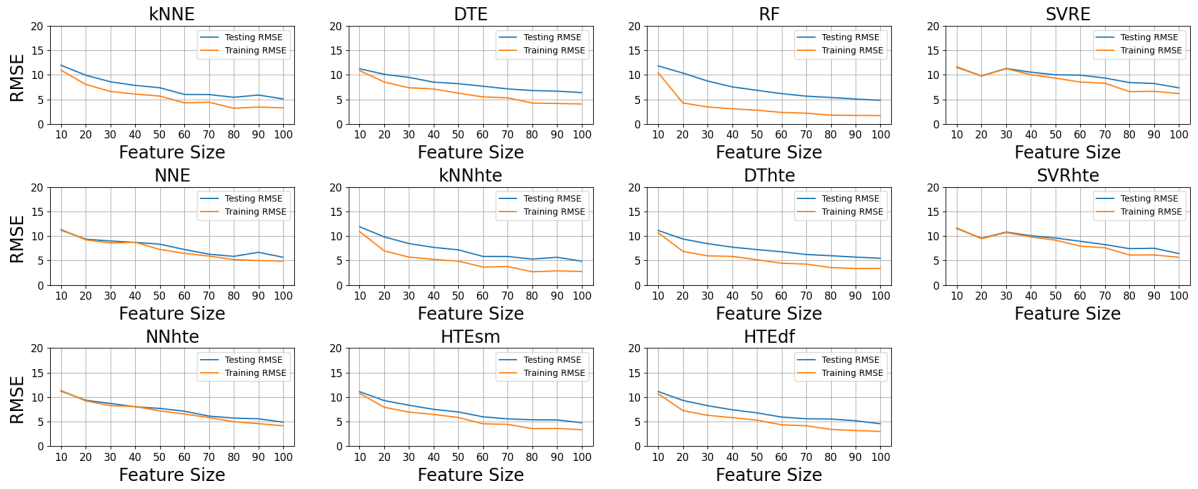

Figure I.9: Ensemble Performance on Feature Subsets of the Gas Turbine Dataset

Yacht Hydrodynamics Dataset

Table I.1: Ensemble Performance on Feature Subsets of the Yacht Hydrodynamics Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	0.834	15.022	1.863	1.562	2.832	6.236	5.839	6.430	6.148	8.752
Training RMSE	1.786	16.407	2.209	2.164	3.824	5.846	5.748	5.085	6.432	8.286
GF	0.218	0.838	0.711	0.521	0.548	1.138	1.032	1.599	0.914	1.116
DTE										
Testing RMSE	0.821	13.266	1.067	0.918	1.268	0.833	0.539	1.312	0.830	0.555
Training RMSE	1.709	15.643	1.494	0.606	0.518	0.518	0.046	0.026	0.518	0.050
GF	0.231	0.719	0.510	2.296	5.999	2.586	139.782	2547.090	2.570	124.783
RF										
Testing RMSE	0.869	13.160	1.040	0.936	0.926	0.885	0.779	0.772	0.816	0.605
Training RMSE	1.722	15.664	1.519	0.910	0.784	0.705	0.509	0.601	0.699	0.589
GF	0.254	0.706	0.469	1.058	1.396	1.574	2.343	1.650	1.361	1.054
SVRE										
Testing RMSE	8.647	12.604	8.942	8.663	8.136	8.100	8.651	8.028	7.699	7.906
Training RMSE	12.147	17.354	11.712	12.054	11.771	11.785	11.569	11.172	11.439	11.077
GF	0.507	0.527	0.583	0.516	0.478	0.472	0.559	0.516	0.453	0.509
NNE										
Testing RMSE	1.640	12.939	1.806	1.601	1.883	2.749	2.787	3.095	2.784	3.516
Training RMSE	2.723	15.698	2.525	2.543	2.861	3.286	3.171	3.270	3.540	3.852
GF	0.363	0.679	0.512	0.396	0.433	0.700	0.772	0.895	0.618	0.833

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
kNNhte										
Testing RMSE	0.799	14.539	1.634	1.309	2.620	4.988	4.996	5.270	5.376	7.520
Training RMSE	1.746	16.386	2.053	2.217	3.271	4.791	4.997	4.549	5.262	6.811
GF	0.209	0.787	0.634	0.349	0.642	1.084	1.000	1.342	1.044	1.219
DThte										
Testing RMSE	0.893	12.409	1.084	0.875	1.063	0.794	0.555	0.921	0.751	0.656
Training RMSE	1.763	16.080	1.544	0.812	0.714	0.682	0.430	0.478	0.704	0.444
GF	0.256	0.596	0.493	1.159	2.213	1.355	1.667	3.712	1.138	2.176
SVRhte										
Testing RMSE	6.311	12.611	6.866	6.483	6.322	6.641	7.019	6.868	6.483	6.834
Training RMSE	8.896	17.338	8.778	9.079	9.074	9.525	9.220	9.240	9.359	9.255
GF	0.503	0.529	0.612	0.510	0.485	0.486	0.580	0.553	0.480	0.545
NNhte										
Testing RMSE	1.411	12.680	1.621	1.353	1.709	2.326	2.453	2.481	2.159	2.859
Training RMSE	2.447	15.816	2.379	2.171	2.282	2.806	2.724	2.573	2.761	2.987
GF	0.333	0.643	0.464	0.388	0.561	0.687	0.811	0.930	0.611	0.916
HTEsm										
Testing RMSE	0.851	13.074	0.783	1.021	1.111	0.824	0.808	0.840	0.703	0.926
Training RMSE	1.760	15.671	1.640	1.514	1.444	0.995	0.726	0.835	0.963	0.845
GF	0.234	0.696	0.228	0.455	0.592	0.686	1.239	1.011	0.533	1.203
HTEdf										
Testing RMSE	0.922	12.945	0.765	1.013	1.092	0.828	0.914	0.877	0.754	0.842
Training RMSE	1.797	15.669	1.682	1.514	1.450	1.019	0.823	0.802	1.039	0.812
GF	0.263	0.683	0.207	0.448	0.568	0.659	1.234	1.195	0.527	1.075

Residential Building Dataset

Table I.2: Ensemble Performance on Feature Subsets of the Residential Building Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	1134.153	540.091	859.934	753.465	772.504	826.705	790.380	773.841	802.849	806.101
Training RMSE	921.455	468.000	756.741	576.256	603.206	678.841	611.322	645.553	648.732	635.210
GF	1.515	1.332	1.291	1.710	1.640	1.483	1.672	1.437	1.532	1.610
DTE										
Testing RMSE	1032.231	307.387	917.450	914.790	326.088	298.106	328.459	315.566	319.963	314.144
Training RMSE	867.198	13.266	89.058	122.994	13.722	11.455	8.869	10.398	8.127	7.927
GF	1.417	536.868	106.126	55.319	564.695	677.204	1371.595	921.101	1550.202	1570.404
RF										
Testing RMSE	1026.980	286.197	723.018	625.803	304.215	244.446	283.004	306.885	277.043	282.167
Training RMSE	873.814	96.648	353.887	220.603	122.417	123.393	116.807	104.812	110.868	126.041
GF	1.381	8.769	4.174	8.047	6.176	3.925	5.870	8.573	6.244	5.012
SVRE										
Testing RMSE	1381.631	1109.382	1191.392	968.794	796.998	774.178	767.075	751.801	753.425	751.679
Training RMSE	1110.253	896.582	940.535	815.651	646.276	636.830	624.886	613.513	614.907	611.687
GF	1.549	1.531	1.605	1.411	1.521	1.478	1.507	1.502	1.501	1.510
NNE										
Testing RMSE	1093.868	393.312	806.447	677.678	396.968	407.978	380.346	383.470	373.760	379.391
Training RMSE	891.652	294.148	496.090	369.096	275.563	286.321	257.280	270.729	270.704	272.376
GF	1.505	1.788	2.643	3.371	2.075	2.030	2.185	2.006	1.906	1.940

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	1140.105	580.989	889.444	739.622	722.406	766.691	735.496	730.962	742.214	752.148
Training RMSE	895.640	424.756	637.498	516.340	520.479	572.880	518.954	534.578	547.893	535.444
GF	1.620	1.871	1.947	2.052	1.926	1.791	2.009	1.870	1.835	1.973
DThte										
Testing RMSE	1095.363	277.812	695.401	707.320	278.836	261.494	279.479	248.707	265.278	283.976
Training RMSE	885.554	55.329	186.260	130.724	53.378	53.791	54.462	49.695	59.625	52.831
GF	1.530	25.211	13.939	29.277	27.289	23.632	26.334	25.047	19.795	28.892
SVRhte										
Testing RMSE	1397.157	1092.101	1150.372	944.582	773.230	757.461	755.540	751.043	746.398	743.073
Training RMSE	1119.585	886.190	903.254	795.284	625.108	621.157	615.493	607.867	607.714	605.883
GF	1.557	1.519	1.622	1.411	1.530	1.487	1.507	1.527	1.508	1.504
NNhte										
Testing RMSE	1107.085	387.726	772.424	651.431	382.890	360.927	331.063	361.228	340.970	353.820
Training RMSE	894.215	252.815	483.016	342.076	233.754	249.126	225.563	227.493	228.974	229.385
GF	1.533	2.352	2.557	3.627	2.683	2.099	2.154	2.521	2.217	2.379
HTEsm										
Testing RMSE	1076.867	226.887	728.188	654.179	209.953	165.040	175.314	208.516	184.895	178.968
Training RMSE	930.836	141.815	631.066	411.227	110.014	112.917	90.054	101.035	115.799	99.659
GF	1.338	2.560	1.331	2.531	3.642	2.136	3.790	4.259	2.549	3.225
HTEdf										
Testing RMSE	1068.634	216.121	760.395	642.149	206.925	169.719	181.175	205.307	192.675	174.688
Training RMSE	930.820	134.352	653.758	400.207	109.342	118.342	97.469	109.045	126.092	99.931
GF	1.318	2.588	1.353	2.575	3.581	2.057	3.455	3.545	2.335	3.056

Student Performance Dataset

Table I.3: Ensemble Performance on Feature Subsets of the Student Performance Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	5.139	2.975	4.758	3.744	3.958	3.814	4.159	4.105	3.949	3.740
Training RMSE	4.813	2.589	3.925	3.326	3.128	3.260	3.598	3.511	3.501	3.230
GF	1.140	1.320	1.469	1.267	1.601	1.369	1.336	1.367	1.272	1.340
DTE										
Testing RMSE	4.662	3.456	6.179	3.933	3.292	2.334	2.257	2.720	3.004	2.411
Training RMSE	4.409	1.587	2.033	0.403	0.622	0.110	0.080	0.319	0.307	0.039
GF	1.118	4.741	9.234	95.398	28.006	452.675	791.312	72.552	95.454	3826.417
RF										
Testing RMSE	4.648	2.887	5.239	2.878	2.549	1.873	1.979	2.324	2.471	1.797
Training RMSE	4.420	1.869	2.466	1.390	1.016	0.553	0.637	1.052	1.052	0.645
GF	1.106	2.385	4.512	4.289	6.290	11.451	9.642	4.881	5.521	7.754
SVRE										
Testing RMSE	4.566	2.663	4.584	3.164	3.247	2.789	3.272	3.541	3.524	3.160
Training RMSE	4.478	2.852	4.025	2.636	2.296	1.844	1.875	1.901	1.907	1.904
GF	1.040	0.872	1.297	1.441	2.001	2.288	3.046	3.471	3.415	2.755
NNE										
Testing RMSE	4.650	3.019	5.194	3.014	2.873	2.129	2.514	2.925	3.083	2.580
Training RMSE	4.429	2.202	3.083	1.456	1.018	0.578	0.656	0.732	0.735	0.501
GF	1.102	1.879	2.838	4.286	7.957	13.571	14.672	15.950	17.582	26.562

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	4.866	2.952	4.793	3.770	3.780	3.816	3.980	4.006	3.923	3.724
Training RMSE	4.574	2.377	3.440	2.804	2.678	2.589	3.089	2.912	2.981	2.767
GF	1.131	1.542	1.941	1.807	1.992	2.172	1.660	1.892	1.731	1.811
DThte										
Testing RMSE	4.612	2.843	5.323	2.928	2.623	1.880	1.967	2.401	2.415	2.032
Training RMSE	4.420	1.762	1.984	0.910	0.684	0.450	0.582	0.682	0.650	0.453
GF	1.088	2.605	7.197	10.346	14.696	17.438	11.415	12.405	13.825	20.098
SVRhte										
Testing RMSE	4.545	2.982	4.636	3.438	3.448	2.911	3.210	3.491	3.493	3.105
Training RMSE	4.520	3.189	4.018	2.782	2.549	2.158	2.126	2.132	2.132	2.067
GF	1.011	0.874	1.331	1.528	1.830	1.820	2.279	2.680	2.684	2.257
NNhte										
Testing RMSE	4.645	2.932	5.208	3.110	2.950	2.214	2.579	2.798	3.191	2.664
Training RMSE	4.434	2.278	3.033	1.648	1.061	0.667	0.550	0.767	0.748	0.471
GF	1.097	1.658	2.949	3.561	7.722	11.021	21.956	13.299	18.210	32.030
HTEsm										
Testing RMSE	4.666	2.800	5.034	2.901	2.599	2.237	2.474	2.704	2.671	2.401
Training RMSE	4.436	2.036	2.702	1.627	1.356	1.138	1.284	1.284	1.282	1.130
GF	1.107	1.892	3.473	3.180	3.673	3.862	3.715	4.438	4.341	4.511
HTEdf										
Testing RMSE	4.636	2.781	4.892	2.919	2.564	2.157	2.397	2.489	2.636	2.248
Training RMSE	4.427	1.976	2.673	1.552	1.352	1.065	1.119	1.173	1.240	1.053
GF	1.097	1.980	3.349	3.538	3.599	4.099	4.588	4.504	4.517	4.558

Real Estate Dataset

Table I.4: Ensemble Performance on Feature Subsets of the Real Estate Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	14.281	12.717	7.966	8.977	10.224	9.053	9.067	8.104	8.369	8.034
Training RMSE	14.293	10.980	6.899	7.794	8.726	8.088	8.115	7.445	8.338	7.812
GF	0.998	1.341	1.333	1.327	1.373	1.253	1.248	1.185	1.007	1.058
DTE										
Testing RMSE	13.340	14.623	8.806	9.430	11.318	7.930	8.162	8.026	8.682	7.918
Training RMSE	13.204	7.596	3.208	4.232	2.164	2.465	2.481	1.757	1.355	1.796
GF	1.021	3.706	7.534	4.965	27.359	10.346	10.824	20.866	41.073	19.440
RF										
Testing RMSE	13.326	13.848	7.578	9.151	11.435	7.046	7.304	6.608	7.249	6.668
Training RMSE	13.220	7.919	4.014	4.414	5.090	3.889	3.777	3.532	3.560	3.545
GF	1.016	3.058	3.564	4.298	5.047	3.283	3.740	3.500	4.147	3.537
SVRE										
Testing RMSE	13.222	13.185	12.636	11.882	11.795	11.236	11.339	10.889	10.609	10.073
Training RMSE	13.624	13.688	12.838	12.254	12.322	11.424	11.405	10.998	11.099	10.636
GF	0.942	0.928	0.969	0.940	0.916	0.967	0.988	0.980	0.914	0.897
NNE										
Testing RMSE	12.987	11.481	8.417	10.045	9.591	8.034	8.095	7.272	7.556	7.275
Training RMSE	13.510	12.596	8.651	10.365	10.263	8.541	8.561	7.450	8.337	7.812
GF	0.924	0.831	0.947	0.939	0.873	0.885	0.894	0.953	0.821	0.867

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	14.879	12.468	7.980	8.087	9.969	8.726	8.760	7.641	8.075	7.780
Training RMSE	14.663	9.960	6.196	6.554	7.387	6.832	6.852	6.248	6.982	6.563
GF	1.030	1.567	1.659	1.522	1.821	1.631	1.635	1.495	1.338	1.405
DThte										
Testing RMSE	13.305	13.155	7.303	8.164	9.686	7.124	7.185	6.875	6.888	7.069
Training RMSE	13.229	7.141	3.697	4.227	2.860	2.958	2.782	2.267	2.249	2.406
GF	1.012	3.394	3.902	3.731	11.468	5.801	6.672	9.202	9.380	8.632
SVRhte										
Testing RMSE	13.215	12.990	10.707	10.530	10.704	9.671	9.764	9.074	9.060	8.427
Training RMSE	13.614	13.521	11.354	11.270	11.224	10.318	10.351	9.542	10.067	9.457
GF	0.942	0.923	0.889	0.873	0.909	0.879	0.890	0.904	0.810	0.794
NNhte										
Testing RMSE	12.963	11.486	8.445	9.693	9.609	8.070	8.087	7.170	7.578	7.236
Training RMSE	13.493	12.592	8.679	10.189	10.240	8.527	8.577	7.372	8.361	7.700
GF	0.923	0.832	0.947	0.905	0.881	0.896	0.889	0.946	0.821	0.883
HTEsm										
Testing RMSE	13.141	12.109	7.699	8.124	9.395	7.687	7.690	6.852	7.163	6.948
Training RMSE	13.264	9.767	6.185	6.965	6.823	6.182	6.132	5.347	5.982	5.632
GF	0.981	1.537	1.550	1.361	1.896	1.546	1.573	1.642	1.434	1.522
HTEdf										
Testing RMSE	13.154	12.402	7.452	7.781	9.435	7.619	7.656	6.787	7.100	6.945
Training RMSE	13.281	9.298	5.820	6.354	6.272	5.573	5.656	5.107	5.524	5.201
GF	0.981	1.779	1.639	1.500	2.263	1.869	1.832	1.767	1.652	1.783

Energy Efficiency Dataset

Table I.5: Ensemble Performance on Feature Subsets of the Energy Efficiency Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	4.739	3.256	1.304	2.848	1.855	2.847	2.010	1.141	1.132	1.096
Training RMSE	4.504	3.184	1.243	2.868	1.700	2.821	1.749	1.130	1.099	1.076
GF	1.107	1.046	1.100	0.986	1.191	1.019	1.320	1.019	1.062	1.039
DTE										
Testing RMSE	4.146	3.612	1.447	2.863	2.055	2.901	1.991	1.398	1.393	1.421
Training RMSE	4.073	3.464	1.187	2.869	1.964	2.896	1.915	1.100	1.128	1.121
GF	1.036	1.087	1.486	0.996	1.095	1.004	1.081	1.616	1.525	1.608
RF										
Testing RMSE	4.000	2.760	1.402	2.759	1.736	2.759	1.740	1.200	1.326	1.222
Training RMSE	3.993	2.643	0.800	2.641	1.557	2.642	1.562	0.637	0.619	0.616
GF	1.004	1.090	3.071	1.092	1.242	1.090	1.241	3.543	4.591	3.932
SVRE										
Testing RMSE	5.407	9.111	3.887	4.429	4.020	4.234	3.862	3.279	3.265	3.301
Training RMSE	5.188	8.680	3.676	4.306	3.769	4.131	3.637	3.088	3.059	3.093
GF	1.087	1.102	1.118	1.058	1.138	1.051	1.127	1.128	1.140	1.139
NNE										
Testing RMSE	4.474	7.213	2.148	3.649	3.032	3.658	3.141	2.047	2.063	2.032
Training RMSE	4.343	6.811	2.000	3.756	3.040	3.764	3.097	1.951	1.994	1.928
GF	1.061	1.122	1.154	0.944	0.994	0.944	1.029	1.100	1.070	1.111

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	4.826	3.060	1.268	2.873	1.876	2.838	1.933	1.173	1.134	1.111
Training RMSE	4.580	2.971	1.073	2.890	1.742	2.847	1.705	0.969	0.936	0.944
GF	1.110	1.061	1.395	0.988	1.159	0.994	1.285	1.466	1.468	1.386
DThte										
Testing RMSE	4.233	4.164	1.411	2.877	2.037	2.898	1.937	1.373	1.371	1.346
Training RMSE	4.125	3.937	1.099	2.872	1.912	2.899	1.824	1.006	1.013	0.992
GF	1.053	1.119	1.649	1.004	1.135	0.999	1.128	1.864	1.831	1.839
SVRhte										
Testing RMSE	4.381	8.351	2.114	3.593	3.106	3.573	3.066	2.224	2.285	2.334
Training RMSE	4.281	7.894	2.048	3.637	3.016	3.615	2.967	2.071	2.139	2.159
GF	1.047	1.119	1.065	0.976	1.060	0.977	1.068	1.153	1.141	1.168
NNhte										
Testing RMSE	4.488	5.149	2.111	3.657	3.059	3.643	3.050	2.055	2.063	1.996
Training RMSE	4.350	4.920	1.989	3.756	3.064	3.706	3.036	1.954	1.972	1.929
GF	1.064	1.095	1.127	0.948	0.997	0.966	1.009	1.106	1.094	1.071
HTEsm										
Testing RMSE	3.993	2.756	1.503	2.756	1.748	2.756	1.748	1.519	1.470	1.519
Training RMSE	3.992	2.639	0.621	2.639	1.555	2.639	1.555	0.455	0.451	0.451
GF	1.001	1.090	5.851	1.090	1.264	1.090	1.264	11.165	10.628	11.361
HTEdf										
Testing RMSE	4.041	2.802	1.434	2.752	1.828	2.755	1.834	1.259	1.364	1.232
Training RMSE	4.007	2.690	0.729	2.656	1.629	2.657	1.642	0.608	0.596	0.652
GF	1.017	1.085	3.863	1.073	1.259	1.075	1.248	4.288	5.242	3.567

Concrete Dataset

Table I.6: Ensemble Performance on Feature Subsets of the Concrete Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	16.215	12.912	12.371	10.572	12.503	12.939	8.903	9.848	9.233	9.284
Training RMSE	12.907	11.335	11.630	7.978	10.717	10.791	6.922	8.428	7.378	7.347
GF	1.578	1.298	1.131	1.756	1.361	1.438	1.654	1.365	1.566	1.597
DTE										
Testing RMSE	15.887	13.599	12.860	10.480	12.617	13.154	7.258	8.712	6.402	7.420
Training RMSE	11.948	10.237	9.788	4.793	10.093	10.052	2.407	3.753	2.409	2.564
GF	1.768	1.765	1.726	4.781	1.562	1.713	9.093	5.389	7.061	8.375
RF										
Testing RMSE	15.876	13.542	11.847	9.389	12.693	13.116	6.141	6.890	6.019	6.429
Training RMSE	11.841	10.370	9.283	4.339	10.189	10.209	2.517	3.172	2.604	2.396
GF	1.797	1.705	1.629	4.683	1.552	1.651	5.954	4.717	5.343	7.202
SVRE										
Testing RMSE	16.038	14.843	16.017	14.753	14.491	13.465	12.646	13.157	11.867	11.486
Training RMSE	16.805	15.091	16.857	15.588	15.004	13.993	13.450	14.120	12.634	12.019
GF	0.911	0.967	0.903	0.896	0.933	0.926	0.884	0.868	0.882	0.913
NNE										
Testing RMSE	16.079	12.863	22.028	12.687	11.881	11.526	12.511	12.677	10.396	6.674
Training RMSE	16.325	13.252	13.904	11.531	12.021	12.220	7.548	8.797	7.379	5.954
GF	0.970	0.942	2.510	1.211	0.977	0.890	2.747	2.076	1.985	1.256

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	15.869	13.093	11.648	10.099	12.190	12.692	8.159	9.141	8.400	8.376
Training RMSE	12.675	10.972	10.852	7.026	10.561	10.557	5.917	7.160	6.285	6.341
GF	1.568	1.424	1.152	2.066	1.332	1.445	1.901	1.630	1.786	1.745
DThte										
Testing RMSE	15.394	13.117	11.861	8.975	12.392	12.736	6.220	6.995	6.522	6.408
Training RMSE	12.098	10.302	9.200	4.487	10.195	10.105	2.902	3.784	2.755	2.786
GF	1.619	1.621	1.662	4.001	1.478	1.589	4.592	3.417	5.603	5.289
SVRhte										
Testing RMSE	15.946	13.519	15.285	13.089	12.963	12.273	10.476	11.465	9.988	9.440
Training RMSE	16.672	13.991	16.257	14.292	13.350	12.866	10.792	12.347	10.263	9.741
GF	0.915	0.934	0.884	0.839	0.943	0.910	0.942	0.862	0.947	0.939
NNhte										
Testing RMSE	16.043	12.790	23.014	10.432	11.891	11.557	10.479	10.423	7.933	6.602
Training RMSE	16.333	13.193	14.027	10.712	11.933	12.245	7.159	8.519	6.529	5.760
GF	0.965	0.940	2.692	0.949	0.993	0.891	2.143	1.497	1.476	1.314
HTEsm										
Testing RMSE	14.986	12.441	12.561	9.350	11.836	11.881	7.520	7.954	6.629	6.522
Training RMSE	13.201	11.204	11.083	7.653	10.702	10.731	5.644	6.494	5.113	4.740
GF	1.289	1.233	1.285	1.493	1.223	1.226	1.776	1.500	1.681	1.893
HTEdf										
Testing RMSE	14.965	12.479	12.581	9.171	11.930	12.136	6.551	7.555	6.377	6.297
Training RMSE	12.737	10.914	10.595	6.982	10.544	10.506	4.692	5.761	4.473	4.173
GF	1.380	1.307	1.410	1.725	1.280	1.334	1.949	1.720	2.033	2.278

Parkinsons Disease Dataset

Table I.7: Ensemble Performance on Feature Subsets of the Parkinsons Disease Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	6.977	11.057	2.619	1.169	1.013	4.356	1.616	0.753	0.953	0.982
Training RMSE	5.048	9.454	2.113	0.707	0.798	3.396	1.142	0.561	0.684	0.691
GF	1.910	1.368	1.536	2.740	1.611	1.646	2.003	1.802	1.942	2.023
DTE										
Testing RMSE	7.548	11.006	2.312	1.272	1.139	3.529	1.224	0.877	0.930	0.917
Training RMSE	6.068	8.985	2.108	1.252	1.100	2.960	1.064	0.783	0.815	0.855
GF	1.547	1.500	1.204	1.032	1.072	1.422	1.325	1.254	1.301	1.151
RF										
Testing RMSE	7.524	11.334	1.536	0.507	0.656	2.001	0.758	0.571	0.523	0.465
Training RMSE	3.057	4.842	0.736	0.217	0.254	0.542	0.335	0.220	0.208	0.220
GF	6.056	5.480	4.356	5.430	6.661	13.611	5.110	6.759	6.320	4.485
SVRE										
Testing RMSE	10.195	10.510	4.837	4.717	3.661	9.197	3.731	3.512	3.392	3.445
Training RMSE	10.566	10.944	5.040	4.943	3.827	9.566	3.832	3.657	3.535	3.530
GF	0.931	0.922	0.921	0.910	0.915	0.924	0.948	0.922	0.921	0.953
NNE										
Testing RMSE	10.214	16.454	3.303	2.959	2.242	8.640	2.229	2.050	2.137	2.467
Training RMSE	10.015	10.632	3.398	3.075	2.199	4.100	2.007	1.811	1.819	2.255
GF	1.040	2.395	0.945	0.926	1.040	4.441	1.233	1.282	1.380	1.197

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	6.889	10.919	2.484	1.014	0.855	4.009	1.488	0.715	0.873	0.826
Training RMSE	4.408	8.120	1.759	0.580	0.601	2.730	0.958	0.458	0.521	0.538
GF	2.443	1.808	1.994	3.055	2.023	2.156	2.412	2.431	2.808	2.356
DThte										
Testing RMSE	7.319	10.646	2.221	1.306	1.225	3.408	1.302	0.908	0.996	0.787
Training RMSE	5.572	8.138	1.933	1.298	1.182	2.902	1.111	0.828	0.872	0.683
GF	1.726	1.712	1.320	1.012	1.075	1.379	1.374	1.203	1.306	1.327
SVRhte										
Testing RMSE	10.171	10.542	3.666	3.651	2.749	8.850	3.107	2.704	2.565	2.536
Training RMSE	10.364	10.906	3.815	3.810	2.881	9.191	3.024	2.718	2.594	2.586
GF	0.963	0.934	0.924	0.918	0.911	0.927	1.055	0.990	0.978	0.962
NNhte										
Testing RMSE	10.422	12.949	3.268	2.787	2.011	6.744	2.226	1.485	1.664	2.094
Training RMSE	9.947	10.644	3.388	2.841	1.908	3.730	2.015	1.181	1.437	1.742
GF	1.098	1.480	0.931	0.962	1.111	3.269	1.221	1.581	1.341	1.445
HTEsm										
Testing RMSE	6.744	12.077	2.067	0.767	0.902	2.276	0.955	0.623	0.729	0.538
Training RMSE	4.931	9.307	1.510	0.548	0.625	1.032	0.588	0.414	0.413	0.422
GF	1.871	1.684	1.873	1.963	2.081	4.868	2.645	2.261	3.124	1.624
HTEdf										
Testing RMSE	6.832	10.671	1.863	1.104	1.089	2.799	1.120	0.951	0.940	0.890
Training RMSE	4.092	7.127	1.441	1.070	0.982	2.156	0.994	0.909	0.869	0.839
GF	2.788	2.242	1.671	1.065	1.231	1.685	1.270	1.094	1.172	1.127

Air Quality Dataset

Table I.8: Ensemble Performance on Feature Subsets of the Air Quality Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	37.959	24.018	29.381	21.541	18.993	18.916	19.127	19.308	19.300	19.557
Training RMSE	34.861	17.492	23.457	14.454	13.713	13.525	14.174	13.563	13.604	13.617
GF	1.186	1.885	1.569	2.221	1.918	1.956	1.821	2.027	2.012	2.063
DTE										
Testing RMSE	36.675	24.895	30.892	23.674	21.054	19.848	20.494	20.937	20.463	19.684
Training RMSE	34.091	16.004	20.556	12.814	12.215	12.151	11.938	11.530	11.785	11.277
GF	1.157	2.420	2.259	3.413	2.971	2.668	2.947	3.297	3.015	3.047
RF										
Testing RMSE	37.936	24.423	28.180	20.266	17.749	17.146	17.233	16.979	16.401	16.299
Training RMSE	33.478	9.575	12.715	7.605	6.933	7.072	7.232	6.407	6.575	6.569
GF	1.284	6.506	4.912	7.102	6.554	5.878	5.678	7.022	6.223	6.156
SVRE										
Testing RMSE	41.756	39.616	37.452	34.970	31.922	32.321	31.214	30.063	29.562	27.998
Training RMSE	40.119	36.795	36.158	33.045	29.995	30.824	29.718	28.194	27.882	26.763
GF	1.083	1.159	1.073	1.120	1.133	1.100	1.103	1.137	1.124	1.094
NNE										
Testing RMSE	35.708	28.278	29.241	23.204	20.404	21.345	18.838	20.222	18.232	17.389
Training RMSE	35.520	23.972	28.613	21.583	18.948	19.975	18.102	18.326	17.187	16.542
GF	1.011	1.391	1.044	1.156	1.160	1.142	1.083	1.218	1.125	1.105

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	37.765	23.535	28.793	20.879	18.537	18.347	18.746	18.592	18.668	18.846
Training RMSE	34.560	15.169	20.406	12.342	11.725	11.593	12.170	11.583	11.569	11.593
GF	1.194	2.407	1.991	2.862	2.499	2.504	2.373	2.576	2.604	2.643
DThte										
Testing RMSE	36.492	23.281	26.940	20.254	18.944	18.073	18.199	18.117	18.047	17.515
Training RMSE	33.872	12.241	17.277	9.871	9.938	10.089	9.614	9.086	9.069	9.219
GF	1.161	3.617	2.431	4.210	3.633	3.209	3.583	3.976	3.960	3.610
SVRhte										
Testing RMSE	38.895	32.438	32.331	28.531	25.465	25.772	24.974	24.648	24.078	22.766
Training RMSE	38.070	30.801	31.608	26.549	23.615	24.434	23.549	22.778	22.465	21.689
GF	1.044	1.109	1.046	1.155	1.163	1.113	1.125	1.171	1.149	1.102
NNhte										
Testing RMSE	35.621	28.430	28.985	21.646	19.389	19.921	18.784	21.480	18.127	17.586
Training RMSE	35.481	23.403	28.541	20.332	18.269	18.861	17.983	18.998	17.124	16.740
GF	1.008	1.476	1.031	1.133	1.126	1.116	1.091	1.278	1.121	1.104
HTEsm										
Testing RMSE	36.135	22.426	27.020	19.367	16.860	16.493	17.176	16.379	16.531	16.020
Training RMSE	33.899	15.912	20.439	13.083	11.605	11.627	11.909	11.250	11.253	10.731
GF	1.136	1.986	1.748	2.191	2.111	2.012	2.080	2.120	2.158	2.229
HTEdf										
Testing RMSE	36.428	22.548	26.789	19.061	16.765	16.463	16.510	16.168	16.020	15.987
Training RMSE	33.763	13.936	18.275	10.915	10.215	10.845	10.337	9.457	9.497	9.580
GF	1.164	2.618	2.149	3.050	2.694	2.305	2.551	2.923	2.846	2.785

Bike Sharing Dataset

Table I.9: Ensemble Performance on Feature Subsets of the Bike Sharing Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	170.247	181.874	88.777	35.457	94.332	31.292	101.679	41.458	44.501	45.708
Training RMSE	179.366	181.860	75.998	20.747	77.297	20.055	84.502	30.963	34.494	34.897
GF	0.901	1.000	1.365	2.921	1.489	2.435	1.448	1.793	1.664	1.716
DTE										
Testing RMSE	159.503	169.474	84.366	26.494	82.581	21.285	80.870	23.210	22.331	21.550
Training RMSE	167.387	168.642	81.467	20.171	74.648	15.337	74.723	15.275	15.251	14.308
GF	0.908	1.010	1.072	1.725	1.224	1.926	1.171	2.309	2.144	2.268
RF										
Testing RMSE	159.589	170.072	91.680	26.920	78.332	18.553	74.815	19.661	18.330	18.394
Training RMSE	167.418	167.818	55.962	13.690	33.722	8.005	31.867	8.066	8.261	7.930
GF	0.909	1.027	2.684	3.867	5.396	5.371	5.512	5.941	4.924	5.380
SVRE										
Testing RMSE	169.227	173.545	145.278	72.014	141.126	54.832	140.125	49.652	50.298	51.989
Training RMSE	177.466	179.795	156.105	71.025	151.949	54.529	151.424	49.697	52.698	52.218
GF	0.909	0.932	0.866	1.028	0.863	1.011	0.856	0.998	0.911	0.991
NNE										
Testing RMSE	162.499	168.949	93.116	24.479	73.978	19.391	73.752	20.182	17.673	18.179
Training RMSE	170.034	173.537	98.005	23.970	73.309	19.077	74.624	17.897	17.840	17.490
GF	0.913	0.948	0.903	1.043	1.018	1.033	0.977	1.272	0.981	1.080

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	168.327	180.380	86.316	35.633	89.725	30.225	96.706	39.694	40.908	42.400
Training RMSE	178.397	179.549	69.078	18.508	65.276	16.902	71.214	25.435	28.386	28.658
GF	0.890	1.009	1.561	3.707	1.889	3.198	1.844	2.436	2.077	2.189
DThte										
Testing RMSE	159.930	168.526	86.926	26.722	79.339	21.302	78.514	23.831	22.108	21.542
Training RMSE	168.514	169.657	70.780	17.627	58.276	15.012	58.343	17.461	16.152	14.374
GF	0.901	0.987	1.508	2.298	1.853	2.014	1.811	1.863	1.874	2.246
SVRhte										
Testing RMSE	176.211	176.130	153.912	112.723	150.452	97.222	148.482	88.903	85.716	86.378
Training RMSE	183.113	181.908	165.407	108.977	161.663	96.407	160.111	90.986	89.800	87.947
GF	0.926	0.937	0.866	1.070	0.866	1.017	0.860	0.955	0.911	0.965
NNhte										
Testing RMSE	162.442	168.668	83.022	24.523	78.617	20.409	76.790	21.418	17.447	18.819
Training RMSE	170.108	173.549	88.172	24.001	79.462	20.074	77.904	17.600	17.489	18.150
GF	0.912	0.945	0.887	1.044	0.979	1.034	0.972	1.481	0.995	1.075
HTEsm										
Testing RMSE	159.782	167.834	81.879	24.532	75.105	17.885	71.665	19.424	17.531	17.230
Training RMSE	167.680	168.889	66.728	16.503	56.239	12.017	51.429	11.197	11.066	10.519
GF	0.908	0.988	1.506	2.210	1.783	2.215	1.942	3.009	2.510	2.683
HTEdf										
Testing RMSE	160.032	167.621	80.565	24.427	73.788	19.273	71.346	19.134	17.227	17.238
Training RMSE	167.823	169.039	73.996	19.047	61.937	14.661	60.336	12.261	12.426	11.851
GF	0.909	0.983	1.185	1.645	1.419	1.728	1.398	2.435	1.922	2.116

Gas Turbine Dataset

Table I.10: Ensemble Performance on Feature Subsets of the Gas Turbine Dataset

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNE**										
Testing RMSE	11.944	9.921	8.580	7.847	7.371	6.002	5.983	5.422	5.883	5.086
Training RMSE	10.947	8.032	6.612	6.071	5.668	4.286	4.412	3.181	3.433	3.268
GF	1.190	1.526	1.684	1.671	1.691	1.961	1.839	2.905	2.937	2.423
DTE										
Testing RMSE	11.241	10.108	9.476	8.511	8.195	7.677	7.131	6.795	6.669	6.380
Training RMSE	10.846	8.532	7.367	7.118	6.258	5.502	5.304	4.250	4.154	4.057
GF	1.074	1.403	1.654	1.429	1.715	1.947	1.807	2.556	2.578	2.473
RF										
Testing RMSE	11.826	10.351	8.733	7.538	6.848	6.163	5.657	5.378	5.067	4.814
Training RMSE	10.412	4.269	3.446	3.081	2.791	2.332	2.180	1.776	1.726	1.695
GF	1.290	5.880	6.421	5.985	6.019	6.981	6.734	9.169	8.623	8.069
SVRE										
Testing RMSE	11.529	9.786	11.290	10.541	10.001	9.945	9.345	8.425	8.228	7.356
Training RMSE	11.648	9.741	11.247	10.031	9.325	8.526	8.295	6.576	6.641	6.177
GF	0.980	1.009	1.008	1.104	1.150	1.361	1.269	1.642	1.535	1.418
NNE										
Testing RMSE	11.189	9.322	8.975	8.687	8.320	7.233	6.275	5.836	6.658	5.655
Training RMSE	11.316	9.208	8.558	8.719	7.251	6.442	5.887	5.178	4.980	4.860
GF	0.978	1.025	1.100	0.993	1.316	1.261	1.136	1.270	1.788	1.354

Measure	Feature Subsets %									
	10	20	30	40	50	60	70	80	90	100
***k*NNhte**										
Testing RMSE	11.870	9.801	8.442	7.667	7.168	5.825	5.811	5.278	5.645	4.857
Training RMSE	10.887	6.946	5.694	5.209	4.865	3.658	3.758	2.692	2.880	2.742
GF	1.189	1.991	2.198	2.167	2.171	2.536	2.390	3.844	3.841	3.138
DThte										
Testing RMSE	11.121	9.371	8.426	7.716	7.212	6.775	6.222	5.965	5.683	5.451
Training RMSE	10.644	6.866	5.939	5.835	5.148	4.435	4.260	3.543	3.360	3.378
GF	1.092	1.863	2.013	1.748	1.963	2.333	2.134	2.835	2.862	2.604
SVRhte										
Testing RMSE	11.518	9.512	10.787	10.035	9.580	8.912	8.236	7.408	7.490	6.410
Training RMSE	11.628	9.435	10.698	9.802	9.136	7.916	7.593	6.110	6.124	5.649
GF	0.981	1.016	1.017	1.048	1.099	1.268	1.177	1.470	1.496	1.287
NNhte										
Testing RMSE	11.212	9.331	8.701	8.008	7.683	7.112	6.118	5.708	5.560	4.889
Training RMSE	11.331	9.206	8.251	8.069	7.185	6.540	5.815	4.978	4.573	4.148
GF	0.979	1.027	1.112	0.985	1.144	1.182	1.107	1.315	1.478	1.389
HTEsm										
Testing RMSE	11.084	9.275	8.329	7.485	6.962	5.972	5.549	5.366	5.339	4.764
Training RMSE	10.733	7.915	6.928	6.461	5.832	4.554	4.427	3.559	3.602	3.343
GF	1.066	1.373	1.445	1.342	1.425	1.720	1.571	2.273	2.198	2.031
HTEdf										
Testing RMSE	11.128	9.314	8.239	7.405	6.775	5.923	5.568	5.521	5.163	4.558
Training RMSE	10.656	7.237	6.280	5.804	5.292	4.339	4.147	3.403	3.175	3.004
GF	1.090	1.656	1.721	1.628	1.639	1.863	1.803	2.632	2.644	2.302