Process Monitoring with Restricted Boltzmann Machines

by

John Matali Moody

Thesis presented in partial fulfillment
of the requirements for the Degree

of

MASTER OF SCIENCE IN ENGINEERING
(EXTRACTIVE METALLURGICAL ENGINEERING)

in the Faculty of Engineering
at Stellenbosch University

Supervisor

Prof C. Aldrich

Co-Supervisor

Dr C. Dorfling

April 2014

DECLARATION

By submitting this thesis 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.

John Moody 19/02/2014

…… ……

Signature Date

ABSTRACT

Process monitoring and fault diagnosis are used to detect abnormal events in processes. The early detection of such events or faults is crucial to continuous process improvement. Although principal component analysis and partial least squares are widely used for process monitoring and fault diagnosis in the metallurgical industries, these models are linear in principle; nonlinear approaches should provide more compact and informative models. The use of auto associative neural networks or auto encoders provide a principled approach for process monitoring. However, until very recently, these multiple layer neural networks have been difficult to train and have therefore not been used to any significant extent in process monitoring.

With newly proposed algorithms based on the pre-training of the layers of the neural networks, it is now possible to train neural networks with very complex structures, i.e. deep neural networks. These neural networks can be used as auto encoders to extract features from high dimensional data. In this study, the application of deep auto encoders in the form of Restricted Boltzmann machines (RBM) to the extraction of features from process data is considered. These networks have mostly been used for data visualization to date and have not been applied in the context of fault diagnosis or process monitoring as yet. The objective of this investigation is therefore to assess the feasibility of using Restricted Boltzmann machines in various fault detection schemes. The use of RBM in process monitoring schemes will be discussed, together with the application of these models in automated control frameworks.

Keywords: RBM, auto encoders, dimensionality reduction, process monitoring

OPSOMMING

Prosesmonitering en fout diagnose word gebruik om abnormale gebeure in prosesse op te spoor. Die vroeë opsporing van sulke gebeure of foute is noodsaaklik vir deurlopende verbetering van prosesse. Alhoewel hoofkomponent-analise en parsiële kleinste kwadrate wyd gebruik word vir prosesmonitering en fout diagnose in die metallurgiese industrieë, is hierdie modelle lineêr in beginsel; nie-lineêre benaderings behoort meer kompakte en insiggewende modelle te voorsien. Die gebruik van outo-assosiatiewe neurale netwerke of outokodeerders bied 'n beginsel gebaseerder benadering om dit te bereik. Hierdie veelvoudige laag neurale netwerke was egter tot onlangs moeilik om op te lei en is dus nie tot 'n beduidende mate in die prosesmonitering gebruik nie.

Nuwe, voorgestelde algoritmes, gebaseer op voorafopleiding van die lae van die neurale netwerke, maak dit nou moontlik om neurale netwerke met baie ingewikkelde strukture, d.w.s. diep neurale netwerke, op te lei. Hierdie neurale netwerke kan gebruik word as outokodeerders om kenmerke van hoë-dimensionele data te onttrek. In hierdie studie word die toepassing van diep outokodeerders in die vorm van Beperkte Boltzmann Masjiene vir die onttrekking van kenmerke van proses data oorweeg. Tot dusver is hierdie netwerke meestal vir data visualisering gebruik en dit is nog nie toegepas in die konteks van fout diagnose of prosesmonitering nie. Die doel van hierdie ondersoek is dus om die haalbaarheid van die gebruik van Beperkte Boltzmann Masjiene in verskeie foutopsporingskemas te assesseer. Die gebruik van Beperkte Boltzmann Masjiene se eienskappe in prosesmoniteringskemas sal bespreek word, tesame met die toepassing van hierdie modelle in outomatiese beheer raamwerke.

Sleutelwoorde: Beperkte Boltzmann Masjiene, outokodeerders, dimensionaliteit vermindering,prosesmonitering

ACKNOWLEDGEMENTS

I hereby express my gratitude to the following for making this work a success:

To my supervisors, Professor Aldrich, for your valuable technical guidance, encouragement and patience; and Dr Dorfling for taking over the project and ensuring it is completed.

For all the assistance throughout my studies, a special thank you to Phillip for all your input.

For financial assistance, a big thank you to the Rio Tinto Rössing Uranium employee bursary scheme.

For emotional support, I am very grateful to my wife Kahundu, friends and family.

For all I am, to my Lord and Saviour Jesus Christ for seeing me through.

Finally, I dedicate this work to my two children, Ray and Nankole.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 Process Monitoring and Fault Diagnosis

Process monitoring and fault diagnosis are used to detect faults or abnormal events in processes. The early detection of these events or faults is crucial to continuous process improvement. Traditional methods have been based on mechanistic or causal process models. However, such models are not always available or may be expensive to construct and therefore alternative approaches based on multivariate statistical process control have been proposed. These models are based on empirical correlations built from normal plant operating data when common cause variation is present.

A fault can be defined as an unpermitted deviation of at least one characteristic property of a variable from its normal acceptable behaviour (Isermann, 1997). Therefore, a fault is a state that may lead to a malfunction or failure of a system, which in turn results in process inefficiencies. Fault diagnosis has increasingly become an area of great importance in process control and automation. It provides a framework in which data is monitored and submitted to fault detection schemes. The fault detection scheme records alarms whenever faults are detected. These faults are then identified and classified according to their nature as well as trace their sources.

The fault detection and diagnosis techniques that are used normally depend on process models. The process data from the plant historians is input to fault detection algorithms, and then comparisons are made with the corresponding plant outputs. A difference in these comparisons is an indication that a fault has occurred, and hence could be investigated. Once the type of fault is known, it can then be classified and corrective measures can be put in place to remedy the fault.

As useful as they are, linear methods (such as Principal Components Analysis) do have significant limitations, although a large range of different nonlinear approaches have been considered to date, none of these approaches solve all problems all the time. A major limitation of current linear feature extraction benchmarks is their linear nature. It has been found that using a linear method to extract features from nonlinear data can be inadequate (Dong & McAvoy, 1996).

The diversity that is found in process data structures motivates the exploration of other feature extraction methods. In light of this, many statistical inference techniques such as neural networks (Dong & McAvoy, 1996; Zhu & Li, 2006), kernel methods (Lee et al., 2004; Cho et al., 2005, 2005), random forests (Auret & Aldrich, 2010b) and many others have been investigated in feature extractive fault diagnosis.

1.2 Restricted Boltzmann Machines

Even though principal component analysis and partial least squares have generally been used in process monitoring and fault diagnosis, these models are linear in principle; therefore nonlinear approaches are more likely to provide more accurate, compact and informative models. The use of auto associative neural networks; or auto encoders, provide a better approach to achieve this. However, until very recently, these multiple layer neural networks have been difficult to train and have therefore not been used to any significant extent in process monitoring.

With newly proposed algorithms that are based on the pre-training of layers of the networks, it is now possible to train neural networks with complex structures, which are referred to as deep neural networks. These neural networks can be used as auto encoders to extract features from high dimensional data.

Restricted Boltzmann machines have been used in many applications as generative

models for different types of data, including images (Hinton et al., 2006). Furthermore, Restricted Boltzmann Machines are very interesting because they are used as the building blocks in Deep Belief Networks, which can have many layers and hence are efficient at representing complicated distributions (Bengio, 2009). The process of learning one hidden layer at a time is in effect a very good way to train deep networks that have many hidden layers and millions of weights to deal with. Even though the learning is completely unsupervised, the highest-level features are usually much more useful for classification tasks than the raw data vectors that the network learns.

These deep networks can then be fine-tuned to be better at classification or even in dimensionality reduction problems by using the backpropagation algorithm (Hinton & Salakhutdinov, 2006). Because of the fact that these RBMs can be stacked in deep learning schemes and are generative models, their use as a nonlinear approach in process monitoring and fault diagnosis will be investigated. In view of the above, the usefulness of the features extracted using these networks would be key in using RBMs for fault diagnosis.

In this study, the application of deep auto encoders in the form of Restricted Boltzmann machines (RBM) to the extraction of features from process data is considered. These networks have mostly been used for data visualization to date and have not been applied in the context of fault diagnosis or process monitoring as yet. The objective of this investigation is therefore to assess the feasibility of using Restricted Boltzmann machines in various fault detection schemes. The use of RBM machines features in process monitoring schemes will be discussed, together with the application of these models in automated control frameworks.

An auto encoder with RBM pre-training will be used to extract features from data, and these features used as a basis for process fault diagnosis in several case studies.

1.3 Problem Statement

Many chemical and metallurgical processes are characterized by highly nonlinear and complex dynamics, with long time constants and significant delays. A lot of research has been done on nonlinear process monitoring techniques over the past two decades. This research is driven by the fact that most processes are nonlinear and the methods used to model, monitor and control them are predominantly linear. Although the linear methods that are being used can model and monitor the processes to some good degree of accuracy, there are instances where they fail to capture the nonlinearity that is inherent in the process. Various nonlinear methods have been developed over the years, some of which are already being used in different applications in the chemical and mineral processing industry. There is no single technique that possesses all the desirable features to accurately model and monitor all processes, hence there is need to find more, and even better monitoring techniques.

1.4 Research Objectives

The overall objective of this study is to assess the feasibility of using Restricted Boltzmann Machines in various fault detection schemes. This objective will be covered by the following tasks:

- A literature review of the feature extraction fault diagnosis and the applications of Restricted Boltzmann Machines
- Numerical work in which features are extracted from process data with Restricted Boltzmann Machines (RBMs) and used as the basis for process fault diagnosis in several case studies.
- Comparison and evaluation of the results with other nonlinear approaches.

1.5 Thesis Layout

The rest of this thesis is organised as follows: In Chapter 2, the literature review of multivariate statistical process control and an overview of fault diagnosis is discussed. Chapter 3 deals with the theoretical framework of the Restricted Boltzmann Machine and how they are used as a basis for multilayer auto encoders. The applications of Restricted Boltzmann Machines for feature extraction are also discussed in this chapter. In Chapter 4, the methodology that was used in the study is discussed. The application of the RBM methodology in several case studies is dealt with in Chapter 5. In Chapter 6, recommendations and conclusions from the study are discussed.

CHAPTER 2 MULTIVARIATE STATISTICAL PROCESS CONTROL – LITERATURE REVIEW

This chapter briefly reviews the basics of MSPC (multivariate statistical process control). An overview of current nonlinear methods used for process monitoring is discussed. The overview is not meant to be exhaustive, but to give an outline of what has been studied in the industry in recent years.

2.1 Basics of MSPC (Multivariate Statistical Process Control)

Statistical process control can be a powerful tool to characterize the chemical process in both normal and abnormal conditions. Once the process is characterized, statistical process control can be used to monitor and give early warning of existing, or developing, abnormal conditions.

2.1.1 Univariate Statistical Process Control

For most metallurgical and chemical processes, there is a process control system where large amounts of data is collected and stored on historian data servers. In order to detect and correct problems and process inefficiencies, this data is used, in which only a single variable is considered. Even though the data is available and can be queried from the databases at any time, it is usually difficult for anyone to use this data and determine whether the process is being controlled according to the set control parameters.

Statistical process control charts, such as the Shewart chart, cumulated sum plot, and the exponentially weighted moving average chart, (Venkatasubramanian et al., 2003a) are well established charts and are used in many plants to determine how

well the process is performing. An example of a univariate statistical process control chart is shown in Figure 2.1. From Figure 2.1, Var represents the variable that is being monitored, LCL is the lower control limit and UCL is the upper control limit.

The confidence limits; the lower and the upper control limits, are usually calculated and then used as a basis for detecting the process deviations. When a confidence limit is exceeded, it shows that a fault has occurred and that the process is no longer operating according to the set conditions. These control limits are usually calculated, based on the normal operating conditions (NOC) data, which is taken when the plant is operating at the desired, optimum conditions.

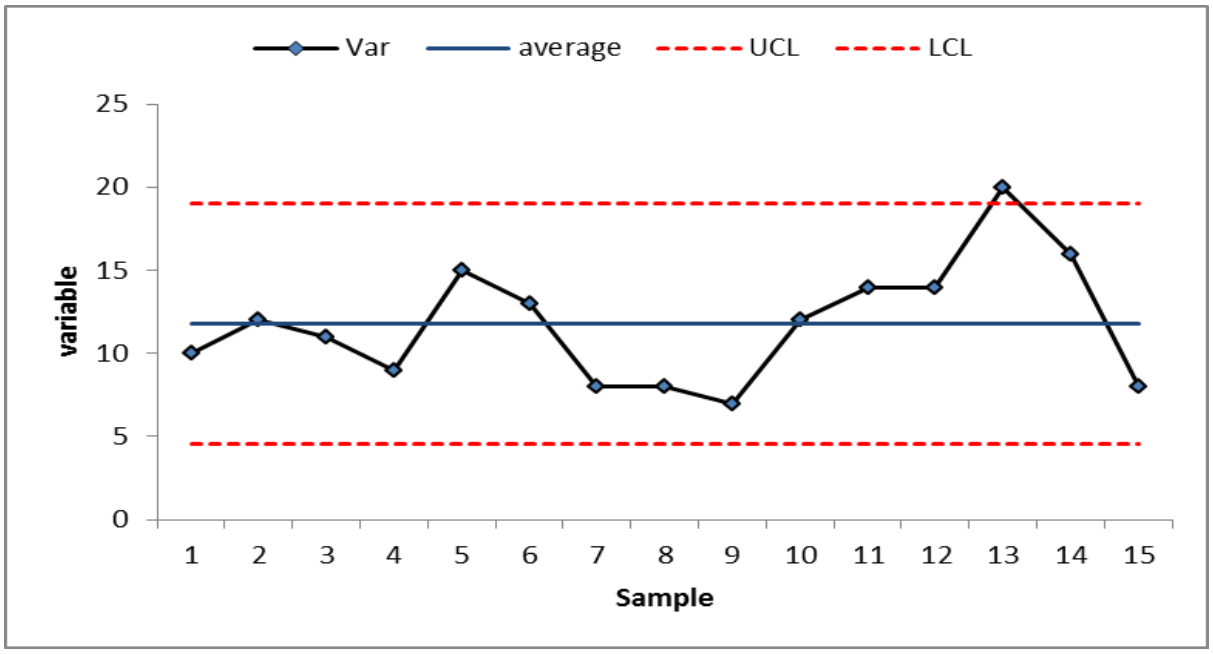


Figure 2.1: Univariate statistical control chart

In this process diagnosis and monitoring scheme, there is only one variable that is measured and hence tested. When this scheme is used, it does not perform well for processes in which there are high correlations among the observed process variables. One of the disadvantages of this scheme is that for a single process, many variables are available that can be monitored and even controlled (Stefatos & Ben Hamza, 2007). This monitoring scheme treats the variables independently, and, as a result, it

only extracts the deviations in each variable independent of all the others. In the process, it ignores the correlation structure between these variables. As a result, process deviations in the process may not be detected at all by the scheme. However, the use of multivariate statistical process control methods can provide a better alternative.

The need to do multivariate data analysis arises when monitoring process performance becomes critical due to that fact that the number of measured process variables increases. This is briefly discussed in the next section (2.1.2).

2.1.2 Multivariate Statistical Process Control

Multivariate statistical process control (MSPC) is an advanced statistical method that attempts to identify the critical variables and patterns in process data. It also shows the relationships between the process variables and how they have an effect on each other. This is important, and applicable when dealing with complex metallurgical and chemical processes.

As done with the Shewart charts, process data is identified and defines the desired normal operating conditions (NOC data). An analysis is then performed, that does not isolate certain variables, so as to ensure that the correlations between the variables are also captured. The major benefit of MSPC compared to univariate monitoring is that the correlation between the original variables is also included in the analysis, which then decreases the chance to omit an out-of-control situation due to the correlation inherent in the data (Thissen et al., 2001).

There are many advantages of multivariate as compared to univariate statistical process control, some of which have already been outlined in the foregoing

discussion. Multivariate analysis can simplify the work of process operators, in that it can show all the process variables, including the relationships that cannot be detected when using univariate statistics. As a result, there is no need to construct process control charts for each variable. Such an analysis is able to reveal the correlation between process parameters and how they are related to faults that are detected in the analysis. MSPC, therefore, assists in understanding the interaction between variables, which makes it possible to create models that can predict effects on the process, before actually implementing these changes.

2.2 Process Monitoring and Fault Diagnosis

2.2.1 Feature extraction process fault diagnosis

In modern chemical and metallurgical plants, process data provides the basis for the monitoring of product quality, process control and improvement. With all the advances in instrumentation and data management technology, large volumes of process data is collected and stored on plant servers. There is a great deal of correlated information in these process variables that are being measured and stored. As a result, the information that is in these stored data should be extracted in such a way that the essential information is retrieved.

To ensure that data that is collected and stored in process and chemical plants is utilized for process control and optimization, it is crucial that the significant features in these data is extracted and analysed. This approach of extracting features from high dimensional data enables plant engineers and metallurgists to better understand the process. Principal component analysis is commonly used for this purpose, as are other techniques such as partial least square, Sammon maps and multidimensional scaling (Zhang, 2009).

Process fault diagnosis can be viewed as a series of mappings of measured process variables. The first mapping is the transformation that is done from the process measurement space, i.e. normal operating data, to the feature space (it should be noted that this is not necessarily the usual case, but in this particular instance, it is). Secondly, a learning algorithm or method is then used to map this feature space unto a decision space. The mapping that is done from the feature space to a decision space is made in such a way that it meets some objective function. There are two categories of methods of developing the feature space from the measurement space, namely, the feature selection and the feature extraction methods. In feature selection, one simply selects a few important measurements of the original measurement space (Venkatasubramanian et al., 2003c).

While in the feature extraction, it is the transformation of high dimensional data into a representation that is useful, but of reduced dimensionality using many different techniques depending on the applications. The technique may be linear, as in principal component analysis, but many other nonlinear techniques do exist that can also be utilised. Dimensionality reduction can be illustrated as follows:

- **Non-linear dimensionality reduction**

Assume that X is a dataset represented in a $n \times D$ matrix that consists of n data vectors with dimensionality D, with intrinsic dimensionality of d<D. The intrinsic dimensionality of data is defined as the minimum number of parameters that are needed to account for the observed properties of the process data (Van der Maaten et al., 2009). During this dimensionality reduction, the reduced dimensionality d contains the features that are extracted and used in process monitoring. This feature space must retain the geometry of the original data as much as possible, and hence contains the significant features that represent the original data.

2.2.2 Fault detection characteristics

In order for one to select a desired feature extraction method to use in fault detection and diagnosis, different approaches are compared. In this comparison, certain characteristics or standards are used to show how these methods perform. All these characteristics are not meant to be satisfied by a fault detection method, but rather to give an indication on how different approaches are comparable. Some of the desirable characteristics looked for in fault detection and diagnostic algorithms are (Venkatasubramanian et al., 2003c):

- **Quick detection and diagnosis:**

  An algorithm should be quick to detect and diagnose faults in a process control system. The time taken to detect these faults normally depends on the process that is being analysed, as the retention time of processes differ. Nevertheless, it is important that the quick detection of the faults does not generate many false alarms, as that becomes a nuisance in the system.

- **Adaptability:**

  Processes have a tendency to change and evolve as a result of changes in external inputs, production quantities, and quality of consumables. The diagnostic system should be able to adapt to these changes, and it has to be designed such that changes in operating parameters can be captured and updated.

- **Explanation facility:**

  Besides the ability to identify the source of the fault, a diagnostic system should also provide explanations on the origin of the fault that is identified. If the source of the fault is known, ways on how to take corrective actions, and design improvements can then be investigated.

- **Modelling requirements:**

  The amount of time and resources spent on modelling has to be kept as minimal as possible for fast and easy deployment of the fault detection scheme. A system that uses a lot of resources in modelling may not be ideal, as more time and resources will be spent on the system than on improving the process.

In the next section, Principal Component Analysis is discussed, as it is the benchmark in multivariate statistical process control.

2.2.3 Principal Component Analysis

Principal Component Analysis (PCA) is a linear multivariate statistical method, generally used for data compression and information extraction by projecting high-dimensional data onto a space with significantly lower dimensions. Specifically, PCA transforms a set of highly correlated variables, into a smaller set of new, uncorrelated variables called principal components (PC). PCA takes advantage of redundant information that exists in highly correlated variables to reduce the dimensionality. Mathematically, PCA relies on an eigenvector decomposition of the covariance or correlation matrix of the process variables.

Principal components are orthogonal to each other and are a linear combination of the original variables. Principal components are traditionally ordered in decreasing order of eigenvalue; are rotated in the directions of maximum variance. In most cases, only the first few principal components that explain most of the variation in the data are retained in the analysis. In order to handle variables with different amplitude and frequency, all the process measurements are usually mean centred and scaled before PCA analysis is done (Rosen & Lennox, 2001). This is standard practise in process monitoring and fault diagnosis.

Principal Component Analysis

- For a data set **X** (n observations by m variables), create a covariance matrix **E**
  - $\boldsymbol{E} = \frac{1}{n-1}\boldsymbol{X}^T\boldsymbol{X}$ **Eqn.1**
- Calculate the eigenvectors **V** and eigenvalues Λ for the covariance matrix **E** using eigenvalue decomposition
  - $\boldsymbol{E} = \boldsymbol{V}\Lambda\boldsymbol{V}^T$ **Eqn.2**
- Determine the reduced dimensionality **a**, that captures significant variance.
- Define the loading matrix (principal components) **P** as the first **a** eigenvectors of **V**
- Calculate the principal component scores
  - $\boldsymbol{T} = \mathbf{XP}$ **Eqn.3**

The columns of the matrix P are known as loadings while elements of the matrix T are called scores. The scores are the values of the original process variables that are mapped into the reduced dimensional space vectors. In the context of feature extraction, the score vectors obtained from projecting the process variables onto the principal components can be considered as the extracted features. The number of principal components to use in calculating the features can be determined by investigating the cumulative variance accounted for by including additional principal components (Zumoffen & Basualdo, 2008).

The scores can then be transformed into the original vector as follows:

$$\widehat{\boldsymbol{X}} = \mathbf{TP^T} \qquad \textbf{Eqn.4}$$

The residual matrix R, is now evaluated as

$$\boldsymbol{R} = \boldsymbol{X} - \widehat{\boldsymbol{X}} \qquad \textbf{Eqn.5}$$

Finally, the original input data can be calculated as

$$\boldsymbol{X} = \mathbf{TP^T} + \mathbf{R} \qquad \textbf{Eqn.6}$$

- **Process Monitoring with PCA**

After the PCA model that is based on historical data is constructed, multivariate control charts that are based on Hotelling's T^2 and square prediction error (SPE), or Q, can now be plotted. The process monitoring scheme is then reduced to only the two variables, T^2 and Q, which characterizes two orthogonal subsets of the original data space. Hotelling's T^2 represents the major variation in the data and Q represents the random noise that is in the original data (Garcia-Alvarez et al., 2009). Hence, T^2 explains the variation within the score space by using all the retained PCs. Hotelling's T^2 value is calculated as

$$T^2 = \boldsymbol{X}^T \boldsymbol{P} \Lambda_\alpha^{-1} \boldsymbol{P}^T \boldsymbol{X} \qquad \textbf{Eqn.7}$$

where Λ_α is the square matrix that is formed by the first a rows and columns of Λ

The process will be considered to be normal if:

$$T_\alpha^2 \leq \frac{(n^2-1)a}{n(n-a)} F_\alpha(a, n-a) \qquad \textbf{Eqn.8}$$

where $F_\alpha(a, n-a)$ is the Fisher-Snedecor distribution with $a, n-a$ degrees of freedom and α the level of significance.

The Q statistic or squared prediction error (SPE) measures the variability that breaks the normal process correlation in the data. Mathematically, Q is obtained as the sum of the squared errors in the residual space or the sum of variations in the residual space, which is defined as

$$\boldsymbol{Q_j} = (\boldsymbol{X_j} - \widehat{\boldsymbol{X}}_j)^2 \qquad \textbf{Eqn.9}$$

for the j^{th} sample

The Q statistic is thus a measure of the amount of the variations in each sample that is not captured by the retained PCA model.

The detection thresholds for the squared prediction errors can be calculated as:

$$\boldsymbol{Q_\alpha} = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\boldsymbol{\theta_1^2}} \right]^{1/h_0} \qquad \textbf{Eqn.10}$$

where $\theta_i = \sum_{j=a+1}^{n} \lambda_j^2$, $h_0 = \frac{2\theta_1\theta_3}{3\theta_2^2}$ and $c_\alpha$is the value of the normal distribution, α is the level of significance and λ_j is the j[th] eigenvalue of E (Alcala & Joe Qin, 2011).

The values of these two statistics are also calculated for the new data set. If, at a specific point, T^2 or Q for the new data set is outside the calculated control limits, the process is said to be out of control at that point. In fact, this may mean that a fault has occurred at that point. When any fault has been detected using any of the T^2 or Q statistics, it is crucial to identify the cause of that fault. This can be done by using contribution plots of the original data. In a PCA model, two types of contribution plots are used to identify the fault since two types of control charts are used, i.e., a chart for residuals and one for Hotelling's (Teppola et al., 1998).

The residual plots show the Q residual values plotted against the samples, and this shows the time when the fault occurs. The contribution plots are computed so that it can be determined what type of fault is detected. The contribution plots are calculated by computing the means of the columns of the residual matrix R that is based on the faulty data set (Ralston et al., 2004). The contribution plots are then used to determine which variables are associated with the faults that are detected.

In determining whether the individual variable contribution to the T^2 value is significant or not, one can calculate the control limits for the contribution plots. It is also possible to compare the size of the variable's contribution during the faulty conditions with the size of the same variable's contribution under the desired normal operating conditions. Therefore, the variables with the largest contribution to the T^2 value normally indicate the source of the fault (Johnson & Wichern, 2007).

Principal component analysis has been applied in many different areas such as science, biology, engineering, etc., but despite all these applications, it has its difficulties as well. Limitations of the PCA methodology include its lack of exploitation of autocorrelation (Venkatasubramanian et al., 2003b) and its linear nature. In order to address these and other drawbacks of PCA, several extensions of PCA have been developed, some of which are discussed in the next section.

2.3 Developments in Nonlinear Feature Extraction Fault Detection

In order to capture the nonlinear nature of measured process data for fault diagnosis, many feature extraction strategies have been investigated and studied over the last couple of years. The overview of these nonlinear feature extraction methods is not meant to be exhaustive, but only to highlight the different approaches to nonlinear feature extraction available in literature. The body of literature is also relatively large and hence only a brief review is given in this section.

2.3.1 Neural Networks

A neural network is an architecture that is made up of large numbers of units that are called neurons. An example of a neuron is shown in Figure 2.2 (Page 17). The neuron that is shown consists of n inputs;$x_1, x_2, x_3, \ldots, x_n$. These inputs come from a variety of sources, not limited to the network structure in which they originate from other units, or may even be from some external sources (Pollard et al., 1992). The output of the unit y in this network is given as:

$$y = \frac{1}{1+e^{-A}} \qquad \textit{Eqn.11}$$

where

$$A = the\ element\ activation = \sum_{i=1}^{n} w_i x_i \qquad \textit{Eqn.12}$$
$$w_i\ is\ a\ weight\ factor$$

These units or neurons are arranged in layers, as shown, Figure 2.2. The network that is shown has three layers. The first layer consists of neurons that have inputs to the network; these inputs come from external sources. This is the layer that interacts with the outside environment. These neurons in the first layer then act as inputs to the second layer. In the same manner, the neurons in the third layer get their inputs from the second layer, and the third layer is the output of the entire network to the outside environment in case of a single hidden layer.

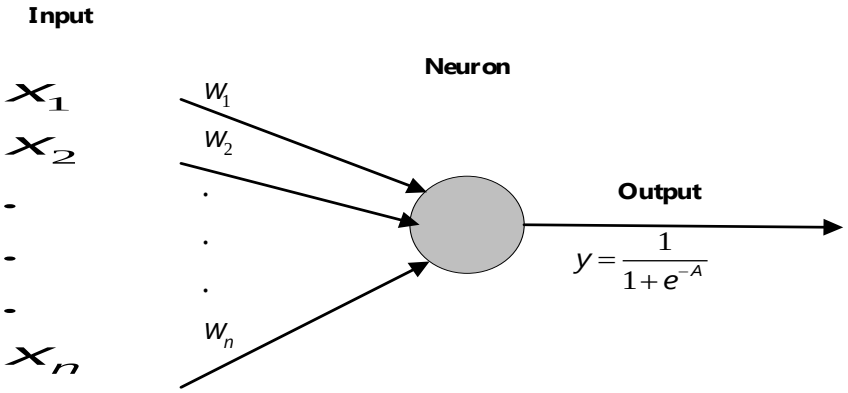


Figure 2.2 Artificial neuron

Since the second layer has no direct connections with the environment as it only interacts with the first (input) and the third (output) layers, it is called the hidden layer. The number of hidden layers can be more than one, as network structures change, and depending on the intended use of the network.

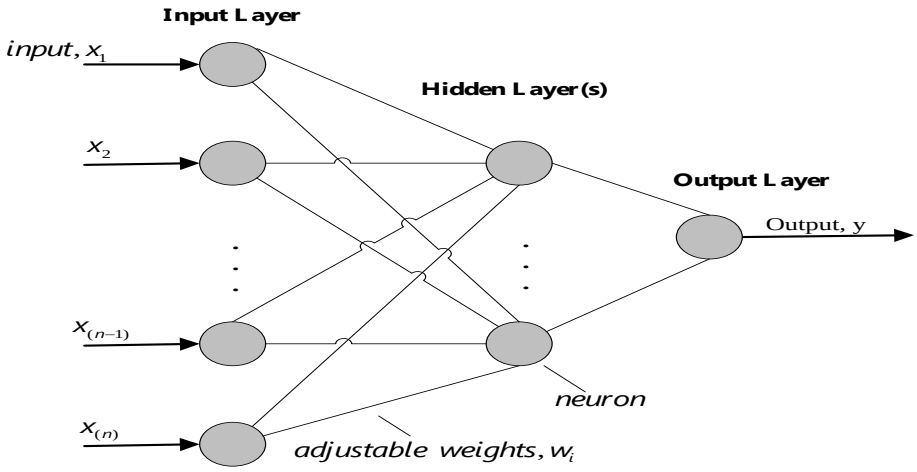


Figure 2.3 An example of a neural network with a single hidden layer

The reason why we require these neural networks is to construct a mapping from a vector X to a vector Y. The size of the input layer X and output layer Y remains fixed by the number of neurons that are contained in them. As for the hidden layer, its size depends on the user's requirements and the purpose for which the network is being used for. Since the network maps the input values to the output, the error between the predicted and the observed values should be as small as possible. During the training phase of the network, the network is presented with examples of the type of mapping that is required. These training examples are referred to as training vectors, and they are pairs that consist of the input and the output (Pollard et al., 1992).

There has been an interest in literature in the application of neural networks in order to have a solution to the fault diagnosis problem. The neural networks that have been studied can be classified into the following categories:

(i) the architecture of the network such as sigmoidal, and
(ii) the learning algorithm in form of either supervised or unsupervised (Venkatasubramanian et al., 2003b).

The standard of applying neural networks in fault diagnosis is used to classify the process data according to the operation of the process. The classification method uses the individual measurement patterns in the process data, and has no information about the direction of changes found in the process measurements. The classification of these individual measurement patterns is a very straightforward fault diagnosis method. When there is sufficient process measurements that are available, this classification can be done. This classification method is an off line fault diagnosis scheme, where process data is collected and the faults are properly defined. A classifier is then designed, and exposed to some test data. After this, the classifier is then used in the process (Sorsa & Koivo, 1993).

2.3.2 Nonlinear PCA with auto associative neural networks

Nonlinear principal component analysis (NLPCA) is a nonlinear generalization of the standard principal component analysis that was discussed earlier. This is used in multivariate data, and generalizes the principle components from straight lines to curves. Therefore, the subspace in the original data space which is described by all nonlinear components is also curved (Scholz et al., 2007). The NLPCA is used to identify and remove correlations found within the problem variables, thereby assisting with the fault diagnosis problem. The main difference of NLPCA compared to PCA is that both linear and nonlinear correlations are uncovered within the data.

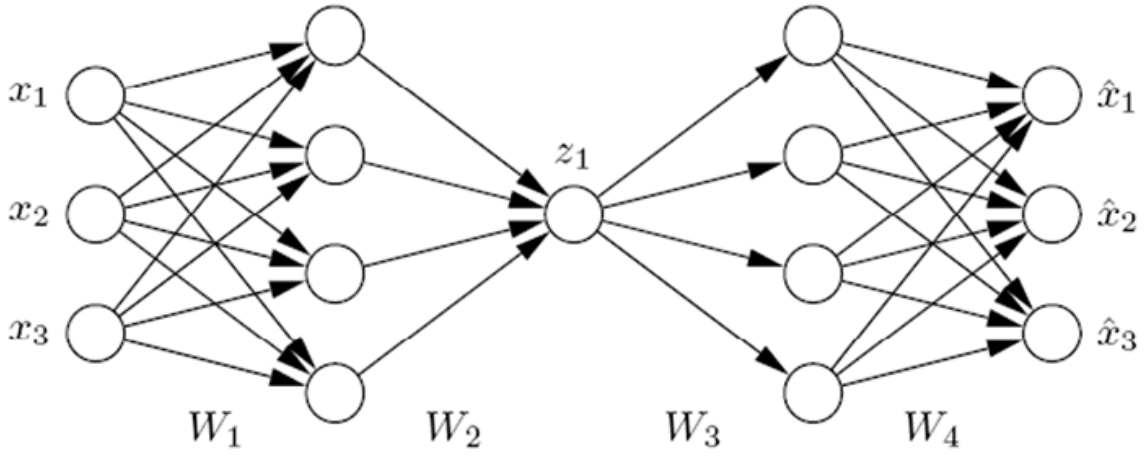


Figure 2.4 Auto associative neural network architecture (from Scholz et al., 2007)

Nonlinear PCA can be implemented by using a neural network (Figure 2.1). NLPCA operates by training a feed forward neural network to perform the identity mapping, where the network inputs are reproduced at the output layer (Kramer, 1991). However, we find in the middle of the network a layer that works as a bottleneck where dimensionality reduction of the data is applied. This bottleneck layer ensures that the network develops a representation of the input data, and that all the features in data are extracted in this layer.

2.3.3 Kernel PCA

When dealing with neural networks for feature extraction, there are difficulties that are encountered. Some of these difficulties arise because one has to predetermine the number of features that must be extracted (Lee et al., 2004). An alternative to network based feature extraction that addresses some of the difficulties is Kernel PCA. What KPCA does is that it tends to transform the original data into a higher dimensional feature space, in which linear PCA can be applied, and then only the significant components are retained.

The calculation of the Kernel principal components is an eigenvalue problem. The number of components that is retained is determined based on the variance decomposition (Cho et al., 2005). Figure 2.5 illustrates the way that Kernel principal components analysis is performed.

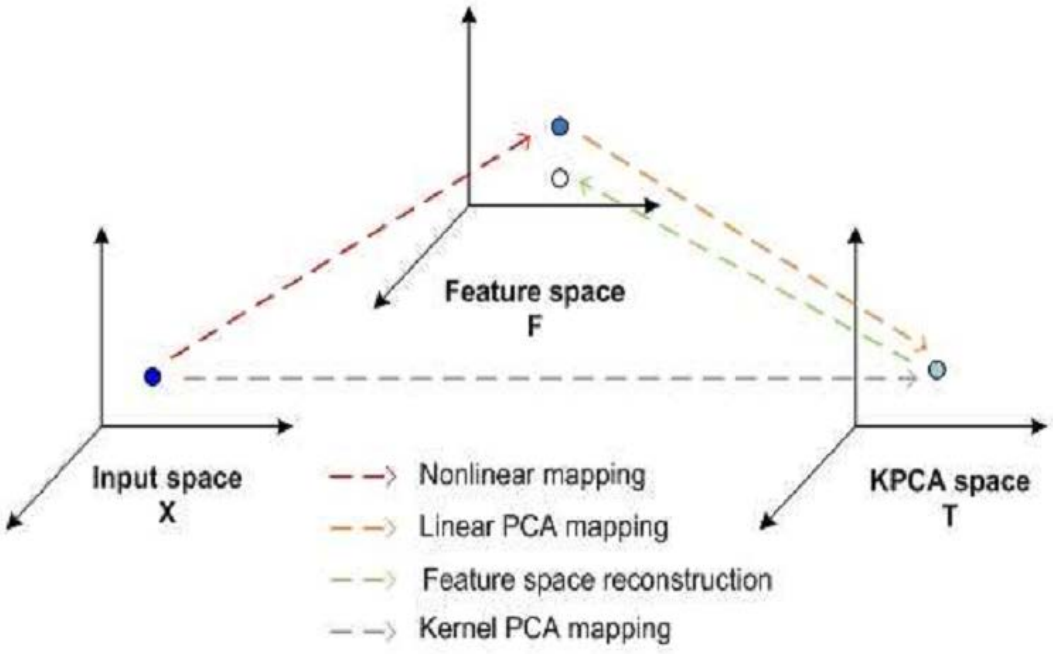


Figure 2.5 Steps of KPCA projection and reconstruction (Lee et al., 2004; Auret, 2010)

The setback that KPCA has is that no explicit demapping function is available to reconstruct the nonlinear principal components to the original input data. Some limitations to the KPCA approach includes the computational expense of calculating

the required dot product for large sample data sets, as well as the lack of interpretability of nonlinear components in the original input space (Cho et al., 2005).

2.3.4 Random forests

A development in statistical learning is the emergence of ensembles of learning machines. “An ensemble is described as a combination of a collection of classifiers in order to enhance the performance of the overall classifier” (Valentini & Masulli, 2002).

It has been shown (Valentini & Masulli, 2002) that these ensembles of classifiers normally perform better than the individual classifiers, even in cases where the base classifiers are considered weak. By constructing an ensemble of classifiers, more thorough exploration of hypotheses can be accomplished (Valentini & Masulli, 2002).

Random forests are nonlinear regression models that consist of ensembles of regression trees, in which each tree depends on a random vector that is sampled independently from the process data (Auret & Aldrich, 2010a). The random forest model is an example of ensemble methods, with the base classifiers consisting of unpruned decision tree classifiers (Breiman, 2001). A decision tree is a recursive subspace partitioning classifier, that works in such a way to reduce the class impurity of successive subsets (Breiman et al., 1993).

The prevalent use of the random forest algorithm can be due to its high accuracy and fast computations (Breiman & Cutler, 2003). The tree ensembles such as these random forests can further provide an added functionality in which one can interpret the variable importance (Breiman & Cutler, 2003) as well as partial dependence analysis (Friedman, 2001). The random forest feature extraction was applied to unsupervised fault diagnosis for process data, and compared to linear and nonlinear

methods. Random forest results were comparable to the existing techniques (Auret & Aldrich, 2010a; Auret, 2010).

2.3.5 Biplots

Gabriel (1971) introduced the concept of the biplot, which is defined as a graphical display which consists of a vector for each row and a vector for each column of a matrix that has a rank of two. The biplot is a multivariate equivalent of the scatter plot (used for univariate). An element of this matrix is then represented by the inner product of the vectors that correspond to both its row and its column (Gardner et al., 2005). Aldrich et al., 2004 and Gardner et al., 2005 proposed a related statistical process monitoring approach that emphasizes on the visualization of process correlations and variations in the process variables by using the biplot. In addition, this approach of bipot provides for an automatic detection and then the visualization of the process disturbances by use of bagplots (Rousseeuw et al., 1999)

CHAPTER 3 RESTRICTED BOLTZMANN MACHINES

This chapter gives the theoretical framework of Restricted Boltzmann Machines, the network architecture and the training algorithm. The auto encoder is also discussed, in which the network is pre-trained with Restricted Boltzmann Machines. A review, on the use of RBMs for feature extraction is discussed in this chapter as well.

3.1 Boltzmann Machines

The Boltzmann machine is a collection of symmetrically connected, neuron-like, stochastic binary units (Figure 3.1, page 24). Each unit in the network selects to be on or off by considering the total input that it receives from all the other units.

For any training set of state vectors, the weights and biases in a Boltzmann machine can be adjusted to assign high probability to vectors in the training data. The units in a Boltzmann machine can be partitioned into two subsets, namely, visible and hidden unit. The visible units are those units of the network whose states can be observed, while the hidden units are those with unobserved states. These visible neurons provide an interface between the network and the environment in which the network operates (Haykin, 1999).

In this study, the focus is in a special type of the Boltzmann Machine, in which there are no connections within layers (with no visible-visible or hidden-hidden connections). This special type is called the Restricted Boltzmann Machine.

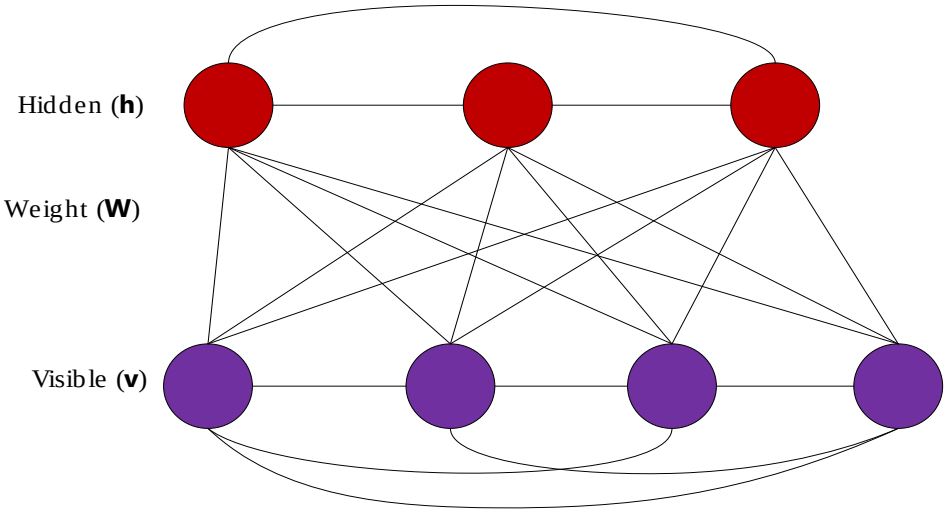


Figure 3.1: The Boltzmann Machine

3.1.1 The Restricted Boltzmann Machine

A Restricted Boltzmann Machine (RBM) (Sejnowski, 1986) is a two-layer neural network that contains a layer of visible, binary stochastic units, connected to a layer of hidden, binary stochastic units, without connections within each layer, i.e. no visible-visible and no hidden-hidden connections, as shown in Figure 3.2. The connections are symmetric, meaning that they have the same weight in both directions (Hinton, 2010).

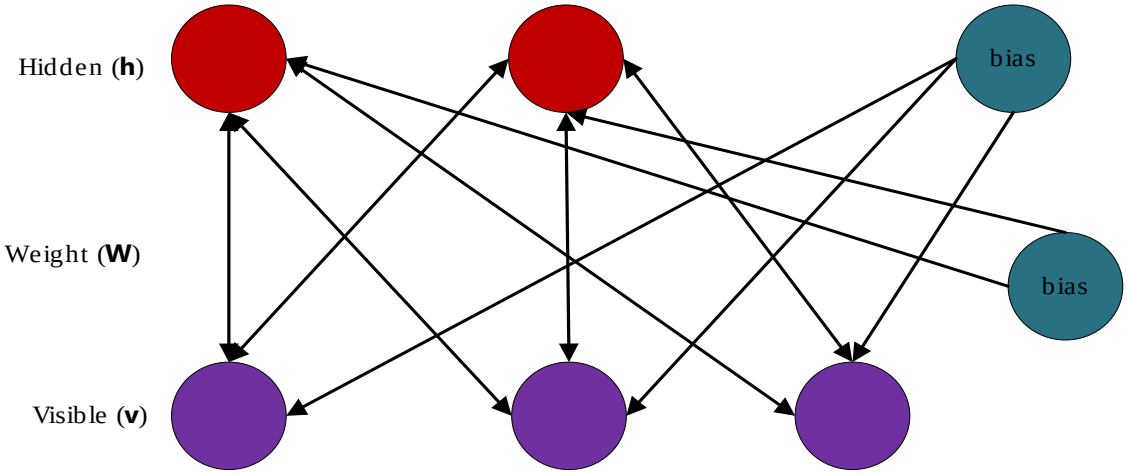


Figure 3.2: Restricted Boltzmann Machine

A configuration $(\boldsymbol{v}, \boldsymbol{h})$ of both the visible and hidden units has the following energy:

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \qquad \textbf{Eqn.13}$$

where v_i, h_j are binary states of visible unit i and hidden unit j, a_i, b_j are their respective biases and w_{ij} is the weight.

This network then assigns the following probability function for every possible pair of visible and hidden vector

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v}, \boldsymbol{h})} \qquad \textbf{Eqn.14}$$

where the partition function, Z, is given by summing over all possible pairs of visible and hidden vectors:

$$Z = \sum_{\boldsymbol{v}, \boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})} \qquad \textbf{Eqn.15}$$

The probability that the network assigns to a visible vector, **v**, is given by summing over all the possible hidden vectors:

$$p(\boldsymbol{v}) = \frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})} \qquad \textbf{Eqn.16}$$

3.1.2 Training the RBM

The derivative of the log probability of a training vector with respect to a weight (as shown in Eqn. 16) is simplified as:

$$\frac{\partial log p(\boldsymbol{v})}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \qquad \textbf{Eqn.17}$$

where $\langle \ \rangle_{data}$ is the value that is expected of that distribution, and $\langle \ \rangle_{model}$ is then the value that is expected of Boltzmann sampling vectors.

Since there are no direct connections between the hidden units in a Restricted Boltzmann Machine, it becomes easy to obtain an unbiased random sample of $\langle v_i h_j \rangle_{data}$. For a randomly selected training data from the input space, v, the binary state, h_j,for each of the hidden unit, j, is set to 1 with a probability of

$$p(\boldsymbol{h_j} = \mathbf{1} \mid \boldsymbol{v}) = \sigma(b_j + \sum_i v_i w_{ij}) \qquad \textbf{Eqn.18}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid function. $v_i h_j$ is then an unbiased sample (Hinton et al., 2012).

Similarly, since there are no direct connections between visible units in an RBM, it is also very easy to get an unbiased random sample of the state of a visible unit, provided the hidden vector is known

$$p(\mathbf{v_i} = \mathbf{1} \mid \mathbf{h}) = \sigma(a_i + \sum_j h_j w_{ij}) \qquad \textbf{Eqn.19}$$

To get an unbiased sample of $\langle v_i h_j \rangle_{model}$ is very difficult, and requires adjustments in the training procedure. To sample from $\langle v_i h_j \rangle_{model}$ still requires multiple iterations that alternates between the updating of all the hidden units, and then updating all of the visible units and both updates are done in parallel. However, this learning still works very well if $\langle v_i h_j \rangle_{model}$ is replaced by the corresponding $\langle v_i h_j \rangle_{recon}$. A much faster learning procedure was proposed by Hinton (Hinton, 2002) which ensures that $\langle v_i h_j \rangle_{recon}$ is obtained as follows:

a. Starting with a training data vector on the visible units, set the states of the visible units to a training vector. Then update all of the hidden units in parallel.
b. Update all of the visible units in parallel to get a “reconstruction”.
c. Update all the hidden units again

From Figure 3.3, it can be seen that if an input vector,$\boldsymbol{x}_i$ is used as a training vector, then the hidden units are all updated in parallel. Then update the visible units again to get a reconstruction, which is shown by the vector, $\boldsymbol{x}_i'$.

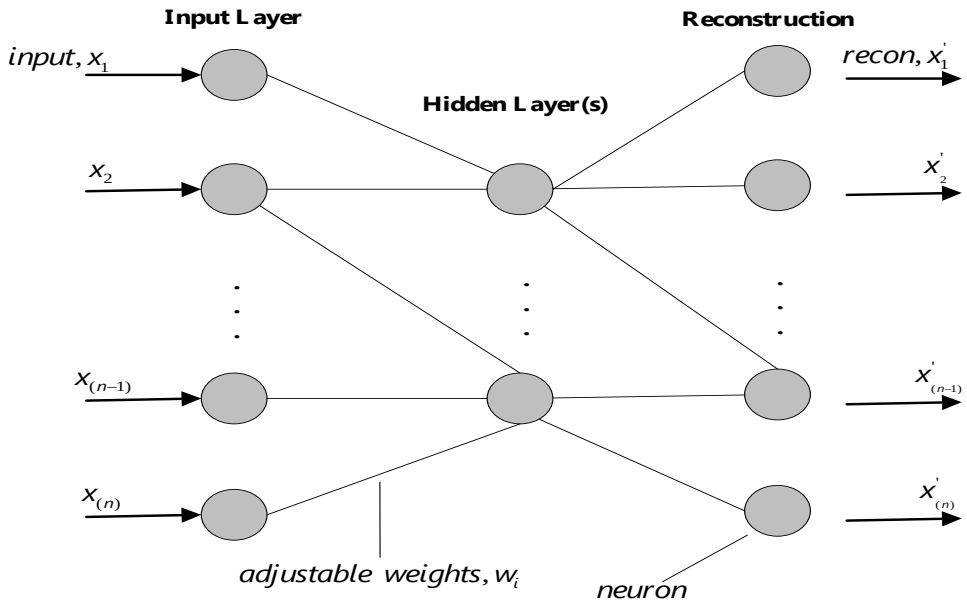


Figure 3.3: Training the Restricted Boltzmann Machine

After all the updates, the change in weight is derived as

$$\Delta w_{ij} = \varepsilon\left(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{recon}\right) \qquad \textbf{Eqn.20}$$

where ε is the learning rate

This learning procedure that is explained in the foregoing discussion approximates gradient descent in what is known as Contrastive Divergence (CD).

3.2 Stacked Restricted Boltzmann Machines

3.2.1 Dimensionality reduction using auto encoders

A multilayer auto encoder is a feed forward neural network that has more than one hidden layer in the network structure. This network attempts to reconstruct the input

data at the output layer of the network (Hinton et al., 1997). The targets at the output layer of the network are normally the same as what you find at the input layer, therefore, the sizes of the input and output layers are the same. Since the hidden layer is smaller compared to the input data in terms of its size, the dimensionality of the original input data is reduced to a smaller dimensional space at this hidden layer (Vishnubhotla et al., 2010). The hidden layer gives a smaller dimensional representation of the data that preserves as much structure as in the original data. This ensures that the low dimensional, nonlinear structure of the data is revealed (Hinton & Salakhutdinov, 2006).

Real world data such as; speech signals, process data, digital photographs, usually has a high dimensionality. In order to handle that type and nature of data effectively, there is a need to reduce its dimensionality to rather a level much lower than the original data. After this transformation, the reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the data. The intrinsic dimensionality of data is the minimum number of parameters that are required to account for the observed properties of the data (Van der Maaten et al., 2009). PCA is widely used in reducing the dimensionality of process data, but, as discussed earlier, its linear nature is a drawback. A neural network that has at least one hidden layer in its network structure can give a nonlinear mapping from input to output layer. However, the normal neural networks are usually unable to reduce the dimensionality of training data to the same extent as that of PCA (Tan & Eswaran, 2008).

High dimensional data can be converted to low dimensional space by training a multilayer network with a small central layer that reconstructs high dimensional input vectors. As Hinton and Salakhutdinov did in (Hinton & Salakhutdinov, 2006), they describe a way that initializes the weights that can allow deep auto encoder networks to learn low dimensional space that works better to reduce the dimensionality of the

training data. This is a nonlinear generalisation of principal components analysis. It uses a multilayer encoder network to transform the high dimensional training data into a low dimensional space, and then also uses a similar decoder network to recover the data from the reduced space, see Figure 3.4 (page 30).

3.2.2 Stacked Autoencoder with RBM pre training

In training this network, first start with a standard one hidden layer auto encoder. The weights are trained with the Restricted Boltzmann Machine. The outputs from this first RBM are used as the inputs for the next encoder. The same training process is done in which the hidden layer is trained, and the outputs used as the input for the next network in the stack. This training process is repeated for as many layers as needed; thereby creating a stack of auto encoders.

After the pre training of multiple layers, the model is unfolded (Figure 3.4, page 30) to produce the encoder and decoder networks that use the same network weights that it has learned during the training. The fine tuning stage of the network then replaces the stochastic activities by deterministic, real valued probabilities and then uses backpropagation through the whole auto encoder in order to fine tune the weights. A multilayer auto encoder is a feed forward neural network which has more than one hidden layer in the entire network structure. This structure uses RBM pre training for each of the hidden layers.

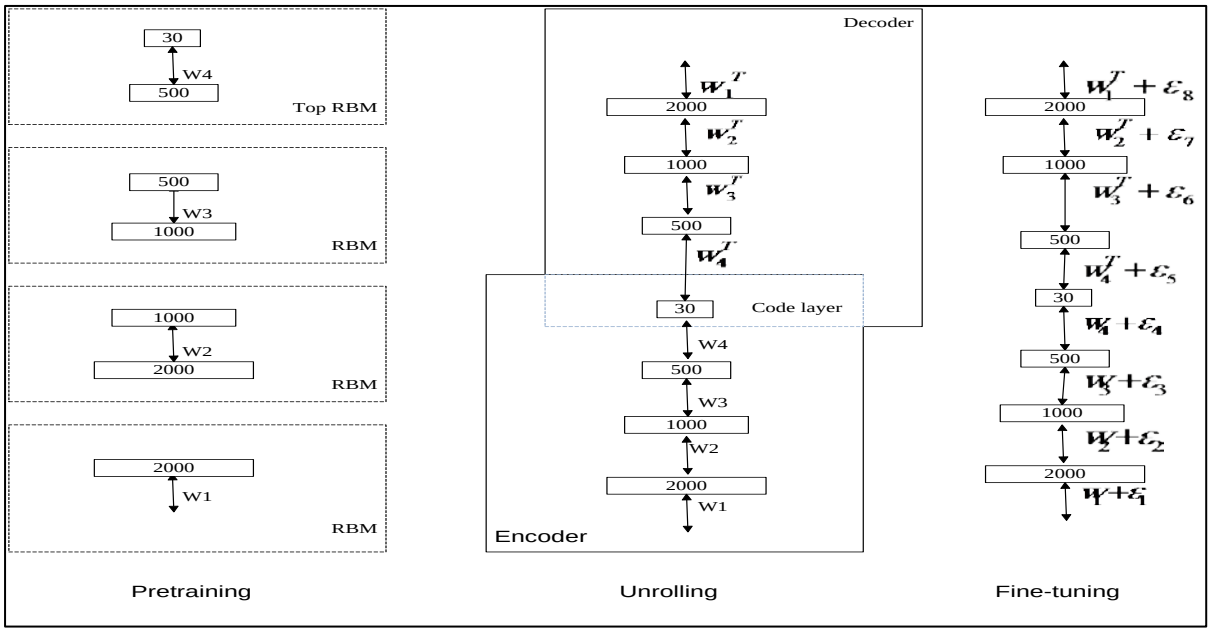


Figure 3.4 Autoencoder with RBM pre-training (Hinton & Salakhutdinov, 2006)

3.2.3 Stacked RBM network architecture

After each RBM has been trained, a new layer is added in which the input is the output of the trained RBM. This new layer is trained as a separate RBM using the normal training process. In the greedy training procedure, one layer is added on top of the network at each stage, and only that top layer is trained (Hinton, 2007) (as an RBM, see Figure 3.5).

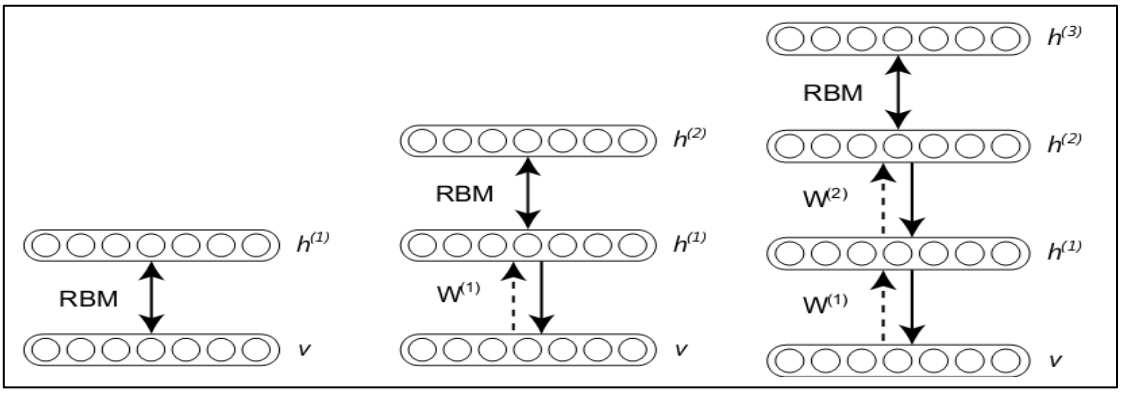


Figure 3.5: Stages of the learning of layers of RBM's (Hinton, 2007).

Using the layer-by-layer learning algorithm of section 3.1 (page 29), first learn a stack of RBM's. After the learning is complete, the stochastic activities of the binary units in each layer are replaced by deterministic, real valued probabilities and the auto encoder is then used to initialize a multilayer, nonlinear mapping as shown in Figure 3.6. This learning is treated as a pre training stage that captures a lot of the higher order structure in the input data. In Figure 3.6, greedy training a stack of RBM's where samples from the lower level RBM are used as the data for training the next RBM is shown. The corresponding deep belief network that is formed after the learning is shown in Figure 3.7 (page 32).

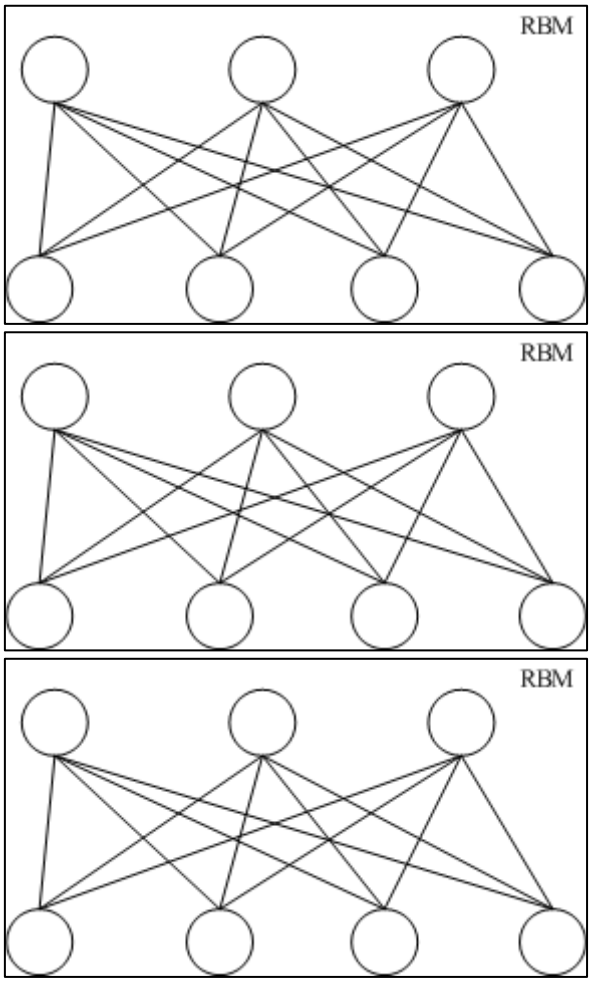


Figure 3.6: Learning a stack of RBMs

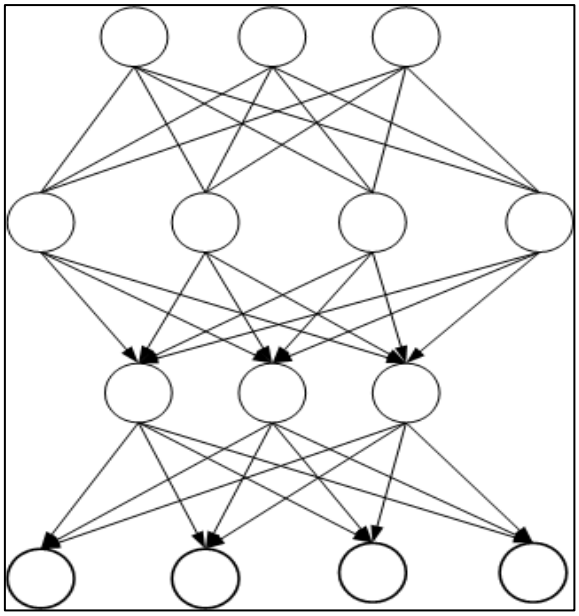


Figure 3.7: A deep multilayer network

Successful applications of these networks have been applied among others, in classification problems (Bengio et al., 2007), regression analysis (Salakhutdinov & Hinton, 2008), dimensionality reduction (Hinton & Salakhutdinov, 2006; Salakhutdinov & Hinton, 2007), modelling textures (Osindero & Hinton, 2008), information retrieval (Krizhevsky & Hinton, 2011), robotics (Hadsell et al., 2008) and natural language processing (Collobert & Weston, 2008). With a few exceptions (Sutskever & Hinton, 2007; Hinton & Brown, 2000), the literature on RBMs is confined to modelling static data. Therefore, in the next section, a review of some of the applications of Restricted Boltzmann machines is given.

3.3 Review of Applications of Restricted Boltzmann Machines

This section briefly describes the review of the applications of Restricted Boltzmann Machines. As already highlighted, the use of RBMs in feature extraction is important, as this will determine its usefulness in process monitoring.

3.3.1 Reconstruction of images

Reconstruction of face and digital images using auto encoders is discussed in this section. The training of the auto encoder using Restricted Boltzmann Machine as building blocks is discussed in section 3.1.1. The first step to consider when dealing with image reconstruction is to start by training the auto encoder. This auto encoder will have an input layer, a hidden layer and the output layer.

The sizes of the hidden layers are set as desired in the experiment. The training images are set as fed into the auto encoder network, which then reduces the dimensionality of the training data in the middle hidden layer. During this dimensionality reduction, the training data is represented into a smaller code space, which is then reconstructed back into the images. The output of the hidden layer in this network is then used as the input to train the next auto encoder network. This process is then repeated for the next network. The output layer always reconstructs the image as input through the training and testing phase.

The experiments were conducted on the ORL (Olivetti Research Laboratory) face data set. The training data had 400 images. The training images are rescaled to size 37×30 by using the nearest neighbour interpolation. The pixel values of these images are then normalised to be in the range from 0 to 1. The dataset is then divided into 200 training images of two sets, one contains the first five (5) images and the other subset the last five images of each person.

The network that was trained remained in such a way that the deepest hidden layer had 30 neurons. The deepest hidden layer in the network uses a linear activation function, whereas all the other layers use sigmoid activation functions (Tan & Eswaran, 2010). All the layers in the network were fully connected after the training was completed.

A standard one hidden layer stacked auto encoder network is initialised with small random weights and biases that range from 0 to 0.1. For the architecture in the experiments conducted, the weights and biases were pre-trained using RBM for 50 epochs, the total number of epochs used for the training being 230. The MSE for the testing phase after 230 epochs was 6.8, which performed better than auto encoders without RBM pre training which had the reconstruction error of 9.1. From these experiments, auto encoders were successfully used to reconstruct images, as was seen from the reconstruction errors in which they outperformed those without RBM pre-training.

The same approach was considered but in this case using the MNIST dataset of handwritten digits (Tan & Eswaran, 2010). The training and testing sets were divided according to most of the other benchmarking experiments carried out by other researchers, in order to make it easier to do comparisons. By using similar network architectures, the MSE for the auto encoder with RBM pre-training was 1.21 compared to 1.685 for the one without.

It was shown in this experiment, that auto encoders with RBM pre-training can be used successfully in image reconstruction, and outperformed the networks without RBM pre-training. Since the MNIST database is a large dataset (with 6000 training images) compared to the ORL (with 400 images), the trained auto encoder has better generation since a good convergence is achieved at the end of the training phase. The reconstruction errors are shown for both data sets in Table 1.

Table 1: MSE for MNIST & ORL datasets for whole image

Model	ORL	MNIST
Autoencoder	9.1	1.685
Autoencoder with RBM pre-training	6.8	1.210

3.3.2 Using Auto encoders for Mammogram Compression

The application of auto encoders for medical image compression was considered by Tan & Eswaran (2009). The paper presents the results obtained for medical image compression using auto encoder neural networks. These experiments show that auto encoders can be trained effectively by using image patches instead of the entire image, and still yield results that are comparable to other approaches (Tan & Eswaran, 2009).

The performance of the auto encoder is based on the parameters mean squared error (mse) and structural similarity (ssim) index. MSE is the one measure of distortion used for images. "The MSE averages the squared intensity differences of compressed and original image pixels" (Cosman et al., 1994). The ssim index varies between o and 1, with 0 being worst as it represents non-identical images and 1 represents identical images.

Experiments were conducted on Images from Digital Database for Screening Mammography (DDSM), a mammogram dataset for breast cancer diagnosis. Three categories of mammograms that consist of 100 patients with normal breasts, 80 patients with breast cancer and 70 patients with benign were selected. The results for the MSE are shown in Table 2. The performances also depend on the size of the hidden layers, since smaller hidden layers decrease the performance as the reconstruction errors are higher. The auto encoder with RBM pre training managed to get the ssim index of 0.98, as compared to the one without pre training with an ssim index of 0.89.

Table 2: MSE for different network architectures

Network architectures	training
Autoencoder	0.1206
Autoencoder with RBM pre-training	0.00974

3.3.3 Face Recognition

The face recognition problem is addressed using an auto encoder with RBM pre-training. The recognition problem using the auto encoder can be implemented using a number of steps that are discussed (Tan & Eswaran, 2010). As with many of these applications, the first step normally involves training the auto encoder. After the auto encoder is trained using the images, feature codes are then obtained from the test images.

In the experiments conducted, the feature codes from the deepest hidden layer are extracted for classification. These experiments were conducted based on the two datasets, namely the MNIST and ORL face dataset. Table 3 shows the recognition rates that were obtained in the experiments. From the results, it is evident that the auto encoder with RBM pre training yielded good results with recognition rates of 86% on the ORL dataset and 93.1% on the MNIST database.

Table 3: Recognition rates (%) of different network architectures

Models	ORL	MNIST
Autoencoder	80.5	92.6
Autoencoder with RBM pre-training	86.0	93.1

3.3.4 Classification & filtering

Collaborative filtering is the process of filtering for patterns (or information) by using techniques involving collaboration among viewpoints, data sources, etc. This involves very large datasets, and can include, but not limited to, sensing and monitoring data, financial data, and movie ratings. A widely used approach to collaborative filtering is to assign a low dimensional feature vector to each user and a low dimensional feature vector to each movie. This is done in order that the rating that each user assigns to each movie is then modelled by the scalar product of the two feature

vectors. In Salakhutdinov (2007), authors showed that RMBs can be used in modelling tabular data, for instance, the user's ratings of movies.

The Restricted Boltzmann Machine was applied to the Netflix data with very good results. The Netflix data represents the distribution of all ratings Netflix collected during the period of October 1998 to December 2005. The training data consists of 100,480,507 ratings from 480,189 randomly chosen, anonymous users done on 17,770 movie titles. Also provided in the research is the validation data that has 1,408,395 ratings and a test set containing 2,817,131 user/movie pairs with the ratings withheld. The pairs that were selected contained the most recent ratings that were available.

The RBM was trained using Netflix training data by means of various network specifications. The weights during the training phase were initialized with small random numbers that were sampled from a normal distribution with zero mean and a standard deviation of 0.01 (Salakhutdinov et al., 2007). The baseline root mean square error (RMSE) provided by Netflix on the same data, is 0.9514 using their own system, and the result of the experiment was an RMSE of 0.92. Restricted Boltzmann Machines can be successfully applied to large data containing approximately over 100 million user/movie ratings.

CHAPTER 4 PROCESS MONITORING WITH RBM METHODOLOGY

This section describes the methodology that was utilised to use the Restricted Boltzmann Machines to extract features from process data. The features will then be the basis for process diagnosis and monitoring. The stacked RBM that was used as generative model for extracting these features is also discussed in this chapter.

4.1 Feature Extraction Overview

A multilayer auto encoder with RBM training was used to extract features from process data. The Van der Maaten's Matlab Dimension Reduction Toolbox (Van der Maaten et al., 2009) was used in this study to extract features. All of the layers of the auto encoder are trained in a single phase as the weights are pre trained with the Restricted Boltzmann Machines (RBM). The RBM is trained using contrastive divergence. All the experiments in the study were performed in MATLAB.

Case studies were made on several data sets that have been studied in literature. The results will then be compared with results of other non-linear approaches. Even though the labels of the data are known, they are not used when features are extracted, the labels will only be used to show the quality of the features extracted.

The methodology used for extracting features is summarized as follows:

Feature Extraction Algorithm:

- Normalize (scale) the data (to be between 0 and 1).
- Train the auto encoder with RBM training
- Reduce the dimensionality as desired, but not less than the intrinsic dimensionality
- Extract features from the reduced dimensionality

The features that will be extracted at this stage will then be used in a process diagnosis & fault detection scheme.

4.1.1 Feature Extraction fault diagnosis

A general outline of fault diagnosis with feature extraction is presented as shown in Figure 4.1, with the aim of designing a framework based on Restricted Boltzmann Machines feature extraction. Process fault detection and identification with unsupervised methods consist of offline and online applications. In the offline routine, normal operating conditions data is used to specify the process and this forms the basis for the model calibration. The online application involves the testing of unseen data against the specified process features to detect whether a fault has occurred.

In fault diagnosis using feature extraction, features are extracted from the original process data **X** of dimension m by a forward-mapping function that reduces this into a feature space **F** of dimension p. A reverse mapping then reconstructs the original variables from the feature space. The residual space **R** is the error between the original variables and the reconstructed variables. See Figure 4.1 (page 40).

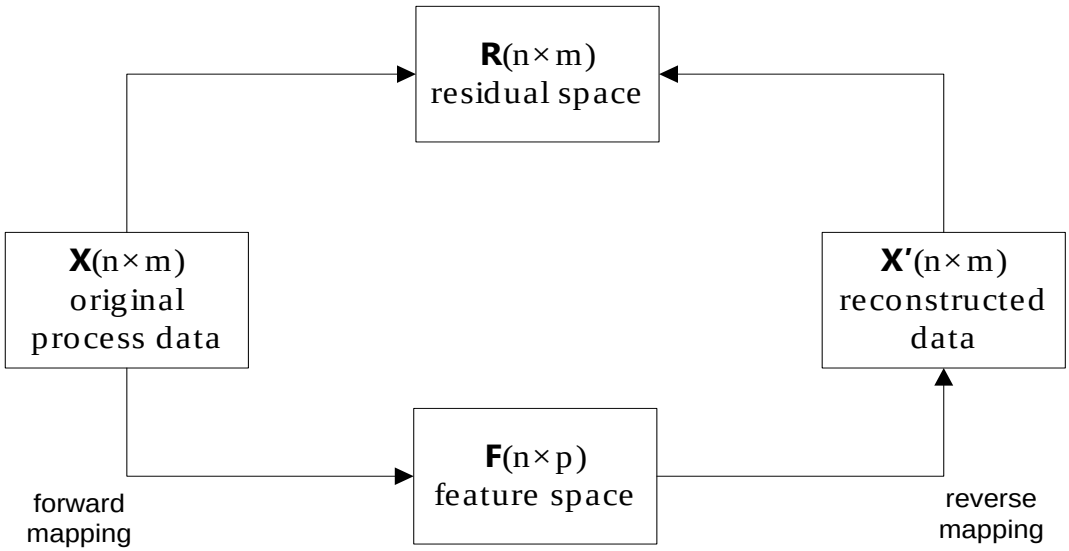


Figure 4.1: Feature Extraction Fault Diagnosis

The algorithm for the feature extraction process is outlined below:

Feature Extraction Fault Diagnosis Algorithm:

- Original process data/variables refer to the physical measurements made during normal operating conditions and used as input to feature extractive algorithms, represented by the matrix **X** (n samples of m dimensions).
- Features refer to the directional components or manifold definitions extracted by dimension reduction.
- Scores are the sample-specific values along the defined features, represented by the matrix **F** (n samples of p dimensions).
- The mapping function **G** (.) calculates the scores from the process variables: **F** = **G**(**X**).
- Reconstructed variables are represented by $\hat{\boldsymbol{X}}$ (n samples of m dimensions), and are the approximation of the original variables from the scores.
- The demapping function **H** (.) calculates the reconstructed variables from the scores:
- $\hat{\boldsymbol{X}} = \boldsymbol{H}(\boldsymbol{F})$

4.2 Design Issues with RBM Fault Diagnosis

Fault diagnosis requires more than simply a feature extraction step, as shown in the general fault detection and identification algorithms. The design issues as applicable to creating a RBM feature extraction fault diagnostic method are presented here.

4.2.1 Reduced feature space dimension

Selecting the number of features that capture significant information is a challenge with no clear solution. One approach, the crossing (Russell et al., 2000), makes use of multidimensional scaling eigenvalues. Through the plotting of both the structured and unstructured eigenvalues, the reduced feature space dimension can be approximated at that point where the structured eigenvalues cross the unstructured eigenvalues.

4.2.2 Mapping and demapping functions

RBM feature extraction only provides feature scores for the training data, but cannot precisely calculate scores for the data that did not feature during the training. This problem is also present in other nonlinear feature extraction techniques.

In order to calculate the scores and reconstructions of newly introduced data, explicit mapping and demapping models need to be constructed. Since an auto encoder is used with RBM pre-training, the decoder part of it is used for demapping.

4.2.3 Feature characterization

Once features have been extracted, the normal operating conditions region within the feature space must be quantifiably characterized, so that a new process data sample can be evaluated in terms of its membership to this normal operating conditions region. The typical PCA process monitoring approach assumes a

multivariate Gaussian distribution of the features, leading to the Hotelling's T^2 statistic thresholds. A more general approach to the definition of the normal operating conditions region in feature space is thus required. The problem of defining a region where data points occur, separating this from a region where data points do not occur, is termed support estimation. "One-class support vector machines provide data-adaptive, non-Gaussian support estimation" (Jemwa & Aldrich, 2006).

4.2.4 Contribution calculations

Process variables related to a fault condition could be determined by investigating the difference between actual values and reconstruction values, as obtained by the demapping function. The residual based contribution of variable j ($C_{r,j}$) is calculated from the actual variable value X_j and the reconstructed variable value X'_j.

- $C_{rj} = (X_j - X'_j)^2$ **Eqn.21**

The calculation of the lack of fit statistics, (i.e. residual space) that is often used, is the squared prediction error (SPE), which is simply the sum of squares for each row of the residual matrix. This is a measure of the residual between an observation and its projection that is retained in the model.

The control limit for the SPE-statistic is calculated as in MacGregor & Kourti (1995), using the normal inverse cumulative distribution function, the confidence threshold and the eigenvalues of the model. Data that falls outside the confidence limit is considered abnormal data. The process data is assumed to be normally distributed. The individual contributions are normalised with the 99^{th} percentile of the contributions of the normal data to obtain individual relative contributions. An indication of the average individual relative contributions to the SPE-statistic is calculated as an indicator of variable importance.

4.3 Process Monitoring Methodology

4.3.1 Principal Components Analysis Fault Diagnosis

In this section, PCA is discussed in detail because it will be used as a basis for comparison. Although PCA is a linear approach and the fact that RBMs are nonlinear, it is used simply because it is an industry standard at present.

"Principal component analysis (PCA) is a vector space transformation often used to transform multivariable space into a subspace which preserves maximum variance of the original space in minimum number of dimensions" (Garcia-Alvarez et al., 2009) . This subspace will be new variables that are uncorrelated and retain most of the original information, where the variation in the signals is considered to be the information. PCA takes advantage of redundant information existent in highly correlated variables to reduce the dimensionality.

The columns of the matrix P are known as loadings while elements of the matrix T are called scores. The scores are the values of the original process variables, which are mapped into the reduced dimensional space vectors. In the context of feature extraction and fault diagnosis, the score vectors obtained from projecting the process measurements onto the principal components can be considered as the extracted features. The number of principal components to use in calculating the features can be determined by investigating the cumulative variance accounted for by adding additional principal components to the score space.

The Process Diagnostic Toolset (Yzelle, 2012) will be used to perform PCA fault diagnosis. The fault diagnosis scheme will involve an offline application for calibrating the model as well as an online application for testing unseen data.

PCA Fault Diagnosis - Offline Application

Feature Extraction

- For a dataset **X** (n observations by m variables), determine a covariance matrix **E**
  - $\boldsymbol{E} = \frac{1}{n-1}\boldsymbol{X}^T\boldsymbol{X}$ **Eqn.22**
- Calculate the eigenvectors **V** and eigenvalues Λ for the covariance matrix **E** using eigenvalue decomposition
  - $\boldsymbol{E} = \boldsymbol{V}\Lambda\boldsymbol{V}^T$ **Eqn.23**
- Determine the reduced dimensionality a, which captures significant variance, n is the number of components accounting for 90% of cumulative variance.
- Define the loading matrix (principal components)**P** as the first **a** eigenvectors of **V**
- Calculate the principal component scores
  - $\boldsymbol{T} = \mathbf{XP}$ **Eqn.24**

Feature characterization

- Calculate the score distance as the Hotelling's T^2 value:
  - $T^2 = \boldsymbol{X}^T\boldsymbol{P}\Lambda_\alpha^{-1}\boldsymbol{P}^T\boldsymbol{X}$ **Eqn.25**

    where Λ_α is a squared matrix formed by the first a rows and columns of Λ
- Determine the detection thresholds for the score distance
  - $T_\alpha^2 = \frac{(n^2-1)a}{n(n-a)}F_\alpha(a, n-a)$ **Eqn.26**

    where $F_\alpha(a, n-a)$ is the Fisher-Snedecor distribution with $a, n-a$ degrees of freedom and α the level of significance.

PCA Fault Diagnosis - Offline Application continued

Variable reconstruction

- Calculate the reconstructed input variables $\mathbf{X'}$
  - $\boldsymbol{X' = TP^T}$ **Eqn.27**
- Calculate the square prediction errors Q
  - $\boldsymbol{Q_j = (X_j - X'_j)^2}$ **Eqn.28**
- Calculate the residual distance error r
  - $\boldsymbol{r = \sum_{j=1}^{m} Q_j}$ **Eqn.29**
- Determine the detection thresholds for the squared prediction errors:
  - $\boldsymbol{Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}\right]^{1/h_0}}$ **Eqn.30**
  - Where $\theta_i = \sum_{j=\alpha+1}^{n} \lambda_j^2$, $h_0 = \frac{2\theta_1\theta_3}{3\theta_2^2}$ and c_α is the value of the normal distribution with α the level of significance.

Contribution calculations

- Calculate the score distance contributions $\boldsymbol{C_s}$
  - $\boldsymbol{C_{s,j} = T\Lambda^{-1}P_j^T X_j}$ for variable j **Eqn.31**
- Calculate the residual distance contributions $\boldsymbol{C_r}$
  - $\boldsymbol{C_{r,j} = Q_j}$ **Eqn.32**

PCA Fault Diagnosis - Online Application Algorithm:

For unseen data

- **Feature calculation**
  - Scale new data using scaling model of normal operating conditions.
  - Calculate the principal component scores
- **Feature characterization**
  - Calculate the score distances
- **Detection in feature space**
  - Compare the score distance to detection threshold s_α
  - Detection if value exceeds detection threshold
- **Reconstruction calculation**
  - Calculate the reconstructed variables
  - Calculate the squared prediction errors
  - Calculate the residual distances
- **Detection in residual space**
  - Compare the residual distance to detection threshold r_α
  - Detection if value exceeds detection threshold
- **Contribution calculations**
  - Calculate the score distance contributions
  - For identified faults, compare with upper limits of score distance contributions of normal operating conditions
  - Calculate the residual distance contributions $\boldsymbol{C_r}$
    - For identified faults, compare with upper limits of residual distance contributions of normal operating conditions

4.3.2 Restricted Boltzmann Machines Fault diagnosis

Model Calibration (Offline Application)

The first part in this process diagnosis scheme is to calibrate the model. This involves the normal operating conditions (NOC) data. The algorithm for calibrating the model is as follows:

Model Calibration (Offline Application) Algorithm:

- **Feature extraction:**
  - Scaled original input variables X representing normal operating conditions are mapped to features **F**.
  - The Autoencoder with RBM training is used for this feature extraction
- **Variable reconstruction:**
  - Features are demapped to the original input space to obtain reconstructed variable values X'.
  - The Multilayer Linear Regression is used for the reconstructions
- **Reconstruction Characterisation**
  - A statistic is calculated to summarize the information not captured in the feature space, the residual distance (r).
  - A detection threshold for the residual distance (r_α) is determined using the percentile approach
  - Calculate the squared reconstruction errors (Q).
- **Contribution calculations:**
  - Feature space contributions (C_s) can be determined by decomposing the score distance into input variable contributions
  - Residual space contributions (C_r) can be determined by decomposing the residual distance into input variable contributions

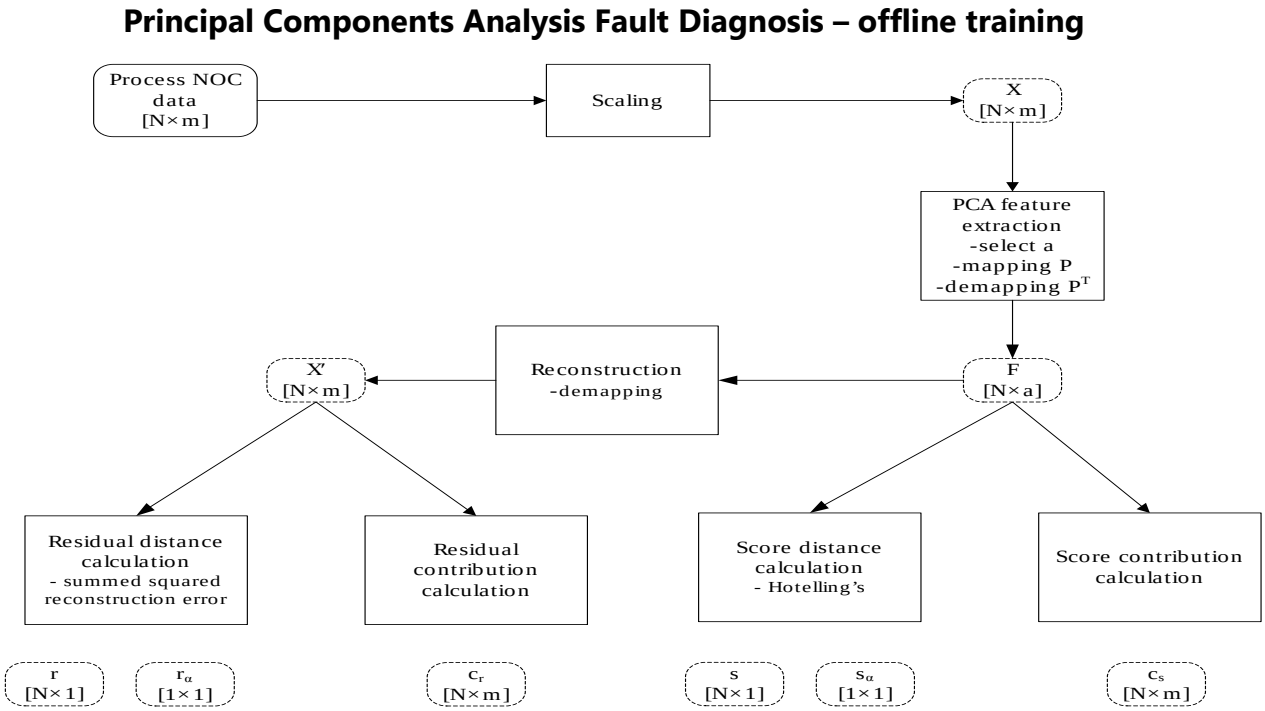
Online Application

Online Application Algorithm:

For unseen data:

- **Feature calculation:**
  - The process inputs for unseen data are scaled using the Restricted Boltzmann Machines for the normal operating conditions data.
- **Reconstruction Characterisation**
  - With the feature scores of the unseen data, reconstructed input variables can be obtained, and squared reconstruction errors (Q) calculated.
  - Reconstructed variable values are used to calculate the residual distance (r)
- **Detection in residual space:**
  - Residual distances are compared to its detection threshold (r_α), and detection is indicated if this statistic exceeds the detection threshold.
- **Contribution calculations:**
  - Feature space and residual space contributions (C_s and C_r) are calculated with the offline algorithm.
  - For detected faults, these contributions can be inspected to identify the fault. The upper limit of contributions for normal operating conditions provides a useful comparison.

A summary of the on-going description of the process methodology used in this study is shown in Figure 4.2 (page 49).



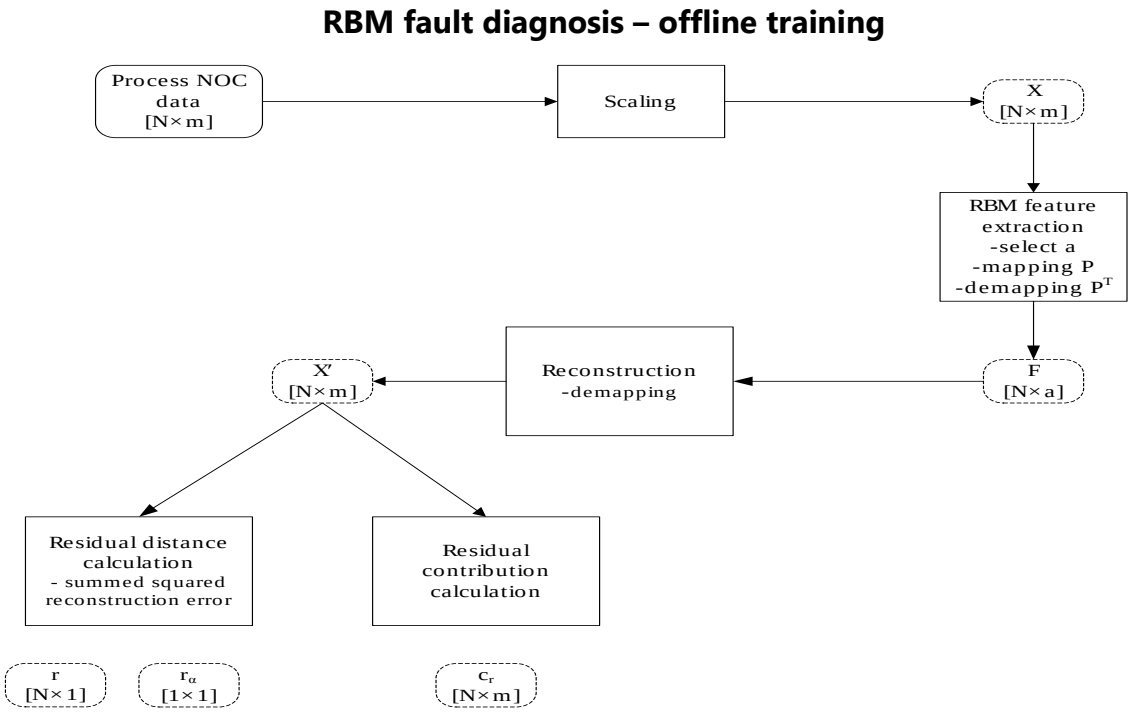


Figure 4.2: Schematic of PCA and RBM fault diagnosis training algorithms

4.4 Experimental Procedures Implemented

In this study, three datasets are used and the various models performed. This section gives an overview of the experimental procedures that were performed. The results that were obtained from each dataset are shown in Chapter 5. Even though PCA and RBM are compared in all the case studies, various other non-linear approaches are compared, using the results which are found in literature. All the experiments that are discussed in the thesis were done using the MATLAB software.

4.4.1 PCA

The PCA execution uses MATLAB's built in function princomp and a custom implementation of the details with regard to process monitoring, the Process Diagnostic Toolset (Yzelle, 2012) MATLAB toolbox.

4.4.2 RBM

The Autoencoder was implemented using a modified code from the dimensionality toolbox written by Van der Maaten. The code was modified to be able to work with Gaussian units and also to allow performing the back-propagation using MATLAB's neural network toolbox. The algorithm uses Levenberg Marquart back propagation for networks with fewer than 100 weights, otherwise the scaled conjugate gradient descent algorithm was used. The auto encoder is pre-trained using the Restricted Boltzmann Machines.

For the RBM that was used, a learning rate of 0.0015 with a weight decay of 0.00015 was chosen. The momentum that was used equals 0.8 with a batch size of 30. All these parameters were chosen with trial and error. Weight decay works by adding an extra term to the normal gradient. Weight decay improves the generalization to new

data by reducing the over fitting to the training data and shrinks useless weights that are found in the hidden units. Weight decay also unsticks the hidden units that have developed larger weights early during the training phase of the algorithm

CHAPTER 5 CASE STUDIES

This section gives the results obtained from various case studies that were done to validate the results.

5.1 PGM Data

In this case study, the feature extraction from textural information of froth structures generated from a platinum flotation plant is considered. The platinum group metals flotation data set consists of five inputs from digitized images of froths obtained from a PGM flotation plant, with three different observed operating regimes present and has a sample size of 297 (Jemwa & Aldrich, 2006).

The PGM data (datapgm) has three categories of data, labelled A, B and C. The data that is in category B was taken as the normal operating condition (NOC) data. This data therefore was used to calibrate the model during the offline mode. The NOC data has five variables with 99 measurements. The Restricted Boltzmann Machine was used to extract features from this data as discussed in section 4.3.2.

The training data consists of five image features characterizing the froth. Class B represents the normal operating conditions, while Classes A and C represent fault conditions. Class A is the data that represents fault 1, whereas Class C is the data that represents fault 2 in the case study.

5.1.1 Feature Extraction

Three features were extracted from the PGM data and used as a basis for process monitoring. An auto encoder with RBM pre-training was used to extract the features. The results for the features that were obtained are shown as follows:

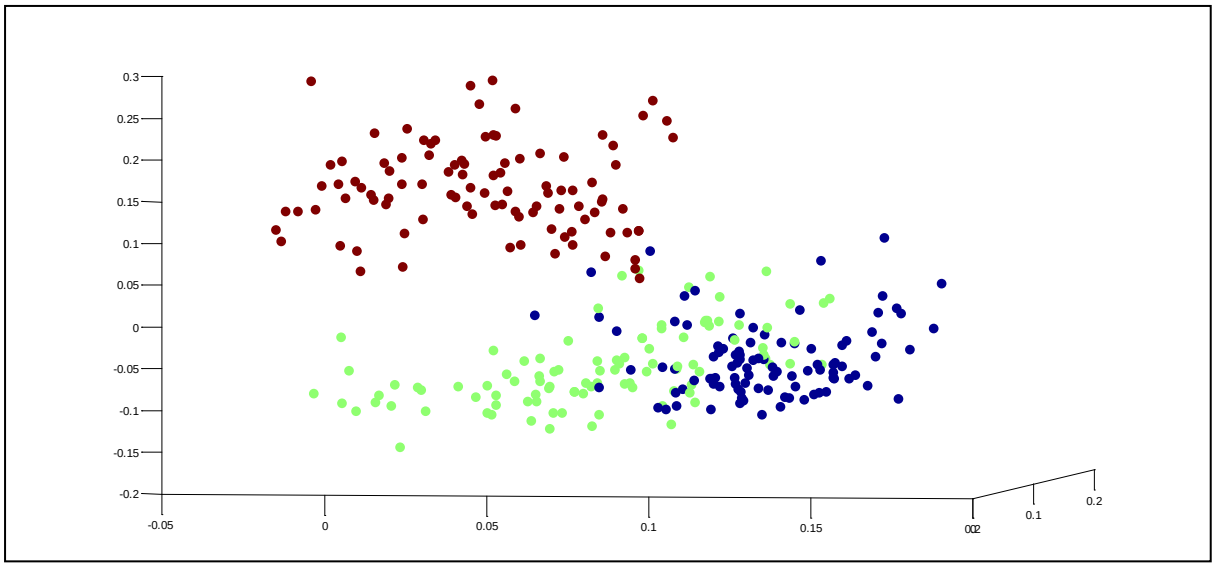


Figure 5.1: Three dimensional features obtained for the copper flotation dataset

It can be seen from Figure 5.1 that there is an almost complete separation of one class. Even though there is an overlap between the two classes represented by blue and green, the auto encoder shows reasonable separability.

5.1.2 Selecting number of features

Variance and cumulative plots were generated to aid in the selection of features to select for both the PCA and the RBM algorithms. Figure 5.3 (page 54) shows the scree plot for the discrete and cumulative latents from the cumulative components analysis. This gives an indication of the variance explained by each principal component. Since the cumulative latent exceeds 0.8 at k=2 as shown in Figure 5.3 (page 54), two features will be used, as they account for 84% cumulative variance in the training NOC data.

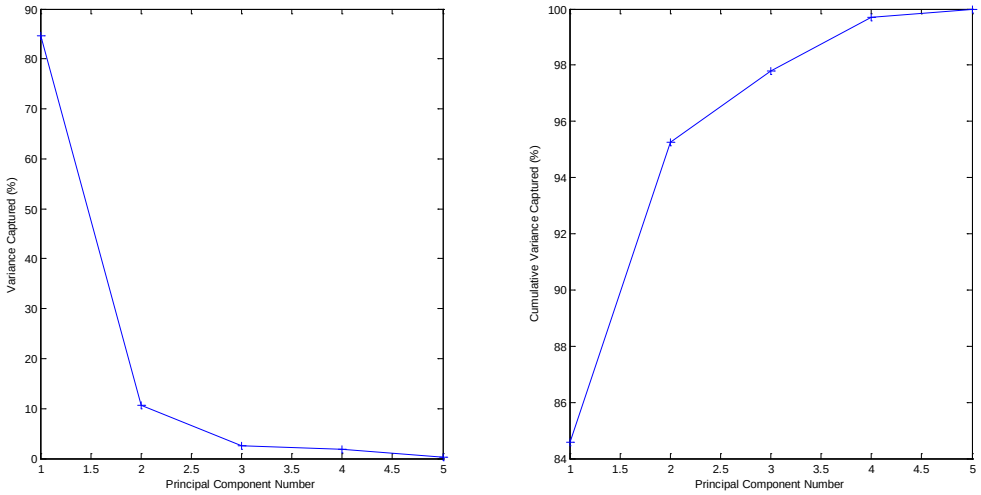


Figure 5.2: Scree plot and cumulative variance from PCA Analysis

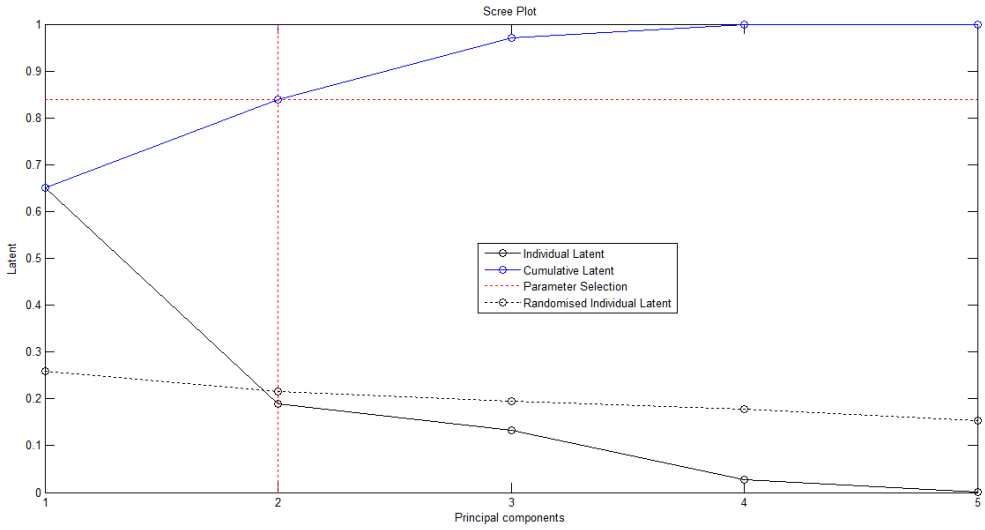


Figure 5.3 Selection of number of features

For this study, three features were used for process monitoring using Restricted Boltzmann Machines (refer to Table 4 (page 56), as it shows the lowest MSE). The reconstruction correlation of the three features is 83.6%, which will give representative models. From the graph in Figure 5.4 (page 55) and Figure 5.6 (page 56), we can see the reconstruction correlations for the NOC data.

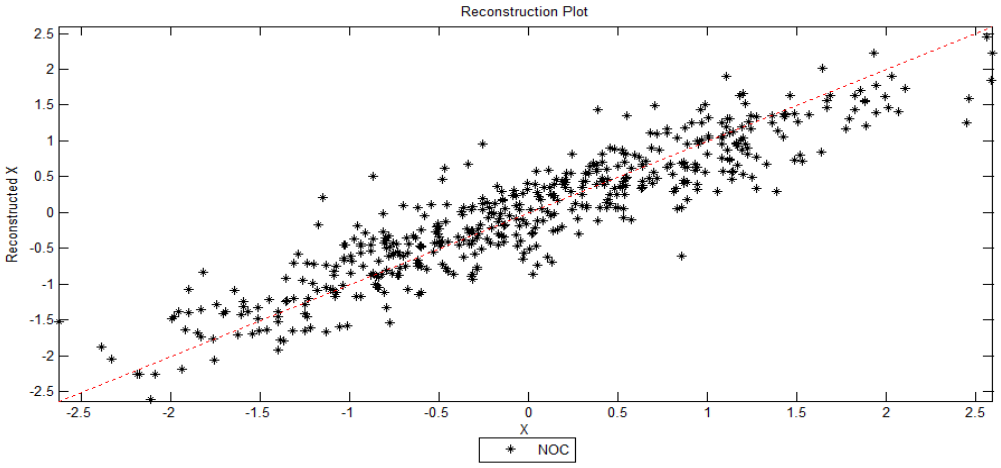


Figure 5.4: Reconstruction Plot using PCA

In this case study, because of the dimensionality of the data, only a simple two layer RBM's were used. The network structure that was used in the training phase is shown in Figure 5.5. The network is trained on NOC data. First, the network was trained with five input nodes and 5 output nodes.

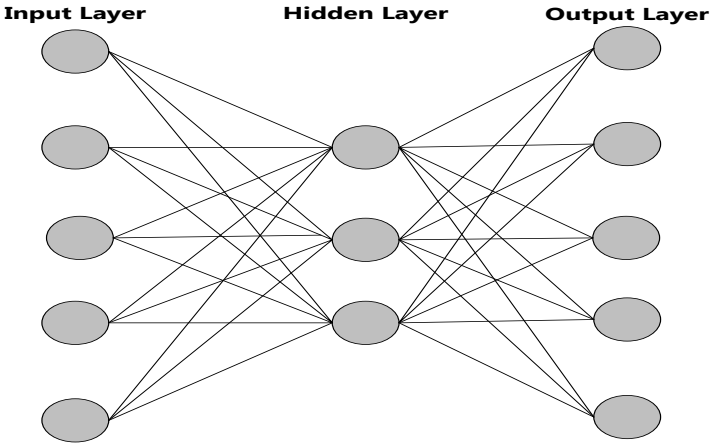


Figure 5.5: The Network (5-3-5 architecture)

The results of the error, for each architecture trained to determine the number of hidden units, are shown in Table 4 (page 56). It is evident from the table that the network with 3 hidden units had the lowest error, hence it was used. The number of neurons in the mapping and demapping layers does not have much of an impact on the performance of the network.

Increasing the number of neurons for the hidden layer improves the network performance (mean square error decreases). The auto encoder with RBM pre training algorithm works well with high dimensional data, because of the nature of the training that is involved. Since, in this case, the dimension of the data is small, adding more hidden layers in the network does not improve the performance of the network significantly.

Table 4: MSE of network structure for PGM

Number of hidden units	MSE
2	0.0354
3	0.0212
4	0.02356
5	0.02319

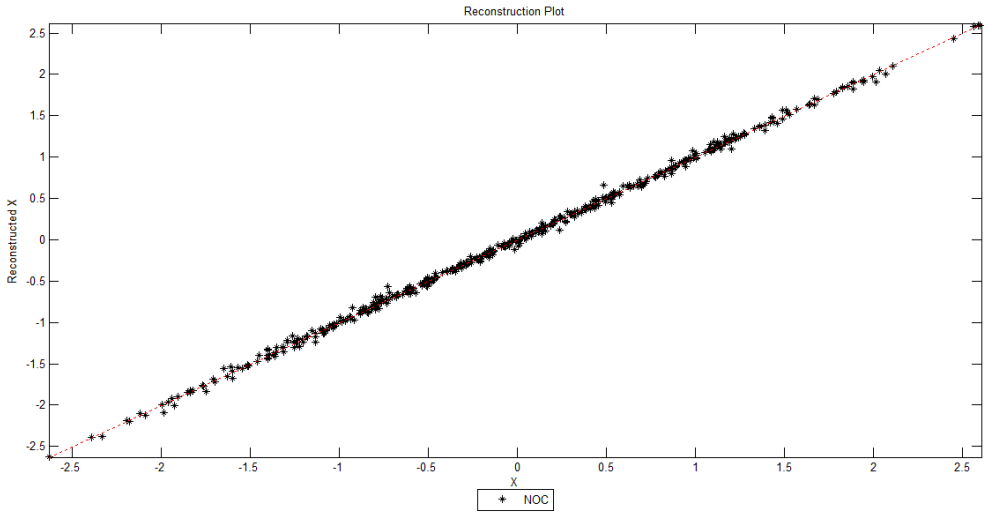


Figure 5.6 Reconstruction Plot using RBM

5.1.3 Fault detection

The missing alarm rates for the two faults of the PGM data are shown in Table 5. From the table, it can be seen that on fault 1, PCA performed better than RBM. This can be expected from data that has low dimensionality, as it means that the RBM

network does not have sufficient data points during training and hence the network gets trapped in local minima. Fault 2 shows a slightly different trend as compared to the first fault. It should be mentioned here that in terms of the chemical process, higher missing alarm rates have negative consequences on the process improvement initiatives. This is because the fault conditions are not identified on time, and hence causes a delay in ensuring that initiatives are done to bring the process back to normal operating conditions. Furthermore, it may also result in the process monitoring methodology to be reviewed, and that the process diagnosis framework has to be continuously trained to cater for many operating conditions.

Table 5: Missing alarm rates for PGM

Fault	PCA	RBM
Fault 1	0.01	0.02
Fault 2	0.95	0.67

The variation of each observation within the PCA model (i.e. PCS) is indicated by Hotelling's T^2 statistic. Figure 5.7 (page 58), shows the T statistic using PCA. The control limits are indicated in red. The first 99 samples represent the NOC samples and the rest represent the fault data samples thereof.

In Figure 5.7 (page 58), both the SPE (Sqaure Prediction Error) and Hotelling' T^2 statistic are displayed for fault 1. It can be clealry observed that the occurance of the fault at t=100 is identified. Both the 95% and the 99% control limits detect the fault. It is evident as shown in Figure 5.7 (page 58), that the SPE did better than the T^2.

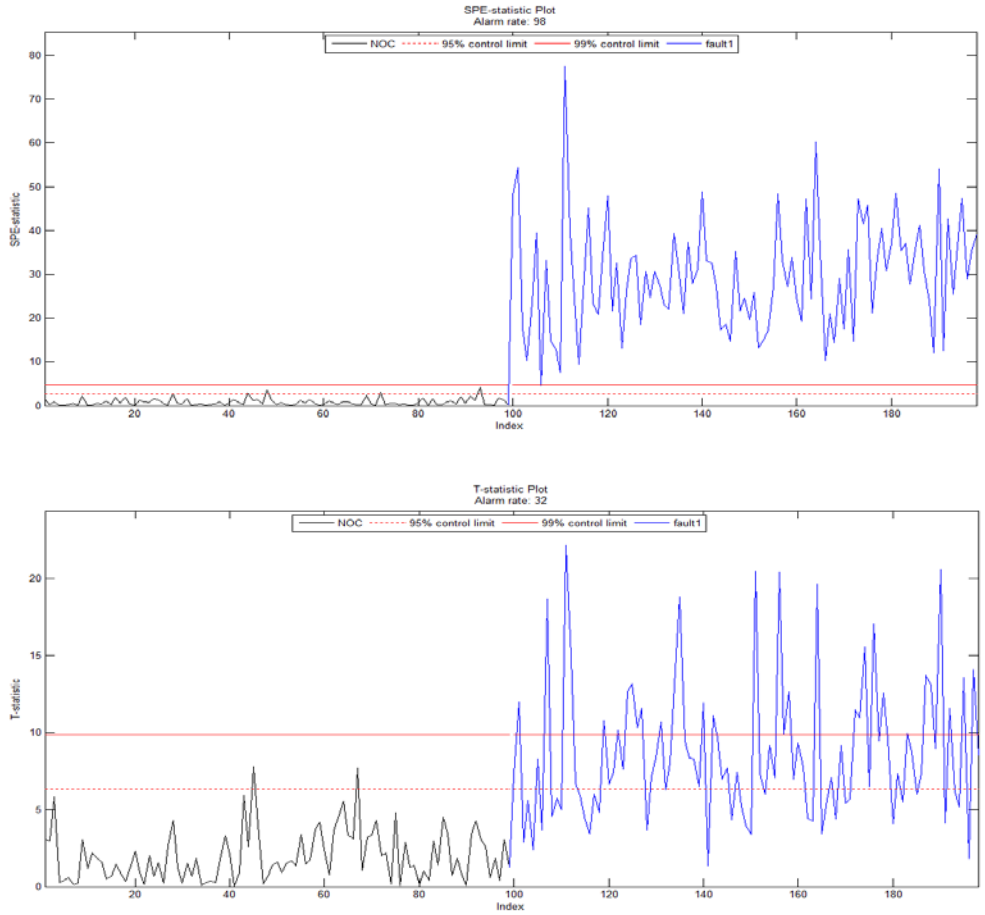


Figure 5.7: SPE and Hotelling's T-statistic using PCA of Fault 1 datapgm

The RBM squared prediction error (Q-statistic) is shown in Figure 5.8 for the PGM flotation data. From the graph, it can be observed that the RBM also was able to identify the fault when it occurred.

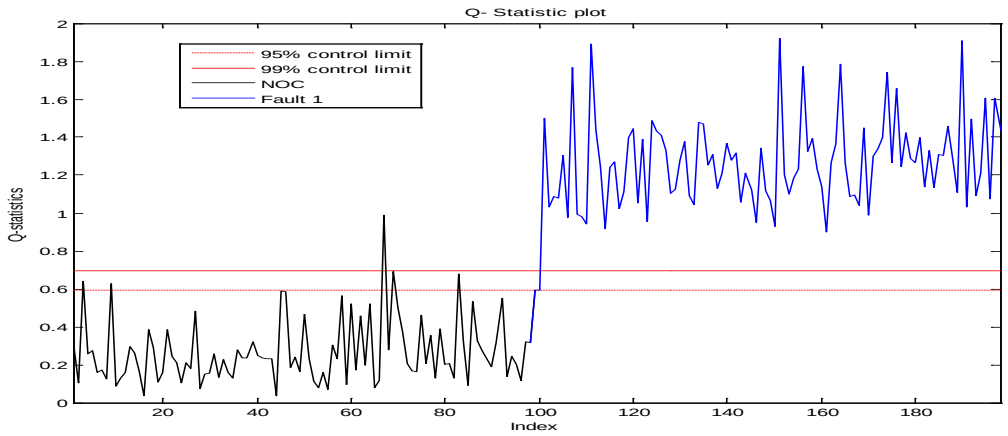


Figure 5.8: Q-statistic using RBM of Fault 1 datapgm

Similarly, the fault diagnosis was carried with the second set of data, to see how fault 2 will be diagnosed. The results are shown in Figure 5.9 and Figure 5.10. Both algorithms did not do well on fault 2.

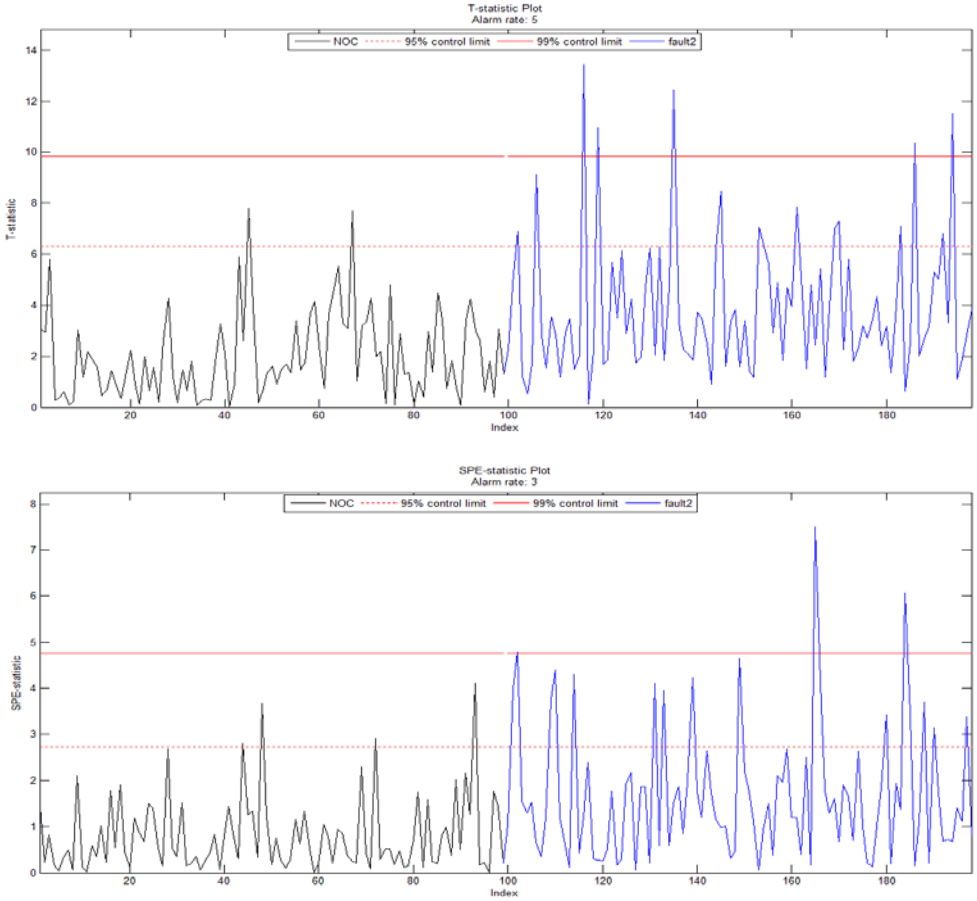


Figure 5.9: SPE and T-statistic using PCA of Fault 2 datapgm

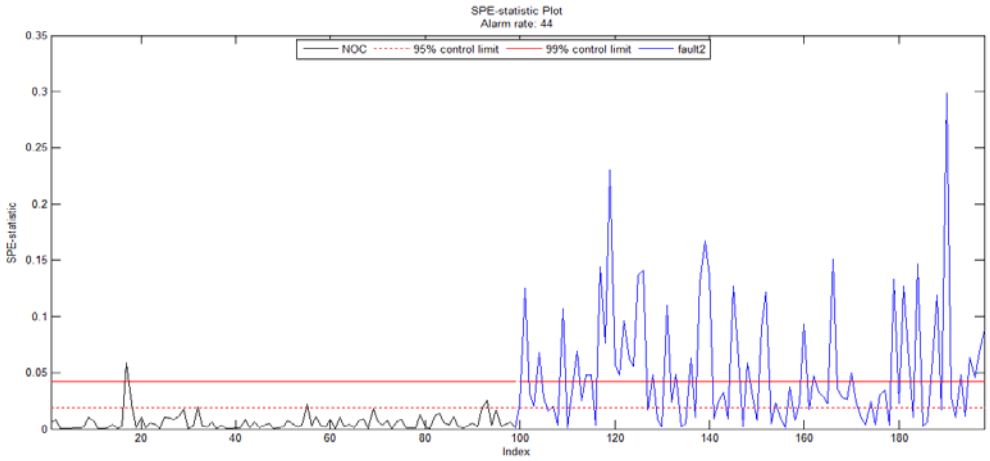


Figure 5.10: Q statistics using RBM of Fault 2 datapgm

5.1.4 Variable Contributions

The variable contributions plots are shown in this section for the two faults, both using PCA and RBM. These were obtained from the Q/SPE or the Hotelling's T^2 statistic. The contribution plots will identify which variables contributed to the Fault that is identified. From Figure 5.11, PCA identified variables 1 and 2 as the most contributing to the Fault represented by the data.

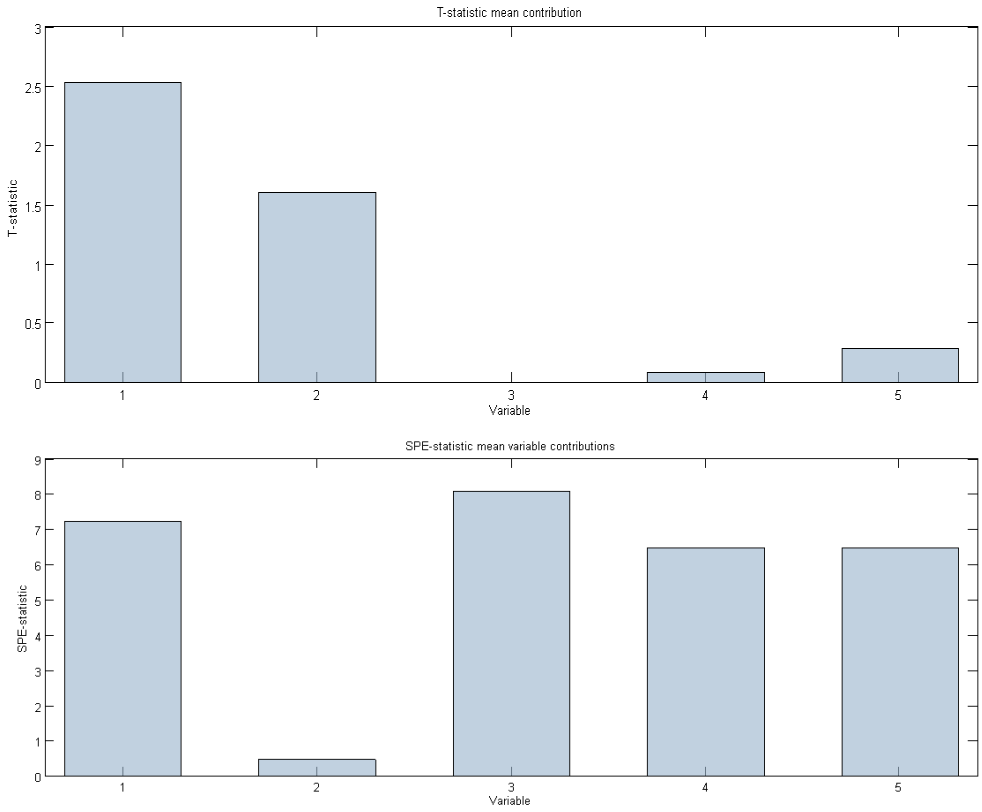


Figure 5.11: SPE and T-statistic contributions using PCA of Fault 1 datapgm

The RBM identified variables 2, 3 and 5 as the ones contributing to the fault condition, at 99% confidence limit. Figure 5.12 (page 61) shows the contribution plots generated using the RBM.

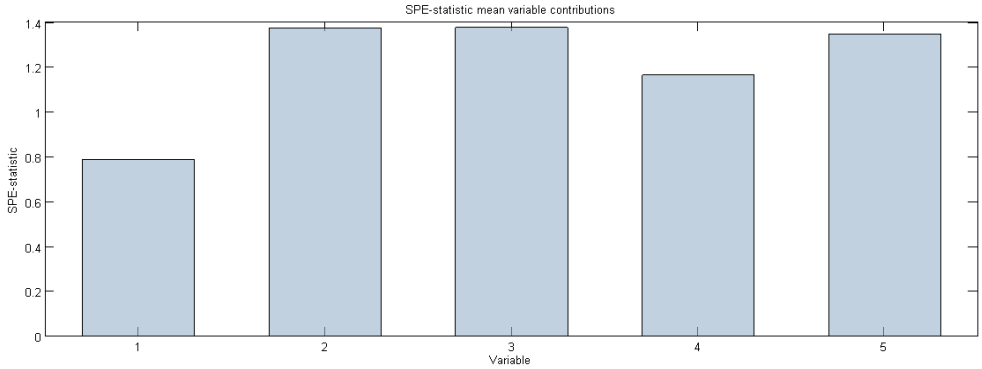


Figure 5.12: Contribution plots using RBM of Fault 1 datapgm

A similar approach was also conducted for fault 2. The results obtained using PCA are displayed in Figure 5.13. The graph shows that variables 2, 3 and 5 contributed most with respect to fault 2.

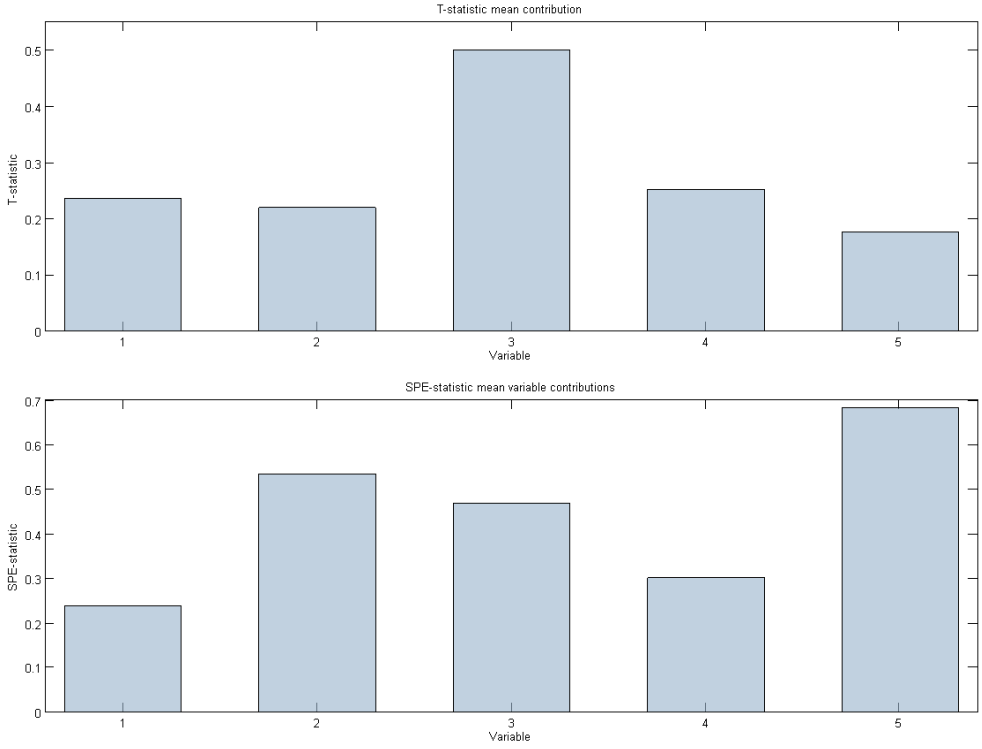


Figure 5.13: SPE and T contribution plots using PCA of Fault 2 datapgm

The RBM was also used to make the variable contribution plots. Figure 5.14 (page 62) shows that RBM identified variable 4 and variable 5 as the variables that contributed most to the fault condition represented by Class A data.

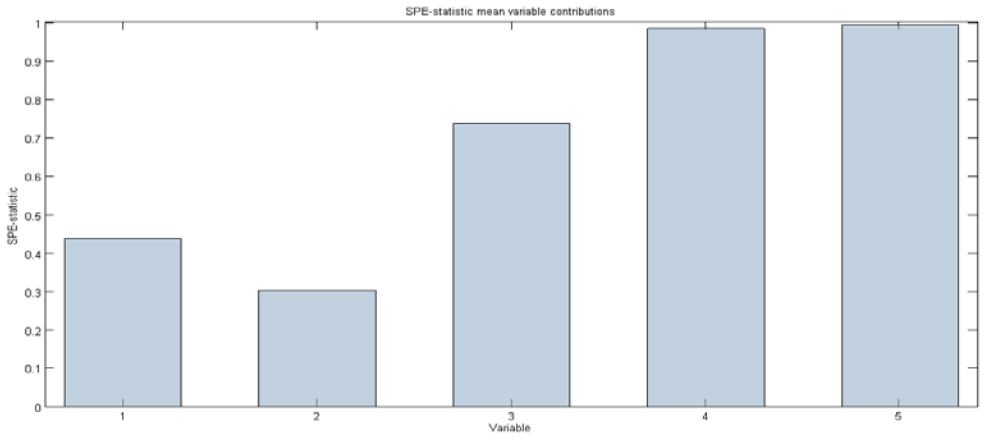


Figure 5.14: Contribution plots using RBM for Fault 2 datapgm

Since there is no prior knowledge of variables that contributes to these faults in this case study, it is difficult to identify which techniques correctly identified the correct variables.

5.2 Copper Flotation Data Set (datacop)

The copper flotation data sets consist of ten inputs extracted from digitized images of froths obtained from a copper flotation plant, with four different observed operating regimes function as class label. This data set contains 490 samples (Jemwa & Aldrich, 2006).

The four cases are briefly described below: Class 1 represents an ideal froth structure that has bubbles that are well loaded with minerals. Class 2 represents a deep, well drained froth with a polyhedral froth structure, while Class 3 represents a tough froth with an ellipsoidal structure, which might have been caused by too low a pulp level, too high specific gravity or flotation of a particular type of particles. Class 4 represents an excessively stable, stiff froth that might be attributed to low pulp levels.

5.2.1 Feature Extraction

Four features were extracted from the Copper flotation data and used as a basis for process monitoring (as shown in Table 6, the lowest MSE network contains 4 features). An auto encoder with RBM pre-training was used to extract the features. The results for the features that were obtained are shown as follows:

5.2.2 Selecting number of features

Variance and cumulative plots were generated to aid in the selection of the number of features to select for both the PCA and the RBM algorithms. Figure 5.15 shows the scree plot for the discrete and cumulative latents from the cumulative components analysis. This gives an indication of the variance explained by each principal component. Since the cumulative latent exceeds 0.8 at k=3 as shown in Figure 5.16 (page 64), two principal components will be used as they account for 90% cumulative variance in the training NOC data.

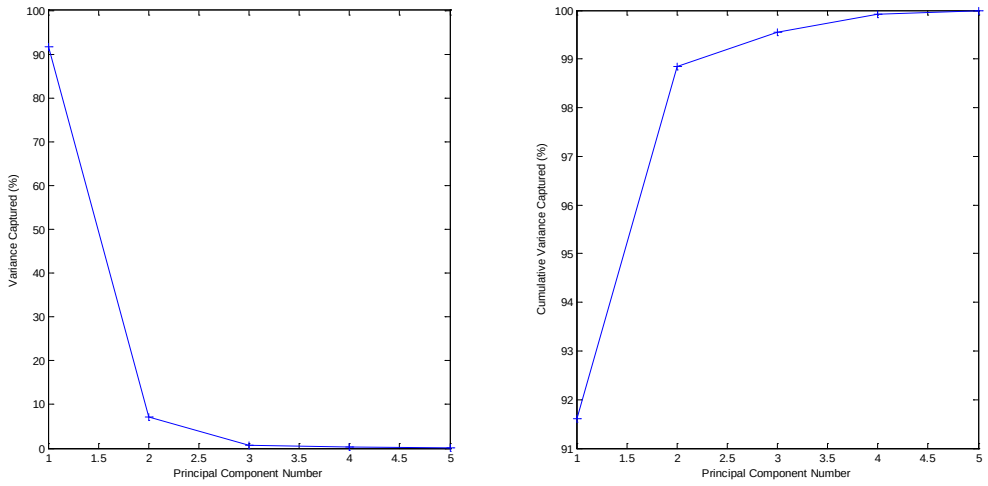


Figure 5.15: Scree plot and cumulative variance from PCA Analysis

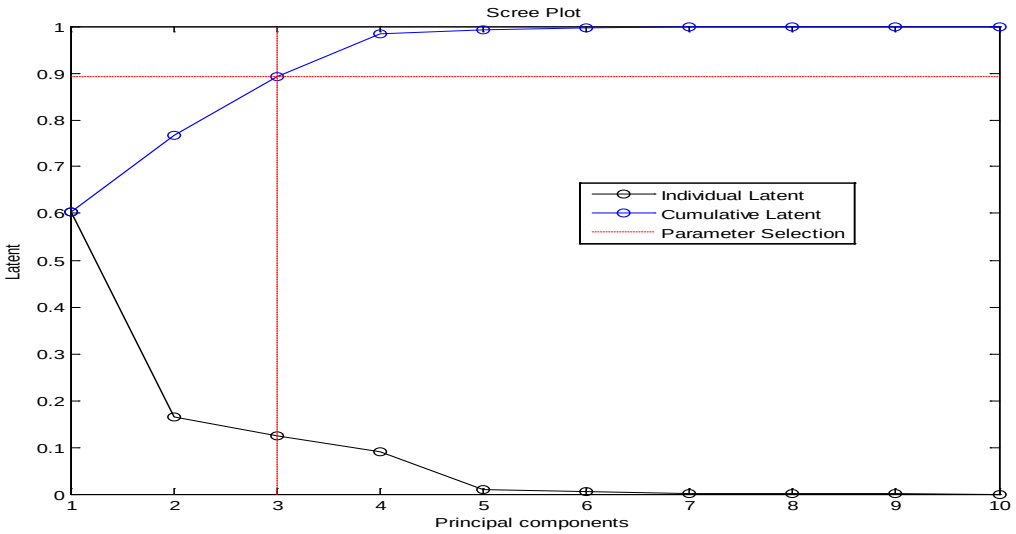


Figure 5.16 Selection of number of features

With the RBM, similar to the previous case study due to the dimensionality of the data, only simple two layer RBMs were used. The results, for each architecture that was trained to determine the number of hidden units are shown in the following table (Table 6). It can be seen from the table that the network structure that contains 4 hidden units had the lowest error, and hence was used. The number of neurons in the mapping and demapping layers does not have much of an impact on the performance of the network. The network performance (mean square error) does not improve much with an increase in the number of neurons for the input and output layers. Increasing the number of neurons for the hidden layer improves the network performance.

Table 6: MSE of network structure for COP

Number of hidden units	MSE
2	0.5392
3	0.5154
4	0.5115
5	0.5237
6	0.52218

5.2.3 Fault detection

The missing alarm rates for the two faults of the Copper flotation data are shown in Table 7. It can be seen that on fault 1, PCA performed better than RBM. These missing alarm rates relate to the similar trends as highlighted in section 5.1.3. Due to the nature of the training of the RBM network, it is possible that the network was jammed in the local minima, as there were not many data points during the training phase. The fact that the network performed better on the second fault means that the nonlinear mapping that was learned during the training phase of the RBM networks had better generalization errors than that of the linear mapping as performed by PCA.

Missing alarm rates do not help the process engineers and operators, as it means that the process monitoring that is designed to identify faults at times does not. As a result, faults are only identified when a lot of damage has been done in the process.

Table 7: Missing alarm rates for COP

Fault	PCA	RBM
Fault 1	0.01	0.61
Fault 2	0.03	0.03

Hotelling's T^2 statistic is a combined variance indicator across variables at each observation. Figure 5.17 (page 66) shows the T statistic using PCA. The control limits are indicated in red. The Q statistic is shown in Figure 5.18 (page 66).

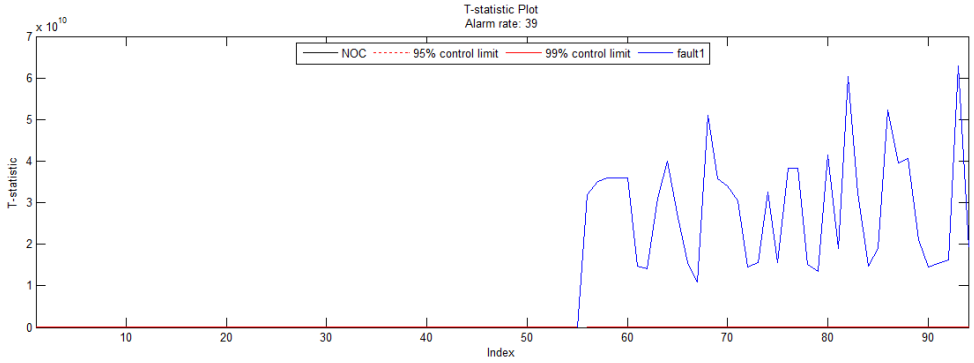


Figure 5.17: Hotelling's T-statistic using PCA of Fault 1 datacop

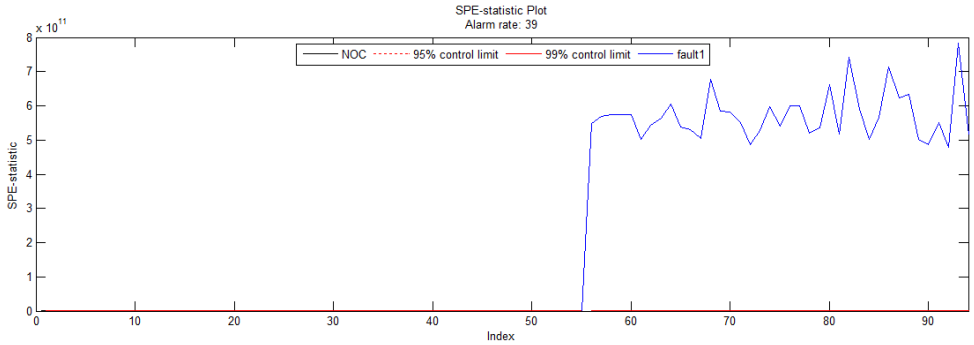


Figure 5.18: Q statistic for PCA of Fault 1 datacop

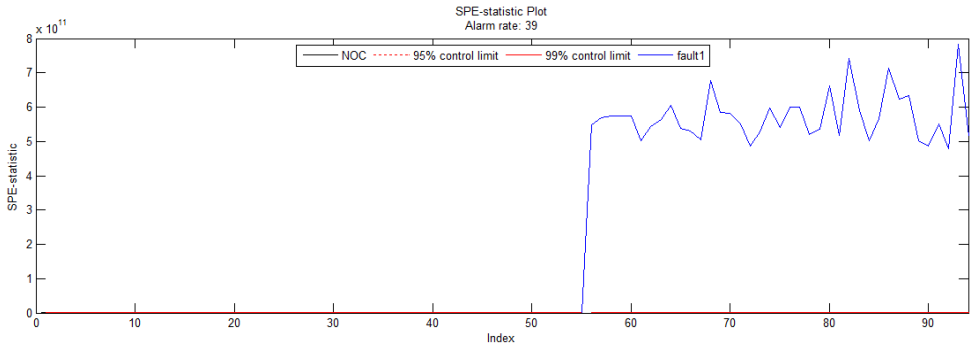


Figure 5.19: Q statistic for RBM of Fault 1 datacop

The RBM squared prediction error (Q-statistic) is shown in figure Figure 5.19 for the Copper flotation data. The graph shows that the fault was diagnosed clearly, though there are some missing alarms, as some of the fault condition data were not detected as faulty.

For fault 2, a similar analysis was done and yielded results that are displayed in Figure 5.19 and Figure 5.20. Both algorithms correctly identified fault 2.

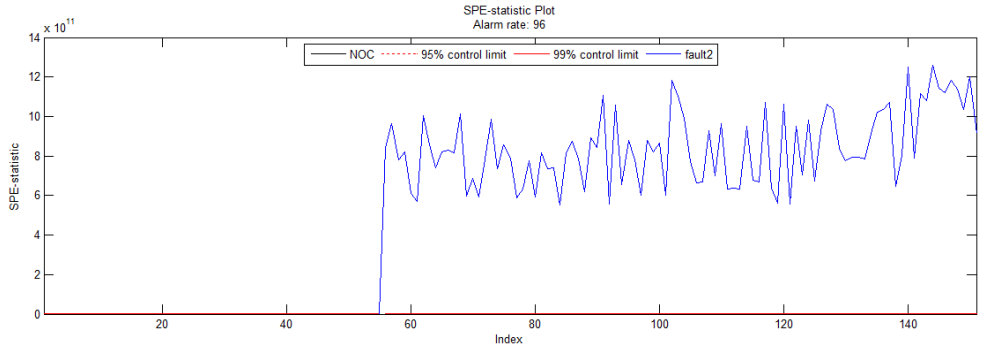


Figure 5.20: Q-statistic using PCA of Fault 2 datacop

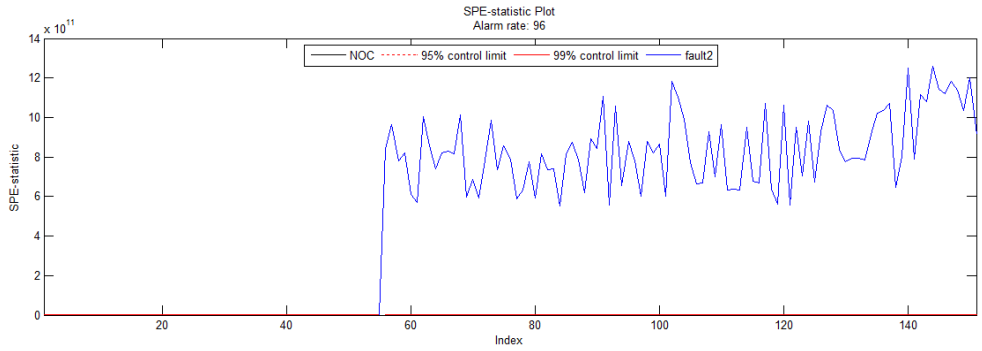


Figure 5.21: Q-statistic using RBM of Fault 2 datacop

5.2.4 Variable Contributions

The variable contributions plots are shown in this section for the two faults, both using PCA and RBM. For fault 1, both PCA (Figure 5.22) and the RBM (Figure 5.23, page 68) identified variables 8 and 9 as the contributing variables.

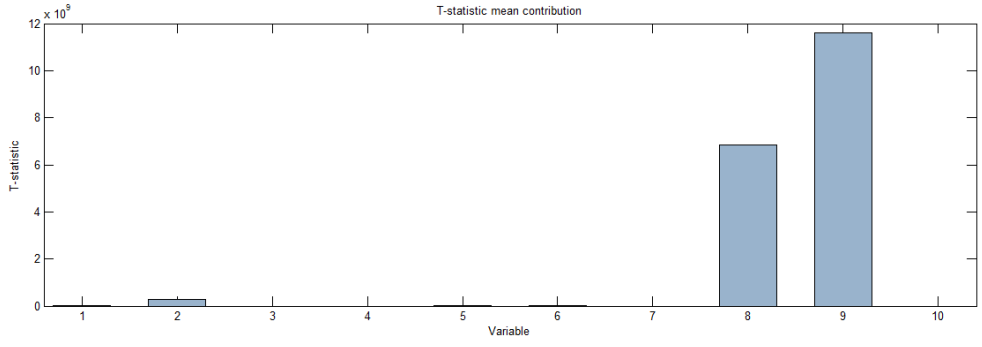


Figure 5.22: T-statistic contributions using PCA for Fault 1 datacop

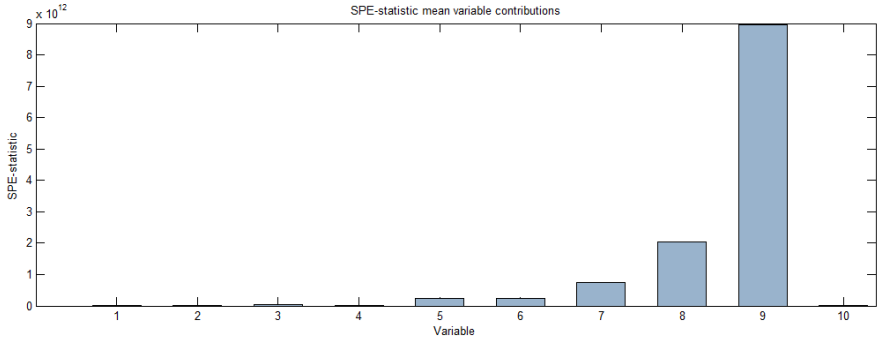


Figure 5.23: Q contributions using RBM for fault 1 datacop

Figure 5.24 and Figure 5.25 show the results for fault 2 where both algorithms identified variables 8 and 9.

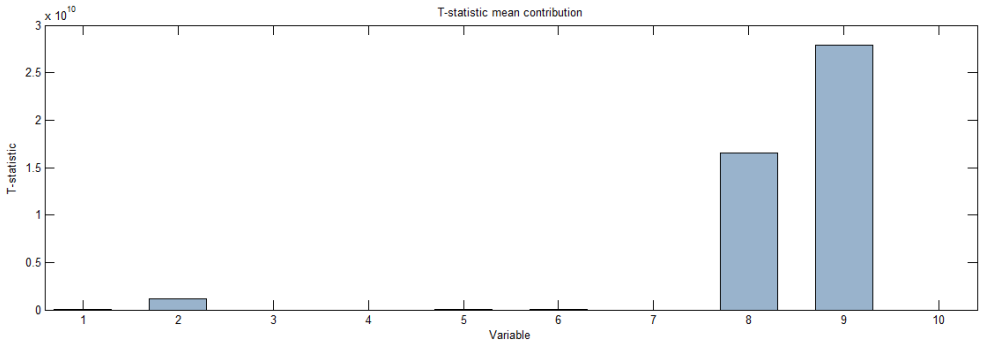


Figure 5.24: Q contributions using PCA for fault 2 datacop

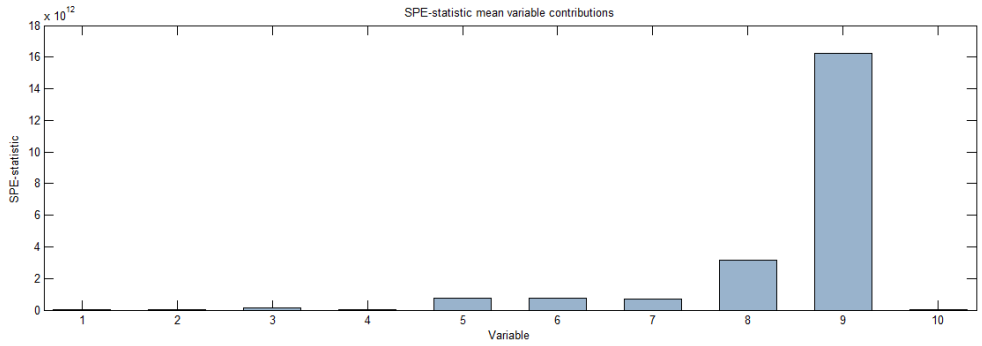


Figure 5.25: Q contributions using RBM for fault 2 datacop

It is important for a fault diagnosis method to be able to correctly identify the variables that are responsible for these faults, as it makes continuous improvements easier. Since there is no prior knowledge of the process, in terms of the variables that contribute to these faults, it is difficult to correctly identify which technique identifies the correct variables. In chemical and process plants, there is knowledge about the

process that can help identify the variables responsible for the faults. In case of the RBM techniques, it may mean, among others, that the network will have to be re-calibrated regularly if the correct variables are not identified, for these faults. More data may also be required to ensure that the network performance is improved during the training phase.

5.3 The Tennessee Eastman Process

The Tennessee Eastman Process (TEP) is a simulation of an actual chemical process developed as a realistic industrial case study useful for plant-wide process control problems, including process monitoring and fault diagnosis (Russell et al., 2000). The process consists of five major units (a reactor, a product condenser, a recycle compressor, a vapor-liquid separator, and product stripper) and eight components labelled A, B, C, D, E, F, G and H. (Downs & Vogel, 1993). The simulation data of the Tennessee Eastman process (with plant-wide control based on proportional (p) and proportional integral (pi)) control is available for normal operating data and the 21 fault conditions (Auret & Aldrich, 2011).

The flow sheet of the TEP is given in Figure 5.26. Components G and H are liquid products produced from the four gaseous reactants, A, C, D and E. The inert product B is also fed to the reactor and the by-product F is produced.

The reactions in the reactor are represented by

Tennessee Eastman Process Equations:

$A(g) + C(g) + D(g) \rightarrow G(liq)$, Product 1

$A(g) + C(g) + E(g) \rightarrow H(liq)$, Product 2,

$A(g) + E(g) \rightarrow F(liq)$, Byproduct,

$3D(g) \rightarrow 2F(liq)$, Byproduct,

The reactions are exothermic, irreversible and approximately first-order with respect to the reactant concentrations. The reaction resulting in G product formation has higher activation energy than reaction of H and, therefore has a high sensitivity to temperature.

Process data contains 41 measured process variables and 11 manipulated variables. Normal operating conditions are represented by 500 samples, with a validation data sample of 960 samples for normal operating conditions. Each of the 21 fault data set consists of 960 samples, with the fault occurring after 161 time steps within the 960 samples.

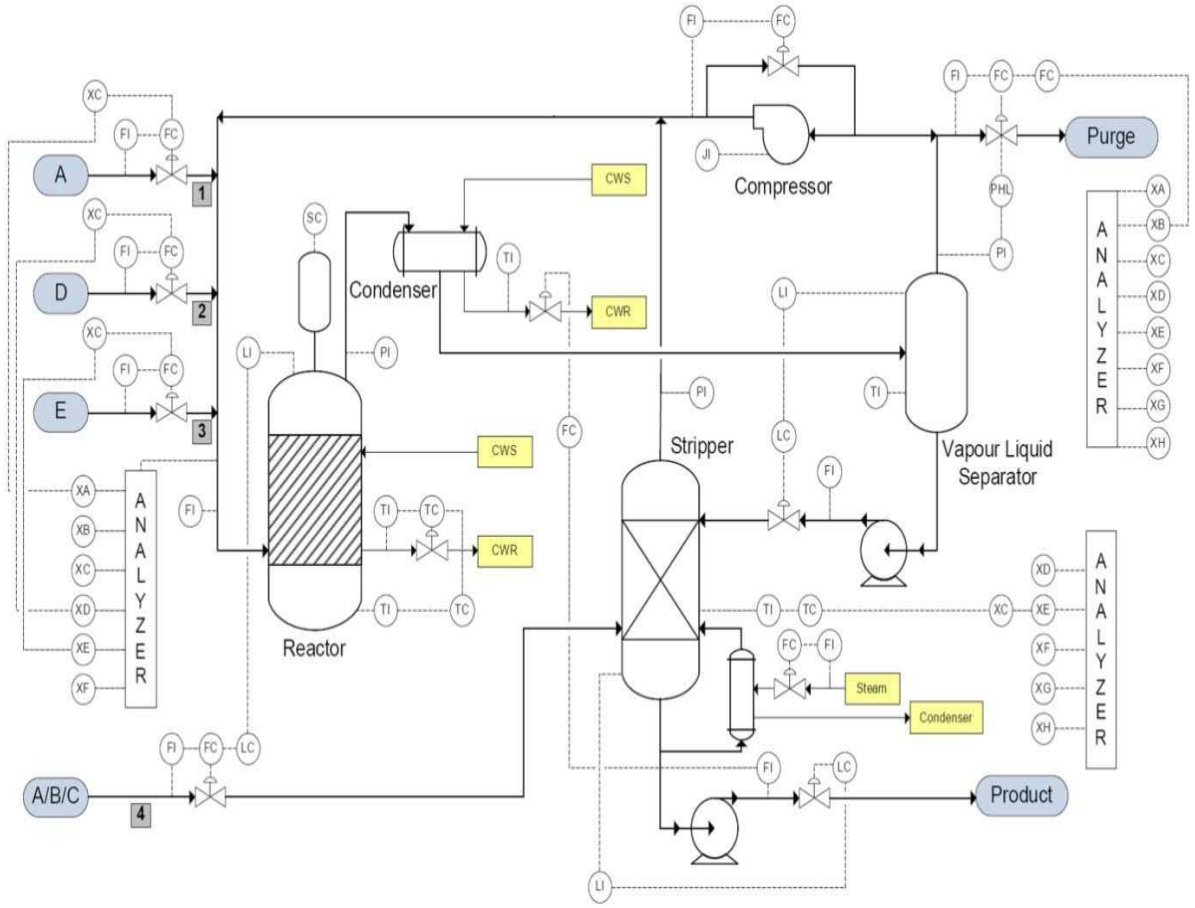


Figure 5.26: Process Flow diagram of the TEP (Russell et al., 2000; Auret, 2010)

Table 8: Process Faults of Tennessee Eastman Process (Russell et al., 2000; Auret, 2010)

Fault	Description	Type
1	A/C feed ratio (B composition constant) – Stream 4	Step change
2	B composition (A/C feed ratio constant) – Stream 4	Step change
3	D Feed temperature – Stream 2	Step change
4	Reactor cooling water inlet temperature	Step change
5	Condenser cooling water inlet temperature – Stream 2	Step change
6	A Feed loss – Stream 1	Step change
7	C header pressure loss (reduced availability) – Stream 4	Step change
8	A, B, C feed composition – Stream 4	Random variation
9	D feed temperature – Stream 2	Random variation
10	C feed temperature – Stream 4	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16 - 20	Unknown	Unknown
21	Valve – Stream 4	Constant position

Table 9: Process Variables of the Tennessee Eastman Process

Variable	Description	Variable	Description
1	A feed – Stream 1(PM)	27	Reactor feed component E (CM)
2	D Feed – Stream 2 (PM)	28	Reactor feed component F (CM)
3	E Feed – Stream 3 (PM)	29	Purge component A (CM)
4	Total Feed – Stream 4 (PM)	30	Purge component B (CM)
5	Recycle flow (PM)	31	Purge component C (CM)
6	Reactor feed rate (PM)	32	Purge component D (CM)
7	Reactor pressure (PM)	33	Purge component E (CM)
8	Reactor level (PM)	34	Purge component F (CM)
9	Reactor temperature (PM)	35	Purge component G (CM)
10	Purge rate (PM)	36	Purge component H (CM)

11	Separator temperature (PM)	**37**	Product component D (CM)
12	Separator level (PM)	**38**	Product component E (CM)
13	Separator pressure (PM)	**39**	Product component F (CM)
14	Separator underflow (PM)	**40**	Product component G (CM)
15	Stripper level (PM)	**41**	Product component H (CM)
16	Stripper pressure (PM)	**42**	D feed Flow – Stream 2 (MV)
17	Stripper underflow (PM)	**43**	E feed Flow – Stream 3(MV)
18	Stripper temperature (PM)	**44**	A feed Flow – Stream 1 (MV)
19	Stripper steam flow (PM)	**45**	Total Feed Flow – Stream 4 (MV)
20	Compressor work	**46**	Compressor recycle valve (MV)
21	Reactor cooling water outlet temp. (PM)	**47**	Purge valve (MV)
22	Separator cooling water outlet temp.(PM)	**48**	Separator product liquid flow (MV)
23	Reactor feed component A (CM)	**49**	Stripper product liquid flow (MV)
24	Reactor feed component B (CM)	**50**	Stripper steam valve (MV)
25	Reactor feed component C (CM)	**51**	Reactor cooling water flow (MV)
26	Reactor feed component D (CM)	**52**	Condenser cooling water flow (MV)

5.3.1 Selecting the number of features

Variance and cumulative plots were generated to aid in the selection of features to select for both the PCA and the RBM algorithms. Figure 5.27 (page 73) shows the scree plot for the discrete and cumulative latents from the cumulative components analysis. This gives an indication of the variance explained by each principal component. From Figure 5.28 (page 73), it can be seen that thirteen principal components will be used in the training NOC data.

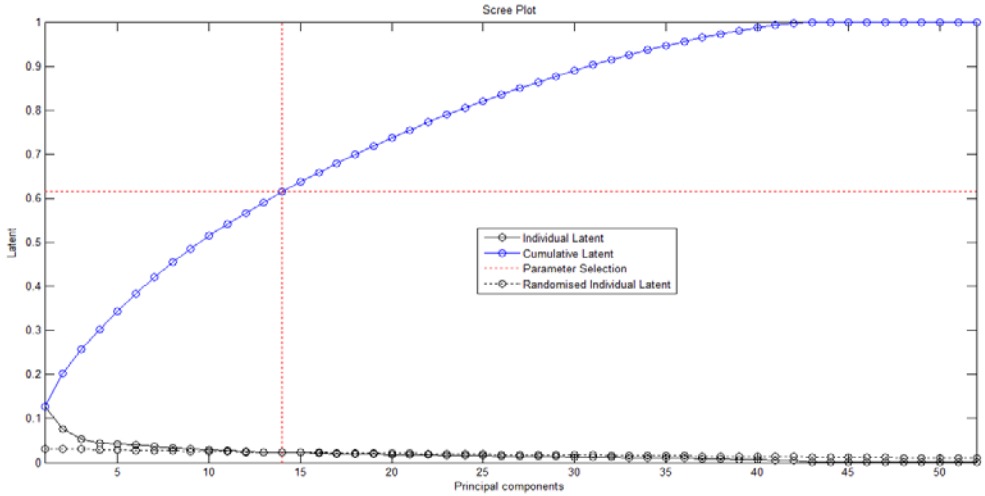


Figure 5.27: Scree plot and cumulative variance from PCA Analysis

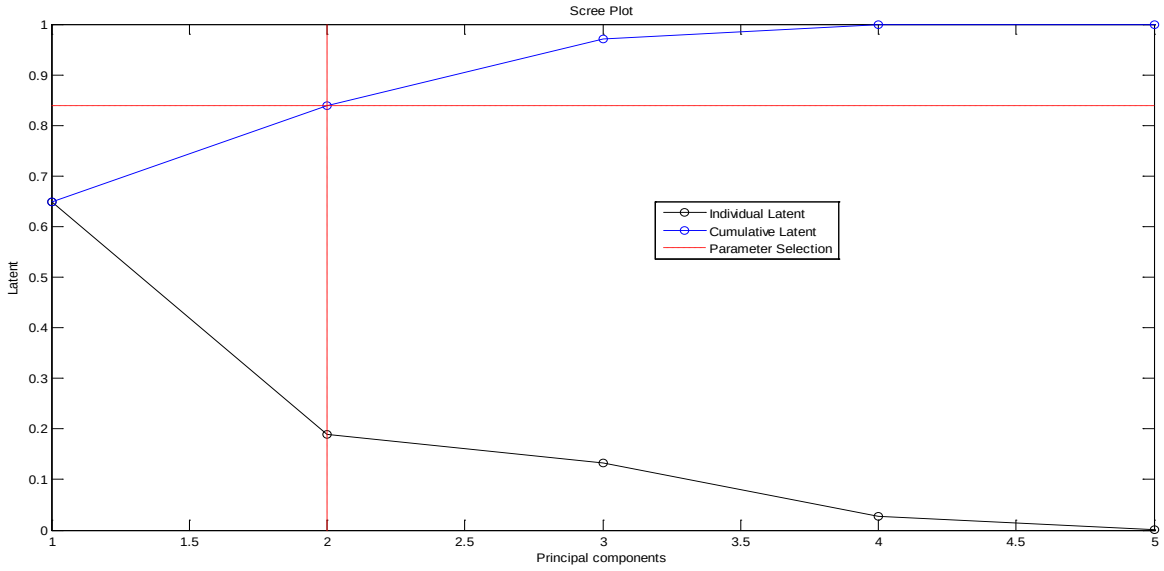


Figure 5.28 Selection of number of features

For the RBM model, ten components were used during the study. The reconstruction correlation of these features is 87.4%, which will give representative models. From the graphs in Figure 5.29 (page 74), we can see the reconstructions for the NOC data. In this case study, stacked RBM's were implemented. Different network architectures were employed to monitor how the features extracted could explain the input data.

Table 10: MSE of network structure for TE

Network Structure	MSE
52-40-10-40-52	0.254
52-10-52	0.390
52-30-10-30-52	0.312
52-30-5-30-52	0.3356
52-20-5-30-52	0.372

The results obtained on different network structures are shown in Table 10. The network structure with the lowest error (MSE) was used in the study. It should be noted however, that due to the nature of the stacking auto encoders and the dimensionality of the data in this case study, increasing the number of hidden layers improves the network performance. The results that are shown in Table 10, is as a result of a variety of trials, but it should be mentioned that a complete exhaustive search would be very expensive to perform due to the computational demands that it would require.

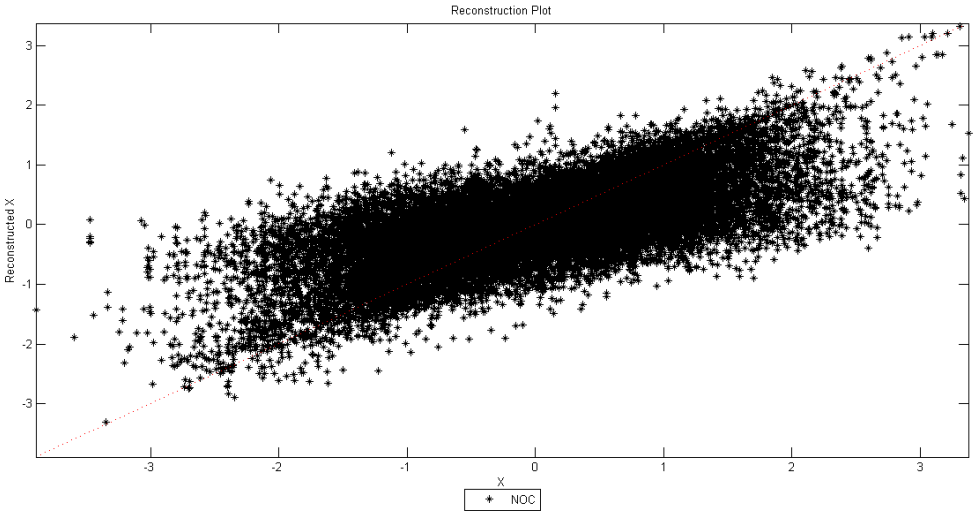


Figure 5.29 Reconstruction plot using RBM for NOC data

5.3.2 Missing alarm rates

The missing alarm rate δ is the fraction of all the known fault samples that are not detected in a data set. The overall missing alarm rate δ is the minimum of the score distance missing alarm rate δ_s and the residual distance missing alarm rate δ_r.

A comparison of the score and residual distance missing alarm rates for PCA with 13 features and RBM with 10 features are given in Figure 5.30. Score and residual distance missing alarm rates for RBM fault diagnosis does not outperform PCA score and residual distance missing alarm rates, but the residual at times performs comparably. PCA score and residual distance alarm rates are similar, with score distance missing alarm rates better (lower) than their residual distance counterpart for some of the faults. A comparison of PCA and RBM missing alarm rates to results obtained in literature for random forests (RF) and Kernel Independent Component Analysis (KICA) are shown in Figure 5.31 (page 76) and Figure 5.32 (page 79) (Auret & Aldrich, 2010b; Zhang & Qin, 2008).

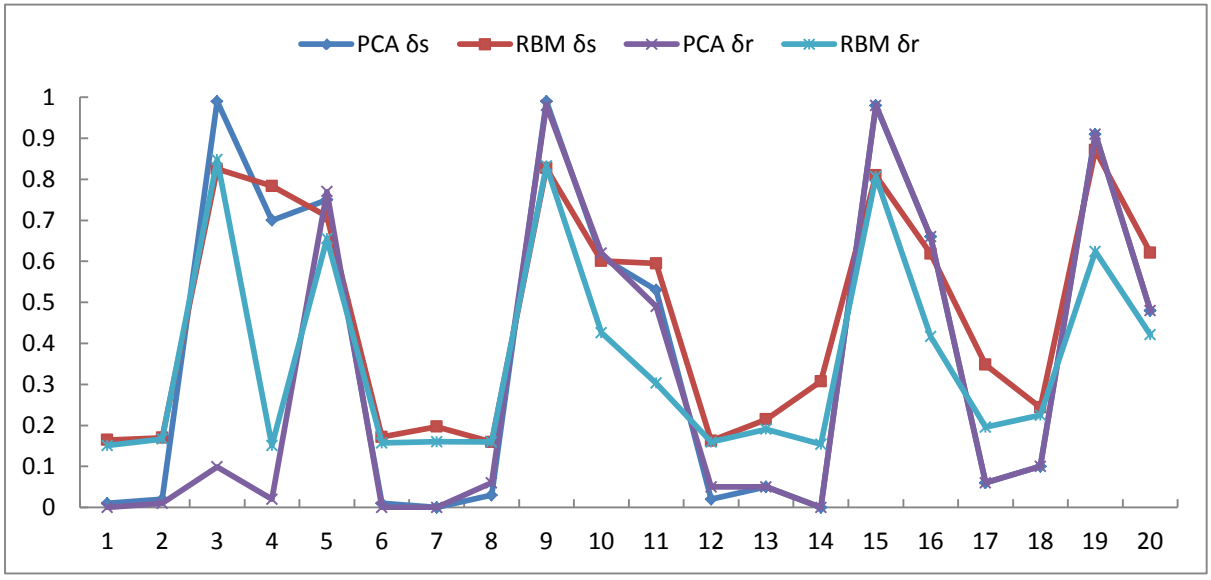


Figure 5.30: Score and residual distance missing alarm rates on fault data

Table 11 (page 77) shows the score missing alarm rates δ_s of the Tennessee Eastman process. The table gives a comparison of four techniques on the score missing alarm rates. For a fault in which a technique scores a 0 as a missing alarm rate, that technique performed best on that fault as it was able to correctly identify the fault. The RBM technique did not perform well on the following faults: 3,9,15 and 19. These high missing alarm rates indicate that the network was not able to correctly detect the abnormal normal, and that corrective measures to improve the process would have been delayed. In terms of the continuous improvement that is normally done to improve process control, this prolongs the time it takes for a process deviation to be rectified. This may also entail that the process engineers re-calibrate the model, so that its performance can improve and this may need more training data.

From Figure 5.31, KICA score distance missing alarm rates are the overall best, and RF score distance missing alarm rates are the overall worst. From Figure 5.32 (page 79), KICA and PCA have better (lower) residual distance missing alarm rates than RBM and RF.

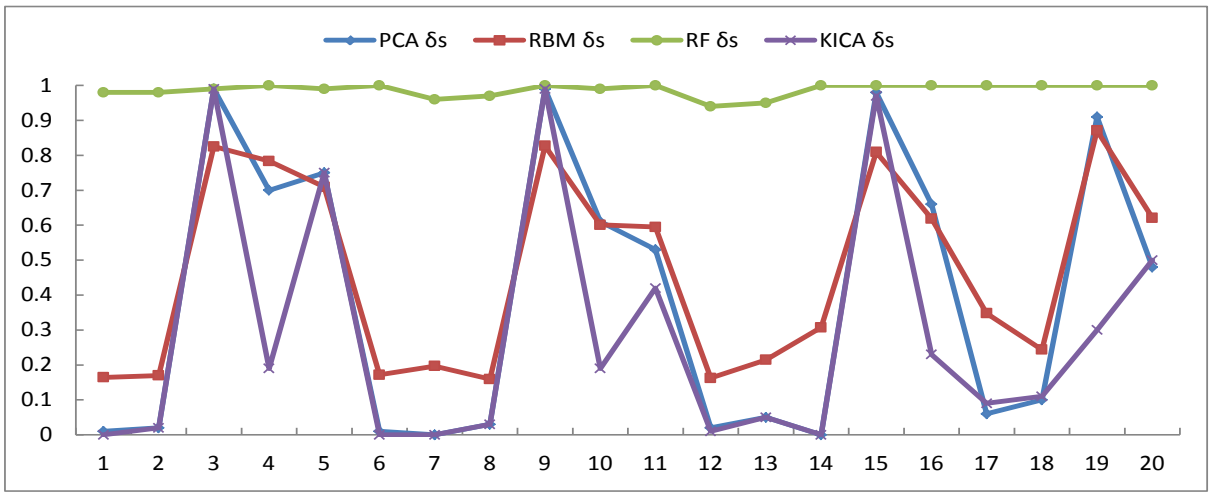


Figure 5.31: Missing alarm rates based on score of the TE process

Table 11: Missing alarm rates (δ_s) for 20 faults of TE

Fault	PCA	RBM	RF	KICA
1	0.010	0.165	0.980	0.000
2	0.020	0.170	0.980	0.020
3	0.990	0.825	0.990	0.990
4	0.700	0.783	1.000	0.190
5	0.750	0.709	0.990	0.750
6	0.010	0.172	1.000	0.000
7	0.000	0.197	0.960	0.000
8	0.030	0.159	0.970	0.030
9	0.990	0.827	1.000	0.990
10	0.610	0.601	0.990	0.190
11	0.530	0.595	1.000	0.420
12	0.020	0.163	0.940	0.010
13	0.050	0.215	0.950	0.050
14	0.000	0.307	1.000	0.000
15	0.980	0.809	1.000	0.970
16	0.660	0.619	1.000	0.230
17	0.060	0.348	1.000	0.090
18	0.100	0.244	1.000	0.110
19	0.910	0.871	1.000	0.300
20	0.480	0.621	1.000	0.500

Table 12 (page 78), shows the residual missing alarm rates δ_r of the Tennessee Eastman process. It can be seen from the table (on page 78), that where the value is 0, those are the techniques that are able to identify that specific fault as the fault data is considered faulty. The results in Figure 5.32 (page 79) show that the RBM algorithm on the known faults has a tendency not to detect the fault data as faulty, especially in faults 3, 9 and 15.

This has adverse consequences on the process, as it means that the intervention that is needed from the technical and operational teams will be delayed to correct the

process deviations. With fault 3, a step change in temperature is done, and if this is not diagnosed earlier, it will result in problems in the formation of product G, since that reaction has a very high sensitivity to temperature. This in turn can lead to losses in process efficiencies and consumables. In order to improve on this performance, more training data will be required. This is because with stacking RBMs during training, more data is required so that the network is not held in local minima.

Table 12: Missing alarm rates (δ_r) for 20 faults of TE

Fault	KICA	PCA	RBM	RF
1	0.000	0.000	0.151	0.000
2	0.020	0.010	0.167	0.020
3	0.970	0.099	0.848	0.950
4	0.000	0.020	0.151	0.000
5	0.720	0.770	0.655	0.720
6	0.000	0.000	0.157	0.000
7	0.000	0.000	0.160	0.000
8	0.020	0.060	0.159	0.020
9	0.970	0.980	0.832	0.960
10	0.220	0.620	0.426	0.220
11	0.230	0.490	0.303	0.190
12	0.010	0.050	0.159	0.010
13	0.050	0.050	0.191	0.050
14	0.000	0.000	0.154	0.000
15	0.950	0.980	0.806	0.940
16	0.130	0.660	0.417	0.130
17	0.030	0.060	0.196	0.030
18	0.090	0.100	0.225	0.090
19	0.150	0.910	0.624	0.150
20	0.350	0.480	0.421	0.350

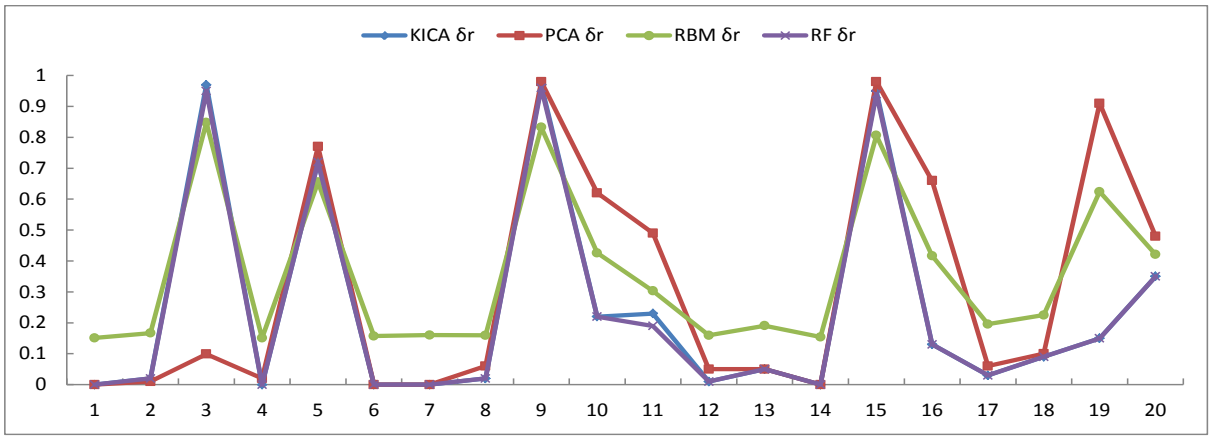


Figure 5.32: Missing alarm rates based on residual distances of the TE process

5.3.3 Variable Contributions

To assess the ability of PCA and RBM fault diagnosis methods and to determine the affected process variables, five of the twenty-one faults, for which causal or affected variables are known, were chosen to assess variable contributions. These faults are faults 4, 5, 11, 12 and 14, and the causal process variables were identified, based on observations made in fault diagnostic literature on the Tennessee Eastman process. The contributions are calculated as the average contributions of samples that are both indicated as faulty and known to be faulty.

The identified variables may not be causal, but may be closely related to the fault. For example, when the reactor temperature increases due to some external disturbance, the cooling water flow rate to the reactor will increase due to closed loop control. Cooling water flow rate is then an affected variable, and not a causal variable.

- **Fault 4**

Fault 4 is simulated by introducing a step change in the reactor temperature (variable 9). As control loops compensate for the temperature increase, a step change in reactor cooling water flow rate (variable 51) is induced, while all other variables

return to steady state. Variable 51 is assumed to be most closely associated with the fault (Russell et al., 2000). From the following contribution plots it is clear that both PCA and RBM can successfully identify variable 51 for this fault. This correct identification will result in time and monetary savings, since it means that corrective action can be taken to remedy the fault. The contribution plots for PCA is displayed in Figure 5.33.

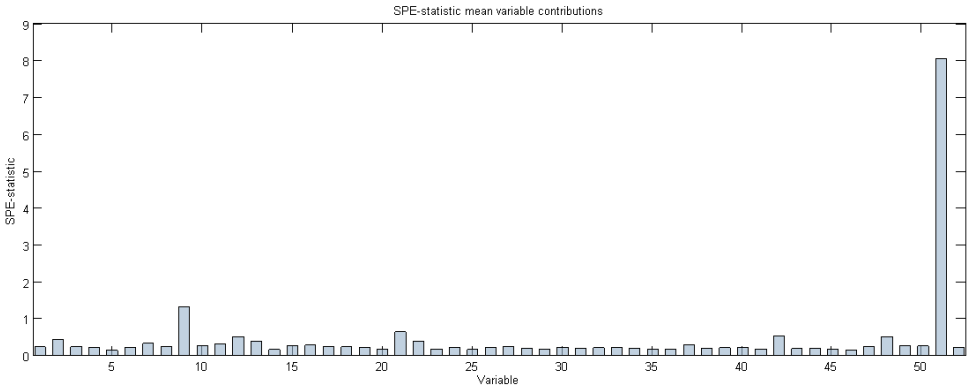


Figure 5.33: Contribution plots for fault 4 using PCA

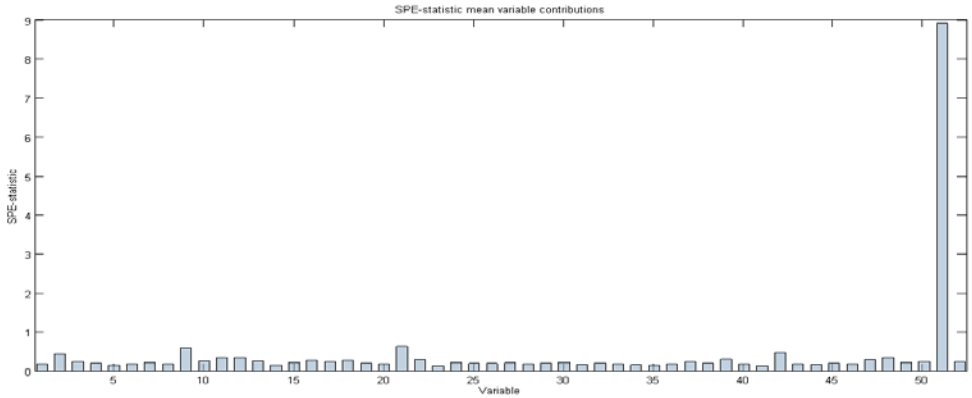


Figure 5.34: Contribution plots for fault 4 using RBM

It can be seen from Figure 5.34 that although the RBM algorithm has correctly identified variable 51 as the variable that contributes to the fault, variables 9, 21 and 42 are also associated with the fault. This will mean that the process engineers who are undertaking fault finding, will also have to ensure that these other variables are also inspected and that they are operating according to the optimum operating conditions.

- **Fault 5**

A step change in condenser cooling water inlet temperature is the defining condition for fault 5. This induces a step change in condenser cooling water flow rate (variable 52), and increases the flow rate of the vapour liquid separator feed, which subsequently results in an increase in the vapour liquid separator temperature (variable 11). This temperature increase further induces an increase in the separator cooling water outlet temperature (variable 22) (Russell et al., 2000; Shao & Rong, 2009).

The contributions plot based on PCA scores (Figure 5.35, page 82) show variable 50 with the most significant contribution, and does not rank variable 22 as contributing to the fault significantly. The contribution plot based on RBM residuals (Figure 5.36, page 82) rank variable 18 as most significant, with more than fifteen other variables also shown as significant. Variable 52 is, however, only shown as significant in the PCA contribution plots. This has a negative effect on the process, as it means that the process engineers and operators will not have identified the correct variable as being responsible for the fault. Such a case will result in process inefficiency, and more time will then have to be invested in figuring out the variables responsible for the fault, as well as training the network again so that it can improve in its performance. Since the network training of the RBM does require data pre-processing, this scenario will disturb the continuous improvement in the process.

One important observation that was made from these experiments is that the RBM pre-training tends to make the auto encoders more focused on the training data (NOC), resulting in a low generalization for faulty data. This results in some of the variables not being correctly ranked in terms of their contribution to the fault, as can be seen in this case.

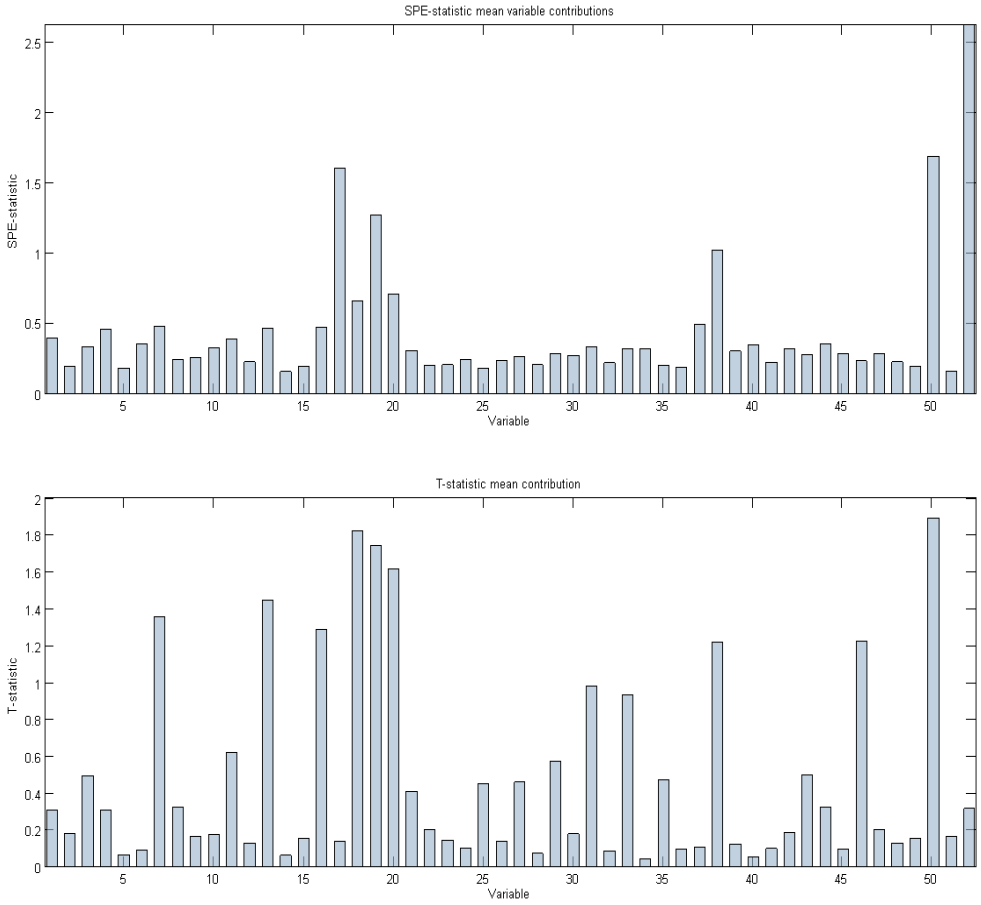


Figure 5.35: Contribution plots for fault 5 using PCA

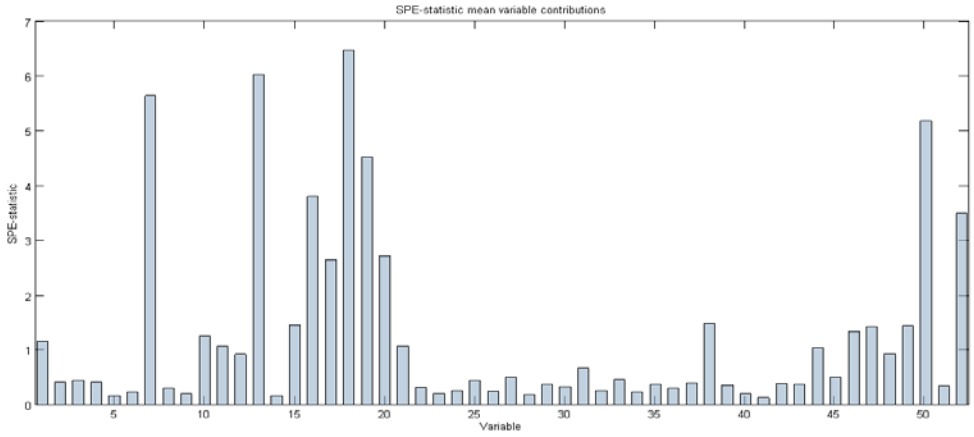


Figure 5.36: Contribution plots for fault 5 using RBM

- **Fault 11**

Fault 11 is simulated as random variation in reactor cooling water inlet temperature. This causes large oscillations in the reactor cooling water flow rate (variable 51) and

the reactor temperature (variable 9) (Russell et al., 2000). From the following contribution plots in Figure 5.37 and Figure 5.38, it is clear that both PCA and RBM successfully identified variables 9 and 51 for this fault.

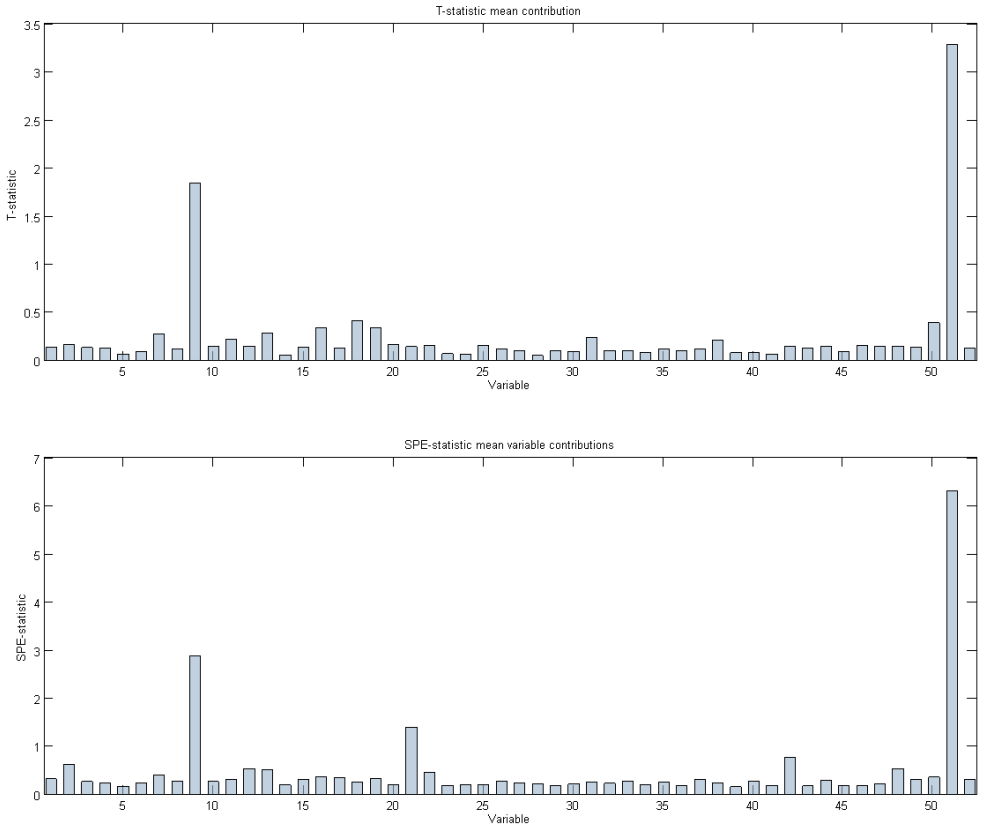


Figure 5.37: Contribution plots for fault 11 using PCA

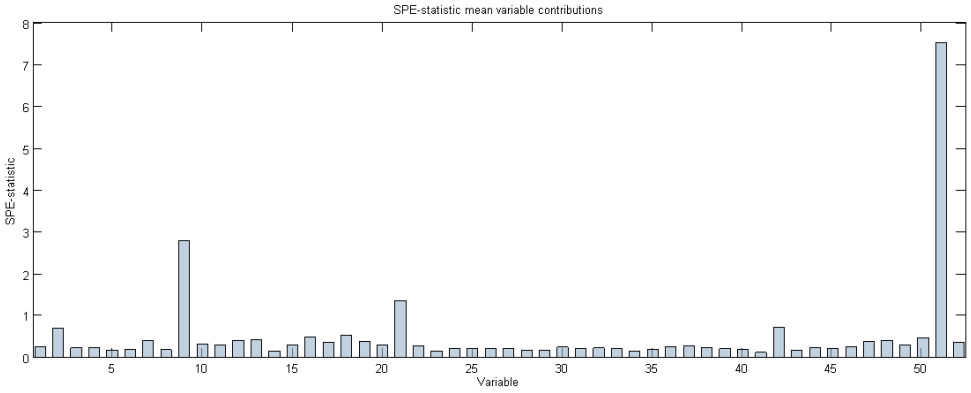


Figure 5.38: Contribution plots for fault 11 using RBM

- **Fault 12**

Fault 12 is simulated as random variation in the condenser cooling water temperature, which induces abnormal behaviour in many variables, including the separator temperature (variable 11), separator pressure (variable 13) and separator outlet cooling water temperature (variable 22) (Russell et al., 2000; Shao & Rong, 2009). PCA score contributions rank variables (Figure 5.39) 13, 7 and 16 as first, second and third most significant, respectively. PCA residual contributions rank variable 11 eighth most significant, while RBM contributions (Figure 5.40, page 85) rank variable 11 and 13 eighth and first most significant.

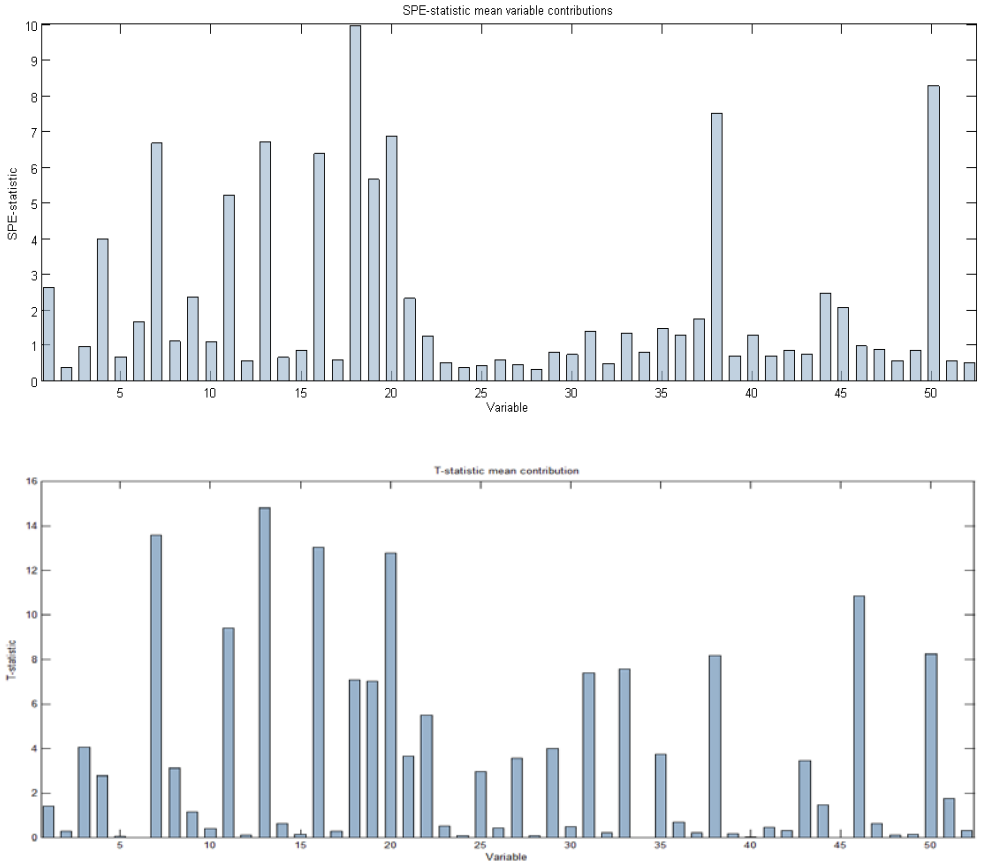


Figure 5.39: Contribution plots for fault 12 using PCA

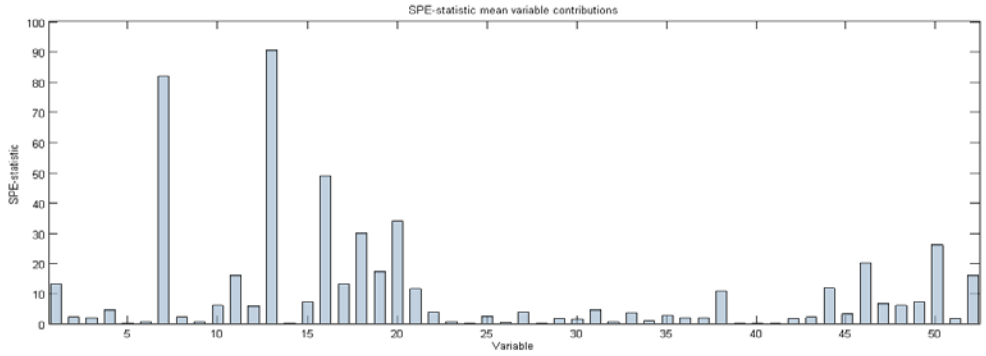


Figure 5.40: Contribution plots for fault 12 using RBM

- **Fault 14**

A sticking valve for reactor cooling water is simulated for fault 14, causing large fluctuations in reactor temperature (variable 9), the reactor cooling water outlet temperature (variable 51) and the reactor cooling water flow rate (variable 21) (Russell et al., 2000; Shao & Rong, 2009). PCA score and residual contributions (Figure 5.41) and RBM contributions (Figure 5.42, page 86) successfully rank these three variables as the three most significant variables related to this fault.

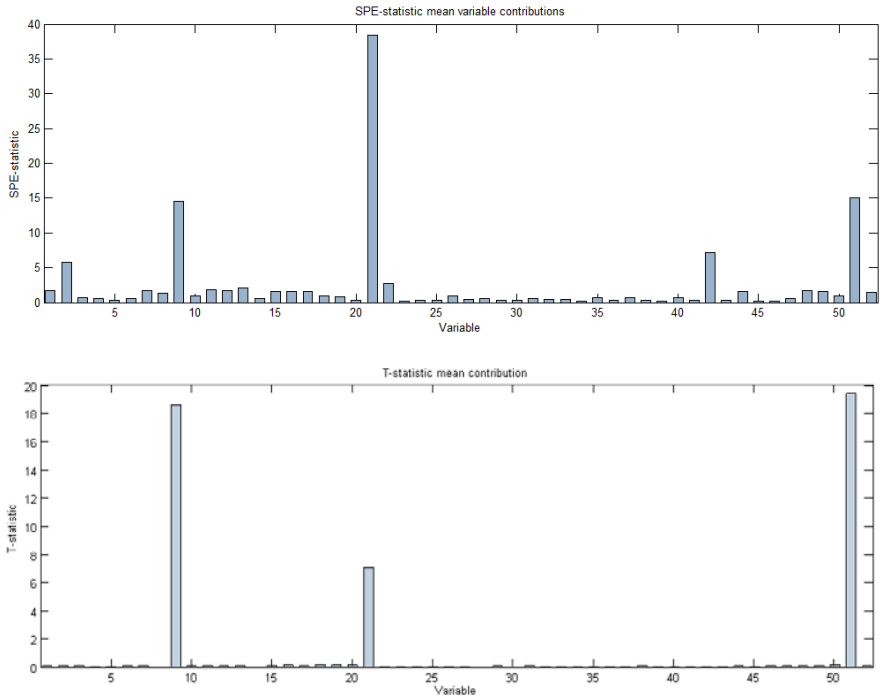


Figure 5.41: Contribution plots for fault 14 using PCA

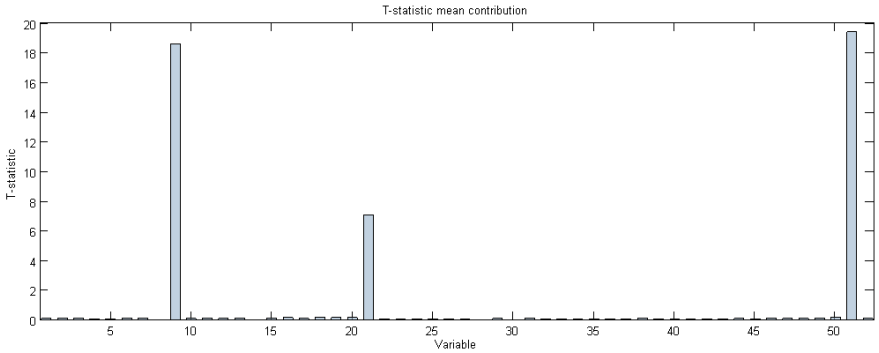


Figure 5.42: Contribution plots for fault 14 using RBM

5.3.4 Discussion of the Tennessee Eastman Process

The Tennessee Eastman has a relatively weak connectivity structure, and contains a lot of statistically unrelated variables. This is mainly attributed to the fact that the defined normal operating condition (NOC) data contains mostly common cause variation (hiding the underlying relational structure within the data). The PCA and RBM fault diagnosis algorithms require the specification of two parameters: the number of model components and the confidence level for limits. A confidence level of 99 % was selected for this study; representing an expected false alarm rate on unseen normal operating conditions data of one false alarm sampled every hundred samples. The number of model components is also very crucial, and was chosen as 13 for the initial PCA model and 10 for the initial RBM model. This selection is based on at least 90 % cumulative variance explained for PCA.

Reconstruction of process variables from features for the PCA and RBM models were expressed as linear reconstruction correlations for both PCA and RBM. RBM showed very high correlations for seen NOC, and very low correlations for unseen NOC, while PCA showed lower correlations than RBM on seen NOC and better correlations than RBM on unseen NOC. These results suggest that, for this data set, PCA generalizes better to unseen NOC. The low generalization ability of RBM may be due to lack of (or representative enough) training data for NOC conditions. As with many deep

learning strategies, the size of the training data influences the performance of the network that is trained.

Comparing PCA and RBM missing alarm rates to RF and KICA results from literature, shows overall superior performance for the KICA model, based on five faults. The other sixteen faults show very similar results for PCA, RBM, RF and KICA models. RBM contributions are fairly successful on the five faults that were investigated, showing correct significance and ranking for three faults and at least one correct variable indication or ranking for the other two faults.

Finally, RBM fault diagnosis is much more computationally expensive than PCA fault diagnosis, and this expense increases with the number of features included. This provides additional motivation for the use of fewer features in using the RBM approach.

CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions on objectives

Conclusions are made in terms of the objectives specified in the introduction to this work (Chapter 1). The overall objective of this study was to assess the feasibility of using Restricted Boltzmann Machines in various fault detection schemes.

The objectives of this work are restated here as follows:

- A literature review of the feature extraction fault diagnosis and the applications of Restricted Boltzmann Machines
- Numerical work, in which features are extracted from process data with Restricted Boltzmann Machines (RBMs) and used as the basis for process fault diagnosis in several case studies.
- Comparison and evaluation of the results with other nonlinear approaches.

The first objective was a literature survey on feature extraction techniques in fault diagnosis and the application of Restricted Boltzmann Machines. Chapter 2 dealt with this survey, with concepts gained from literature incorporated in the development of the RBM algorithms. From a survey of process monitoring methods, the benefits of feature extraction, as well as the limitations of linear feature extraction, were discussed.

RBM feature extraction did not outperform all other feature extraction methods on all data sets, and this should be expected. In feature extraction applications, certain techniques are suited better to certain types of data structures.

The second objective considered the development of an unsupervised fault diagnostic scheme using RBM feature extraction. The result of this development is found in Chapter 4. Another aspect of the second task was the testing of the RBM fault diagnostic scheme on a benchmark process engineering problem and real-world mineral processing data. Flotation (copper and PGM) data sets and the Tennessee Eastman process were employed for this purpose, and relevant performance measures evaluated (Chapter 5).

Overall, from the fault diagnosis method criteria, the RBM approach developed in Chapter 4 could be considered a suitable option for fault diagnosis. However, the RBM approach is computationally expensive, and this expense increases with the number of features extracted, as can be expected with all deep learning training strategies.

6.2 General conclusion

With the creation of process data, the increasing complexity of process plants and the escalating demands of profitability and safety standards, automated process monitoring is an important tool for acquiring valuable information and enabling efficient operation of process plants. Data-driven fault diagnosis schemes aim to exploit the availability of large databases of process data to detect and identify abnormal process conditions. Due to the limitations of linear feature extraction methods, the application of nonlinear feature extraction is a growing topic of interest.

The suitability of nonlinear feature extraction methods to detect abnormal data is considered here. Given a distribution of normal operating conditions data, a nonlinear feature extraction method will attempt to find some nonlinear manifold or transformation that captures the structure of the data. If the distribution of new data

representing fault conditions conforms to the same manifold or transformation, but is located in a sparse region of the normal operating conditions data; this data points will be flagged as faults in the nonlinear pace. However, if the distribution of the new data representing the fault conditions does not conform to the same nonlinear manifold or transformation as the normal operating conditions data, the projection of this data unto the normal operating conditions manifold will give rise to reconstruction errors in the original variable space, and the faulty data are flagged as faults.

This validates the use of nonlinear feature extraction techniques for fault diagnosis in general, and the use of RBM feature extraction, specifically. The performance of the fault diagnosis method depends on the suitable selection of model parameters. In the case of RBM feature extraction, the selection of the number of features has been discussed, and a heuristic proposed.

6.3 Recommendations

From the above conclusions, the application of the proposed RBM feature extraction frameworks for fault diagnosis is recommended as a tool for process monitoring. The effectiveness of these frameworks, as for all data-based process monitoring schemes, depends on a number of factors.

Firstly, for a fault to be detected from process data, the process data must show some distributional change from normal operating conditions to the fault conditions. If the data shows no changes, no data-based fault diagnosis scheme will detect a change. The availability of representative process data is then a necessary, but not sufficient, condition for fault detection with data-based fault diagnosis. Data pre-processing then remains a vital component of process monitoring.

Secondly, the estimation of expected false alarm rates for a fault diagnosis scheme is vital. Given the labelling of new data as representing fault conditions, an expected false alarm rate will aid in risk assessment when decisions need to be made in terms of process recovery. This work has not explicitly investigated the calculation of expected false alarm rates.

An important parameter for the RBM fault diagnosis framework is the number of features to extract to represent normal operating conditions. A disadvantage of deep learning strategies, compared to linear approaches, has been its high computational expense. There are various strategies that can be used in training Restricted Boltzmann Machines, exploring the effect of the different training techniques can be considered to investigate if the performance of the RBM fault diagnosis scheme can be improved (Breiman & Cutler, 2003).

To ensure that the benefits of the deep learning approach involved with stacking RBMs is realised, it is ideal to ensure that high dimensional data is used as the RBMs will tend to perform better as more layers are added to the network structure, as compared to data whose dimension is smaller, as discussed in this work. In plants where process models cover large numbers of variables, auto encoder with RBMs are recommended, as successful training of auto encoders require availability of large amounts of data.

REFERENCES

Alcala, C.F. & Joe Qin, S. (2011). Analysis and generalization of fault diagnosis methods for process monitoring. *Journal of Process Control*. 21 (3). p.pp. 322–330.

Aldrich, C., Gardner, S. & Le Roux, N.J. (2004). Monitoring of metallurgical process plants by using biplots. *AIChE Journal*. 50 (9). p.pp. 2167–2186.

Auret, L. (2010). *Process Monitoring and fault diagnosis using Random forests*. Department of Process Engineering, University of Stellenbosch, Private Bag X1, Matieland, Stellenbosch 7602, South Africa.

Auret, L. & Aldrich, C. (2010a). Change point detection in time series data with random forests. *Control Engineering Practice*. 18 (8). p.pp. 990–1002.

Auret, L. & Aldrich, C. (2011). Empirical comparison of tree ensemble variable importance measures. *Chemometrics and Intelligent Laboratory Systems*. 105 (2). p.pp. 157–170.

Auret, L. & Aldrich, C. (2010b). Unsupervised Process Fault Detection with Random Forests. *Industrial & Engineering Chemistry Research*. 49 (19). p.pp. 9184–9194.

Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*. 2 (1). p.pp. 1–127.

Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in neural information processing systems*. 19. p.p. 153.

Breiman, L. (2001). Random forests. *Machine learning*. 45 (1). p.pp. 5–32.

Breiman, L. & Cutler, A. (2003). *Manual on Setting Up, Using, and Understanding Random Forests,V4.0. ftp://ftp.stat.berkeley.edu/pub/users/breiman/Using_random_forests_v4.0.pdf*. [Online]. Available from: ftp://ftp.stat.berkeley.edu/pub/users/breiman/Using_random_forests_v4.0.pdf. [Accessed: 1 August 2012].

Breiman, L., Olshen, R.., Friedman, J.H. & Stone, C.. (1993). *Classification and Regression Trees*. Chapman & Hall.

Cho, J.-H., Lee, J.-M., Wook Choi, S., Lee, D. & Lee, I.-B. (2005). Fault identification for process monitoring using kernel principal component analysis. *Chemical Engineering Science*. 60 (1). p.pp. 279–288.

Collobert, R. & Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In: *Proceedings of the 25th international conference on Machine learning*. 2008, Helsinki, Finland: ACM, pp. 160–167.

Cosman, P.., Gray, R.. & Olshen, R.. (1994). Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. *Proc. of the IEEE*. 82 (6). p.pp. 919–932.

Dong, D. & McAvoy, T.J. (1996). Nonlinear principal component analysis—Based on principal curves and neural networks. *Computers & Chemical Engineering*. 20 (1). p.pp. 65–78.

Downs, J.J. & Vogel, E.F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*. 17 (3). p.pp. 245–255.

Duin, R., Juszczak, P., Paclik, E., Pekalska, D. & de Ridder, D. (2007). *PRTools 4, A Matlab Toolbox for Pattern Recognition*. Delft University of Technology.

Friedman, J.H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*. 29 (5). p.pp. 1189–1232.

Gabriel, K.R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*. 58 (3). p.pp. 453–467.

Garcia-Alvarez, D., Fuente, M.J., Vega, P. & Sainz, G. (2009). Fault Detection and Diagnosis using Multivariate Statistical Techniques in a Wastewater Treatment Plant. In: *Proc. of the 7th IFAC International Symposium on Advanced Control of Chemical Processes, Turkey*. 2009.

Gardner, S., Le Roux, N.J. & Aldrich, C. (2005). Process data visualisation with biplots. *Minerals Engineering*. 18 (9). p.pp. 955–968.

Hadsell, R., Erkan, A., Sermanet, P., Scoffier, M., Muller, U. & Yann LeCun (2008). Deep belief net learning in a long-range vision system for autonomous off-road driving. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. 2008, pp. 628–633.

Haykin, S. (1999). *Neural networks: a comprehensive foundation*. [Online]. Prentice Hall. Available from: http://books.google.com/books?id=M5abQgAACAAJ.

Hinton, G., Li Deng, Dong Yu, Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *Signal Processing Magazine, IEEE*. 29 (6). p.pp. 82–97.

Hinton, G.E. (2007). Learning multiple layers of representation. *Trends in cognitive sciences*. 11 (10). p.pp. 428–434.

Hinton, G.E. (2010). Learning to represent visual input. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 365 (1537). p.pp. 177 –184.

Hinton, G.E. (2002). Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*. 14 (8). p.pp. 1771–1800.

Hinton, G.E. & Brown, A.D. (2000). Spiking boltzmann machines. In: *Advances in neural information processing systems 12: proceedings of the 1999 conference*. 2000, p. 122.

Hinton, G.E., Dayan, P. & Revow, M. (1997). Modelling the manifolds of images of handwritten digits. *Neural Networks, IEEE Transactions on*. 8 (1). p.pp. 65–74.

Hinton, G.E., Osindero, S. & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*. 18 (7). p.pp. 1527–1554.

Hinton, G.E. & Salakhutdinov, R.R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*. 313 (5786). p.pp. 504 –507.

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods -- An introduction. *Control Engineering Practice*. 5 (5). p.pp. 639–652.

Jemwa, G.T. & Aldrich, C. (2006). Kernel-based fault diagnosis on mineral processing plants. *Minerals Engineering*. 19 (11). p.pp. 1149–1162.

Johnson, R.A. & Wichern, D.W. (2007). *Applied multivariate statistical analysis*. Upper Saddle River, N.J.: Pearson Prentice Hall.

Kramer, M.A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*. 37 (2). p.pp. 233–243.

Krizhevsky, A. & Hinton, G.E. (2011). *Using Very Deep Autoencoders for Content-Based Image Retrieval*. In: 2011.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J. & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In: *Proceedings of the 24th international conference on Machine learning*. 2007, pp. 473–480.

Lee, G., Han, C. & Yoon, E.S. (2004). Multiple-Fault Diagnosis of the Tennessee Eastman Process Based on System Decomposition and Dynamic PLS. *Ind. Eng. Chem. Res.* 43 (25). p.pp. 8037–8048.

Van der Maaten, L.J.P., Postma, E.O. & Van den Herik, H.J. (2009). Dimensionality

reduction: A comparative review. *Tilburg University*. [Online]. Available from: http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html.

MacGregor, J.F. & Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*. 3 (3). p.pp. 403–414.

Osindero, S. & Hinton, G. (2008). Modeling image patches with a directed hierarchy of Markov random fields. In: *Advances in Neural Information Processing Systems 20*. MIT Press, pp. 1121–1128.

Pollard, J.F., Broussard, M.R., Garrison, D.B. & San, K.Y. (1992). Process identification using neural networks. *Neutral network applications in chemical engineering*. 16 (4). p.pp. 253–270.

Ralston, P., DePuy, G. & Graham, J.H. (2004). Graphical enhancement to support PCA-based process monitoring and fault diagnosis. *ISA transactions*. 43 (4). p.pp. 639–653.

Rosen, C. & Lennox, J.A. (2001). Multivariate and multiscale monitoring of wastewater treatment operation. *Water Research*. 35 (14). p.pp. 3402–3410.

Rousseeuw, P.J., Ruts, I. & Tukey, J.W. (1999). The bagplot: a bivariate boxplot. *The American Statistician*. 53 (4). p.pp. 382–387.

Russell, E.L., Chiang, L.H. & Braatz, R.D. (2000). *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*. Springer.

Salakhutdinov, R. & Hinton, G. (2007). Learning a nonlinear embedding by preserving class neighbourhood structure. In: *AI and Statistics*. 2007.

Salakhutdinov, R. & Hinton, G. (2008). Using deep belief nets to learn covariance kernels for gaussian processes. *Advances in neural information processing systems*. 20. p.pp. 1249–1256.

Salakhutdinov, R., Mnih, A. & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In: *Proceedings of the 24th international conference on Machine learning*. 2007, Corvalis, Oregon: ACM, pp. 791–798.

Scholz, M., Fraunholz, M. & Selbig, J. (2007). *Principal Manifolds for Data Visualization and Dimension Reduction*. In: A. N. Gorban, B. Kégl, D. C. Wunsch, & A. Y. Zinovyev (eds.). Lecture Notes in Computational Science and Engineering. Springer Berlin Heidelberg, pp. 44–67.

Sejnowski, T.J. (1986). Higher-order Boltzmann machines. In: *AIP Conference Proceedings*. 1986, p. 398.

Shao, J.-D. & Rong, G. (2009). Nonlinear process monitoring based on maximum variance unfolding projections. *Expert Syst. Appl.* 36 (8). p.pp. 11332–11340.

Sorsa, T. & Koivo, H.N. (1993). Application of artificial neural networks in process fault diagnosis. *Automatica*. 29 (4). p.pp. 843–849.

Stefatos, G. & Ben Hamza, A. (2007). Statistical process control using kernel PCA. *Control & Automation, 2007. MED '07. Mediterranean Conference on*. p.pp. 1–6.

Sutskever, I. & Hinton, G. (2007). Learning Multilevel Distributed Representations for High-dimensional Sequences. *Proceeding of the Eleventh International Conference on Artificial Intelligence and Statistics*.

Tan, C.C. & Eswaran, C. (2008). *Performance Comparison of Three Types of Autoencoder Neural Networks*. In: May 2008, IEEE, pp. 213–218.

Tan, C.C. & Eswaran, C. (2010). Reconstruction and recognition of face and digit images using autoencoders. *Neural Computing and Applications*. 19 (7). p.pp. 1069–1079.

Tan, C.C. & Eswaran, C. (2009). Using Autoencoders for Mammogram Compression. *Journal of Medical Systems*. 35 (1). p.pp. 49–58. Available from: [Accessed: 30 May 2012].

Teppola, P., Mujunen, S.-P., Minkkinen, P., Puijola, T. & Pursiheimo, P. (1998). Principal component analysis, contribution plots and feature weights in the monitoring of sequential process data from a paper machine's wet end. *Chemometrics and Intelligent Laboratory Systems*. 44 (1–2). p.pp. 307–317.

Thissen, U., Melssen, W.J. & Buydens, L.M.C. (2001). Nonlinear process monitoring using bottle-neck neural networks. *Analytica Chimica Acta*. 446 (1–2). p.pp. 369–381.

Valentini, G. & Masulli, F. (2002). Ensembles of Learning Machines. In: M. Marinaro & R. Tagliaferri (eds.). *Neural Nets*. Springer Berlin Heidelberg, pp. 3–20.

Venkatasubramanian, V., Rengaswamy, R. & Kavuri, S.N. (2003a). A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. *Computers & Chemical Engineering*. 27 (3). p.pp. 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. & Yin, K. (2003b). A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & Chemical Engineering*. 27 (3). p.pp. 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K. & Kavuri, S.N. (2003c). A review of process fault detection and diagnosis: Part I: Quantitative model-based

methods. *Computers & Chemical Engineering*. 27 (3). p.pp. 293–311.

Vishnubhotla, S., Fernandez, R. & Ramabhadran, B. (2010). An autoencoder neural-network based low-dimensionality approach to excitation modeling for HMM-based text-to-speech. In: *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. 2010, pp. 4614–4617.

Yzelle, C. (2012). *Process Diagnostic Toolset*. Centre for Process Monitoring, Department of Process Engineering, University of Stellenbosch, Private Bag X1, Matieland, Stellenbosch 7602, South Africa.

Zhang, Y. (2009). Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM. *Chemical Engineering Science*. 64 (5). p.pp. 801–811.

Zhang, Y. & Qin, S.J. (2008). Improved nonlinear fault detection technique and statistical analysis. *AIChE Journal*. 54 (12). p.pp. 3207–3220.

Zhu, Q. & Li, C. (2006). Dimensionality reduction with input training neural network and its application in chemical process modelling. *Chinese Journal of Chemical Engineering*. 14 (5). p.pp. 597–603.

Zumoffen, D. & Basualdo, M. (2008). From Large Chemical Plant Data to Fault Diagnosis Integrated to Decentralized Fault-Tolerant Control: Pulp Mill Process Application. *Ind. Eng. Chem. Res.* 47 (4). p.pp. 1201–1220.

APPENDIX A: NOMENCLATURE

$\langle \ \rangle_{data}$ = expected value of the data distribution

h_j =binary state of hidden unit j

a_i = bias of visible unit i

b_j =bias of hidden unit j

v_i = binary state of visible unit i

w_{ij} = weight between unit i and j

Δw_{ij} = change in weight between unit i and j

h = hidden vector

v = visible vector

Z = partition function

E = Energy function

p = probaility

ε = learning rate

δ =missing alarm rate

δ_s =score distance missing alarm rate

δ_r =residual distance missing alarm rate

λ_j is the j^{th} eigenvalue

LCL=Lower control limit

UCL= Upper control limit

Var = Variable that is monitored

APPENDIX B: LIST OF FIGURES

Appendix B: List of Figures

APPENDIX C: LIST OF TABLES

APPENDIX D: DATA CHARACTERISTICS

D.1. Introduction

This section discusses some data types in order to show how auto encoders perform on different data structures. This is not a comprehensive list of all the suitable and unsuitable data sets, but rather a list that highlights some of the statistical properties of the data, where the auto encoders can either be very useful or not. This will give an idea of what type of data stacked RBM auto encoders can perform and that it is better than other approaches, since the non-linear techniques performance depend on the statistical properties of the underlying data. Training times for the auto encoders are also given, only as an indication as these depend on the computational platforms that are used during experiments.

The experiments are done on data that is often used in the manifold learning literature, as comparisons with other results can be made. The dataset was selected to check how RBM network structure deal with

(i) Data that lies on a low dimensional manifold
(ii) Data that lies on or near a discontinuous manifold
(iii) Data forming a manifold with a higher intrinsic dimensionality

D.2. Experimental Setup

In the experiments conducted on the datasets, auto encoder with RBM is used to extract features from the high dimensional representation of the data. The quality of the resulting low dimensional data representation is then assessed by evaluating to what extent the local structure of the data is retained. This evaluation is performed by measuring the generalization errors of 1-nearest neighbour classifiers that are

trained on the extracted features. The use of 1-nearest neighbour classifiers as a numeric evaluation criterion method has been used, and it requires prior knowledge in terms of class membership (Van der Maaten et al., 2009). This method classifies each data point to the same class as its nearest neighbour. If nearest neighbours in the original input space still remain nearest neighbours in the reduced dimension space, it serves as an indication that local structure has been preserved.

The Matlab Pattern Recognition Toolbox (Duin et al., 2007) is used to calculate the 1-nearest neighbour generalization error.

The data that is used in these experiments is as follows:

(a) SWISS roll

(b) Broken SWISS roll dataset

(c) High dimensional (HD) dataset

(d) MNIST database

The HD dataset consists of points that are randomly sampled from a 5-dimensional non-linear manifold that is embedded in a 10 dimensional space. Each of the datasets contains 5000 samples. The MNIST dataset is a database of 60 000 handwritten digits. The images in the MNIST database have 28x28 pixels, and hence can be considered as points in a 784-dimensional space.

Table 13: Experimental Results

Dataset	PCA	KPCA	RBM
Swiss roll	0.30	0.29	0.48
Broken swiss	0.27	0.31	0.29
HD	0.22	0.28	0.30
MNIST	0.06	0.13	0.07

Appendix D: Data characteristics

The experimental results are shown in Table 13 (page 104), showing the 1-nearest neighbour generalization errors. The high generalization errors on the broken swiss roll data indicate that many nonlinear dimensionality reduction methods do not perform well on the presence of disconnected (i.e., non-smooth) manifolds. The RBM network performed slightly better than KPCA, but linear PCA still did better. This is an indication that RBM's perform better on highly dimensional data, than data that lies on a low manifold and data that lies on a discontinuous manifold as with the broken swiss data.

From the results on the HD dataset, the RBM has a 1-nearest neighbour generalisation error of 0.3 (i.e. 30%) compared to PCA that has 0.22. From the results on the MNIST data, it is evident that with highly dimensional data, the stacked auto encoders perform much better as can be seen with lower generalisation errors than the rest of the datasets. The stacked auto encoder with RBM pre training provides deep architectures (with multiple nonlinear layers). The main advantage of such deep network architectures is that it will require less data points to learn the structure of highly varying manifolds.

D.3. Training times

In order to give an idea of the training time of deep networks, the training time on the MNIST data, using a machine with the following specifications: Intel (R) Core 64 (TM) i3-370M CPU @ 2.40 GHz: is 1700 minutes (28 hours). Depending on the network architecture that is employed, this training can last a couple of days as the network is trying to come to equilibrium.