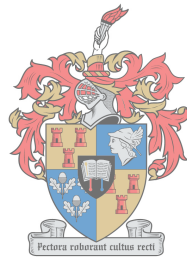


# Load Management of Electric Water Heaters in a Smart Grid Through Forecasting and Intelligent Centralised Control

by

Marcel Roux



UNIVERSITEIT  
iYUNIVESITHI  
STELLENBOSCH  
UNIVERSITY

*Thesis presented in partial fulfilment of the requirements for  
the degree of Master of Engineering (Electronic) in the  
Faculty of Engineering at Stellenbosch University*

Supervisor: Prof. M.J. Booysen

March 2018

# 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.

Date: ..... **March 2018** .....

Copyright ©2018 Stellenbosch University  
All rights reserved

# Abstract

Globally utilities are facing increasing demand and numerous challenges arise with the supply and management thereof. The South African electricity utility Eskom is at present still facing difficulty with meeting the country's growing demand. One of the largest consumers of energy in the residential sector has been identified as the domestic electric water heater (EWH). To manage significant peak loads, electricity utilities employ demand side management (DSM) strategies to throttle demand in order to maintain stability in the electrical grid. Ripple control is such a strategy which is a blunt, unidirectional control scheme which toggles electrical supply to zones of EWHs at times with no consideration for the comfort of individual consumers.

Smart grid (SG) technology is on the rise and emerging Internet of things (IoT) technology augments the adoption of SG to address the problem of DSM. The data collected from a SG is of high value for knowledge discovery and many advantages can be obtained from effective analysis of this data. This study utilises data obtained through the Geasy project which presents a smart EWH controller to enable the monitoring and control of EWHs with resolution of 1 minute. This study presents a three-part look at different aspects of the data aimed towards the development of a cogent, data-driven bidirectional DSM application.

Of fundamental importance to data analysis is to assess the current quality of the data, due to the "garbage in, garbage out" principle. High quality data is required for analysis. After investigating potential data quality impacting factors, the Geasy data was used to develop a numerical data cleaning framework with scalability in mind. The implemented routines were tailored to the specific needs of the data fields considered, such as removing erroneous spikes and filling in missing data according to the most suitable processes. The cleaned data had vastly superior data quality and indicates that the developed data cleaning framework may provide a baseline for more advanced data cleaning steps to be employed before data warehousing.

Next, the aspect of predictive scheduling was investigated. The temporal structure of one of the largest drivers of EWH usage, the hot water usage, was investigated using statistical methods including time series decomposition, autocorrelation and partial autocorrelation plots. The decomposition of the usage data indicated a strong seasonal component that indicated potential for forecasting. Linear seasonal autoregressive integrated moving average models were used to create models of the temporal structure of the usage data. Box-Jenkins parameter identification proved highly effective in estimating good, general-purpose seasonal forecasting models. The obtained forecasting results were shown to predict a daily water volume of 225 L, compared to the observed 272 L, which indicates an error of 17.3 %. However, correcting the forecast volume with the normalised observed training volume reduced the volume error to 0 %.

Continuing the exploration of the value of the SG data, a DSM application was developed to balance the utility and consumer need in real time. During the development of

the algorithms, a computationally efficient EWH thermal model was revised to provide improved scalability through vectorisation which also enabled the algorithms to consider multiple, micro-simulated EWHs during the macro evaluation of a microgrid. The approach uses actual individual hot water consumption patterns, measured real-time water heater temperatures and individual EWH properties as the main determinants in a cost function for a centralised scheduler. The application was evaluated against various demand and temperature limits, with actual consumption measured in a field trial of 34 EWHs for a period of measurements spanning 28 days at 1 minute resolution. For a temperature limit of 60 °C, the application reduces the peak load from a measured 47 kW to 20 kW (vs. 106 kW for full ripple control). The number of undesired cold events decreases by 83.3 %, improving consumer experience, while the total grid energy consumption only increases by 12 %.

# Uittreksel

Wêreldwyd ondervind die elektrisiteit- en waterverskaffers toenemende aanvraag en veelvuldige struikelblokke wat opduik tydens die verskaffing en bedryf van die dienste. Die Suid-Afrikaanse elektrisiteitsverskaffer, Eskom, ondervind geweldige druk om aan die land se groeiende aanvraag te voorsien. Een van die grootste energieverbruikers in die residensiële energieverbruik was geïdentifiseer as die elektriese warmwater verhitte (EWW). Om die hoë aanvraag na krag van die EWW's en die gevolglike druk op die kragnetwerk aan te spreek, maak elektrisiteitsverskaffers gebruik van aanvraagbestuur (AB) om die aanvraag te beperk om sodoende die stabiliteit van die kragnetwerk te behou. Rimpel-effekbeheer is 'n voorbeeld van 'n AB tegniek wat in een rigting beheer uit oefen deur elektrisiteit na sones van EWW's te beperk sonder oorweging van die gerief van individuele verbruikers.

Slim kragnetwerk (SK) -tegnologie ondervind toenemende groei. Die ontluikende Internet van Dinge (IoD) -tegnologie dra by tot toenemende toepassing van SK om AB te implimenteer. Data wat vanaf 'n SK versamel is, is waardevol vir die naspeuring van voorheen onbekende inligting. Veelvuldige voordele kan verkry word deur effektiewe data ontleding. Dié studie maak gebruik van data wat as deel van die “Geasy” projek versamel is. Die “Geasy” projek is 'n slim EWW-moniteringstoestel wat die monitering en beheer van EWW's implimenteer teen 'n resolusie van 1 minuut. Dié studie bied 'n driedelige ondersoek oor 'n verskeidenheid aspekte van die data aan met die doelwit om 'n data-bestuurde tweerigting AB-toepassing te ontwikkel.

Die evaluering van data-gehalte is van fundamentele belang vir data analise weens die *garbage in, garbage out* (gemors in, gemors uit) beginsel. Hoë gehalte data is noodsaaklik vir ontleding. Na 'n ondersoek oor potensieële faktore wat die datagehalte mag beïnvloed, is die “Geasy” data gebruik om 'n numeriese data-skoonmaak raamwerk te ontwikkel met die oog op skaalbaarheid. Die toegepaste metode is ontwikkel om voorsiening vir die individuele behoeftes van die data-velde te maak, soos die verwydering van foutiewe uitskieters en om verlore data volgens die mees toepaslike metode te vul. Na toepassing, is die skoongemaakte data se gehalte aansienlik verbeter en dui dus aan dat die ontwikkelde data-skoonmaak raamwerk 'n goeie beginpunt mag wees vir toekomstige meer gevorderde data-skoonmaak stappe wat gebruik kan word ter voorbereiding vir die langtermyn stoor van data.

Vervolgens is voorspellende skedulering ondersoek. Die tydstruktuur van warm watergebruik (een van die grootste EWW-gebruik aandrywers) is ondersoek deur middel van statistiese metode wat tydreeksontbinding, outokorrelasie en partiële outokorrelasie-grafieke insluit. Die tydreeksontbinding het 'n sterk seisoenale komponent uitgewys wat die potensiaal vir voorspelling aandui. Lineêre seisoenale outoregressiewe geïntegreerde bewegende gemiddelde modelle is gebruik om die tydstruktuur van die data vas te vang. Die Box-Jenkins parameter-identifisering was hoogs doeltreffend met die skatting van goeie, algemene-doel voorspellende modelle. Die resultaat het 'n daaglikse warmwater volume gebruik van 225 L voorspel, waar die eintlike waarde 272 L was. Dit dui 'n fout

van 17.3 % aan. Die fout is egter verminder na 0 % na die voorspelde daaglikse volume geskaleer is om dieselfde te wees as die opleidingsdata se daaglikse volume.

Die nut van SK data is verder ontgin deur die ontwikkeling van 'n AB toepassing met die hoof doel om die behoeftes van die elektrisiteitsverskaffer en die verbruiker intyds te balanseer. Tydens die ontwikkeling van die algoritmes is 'n lae berekeningskompleksiteit termiese EWW model hersien om verbeterde skaalbaarheid te bekom deur middel van vektorisering. Die vektorisering het die algoritmes toegelaat om veelvoudige, mikro-gesimuleerde EWWs te oorweeg tydens die evaluering van 'n mikro elektriese krag-netwerk. Dié benadering maak gebruik van gemete individuele warmwater verbruikspatrone, gemete intydse EWW temperature en individuele EWW eienskappe as bepalende faktore vir die kostefunksie vir 'n sentrale skeduleerder. Die toepassing is geëvalueer teenoor verskeie aanvraag en temperatuur perke, met die werklike verbruiksdata gemeet in 'n veldtoets van 34 EWWs oor 'n tydperk van 28 dae, teen 'n resolusie van 1 minuut. Vir 'n temperatuurperk van 60 °C het die toepassing die piek aanvraag verminder vanaf 'n gemete 47 kW tot 20 kW (teenoor 106 kW vir rimpeleffekbeheer). Die aantal ongewenste koue water gebeurtenisse neem af met 83.3 %, wat die verbruiker se gerief verbeter, terwyl die totale kragnetwerk se energieverbruik met net 12 % toeneem.

# Publications

The work in this manuscript has been published or accepted for publication as follows:

- M. Roux, N.H. Naudé, M.J. Booysen, A. Barnard, "Electric Water Heaters in Smart-grids: Individual Savings Versus Network Peak Load Management", SAUPEC 2017, January 2017, Stellenbosch, South Africa [1].
- M. Roux, M.J. Booysen, "Use of Smart Grid Technology to Compare Regions and Days of the Week in Household Water Heating", DUE 2017, April 2017, Cape Town, South Africa [2].
- L. Leeuwner, N.H.Naudé, M. Roux, M.J. Booysen, "Evaluation of the Energy Model of a Horizontally-Mounted Electric Water Heater Through Internal Temperature Measurement", IEEE PES, ISGT, Asia 2017.

Additionally, the work in this manuscript has been submitted for publication as follows:

- M.Roux, M.J. Booysen, "Hot Water Demand as a Driver for Direct Load Control of Water Heaters in a Smart Microgrid", IEEE Transactions on Smart Grid, September 2017.

Finally, the data sets used during this study are available:

- <http://staff.ee.sun.ac.za/mjbooyesen/SAUPEC2017Dataset/>
- <http://staff.ee.sun.ac.za/mjbooyesen/DUE2017Dataset/>

# Acknowledgements

I would like to express my sincere gratitude to the following people and organisations:

- Prof. Thinus Booysen for his invaluable leadership and for providing opportunities for personal and professional growth.
- My colleagues involved with the Mobile Intelligence Lab for the numerous cups of coffee and frequent technical discussions.
- Family and friends for their moral support and encouragement throughout the years.
- MTN for their support and funding through the MTN Mobile Intelligence Lab.
- The Water Research Commission for their research grant which facilitated the expansion of the scale of the Geasy project, ensuring this research had plenty of data to work from.



# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Uittreksel</b>	<b>iv</b>
<b>Publications</b>	<b>vi</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>Contents</b>	<b>viii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>Nomenclature</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Energy and Water Utility Industries . . . . .	1
1.2 Demand-Side Management and Demand Response . . . . .	2
1.3 Smart Grids and IoT . . . . .	2
1.4 Data: The New Oil . . . . .	3
1.5 Foundational Work . . . . .	4
1.6 Problem Statement and Objectives . . . . .	4
1.6.1 Problem Statement . . . . .	4
1.6.2 Proposed Solution and Objectives . . . . .	5
1.7 Scope of work . . . . .	6
1.7.1 Data Wrangling . . . . .	6
1.7.2 Data Analysis and Forecasting . . . . .	6
1.7.3 Demand-Side Management Application . . . . .	6
1.8 Contributions . . . . .	7
1.9 Dissertation Structure . . . . .	7
<b>2 Data Wrangling</b>	<b>8</b>
2.1 Current Data Wrangling Status and Challenges . . . . .	8
2.1.1 The Concept of Data Quality . . . . .	8
2.1.2 Improving Data Quality with Data Cleaning . . . . .	10
2.2 Geasy Data Flow . . . . .	11
2.2.1 Geasy SEC . . . . .	11

2.2.2	Communication Path . . . . .	12
2.3	Data Cleaning Methods Used . . . . .	14
2.3.1	Sampling Period Regularisation . . . . .	14
2.3.2	Missing Value Imputation . . . . .	15
2.3.3	Spectral Analysis . . . . .	15
2.3.4	Digital Filtering . . . . .	16
2.4	Data Cleaning Tools Used . . . . .	18
2.4.1	Python . . . . .	18
2.4.2	MongoDB . . . . .	18
2.5	Data Tool Developed . . . . .	19
2.5.1	Data Interface Requirements . . . . .	19
2.5.2	Data Interface Implementation . . . . .	19
2.6	Data Cleaning . . . . .	23
2.6.1	Initial Data Investigation . . . . .	24
2.6.2	Sampling Period Regularisation . . . . .	29
2.6.3	Outlier Mitigation . . . . .	29
2.6.4	Missing Value Imputation . . . . .	29
2.6.5	Filter Design . . . . .	31
2.6.6	Summary . . . . .	33
2.7	Results . . . . .	33
2.7.1	Power Data . . . . .	34
2.7.2	Water Data . . . . .	34
2.7.3	Temperature Data . . . . .	36
<b>3</b>	<b>Data Analysis and Forecasting</b>	<b>38</b>
3.1	Time Series Analysis . . . . .	38
3.2	Methods Used . . . . .	39
3.2.1	Time Series Decomposition . . . . .	39
3.2.2	Measure of Dependence: Correlation . . . . .	39
3.2.3	Stationary Time Series . . . . .	41
3.2.4	Residual Diagnostics . . . . .	43
3.2.5	Forecast Models . . . . .	44
3.2.6	Parameter Grid Search . . . . .	45
3.2.7	Performance Metrics . . . . .	45
3.2.8	Statistical Diagnostic Tests . . . . .	46
3.3	Statistical Analysis Tools Used . . . . .	48
3.4	Usage Data Exploration . . . . .	48
3.4.1	Water Usage Decomposition . . . . .	48
3.4.2	Water Usage Correlation . . . . .	49
3.5	SARIMA Modelling . . . . .	50
3.5.1	Selecting Parameters . . . . .	50
3.5.2	Model Diagnostics . . . . .	52
3.6	Results . . . . .	52
3.6.1	SARIMA Forecast . . . . .	53
3.6.2	Volume-Corrected SARIMA Forecast . . . . .	53
<b>4</b>	<b>Data Application: Demand-Side Management</b>	<b>56</b>
4.1	Smart Grid Demand Side Management . . . . .	56
4.1.1	Schedule Optimisation Techniques . . . . .	56

4.1.2	Grid-Centric Scheduling . . . . .	57
4.1.3	User-Centric Scheduling . . . . .	58
4.2	EWH Modelling Overview . . . . .	58
4.2.1	Thermal Principles of Nodal Models . . . . .	58
4.2.2	EWH Modelling . . . . .	60
4.2.3	EWH Nodal Models . . . . .	61
4.3	Methods Used . . . . .	62
4.3.1	Vectorisation . . . . .	62
4.3.2	Cost Function . . . . .	64
4.4	Developed Tools . . . . .	65
4.4.1	Aggregate Data Container . . . . .	65
4.4.2	Event Detection . . . . .	65
4.4.3	Mean Event Temperature Calculation . . . . .	67
4.5	Simulator Design . . . . .	68
4.5.1	Simulator Rev0 . . . . .	68
4.5.2	Simulator Rev1 . . . . .	69
4.5.3	Simulator Rev2 . . . . .	70
4.5.4	Simulator Rev3 . . . . .	70
4.6	Numerical Processing Tools Used . . . . .	71
4.7	Performance Metrics for Oracle . . . . .	71
4.7.1	Peak Demand and Demand Profile . . . . .	71
4.7.2	Mean Event Temperature . . . . .	72
4.7.3	Cold Events . . . . .	72
4.7.4	Total Energy Consumption . . . . .	72
4.7.5	Simulation Performance . . . . .	72
4.8	Oracle Development . . . . .	73
4.8.1	EWH Prioritisation . . . . .	73
4.8.2	MG Power Limit Enforcement . . . . .	76
4.9	Results . . . . .	77
4.9.1	Evaluation Parameters . . . . .	78
4.9.2	Peak Demand and Demand Profile . . . . .	78
4.9.3	Mean Event Temperature . . . . .	79
4.9.4	Cold Events . . . . .	80
4.9.5	Total Energy Consumption . . . . .	81
4.9.6	Simulation Performance . . . . .	82
<b>5</b>	<b>Conclusion</b>	<b>85</b>
5.1	Evaluation of Work . . . . .	85
5.2	Recommendations . . . . .	87
5.2.1	Data Wrangling . . . . .	87
5.2.2	Data Analysis and Forecasting . . . . .	88
5.2.3	Data Application: Demand-Side Management . . . . .	88
	<b>List of References</b>	<b>89</b>

# List of Figures

2.1	The arrangement of the Geasy SEC sensors. Adapted from [1]. . . . .	12
2.2	Geasy topology showing various nodes in the communication. Adapted from [24]. . . . .	13
2.3	Ideal case run-length encoding. . . . .	14
2.4	Data interface operational flow. . . . .	19
2.5	High level of data completeness. . . . .	25
	(a) High data completeness chart. . . . .	25
	(b) High data completeness bar chart. . . . .	25
2.6	Low level of data completeness. . . . .	26
	(a) Low data completeness chart. . . . .	26
	(b) Low data completeness bar chart. . . . .	26
2.7	Investigation of the temperature outliers through frequency plots, suffered by the Geasy SECs. . . . .	27
	(a) Outlet temperature. . . . .	27
	(b) Inlet temperature. . . . .	27
	(c) Ambient Temperature. . . . .	27
2.8	Raw temperature plot showing erroneous spike of 100 °C. . . . .	28
2.9	Investigation of the time stamp drift suffered by the Geasy SECs. . . . .	28
	(a) Time stamp drift experienced for 60 consecutive observations, shown by the recorded seconds value of the time stamp. . . . .	28
	(b) Time stamp drift seconds frequency over the period of a day. . . . .	28
2.10	Energy spectra of vectors that do not qualify for filtering. . . . .	32
	(a) Water vector spectrum. . . . .	32
	(b) Power vector spectrum. . . . .	32
2.11	Energy spectra of outlet temperature vector with and without DC component. . . . .	32
	(a) Temperature vector spectrum with the DC component. . . . .	32
	(b) Temperature vector spectrum without the DC component. . . . .	32
2.12	A typical day of an EWH's power demand cleaned. The original data is indicated in blue, with the filled values in red. . . . .	34
2.13	A less complete day of an EWH's power demand with no starting or stopping times recorded, cleaned. The original data is indicated in blue, with the filled values in red. . . . .	35
2.14	A typical day of an EWH's water usage cleaned. The original data is indicated in blue, with the filled values in red. . . . .	35
2.15	An excerpt of temperature indicating an anomalous spike in recorded value and the cleaned version. The original data is indicated in blue, with the filled values in red. . . . .	36
2.16	Before and after of a typical day's EWH outlet temperature. Noteworthy is how much smoother the cleaned signal is. . . . .	37

(a)	Before. . . . .	37
(b)	After. . . . .	37
2.17	A typical day of an EWH's outlet temperature cleaned. The original data is indicated in blue, with the filled values in red. . . . .	37
3.1	Autocorrelation of a random data set. . . . .	40
3.2	Autocorrelation of a non-random data set. . . . .	41
3.3	Partial autocorrelation of a random data set. (Lag 82 is shorter than that of ACF.) . . . . .	42
3.4	Partial autocorrelation of a non-random data set. . . . .	42
3.5	Time series decomposition of a single EWH's water usage over the period of one month. . . . .	48
3.6	Autocorrelation of the water data set. . . . .	49
3.7	Partial autocorrelation of the water data set. . . . .	50
3.8	AIC results vs total number of parameters in the model. . . . .	51
3.9	Time series forecast of a single EWH's water usage over the period of one month using only lowest AIC score as selector. . . . .	52
3.10	Time series model diagnostics of a single EWH's water usage over the period of one month. . . . .	53
3.11	Time series forecast of a single EWH's water usage over the period of one month using 75:25 data split. . . . .	54
3.12	Time series forecast of a single EWH's water usage with scaling coefficient of 1.1338, over the period of one month using 75:25 data split. . . . .	55
3.13	Time series forecast of a single EWH's water usage with scaling coefficient of 1.1338, versus the measured data. . . . .	55
4.1	Scalar instruction vs vectorised instruction. . . . .	63
4.2	Scalar conditional flow vs vectorised conditional flow. . . . .	63
4.3	Conditional evaluation of vector with subsequent instruction execution and result. . . . .	64
4.4	SISM and SIMD stream representations. . . . .	64
4.5	Example of a resetting cumulative sum. . . . .	67
4.6	Example of detected events from water flow rate data. . . . .	68
4.7	3D array structure. . . . .	71
4.8	Grid power comparison for temperature limit of 60 °C at various power limits for a day period. . . . .	79
4.9	Number of cold events for temperature limit of 60 °C at various power limits for a day period. . . . .	81
4.10	Total energy consumption for temperature limit of 60 °C at various power limits for a day period. . . . .	83
4.11	Oracle simulation times for various amounts of EWHs and time periods compared to the original simulator time for a period of a single day. . . . .	84

# List of Tables

2.1	Data interface requirements. . . . .	20
2.2	Sensor identification in database along with sensor measurement unit. . . . .	22
2.3	Data cleaning stages. . . . .	23
2.4	High and low data integrity comparison. . . . .	24
2.5	Parameter descriptions as used in the data set. . . . .	27
2.6	Potential reasons for failure of various nodes. . . . .	30
2.7	Filter design parameters. . . . .	33
2.8	Data cleaning routines summary. . . . .	33
2.9	Data cleaning summary of routines applied to each feature vector. . . . .	33
3.1	Identification of purely SARMA models from of ACF and PACF plots. Lags are integers, $k = 1, 2, \dots$ and $s$ indicates the seasonality of the lags. Adapted from [52]. . . . .	45
3.2	Summary of selected SARIMA model attributes. . . . .	53
4.1	Parameters of sample EWH used for cost function design. . . . .	60
4.2	Simulator requirements. . . . .	69
4.3	Oracle requirements. . . . .	73
4.4	Parameters of sample EWH used for cost function design. . . . .	74
4.5	Parameter energy values for a 1 minute period of the sample EWH. . . . .	75
4.6	Parameters used during Oracle simulation. . . . .	78
4.7	Peak demand in kW experienced for various combinations of temperature and power limits compared against the measured values. . . . .	79
4.8	Mean event temperature in °C experienced for various combinations of temperature and power limits compared against the measured values. . . . .	80
4.9	Number of cold events experienced for various combinations of temperature and power limits compared against the measured values. . . . .	81
4.10	Mean event temperature of cold events in °C experienced for various combinations of temperature and power limits compared against the measured values. . . . .	82
4.11	Energy consumption in kWh experienced for various combinations of temperature and power limits compared against the measured values. . . . .	82
4.12	Simulation execution time in seconds for various numbers of EWHs considered. . . . .	83
4.13	Normalised simulation time in milliseconds for 1 EWH over a period of 1 day. . . . .	84

# Nomenclature

## Acronyms and Abbreviations

ACF	Autocorrelation Functions
ADF	Augmented Dickey-Fuller
AIC	Akaike Information Criterion
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BIC	Bayesian Information Criterion
DB	Database
DFT	Discrete Fourier Transform
DLC	Direct Load Control
DR	Demand Response
DSM	Demand-Side Management
DTL	Data Translation Layer
EBW	Essential Bandwidth
EWH	Electric Water Heater
FFT	Fast Fourier Transform
GDP	Gross Domestic Product
GPS	Global Positioning System
GSM	Global System for Mobile communications
HVAC	Heating, Ventilation and Air Conditioning
IID	Independent and Identically Distributed
ILC	Indirect Load Control
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
JB	Jarque-Bera
KDD	Knowledge Discovery in Databases
LP	Linear Programming
M2M	Machine-to-Machine
MA	Moving Average
MG	Microgrid
MILP	Mixed Integer Linear Programming
MQTT	Message Queuing Telemetry Transport
MSE	Mean Squared Error
NoSQL	Not only Structured Query Language
PACF	Partial Autocorrelation Function
QQ	Quantile-Quantile

REIPPPP	Renewable Independent Power Producer Procurement Programme
RLE	Run-Length Encoding
ROI	Return On Investment
SARIMA	Seasonal Autoregressive Integrated Moving Average
SEC	Smart Electric Water Heater Controller
SG	Smart Grid
SIC	Schwarz Information Criterion
SIMD	Single Instruction stream, Multiple Data stream
SISD	Single Instruction stream, Single Data stream
SKA	Square Kilometre Array
SQL	Structured Query Language
STL	Robust Seasonal and Trend Decomposition using Loess
TDQM	Total Data/Information Quality Management
TOU	Time-Of-Use
UTC	Coordinated Universal Time
WRC	Water Research Commission

### Constants

$$c = 4184 \text{ J}/(\text{kg } ^\circ\text{C})$$

$$\rho = 1000 \text{ kg}/\text{m}^3$$

### List of Symbols

$cov$	Covariance of signals . . . . .	[ Unitless ]
$e_i$	Forecast residual value at position $i$ . . . . .	[ ]
$\hat{e}_i$	Forecast signal value at position $i$ . . . . .	[ ]
$E_s$	Essential bandwidth energy . . . . .	[ J ]
$E_x$	Total signal energy . . . . .	[ J ]
$E$	Internal energy contained in water . . . . .	[ J ]
$H$	Enthalpy . . . . .	[ J ]
$H_0$	Null hypothesis . . . . .	[ Unitless ]
$H_A$	Alternate hypothesis . . . . .	[ Unitless ]
$h(n)$	Discrete filter impulse response . . . . .	[ Unitless ]
$I$	AC current . . . . .	[ A ]
$K$	Kurtosis . . . . .	[ Unitless ]
$L$	Lag operator . . . . .	[ Unitless ]
$\hat{L}$	Maximum likelihood function . . . . .	[ Unitless ]
$M$	Statistical model . . . . .	[ Unitless ]
$P$	AC power . . . . .	[ W ]
$P$	Pressure . . . . .	[ $\text{kg}/(\text{m } \text{s}^2)$ ]
$Q$	Ljung-Box goodness of fit statistic . . . . .	[ Unitless ]
$R(t)$	Time series decomposed residual . . . . .	[ ]
$r_x$	Autocorrelation . . . . .	[ Unitless ]



$S$	Skew . . . . .	[ Unitless ]
$S(t)$	Time series decomposed seasonality . . . . .	[ ]
$t$	Time period . . . . .	[ ]
$T$	Temperature . . . . .	[ °C ]
$T(t)$	Time series decomposed trend . . . . .	[ ]
$U$	Internal energy . . . . .	[ J ]
$V$	AC voltage . . . . .	[ V ]
$V$	Volume . . . . .	[ L ]
$X_k$	Complex Fourier value at position $k$ . . . . .	[ ]
$x_n$	Original signal value at position $n$ . . . . .	[ ]
$x(n)$	Discrete signal . . . . .	[ ]
$y(n)$	Discrete filtered signal . . . . .	[ ]
$y(t)$	Time series decomposition result . . . . .	[ Unitless ]
$\alpha$	Cost function coefficient . . . . .	[ Unitless ]
$\beta$	Cost function coefficient . . . . .	[ Unitless ]
$\gamma$	Percentage of signal energy to retain . . . . .	[ % ]
$\epsilon_i$	Residual component . . . . .	[ ]
$\theta_i$	Moving average coefficient . . . . .	[ Unitless ]
$\hat{\theta}$	Optimal parameters for maximum likelihood . . . . .	[ Unitless ]
$\Theta_i$	Seasonal moving average coefficient . . . . .	[ Unitless ]
$\mu_a$	Central moment $a$ . . . . .	[ Unitless ]
$\rho_{X,Y}$	Correlation of signals X and Y . . . . .	[ Unitless ]
$\sigma_X$	Signal standard deviation . . . . .	[ Unitless ]
$\sigma_X^2$	Signal variance . . . . .	[ Unitless ]
$\phi_i$	Autoregressive coefficient . . . . .	[ Unitless ]
$\Phi_i$	Seasonal autoregressive coefficient . . . . .	[ Unitless ]
$\chi^2$	Chi-square distribution . . . . .	[ Unitless ]
$\Psi_x$	Signal spectral density . . . . .	[ ]

### Subscripts

$AVG$  Average

$RMS$  Root mean squared

$i$   $i^{th}$  position

$n$   $n^{th}$  position

$k$   $k^{th}$  position

# Chapter 1

## Introduction

### 1.1 Energy and Water Utility Industries

Globally there is a growing demand for energy and water [3]. This is attributed to numerous factors of which the increased population size and rapid urbanisation are some of the largest contributing factors. Global water demand is estimated to increase by 55 % by the year 2050, of which the domestic consumption will contribute and increase in demand of 130 % [3]. Along with this, global electricity demand is expected to increase by 70 % by 2035 [3].

On a national scale, South Africa has seen an increased economic growth which is reflected in the growing gross domestic product (GDP), which has seen an increase in electricity consumption since 1993 to 2006 [4]. The phenomenon where a country's electricity consumption is caused by its GDP is described under the 'conservation hypothesis' and is typically found in emerging economies [4]. Social and economic progress could be hampered by energy conservation policies given a unidirectional relationship between electricity consumption and real GDP [4]. This has placed repeated strain on the electricity utility Eskom which faced an energy reserve margin decrease from 25 % in 2002, to 20 % in 2004 and as low as 16 % in 2006 [5].

Due to further increasing electricity demands but being unable to sufficiently supply, in 2008 Eskom [6] started implementing rolling blackouts (load shedding) in order to maintain grid stability and mitigate the risk of a complete network blackout [7]. This supply constraint has been ongoing and support has been procured from renewable independent power producer procurement programme (REIPPPP) since 2010 [8]. The Department of Energy states in their 2016/2017 annual report that the national development plan requires the development of an additional 210 000 MW by 2025 against the baseline of 44 000 MW in 2013 [8]. Furthermore, Cape Town, the capital of the Western Cape Province in South Africa, is at the time of writing facing a severe drought [9]. The City of Cape Town anticipates that the municipal water will be depleted by March 2018 [9].

Effective management of infrastructure to ensure utilities are able to adequately supply services is a challenge [3]. Challenges experienced by the various utilities are as a result of a multitude of factors, such as economical, environmental and available skills [3]. These challenges may be conflicting, as solving one may negatively impact another - a balance must be found [3].

## 1.2 Demand-Side Management and Demand Response

Reliable operation of electricity infrastructure necessitates a perfect balance between supply and demand in real time [10]. Due to various contributing factors, rapid fluctuations of both supply and demand may occur, adding complexity to the balancing problem. Electricity and water utilities in developing countries typically have difficulty meeting increasing demands [3]. Conventionally an increase in demand has been met with an increase in supply in the form of supply side management - typically this includes the expansion and upgrading of the infrastructure to increase generation capacity. Modern energy management systems, backed by smart grid technology, have shifted focus to demand-side management (DSM), which has the primary objective of adjusting the demand load curve through peak shifting, peak shaving and valley filling. This is achieved through intentionally modifying consumer demand profiles [11]. One of the best known forms of DSM practised in many countries, including South Africa, is ripple control. Ripple control is typically used in residences to control the power allocation to the domestic electric water heater (EWH), such as disabling access to power in times of peak demand.

Demand response (DR) capacity can be defined as “the potential for flexible response from end-use appliances across the commercial, industrial, and residential sectors” [12]. There are two main types of DR employed: direct load control (DLC) and indirect load control (ILC) [12]. DLC is already being actively and effectively used by utility providers typically for peak load shaving during critical periods, such as with ripple control. It is also used for frequency regulation in the case of renewable energy due to the volatility. This type of DR requires a more demanding communication and control infrastructure. ILC is typically employed through price-based systems such as time-of-use (TOU) strategies. DR offers benefits for all stakeholders, each receiving particular benefits [10]. The participant may receive incentive payments and bill saving by participating. Market-wide there may be price reductions and capacity increases which may lead to avoided infrastructure costs. By managing the demand, DR increases the reliability of a utility by reducing the outages through customer participation and diversified resources. Additionally, considering the market performance, the customer has more options available which reduces the market power and reduces the price volatility.

## 1.3 Smart Grids and IoT

Conventional power grids are typically based on a hierarchical structure containing four key operations: generation, transmission, distribution and control [13]. This structure implies that the flow of electricity typically starts at a few central generation facilities from where it is transported to the many customers. A smart grid (SG) is an enhancement of the conventional power grid in that it enables two-way flow of both electricity and information, which creates an automated and distributed advanced energy delivery network. Modern information technologies enable SGs to deliver power in more efficient ways by responding to wide ranging conditions and events. The SG is capable of encompassing any layer of the power grid and respond appropriately with corresponding strategies. The SG is an electrical system which uses information, two-way communication technology and computational intelligence in an integrated fashion across all the layers of the power grid [13]. This results in a system that is clean, safe, secure, reliable, resilient, efficient and sustainable. Typically there are three major systems of SG technology: smart infrastructure system, smart management system and the smart protection system [13]. The

smart infrastructure system pertains to the energy, information and communication infrastructure underlying the SG that supports: advanced electricity generation, delivery and consumption; advanced information metering, monitoring, and management; and advanced communication technologies. The smart management system is a subsystem in the SG that provides advanced management and control services. The smart protection system is a subsystem in the SG that provides advanced grid reliability analysis, failure protection, as well as security and privacy protection services.

The Internet of things (IoT) has been defined as a world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in the business process. Services are available to interact with these 'smart objects' over the Internet, query their state and any information associated with them, taking into account security and privacy issues [14]. IoT is a world-wide trend to establish a pervasive communication system spanning all domains [15]. IoT is a paradigm whereby physical 'things' are connected to the Internet for the express purposes of information exchange and intelligence. These 'things' consist of physical objects or devices embedded with sensors, actuators and network connectivity. In 2010 the number of internet connected devices exceeded the human population. This number is anticipated to grow to 212 billion IoT devices deployed by the end of 2020 [15]. By 2020 the value created by the industrial Internet is estimated to be around \$1.279 trillion, with return on investment (ROI) growing to 149 % compared to 13 % in 2012. By 2022, 45 % of the whole Internet will be machine to machine (M2M) traffic. And by 2025, IoT is expected to have an annual economic impact ranging from \$2.7 trillion to \$6.5 trillion.

IoT consists of a number of building blocks, including identification, sensing, communication, computation, services and semantics. IoT promises many opportunities and socio-economic benefits [14], with applications in industries including supply chain integrity, energy, health, automotive and insurance. Energy industry applications include DSM and DR.

## 1.4 Data: The New Oil

IoT is an enabler for a data-driven world by providing the means to digitise the analogue world and connect all aspects. Automated sensing enables a new and much finer granularity of management, since it helps to control which was before uncontrollable. Data processing may take place either with computational software (such as in cloud processing) or in services (as part of smart home systems). IoT enables real world monitoring which generates data to be used to better control and manage business processes [14]. Vast amounts of data are produced from the billions of IoT devices. Sensory information increases the accuracy of real world checks and is therefore fundamental for event-driven management. With a network architecture that supports the easy discovery, communication, and usage of events across the enterprise, events enriched with contextual business information will improve business data quality on all levels.

Data has become the latest lucrative commodity as part of a fast-growing industry. The five giants that deal in data, which are Alphabet (Google's parent company), Amazon, Apple, Facebook and Microsoft, reported profits collectively totalling \$25 billion in the first quarter of 2017 [16].

## 1.5 Foundational Work

This section presents an overview of the work done as part of the Geasy project [17] on which this study is based.

A proof of concept discussed in [18] established a foundation of a smart infrastructure aimed at EWHs. This concept introduced the scalability of connecting multiple EWHs to a central server via the internet and providing a means to remotely manage or monitor the consumption patterns of the EWHs. This system had no means of capturing water volumetric consumption, a key aspect required to understand consumer hot water consumption. Using this foundation, research was directed towards implementing the controller [19] and improved EWH modelling [20]. Brown [19] presents the development of a unified monitoring and controlling device to be fitted to an EWH. This device enables the measurement of temperatures, power consumption and water consumption. Furthermore, this work enabled remote access to the device through a mobile service provider via the internet. Nel [20] developed a physics-based EWH model for estimating EWH temperatures and the related energy flow in and out of the EWH. Furthermore, Nel [20] validated his findings through physical experimentation to ensure sufficiently accurate models of the EWH.

The project continued to grow as a research project, using 5 EWH prototype units installed in volunteer households in the Cape Winelands district with the primary aim of evaluating scheduling benefits [21]. The project received additional funding from the Water Research Commission (WRC) [22] in order to install 450 smart EWH controllers in the local municipality of Mkhondo [23] in Mpumalanga, South Africa. As a result of the increased scale of the project, Cloete [24] developed a more advanced system implementation using SmartM2M standards over the existing cellular network which allowed the project to effectively scale to the required amount of new unit installations. This newly developed implementation utilised lightweight communication protocols, such as MQTT, to ensure good scalability for any system growth. Like the previous system, this improved system also included a centralised database server which logs usage data of all the EWHs for further processing and analysis.

## 1.6 Problem Statement and Objectives

This section presents the main problem statement to be investigated in this study, along with the objectives identified to systematically address the problem.

### 1.6.1 Problem Statement

EWHs typically contribute to the largest amount of residential energy consumption. Furthermore, EWHs are prone to putting stress on the grid periodically with the peak demand periods of most EWHs occurring simultaneously. Due to the thermal storage capacity of an EWH and the typical work schedule of many consumers, there is potential for intelligent DSM. Some consumers freely opt for some control scheme or device which could enable them to save money and time while employing some form of scheduling on their EWH. Utilities typically respond to peak demand periods with ripple control, a unidirectional control scheme.

This blunt instrument disconnects zones of EWHs, regardless of the individual consumer's efforts to reduce their energy footprint with scheduling, leading to scenarios where hot water may not be available when needed.

Furthermore, smart controllers with a developing infrastructure typically face a number of challenges during their initial usage period that may lead to a reduction in data quality obtained from the devices under control. These challenges are augmented in a developing country where the infrastructure is also not mature and still evolving. This loss of data leads to lower quality analysis which may ultimately reduce the comfort of the consumer who chose the controller.

## 1.6.2 Proposed Solution and Objectives

The problems mentioned in section 1.6.1 are addressed in this study from a data perspective. First, the data quality of the Geasy Project needs to be assessed and possible improvements investigated. The developed methods of data improvement should have a low computational complexity.

Second, the temporal structure of EWH usage needs to be investigated to identify the possible patterns that may provide insight into more advanced scheduling techniques for individual consumers. As an initial investigation, the feasibility of linear models to represent the temporal structure should be established.

Finally, the consumer-utility misunderstanding of personal and grid scheduling is addressed by utilising emerging SG technology to create an intelligent scheduling algorithm which takes into consideration the needs of both the utility and the consumer. This algorithm should operate on the grid level, whereby individual EWHs are controlled instead of zones of EWHs while maintaining good scalability.

### **Objective 1: Assess and improve current data quality.**

**1(a):** Investigate data flow from the Geasy device until it reaches the database for long term storage to identify possible causes of data quality reduction.

**1(b):** Assess the current data quality to establish baseline metrics and identify the consequences of the data quality reductions in the normal course of Geasy operation.

**1(c):** Develop numerical data cleaning routines as part of a data cleaning framework to address the identified data quality shortcomings.

### **Objective 2: Analyse and model temporal structure of EWH usage.**

**2(a):** Investigate EWH usage for temporal patterns that may lead to usage modelling.

**2(b):** Develop linear statistical models to model EWH usage and attempt forecasting.

**2(c):** Evaluate feasibility of linear statistical models for EWH usage forecasting.

### **Objective 3: Develop a smart grid demand-side management application.**

**3(a):** Revise the nodal EWH thermal models presented by Nel [25] to allow concurrent simulation for algorithm evaluation.

**3(b):** Investigate the main drivers for establishing a single EWH's schedule.

**3(c):** Develop a control algorithm that balances consumer comfort with utility supply for a grid of EWHs. The algorithm must be computationally inexpensive to enable good scaling performance.

## 1.7 Scope of work

This section provides the scope which restricted the focus of this study.

### 1.7.1 Data Wrangling

- The state of the numerical data quality obtained from the device sensors will be assessed; no additional text meta data quality will be assessed.
- Not all data sets will be used during development; only data sets that are deemed salvageable will be selected for this study.
- A data cleaning framework will be developed only for the numerical sensor data sets; no data cleaning for additional text meta data (such as EWH installation parameters, insulation, location etc.) will be developed.

### 1.7.2 Data Analysis and Forecasting

- Statistical time series analysis of EWH data will be performed only on the hot water usage data sets of individual EWHs.
- Linear statistical models will be developed based on previous analysis instead of non-linear models.
- Feasibility of linear statistical models for EWH usage forecasting will be assessed particularly in terms of how labour intensive and automated the model creation process is.

### 1.7.3 Demand-Side Management Application

- Existing EWH thermal model will be used and no model development performed; the model is assumed sufficiently accurate for the purposes of this study.
- Existing EWH thermal model will only be expanded from a scalar model to a vector model to allow concurrent, independent thermal modelling of EWHs.
- Existing field data will be used to evaluate the developed algorithm; no physical experiment is used to verify findings.
- No consideration will be made for health related aspects, such as Legionella growth in EWHs at lower set temperatures.

## 1.8 Contributions

This section provides an overview of the contributions of this study, which:

- Investigates the current numerical data quality of the data obtained during the Geasy project.
- Presents an overview of potential reasons for suffered data loss.
- Presents a novel numerical data cleaning framework for the Geasy data sets.
- Uses statistical techniques to analyse hot water usage data.
- Presents guidelines for building linear statistical models for time series data with the purpose of forecasting.
- Investigates the feasibility of linear statistical models for forecasting hot water usage.
- Expands the capabilities of an existing EWH thermal model to enable concurrent simulation while improving execution speed.
- Presents a novel DSM DLC algorithm for near real-time control of EWHs in a SG which considers the needs of both the utility and consumer.
- Presents a novel DSM DLC algorithm evaluation procedure which operates on the principle of macro simulation on grid level by micro-modelling individual EWHs.

## 1.9 Dissertation Structure

**Chapter 2:** presents a literature review of data quality and cleaning followed by an investigation of the current state of the data quality obtained from the Geasy project. Based on the findings, a numerical data cleaning framework is developed to improve the data quality in preparation for further processing. Performance metrics are established and used to evaluate the efficacy of the data cleaning framework on the required feature vectors.

**Chapter 3:** presents a literature review of statistical time series modelling followed by an investigation of statistical analysis and forecasting of consumer hot water usage. The linear SARIMA model is used to represent the temporal structure found in the time series. Statistical and visual performance of multiple grid parametric searched models are evaluated.

**Chapter 4:** presents a literature review of EWH modelling and grid demand scheduling followed by the development and evaluation of a novel peak demand manager algorithm. Performance metrics are established to evaluate the performance of the algorithm from both a utility and consumer perspective.

**Chapter 5:** concludes the study by evaluating the X objectives described in section 1.6.2. Recommendations for future work regarding each of the proposed solutions to the objectives are presented.



# Chapter 2

## Data Wrangling

Today data plays a key role in people's daily lives [26]. Global positioning system (GPS) coordinates from mobile phones [27], customer relationship management systems to the square kilometre array (SKA) which will experience a raw data flow volume of around 159 TB/s. Useful information or knowledge can be derived from these large quantities of data. However, many data applications fail to work successfully [28]. This may be due to a range of factors, including poor system infrastructure design, poor query performance and a lack of concern for the issue of data quality. Of these numerous failures, the aspect most certain to yield failure is the lack of concern for data quality [28].

There has been a growing awareness that high data quality is key to today's business success. The quality of any large real world data set depends on a number of factors of which the source is often a crucial factor. Yet, an inordinate proportion of data in most data sources is dirty [29]. Due to the principle of "garbage in, garbage out", dirty data will distort information obtained from it [30]. An important stage of working with data is data warehousing, where data is kept for long term storage and reliable access for data mining and deriving business intelligence [31]. All of these practices are directly sabotaged if low data quality is warehoused.

This chapter provides the research, design and results of a systematic data cleaning framework for Geasy data. In the following sections the fundamental concepts related to data quality and data cleaning are presented. This is followed by the design of a cleaning framework and ultimately the results obtained from this framework.

### 2.1 Current Data Wrangling Status and Challenges

This section gives an overview of the current status of data wrangling and the challenges that are faced during the process. Data wrangling refers to the processes of finding, interpreting, extracting, preparing and recombining data to be analysed and typically accounts for up to 70 % of the time spent on data analytics [32].

#### 2.1.1 The Concept of Data Quality

For effective data analysis, the data has to be clean. The qualities of data that determine whether data is clean is captured in the data quality. There are various definitions for what data quality is [33, 34, 35], however no formal definition of data quality exists [34].

### 2.1.1.1 Data Quality

Data quality can be defined as “fitness for use”, or the ability of data to meet the user’s requirements [34]. This definition implies the concept is relative to the intended purpose of the data. There are typically three approaches toward the study of data quality [36]. The intuitive approach is the most common and follows that the selection of data quality attributes for a particular study is based on the researcher’s experience or intuitive understanding about what attributes are considered important. This has the advantage that feature selection of attributes results in the most relevant attributes for the study [36]. Using the theoretical approach, the process responsible for causing the data deficiency during the data manufacturing process is studied. This has unfortunately proved to not be an adequate basis for improving data quality and is typically significantly worse than empirical approaches [36]. An empirical approach analyses data collected from data consumers to determine the characteristics they use to assess whether the data are fit for use in their tasks. This may reveal characteristics that researchers have not considered as part of data quality. However, the correctness or completeness of the results from the empirical study cannot be proven via fundamental principles [36].

### 2.1.1.2 Data Quality Dimensions

Data quality is measured by means of data quality dimensions. A data quality dimension is defined as a set of data quality attributes that represent a single aspect or construct of data quality [36]. Some commonly used data quality dimensions [37] are briefly discussed next.

**Accuracy:** The accuracy of the data is a measure of how "accurate/truthful" the physical phenomenon is recorded. The accuracy is affected by the sensors during measurement, the installation parameters of the sensors (how and where the sensors are connected) and the value processing done on the data.

**Completeness:** Data completeness is, as the term indicates, a measure of how complete a data set is. The more complete a data set is, the less missing values are present. This is often the more troubling aspect when it comes to data cleaning, as the true data may never be obtainable.

**Consistency:** The consistency of a data set indicates how consistently data types were used during storage. If there is no change between data types in a data set over a period of time, it is an indication of a high level of consistency.

**Relevance:** Data relevance indicates a measure of importance the data set has pertaining to its intended purpose. If values are critical to perform certain calculations, they are considered highly relevant. If values may be omitted with minimal to no impact on calculations, they are considered to have low relevance.

**Reliability:** The reliability of data is a measure of how truthful the recorded values are to the original measured values. This indicates whether the data has been tampered with, having values altered or removed while in storage.

### 2.1.1.3 Impacts and Costs of Data Quality

Data quality problems have become increasingly prevalent in practice with most organisations facing data quality problems [38]. High data quality is critical to an organisation's success; however, not many sufficiently deal with the experienced data quality problems. Low data quality may lead to numerous consequences which include loss of customer satisfaction, high running costs, inefficient decision making processes and low performance.

## 2.1.2 Improving Data Quality with Data Cleaning

In an effort to improve the data quality, data cleaning steps may be followed. Data cleaning is a highly involved and intricate process, to the degree of doctoral research [31]. To ensure high quality data, enterprises require a process which includes methodologies for the monitoring and analysis of data quality as well as methodologies for preventing or detecting and subsequently cleaning dirty data.

Data wrangling has multiple goals which include revealing deeper intelligence; providing accurate, actionable data in the hands of business analysts in a timely manner; reducing the time spent collecting and organising unruly data before it can be utilised as well as enabling data scientists and analysts to focus on the analysis of the data, rather than the wrangling [39].

### 2.1.2.1 Data Cleaning Applications

There is no commonly agreed formal definition of data cleaning. Definitions of data cleaning are area of application specific. Data cleaning is commonly included as a defining process in: data warehousing, knowledge discovery in databases (KDD) and total data/information quality management (TDQM) [31]. Data warehousing typically utilises data cleaning when several databases are merged to mitigate duplicate records [40]. Before KDD, data cleaning is performed as a preprocessing step to ensure high data quality for analysis.

### 2.1.2.2 Data Cleaning Stages

Comprehensive data cleaning has been defined as the entirety of operations performed on existing data to remove anomalies and receive a data collection which is an accurate and unique representation of the mini-world [39]. Typically, there are three major stages within the data cleaning process [39]. The first stage involves defining and determining error types. The second stage incorporates the error definitions and a search is done to identify error instances. The third stage involves correcting the uncovered errors.

These three stages are completed in the four steps of the cleaning process [39]. The first step is the data audit, where anomalies are detected typically through statistical methods. In this step the integrity constraints are specified by the domain expert [39]. The second step is the workflow specification, where the workflow is the sequence of operations that detects and eliminates anomalies in the data. This step develops the data cleaning algorithms required to correct the anomalies found in step one. One of the main challenges of this step is to specify a data cleaning workflow that is to be applied to dirty data automatically in order to eliminate all anomalies in the data [39]. The third step is the workflow execution, when the workflow has been verified to satisfactorily clean the data. This implementation should enable efficient cleaning to ensure good performance even on large data sets. This performance may be achieved through a heuristic to achieve

the best accuracy while maintaining satisfactory execution speed [39]. The final step is post-processing and control, where the results of workflow execution are inspected to verify the correctness of the specified operations. This may also include exception handling for data that was not corrected during the workflow execution [39].

## 2.2 Geasy Data Flow

The data to be used in this project was acquired through the Geasy Smart EWH Controller project from BridgIoT [41]. The Geasy project spans over 400+ SECs installed throughout South Africa. This section will provide a brief overview of the Geasy system topology.

### 2.2.1 Geasy SEC

The Geasy (a portmanteau of ‘geyser’ and ‘easy’), is a Smart EWH Controller (SEC) with monitoring capability [17]. This SEC encapsulates a computation unit, communication unit, sensors and actuators [24]. An overview of these elements follows here.

#### 2.2.1.1 Computation Unit

The Geasy utilises an Atmel Xmega 128A4U 8-bit microcontroller at a clock speed of 38 MHz. This unit is programmed in C, allowing lower overheads while interfacing between the functional units of the SEC.

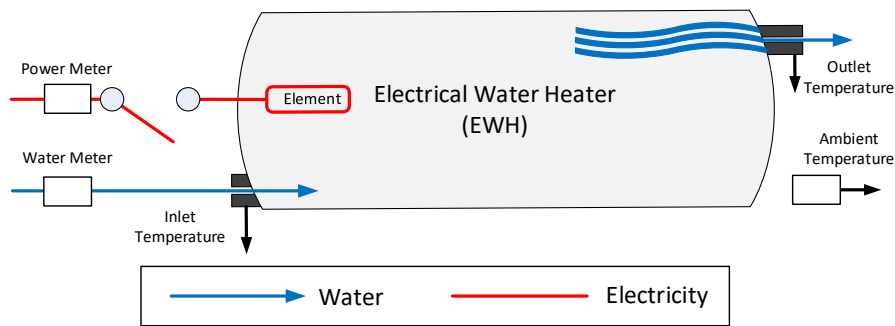
**Data Saving:** Amongst the functions of the SEC is data preparation for communication. Due to the reliance on cellular communication, the design decision was made to minimise the amount of data to be transmitted. To realise this reduction in data, a compression technique was implemented whereby only certain values would be reported. The temperature sensor data is not compressed. The compression technique uses a default value for comparison to determine whether a value should be transmitted. For the water flow sensor and power sensor, the default value is 0. If either the water flow sensor or the power sensor detects successive readings of 0, only the first 0 is transmitted, with the following 0s being suppressed.

#### 2.2.1.2 Communication

The Geasy SEC utilises a 2G modem for communication via a cellular network. This modem also communicates via serial communication with the aforementioned microcontroller. The communication mode is opaque TCP which allows lightweight data package transmission. Each modem has a unique international mobile equipment identity (IMEI) number which identifies the specific device which tries to establish communication. Typically this number is used by Global System for Mobile Communications (GSM) networks to identify valid devices. Communication is designed to transmit measured values every minute.

#### 2.2.1.3 Sensors

The SEC is equipped with sensors to observe the physical phenomenon surrounding the EWH as shown in Fig. 2.1; those of interest are discussed below.



**Figure 2.1:** The arrangement of the Geasy SEC sensors. Adapted from [1].

**Temperature sensors:** Four locations of interest on the EWH are monitored by analogue temperature sensors. These areas are measured extrinsically in order to provide sufficient accuracy while remaining non-invasive. This is done mainly to keep installation cost and complexity down. Furthermore, the sensors are accurate to 1 °C (with calibration), which may hamper the overall accuracy of the measurements and subsequent calculations. The four temperature sensor locations are:

1. **Outlet:** On the hot water outlet pipe, as close as possible to the EWH housing.
2. **Far:** On the hot water outlet pipe, around 500 mm from the EWH housing.
3. **Inlet:** On the cold water inlet pipe, around 500 mm from the EWH housing.
4. **Ambient:** Within 1 m radius from the EWH, in the same environment.

**Flow meter:** The water flow is measured by an in-line flow-meter connected to the cold water inlet pipe. The meter utilises a turbine design whereby a turbine rotates when positive flow (into the EWH) is detected. Flow vanes prevent negative flow from rotating the turbine in the opposite direction. The rotation of this turbine is measured as pulses via a hall-effect sensor. This flow meter has a resolution of 0.5 L.

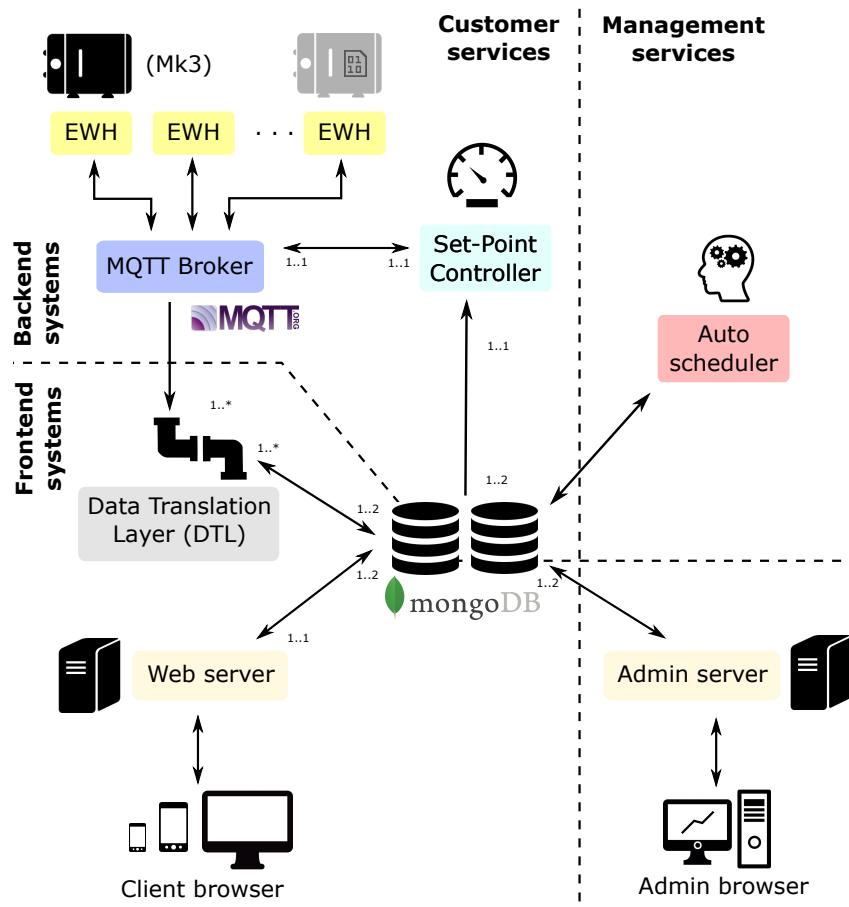
**Power meter:** A current transformer with 5 % verified accuracy is used to measure the supply current to the EWH at a sampling rate of 500 Hz. The RMS current is then calculated from these samples over a period of 160 ms by the microcontroller. The voltage is not measured, but is instead assumed to be constant at 230 V RMS. From these figures, the average power is calculated by:

$$P_{AVG} = V_{RMS} I_{RMS} \quad (2.1)$$

Where  $P_{AVG}$  indicates the average power, in watt (W);  $V_{RMS}$  represents the RMS voltage, in volts (V); and  $I_{RMS}$  is the measured RMS current, in ampere (A).

## 2.2.2 Communication Path

A key aspect of the Geasy SEC, is the ability to provide bi-directional communication between the user and the EWH. The communication infrastructure is achieved through a



**Figure 2.2:** Geasy topology showing various nodes in the communication. Adapted from [24].

marriage of multiple protocols, each fulfilling a role at various layers of abstraction. The Geasy topology can be seen in Fig. 2.2. The communication aspects are briefly discussed below.

### 2.2.2.1 TCP

As briefly mentioned in section 2.2.1.2, the SEC uses a 2G modem for communication along with the TCP protocol. The TCP protocol is used to establish a connection over the internet. This protocol provides data packets in a reliable, ordered and error-checked stream. This robustness is achieved at the expense of lower latency operation.

### 2.2.2.2 MQTT

Message Queue Telemetry Transport (MQTT) is a lightweight M2M connectivity protocol. This protocol has an acute focus on keeping the messaging transport footprint to a minimum, making it especially useful when network bandwidth is at a premium. It works on a publish/subscribe methodology whereby a central broker links clients to various topics. These clients may either be subscribed to or publish to the various topics. The SEC design utilises a data translation layer (DTL) which acts as a proxy between the MQTT broker and the database (DB). The DTL is essentially a client subscribed to all the SEC topics. These messages are then captured for storage in the DB.

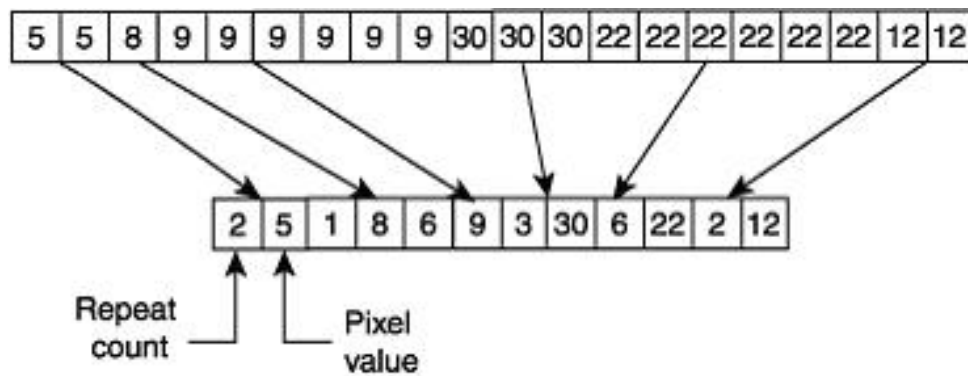


Figure 2.3: Ideal case run-length encoding.

### 2.2.2.3 Run-Length Encoding

Run-length encoding (RLE) is a lossless compression technique that is particularly effective with data that often repeats in long sequences. The operating principle is that a successive sequence of equivalent values is encoded into a repeat count and the value. This operation is shown in Fig. 2.3. Only on transition of the sequence of values will a new value be encoded, leading to an effective data compression that is simple to decode. In this system, an additional field provides a key to whether this is the first observation or a changed observation, for use in decoding.

### 2.2.2.4 MongoDB

The Geasy infrastructure provides a web application for the consumer to monitor and control their EWH via a browser. A more powerful version of this application is available for administration personnel to monitor the health of the network as well as maintain and improve individual connected SECs. This application was built using the Meteor.js framework [42] which integrates Node.js [43] for web server capability and MongoDB [44] for database functionality.

## 2.3 Data Cleaning Methods Used

The data from the Geasy project is primarily numeric and as a result, the cleaning methods will focus on numerical cleaning. Due to the source of data being already stored in a DB, the processing and analysis will be of a non-causal nature. The selected methods were chosen to fulfil various tasks while remaining relatively computationally inexpensive. This section describes the various methods employed to analyse and process the data.

### 2.3.1 Sampling Period Regularisation

To correct the time stamp drift which is observed in the recorded data, the sampling period is corrected to 1 minute intervals. This is a requirement for many of the further processing and analysis techniques that will be employed. The specific method used to achieve the regularised sampling period is time grouping, whereby all observations are processed with the split-apply-combine methodology. First, the observations are split into groups based on the time stamp value at a 1 minute resolution. These groups are

expected to have only one value each; however, missing or duplicate observations may exist. The apply stage applies a mean function to the data, which should return either a missing or single value. In the case of a single value this returns the original value, but in the case of duplicate values the mean is obtained and returned. Due to duplicate values having the same value, the mean is a single observation with the original value. In this way, the sampling period regularisation also removes duplicate data. In the final step, all these per minute observations are combined back into the original data structure which is ready for further processing.

## 2.3.2 Missing Value Imputation

In order to provide uniformly sampled data, the missing values need to be estimated. Uniformly sampled data is important to avoid bias, simplify the analysis of the data and to improve overall efficiency. The estimation is done with various methods and rules, based on the particular feature vector under consideration.

### 2.3.2.1 Persistence

In typically constant-valued feature vectors, short spans of missing data may be assumed to have a persistent value. Depending on the type of data, the missing values may be imputed by persisting the last valid value for a maximum amount of intervals. Once the previous valid value has persisted for the maximum amount of intervals, a default value is imputed for the remaining missing values.

### 2.3.2.2 Interpolation

Interpolation is used to impute new data points within the range of a discrete set of known data points. This method is a popular choice for dealing with missing values. The simplest and most computationally inexpensive form of interpolation is linear interpolation. This method fits a curve to the known data points using linear polynomials. From this generated curve, the missing data points are imputed. As a trade-off for its computational requirement, the error resulting from this method is proportional to the square of the distance between the known data points.

## 2.3.3 Spectral Analysis

Spectral analysis is useful during data processing to identify frequency distributions of interest within a signal. This allows the frequency bands which contain the main information to be identified. Once the identification is complete, steps to extract the information may be designed to obtain the data.

### 2.3.3.1 Fast Fourier Transform

To decompose a discrete time signal with constant period, the discrete Fourier transform (DFT) is used. This transform results in the discrete complex frequency domain representation of the original signal. The mathematical representation of this transform can be written as:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \quad (2.2)$$



Where  $X_k$  is the complex number at position  $k$  of the Fourier sequence;  $x_n$  is the original signal measurement at position  $n$ ; and  $N$  is the total number of elements in the discrete signal. The DFT, however, has a computational order complexity of  $O(n^2)$ . Due to this, the fast Fourier transform (FFT) was developed to be a more efficient implementation of the DFT, with a computational complexity of  $O(n \log n)$ . Furthermore, the DFT for a purely real input results in a Hermitian-symmetric output. This means the negative frequency components are complex conjugates of the positive frequency components, and therefore the negative frequency components are redundant. The frequency spectra of each sample is then calculated and finally all the spectra are then synthesised together into a single frequency spectrum.

### 2.3.3.2 Energy Spectral Density

The energy spectral density is obtained from the FFT and is an indication of how the energy of the signal is distributed with frequency. The spectral energy density of a signal  $x(t)$  is calculated by using:

$$\Psi_x(f) = |X(f)|^2 \quad (2.3)$$

Where  $\Psi_x(f)$  represents the signal's spectral energy density and  $X(f)$  is the Fourier transform of  $x(t)$ . Using the discrete decomposition of a signal into respective frequency components, this method can be used to identify underlying signal components among noise. Being able to identify underlying signal components provides insight into filter design which may help with extracting the original signal.

### 2.3.4 Digital Filtering

Digital filtering can be used for either signal separation or signal restoration. Typically, a digital filter is used for signal separation when noise and interference need to be removed. There are multiple considerations regarding filter design, such as filter type, filter parameters and lag tolerance of the resulting filtered signal [45].

#### 2.3.4.1 Essential Bandwidth

The energy of practical signals is finite, which means that as the frequency tends to  $\infty$ , the signal energy spectrum must approach 0. Due to this phenomenon, most of the signal energy is contained within a bandwidth such that negating the energy content of components above this bandwidth will have minimal impact on the signal shape and energy. This bandwidth is called the essential bandwidth (EBW) of the signal and is typically selected to be 95 % or higher [46]. To calculate the energy contained within the complete signal, the following discrete time to DFT relationship based on (2.3) can be used:

$$E_x = \sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 \quad (2.4)$$

With  $E_x$  representing the total energy;  $N$  represents the total number of samples;  $xn$  represents the discrete time sample at position  $n$ ; and  $X_k$  is the complex number at position  $k$  of the Fourier sequence. To obtain the bandwidth which satisfies the EBW criterion, the following may be used:

$$E_s = \gamma E_x \quad (2.5)$$

Where  $\gamma$  represents the percentage of energy which is to be retained in the essential bandwidth calculations. The essential bandwidth is then calculated from (2.4) as:

$$E_s = \frac{1}{N} \sum_{k=0}^m |X(k)|^2 \quad (2.6)$$

Where  $m$  indicates the number of components required which satisfies (2.5).

### 2.3.4.2 Low Pass Filter

A low-pass filter is used to extract low frequency information from a signal, while suppressing high frequency information. Noise typically manifests as high frequency information which was not present in the original signal. Sharp discontinuities present in a time signal are also removed with a low pass filter.

### 2.3.4.3 Butterworth Filter

The Butterworth low-pass filter produces a maximally flat amplitude response in the pass-band while providing a good rate of attenuation in the cut-off band. Another benefit of this filter is the absence of ripple in both the pass- and reject bands. Typical parameters required during the design of a Butterworth filter include the order of the filter, which indicates the rate of attenuation, and the cut-off frequency where the gain drops to  $\frac{1}{\sqrt{2}}$  (the -3 dB point). With digital filter design, the cut-off frequency is typically expressed as a normalised value between 0 and 1, where 1 represents the Nyquist frequency times pi radians/sample.

### 2.3.4.4 Zero-Phase Filter

A zero-phase slope is achieved by first reverse passing a filter over a signal, inducing a phase offset, then forward passing the same filter over the signal, undoing the phase offset but retaining the amplitude response of both filter sweeps. This equates to the square of the filter's amplitude response. With  $h(n)$ ,  $x(n)$  and  $y(n)$  representing the filter's impulse response, the input signal and the filtered signal respectively, the corresponding Fourier transforms are obtained as  $H(e^{j\theta})$ ,  $X(e^{j\theta})$  and  $Y(\theta)$  respectively. The working principle is illustrated below. Equation 2.7 indicates the reverse pass of the filter and 2.8 indicates the forward pass of the filter added to the previous result. This finally results in the square of the magnitude of the filter affecting the original signal, as shown in 2.9. This type of filtering is non-causal, as all the values are required at time of processing.

$$Y(\theta) = X(e^{-j\theta})H(e^{-j\theta}) \quad (2.7)$$

$$Y(\theta) = X(e^{-j\theta})H(e^{-j\theta})H(e^{j\theta}) \quad (2.8)$$

$$Y(\theta) = X(e^{-j\theta})|H(e^{j\theta})|^2 \quad (2.9)$$

## 2.4 Data Cleaning Tools Used

This section will provide a brief overview of the tools used throughout the development of the data interface and data cleaning routines.

### 2.4.1 Python

Python is a popular high-level programming language for general-purpose programming. The design philosophy of Python emphasises code readability with syntax that allows concepts to be expressed in fewer lines of code. Furthermore, Python has seen large interest from the data science community, having been rated as the most popular data analysis language in 2017 [47]. Python boasts a large set of libraries with over 113k packages [48] available at time of writing.

Of particular note are the Pandas, Numpy and Matplotlib libraries. Pandas is a data analysis toolkit for Python, Numpy is a numerical processing library with great optimisations being implemented and a lot of execution time spent in cython. Matplotlib is the default plotting library for standard Python.

#### 2.4.1.1 Missingno

Missingno is a Python data completeness visualisation tool. The tool provides multiple ways to indicate the state of completeness of a data set. The two most relevant and appropriate methods are the matrix plot and bar chart. The matrix plot provides a chronological data presence view, where the vector fields are represented as columns and observations are represented as rows, flowing from top to bottom. On the right is a completeness sum, which indicates the total number of observations present during each observation. The bar chart indicates nullity per vector field, showing the percentage of the data that is present.

### 2.4.2 MongoDB

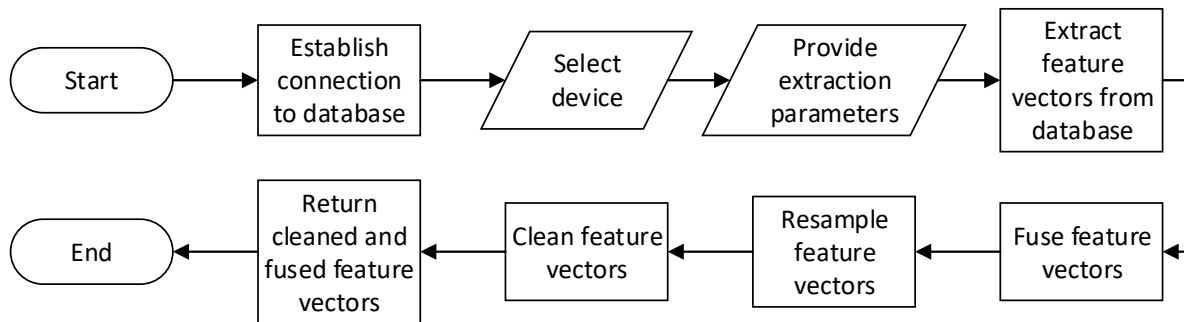
MongoDB is a document database built on the not only structured query language (NoSQL) mechanism with focus on scalability and flexibility. This is in contrast to structured query language (SQL), whereby data is stored in relational tables. NoSQL provides a variety of database types, which adds to the flexibility of NoSQL. The most common working principle of a NoSQL database is that each key is paired with a complex data structure called a document. This enables a key-value pair storage paradigm whereby each document in the database has a unique key. NoSQL DBs are gaining popularity in big data and real-time web applications [49].

#### 2.4.2.1 Collections

Similar documents in a NoSQL database are stored in collections, analogous to SQL tables. The main difference is that the documents are not required to have the same exact structure, allowing more flexible data storage in agile environments.

#### 2.4.2.2 Documents

The main storage elements in a NoSQL database are documents, stored in JSON-like key-value pairs. The NoSQL engine generates a random unique key for each document,



**Figure 2.4:** Data interface operational flow.

ensuring that documents have a unique identifier as well as ensuring each document is unique. If another unique key is required for a document, this can be changed afterwards.

NoSQL documents are highly flexible in the sense that they can be used to conform to a template (schema), as with SQL; however, the ability to modify future documents as the data becomes more complex is available. This adds power to the NoSQL paradigm, especially in rapid development environments where the complexity and volume of data could change rapidly. A document within a collection could store a single integer initially and as the requirements evolve, future documents of the same collection could store a combination of integers, strings and arrays.

## 2.5 Data Tool Developed

During the design and development of the data cleaning framework, a more convenient data handling interface was developed. This interface consolidated a number of data acquisition and preprocessing routines to speed up the process of obtaining viable data. This section briefly covers this development.

### 2.5.1 Data Interface Requirements

The development of the interface was guided by the idea of automating the extraction and preprocessing of the data. This would result in a consistent data extraction experience regardless of the selected device to be analysed. Fig. 2.4 gives an overview of the operational flow of this interface.

Based on these desired operations, the requirements were established as seen in Table 2.1.

### 2.5.2 Data Interface Implementation

This section will give an overview of how the data interface was implemented in order to satisfy the requirements of Table 2.1. The interface was developed in Python to conveniently integrate with the rest of analysis ecosystem.

#### 2.5.2.1 Connectivity

The connectivity requirements include credential management as well as database location agnostic operation.

**Credential Mangement:** As a result of the server-client paradigm of MongoDB, each client requires valid credentials to access the data contained within the DB. Along with valid credentials, the server host IP address and specific DB are required. To manage all these connection parameters in a multi-user application, a credential file was developed using JSON notation. The structure of the file is as follows:

- "MONGO\_HOST": host\_ip,
- "MONGO\_PORT": port,
- "MONGO\_DB": db\_name,
- "MONGO\_USER": user\_name,
- "MONGO\_PASS": password

This file is read in when the data interface is loaded and provides user-specific access to the DB. This simplifies the connection aspect of the data extraction stage.

This satisfies requirement 1.1 from Table 2.1.

**Database Location:** The interface was designed with portability in mind, to enable the usage of the interface with a local or remote database. The major difference between these two connections comes down to correct credential management, as shown in section 2.5.2.1.

**Local:** A local copy of the DB was used during the lifetime of this project. This provided numerous benefits which include significant speed improvement while keeping the remote production DB free from persistent requests. Furthermore, the evolving DB would greatly increase the development time of a suitable interface and cleaning routines. During development the local copy was used in a sandbox mode by not forwarding the port

**Table 2.1:** Data interface requirements.

ID	Requirement
<b>1</b>	<b>Connectivity</b>
1.1	Correctly manage MongoDB credentials.
1.2	Connect to local MongoDB instance.
1.3	Connect to remote MongoDB instance.
<b>2</b>	<b>Data Selection</b>
2.1	Select data for single device.
2.2	Select device based on device IMEI.
2.3	Select device based on device alias.
2.4	Extract required device feature vectors.
2.5	Extract additional device details.
2.6	Extract device feature vectors within specified time range.
2.7	Correctly translate region specific time zone requirement.
<b>3</b>	<b>Vector Aggregation</b>
3.1	Fuse feature vectors on time stamp.
<b>4</b>	<b>Data Cleaning</b>
4.1	See Table 2.3.

through the firewall, allowing a secure environment for the development of the interface. The local version of the DB is up to date until 2016-10-20 and all development was done on the data up until this point.

This satisfies requirement 1.2 from Table 2.1.

**Remote:** The remote DB is accessible with this interface, just like a local DB is accessed, the difference being only the provided credentials. Though successfully implemented, the throughput rate over the network would cause a bottleneck in execution time and keep the production remote DB busy servicing requests.

This satisfies requirement 1.3 from Table 2.1.

### 2.5.2.2 Data Selection

The Geasy network is fed data from the remote Geasy nodes, each node representing an EWH. In the database, the metrics are separated into collections, with each document containing a time stamp, the metric value and a reference to the responsible EWH. Using this basic idea, the interface was designed to extract pertinent data from selected EWHs. A per EWH basis was selected as a suitable container to group all related data, i.e. data was extracted for a single EWH at a time. This allows ease of use when analysing data from a selected EWH and further MG analysis can be achieved by aggregating the extracted data.

This satisfies requirement 2.1 from Table 2.1.

**Device Selection:** The Geasy devices are identified in two ways; first is a hardware specific IMEI number and the second is the administrator assigned device alias. Partial identifier selection is achieved through regular expression searching. If no device matching the identifiers is found, the selection process indicates that there is no record matching the sought after device. Furthermore, in the case where a non-unique identifier is supplied, the selection process will indicate that an error has occurred and the search term needs to be refined.

**IMEI:** The IMEI number is a 15 digit sequence of decimal numbers. This value is recorded during registration of a Geasy unit as a form of identification. The specific number is stored as a string in each document which is related to that specific unit. To select a device based on its IMEI number, either the full number or a unique partial of the number can be used as input.

This satisfies requirement 2.2 from Table 2.1.

**Alias:** The alias of a device is a unique, administrator-assigned name for a Geasy device. This value is assigned during device registration, but can be changed at any time by an administrator. As with the IMEI number, to select a device based on its alias, either a full number or a unique partial of the alias can be used as input. The search term is case-insensitive to improve user-friendly usage. The alias is not used as an identifier for a device in each document and requires a lookup from the devices collection to obtain the IMEI.

This satisfies requirement 2.3 from Table 2.1.

**Table 2.2:** Sensor identification in database along with sensor measurement unit.

Sensor	ID	Unit
Outlet	T1	°C
Far	T2	°C
Inlet	T3	°C
Ambient	T4	°C
Flow	Hm	L/min
Power	W	W

**Feature Vector Selection** The sensors of the Geasy each record a feature of interest. Successive periodic records of these features create temporal feature vectors which are to be further analysed. The sensors are covered in section 2.2.1.3 with a layout of the connections provided in Fig. 2.1. Of these six sensors, only five are required for the purposes of this thesis. These five features are determined from the EWH energy flow equations in section 4.2 and include the power, water flow, outlet temperature, inlet temperature and ambient temperature. The far outlet temperature sensor is not relevant to the energy balance as used in this thesis as the original purpose was for a usage detection algorithm [20].

**Selection of Feature Vectors:** To identify the sensor measurements, each is provided a unique key and stored in the relevant collection. The collections include electricity, temperature and water. The sensors, their unique identifiers and units are shown in Table 2.2. Successful extraction of the feature vectors requires knowledge of the data locations within the collections during the MongoDB request. This satisfies requirement 2.4 from Table 2.1.

**Additional Device Details:** During SEC installation and registration, some additional information was recorded about the physical EWH. This includes EWH volume, element rating, orientation and geographic location. These attributes provide additional dimensions for a more accurate comparative analysis. Extracting these attributes requires the selection of the devices collection and selecting the specific device based on IMEI. This satisfies requirement 2.5 from Table 2.1.

**Time Slicing:** The temporal information of each value in the feature vectors is stored as a Unix time stamp. Unix time describes a point in time by the number of seconds that have elapsed since 00:00:00 coordinated universal time (UTC), Thursday, 1 January 1970. This provides a method of encoding time as a number for each time stamp. With UTC as the reference time zone, it is important to consider the offset from UTC, depending on the location under consideration. In South Africa the offset is +2 hours, indicating the local time zone is UTC +02:00.

To select observations during a certain time period, the starting and ending time is required along with the UTC offset. For convenience, the time is expected in a Python datetime format and the offset as a signed integer. The Python datetime format allows a user to specify the time in a more natural format which includes the standard time elements starting from a year down to a second. The combinations of the datetime start and end times, along with the UTC offset, are parsed into Unix time stamps. These time stamps are then used during the selection of feature vectors for a specified period of time.

**Table 2.3:** Data cleaning stages.

ID	Stage
<b>DC1</b>	<b>Investigation of Data Quality</b>
DC1.1	Completeness
DC1.2	Accuracy
DC1.3	Consistency
DC1.4	Reliability
DC1.5	Relevance
<b>DC2</b>	<b>Transformation</b>
DC2.1	Sampling period regularisation
<b>DC3</b>	<b>Outlier Management</b>
DC3.1	Outlier mitigation
<b>DC4</b>	<b>Missing Value Imputation</b>
DC4.1	Complete data sets
<b>DC5</b>	<b>Sensor Noise</b>
DC5.1	Filtering

A design choice was made to time slice data an additional 4 days from what is specified. An overlap of 2 days preceding and succeeding the data was selected in an attempt to mitigate edge data inconsistencies. Once the data has been processed, the results from the selected (non-extended) data set is returned.

Successful selection of a single measurement value requires the correct specification of database collection, sensor ID and UTC Unix time stamp.

This satisfies requirements 2.6 and 2.7 from Table 2.1.

### 2.5.2.3 Vector Aggregation

Leading towards further preprocessing, the extracted vectors are consolidated into a single data structure. This provides a more intuitive tabular data format to work with. All the vectors are merged on the recorded time stamp to ensure all observations align on the correct time of measurement. All data cleaning routines are to be run on this data structure, with the fully cleaned data being available in this tabular format for easier analysis and processing.

This satisfies requirement 3.1 from Table 2.1.

### 2.5.2.4 Data Cleaning

One of the most significant routines to be run by the interface is the cleaning of the extracted data. This topic is vast, spanning multiple aspects for each feature vector. Due to this complexity, the data cleaning routines are covered in section 2.6.

This satisfies requirement 4.1 from Table 2.1.

## 2.6 Data Cleaning

In section 2.2 the data flow of the Geasy infrastructure was investigated which acts as a precursor to investigating the data quality. This is important as the nature of both the missing data and the data overall plays a vital role in the methodology used to preprocess the data. This section will cover the various methods of cleaning the extracted data. The



**Table 2.4:** High and low data integrity comparison.

Label	Complete (%)	
	High	Low
<b>W</b>	8.3	4.9
<b>T1</b>	93.0	51.5
<b>T3</b>	93.0	51.5
<b>T4</b>	93.0	51.5
<b>Hm</b>	8.0	5.9

focus of this procedure is to automate the process as much as possible due to a large amount of data that may be involved [39] by developing a data cleaning framework.

## 2.6.1 Initial Data Investigation

The initial data investigation aims to assess the current data quality. The quality depends on dimensions that include accuracy, completeness, consistency, relevance and reliability, as covered in section 2.1.1.2. Higher data quality enables higher quality analysis.

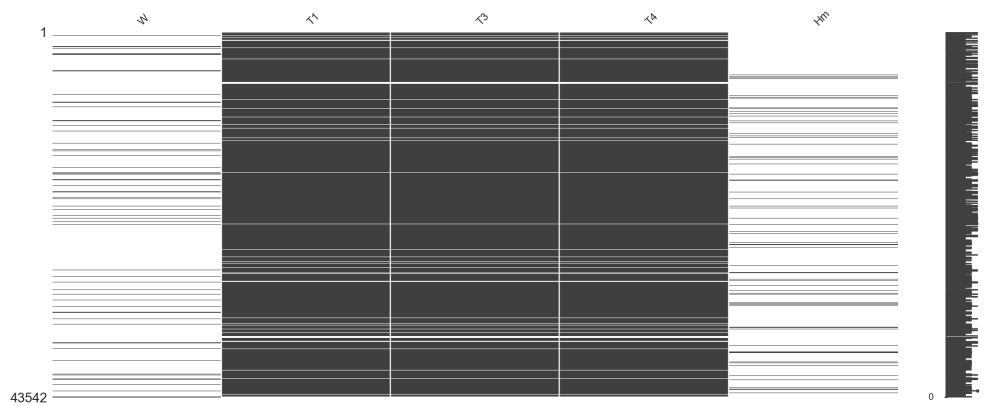
Before considering the required data cleaning steps, an overview of the current data fitness is required. Data cleaning is typically an iterative process and unique to each data set. The requirements for the preprocessed data can be found in Table 2.3.

### 2.6.1.1 Completeness

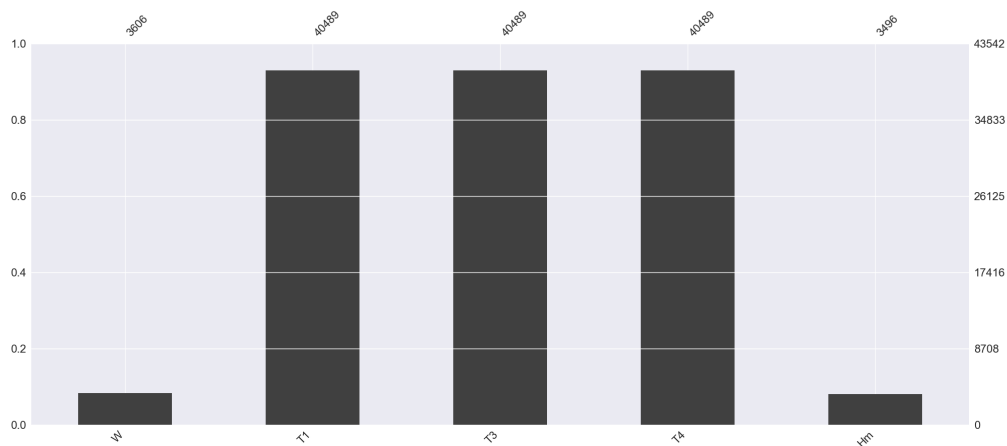
The completeness of data is directly measured by the amount of data present. This is typically the first aspect that suffers when a system experiences technical difficulties. It is also one of the largest contributors to the quality of data before preprocessing and therefore has a large impact on the accuracy and quality of the cleaned data.

Using the Missingno tool, as described in section 2.4.1.1, the data sets were investigated and evaluated for fitness for further analysis. From the sets, a representative from a high level of completeness and a low level of completeness were selected for comparison. Starting with the data set which has a high level of completeness, the resultant Missingno matrix plot and bar chart can be seen in Fig. 2.5(a) and 2.5(b). From these figures, it is evident that the data set, specifically the temperature vectors, reach a completeness of 93%. As discussed in section 2.2.1.3, it is expected that the temperature data sets will be the most complete as the temperature sensor data is always sent; no data saving is performed on these values. This makes the temperature data completeness representative of the SEC's data completeness. In contrast, the power and hot water usage data values are compressed by not reporting the values if they are the default value. As a result of this, the power and hot water usage data values have completeness levels of 8.3 % and 8.0 % respectively.

On the other end of the spectrum, besides units that flat out have no reported data values, there are units which have large amounts of data missing. Representative of such a data set is Fig. 2.6(a) and 2.6(b). Evidently there is intermittent data in the temperature data vectors, with a large outage of data experienced during the early stages of operation. The overall data completeness clearly indicated that the data experienced significant losses during operation. The temperature, power and hot water data set completeness for this unit was found to be 51.5 %, 4.9 % and 5.9 % respectively.



(a) High data completeness chart.



(b) High data completeness bar chart.

**Figure 2.5:** High level of data completeness.

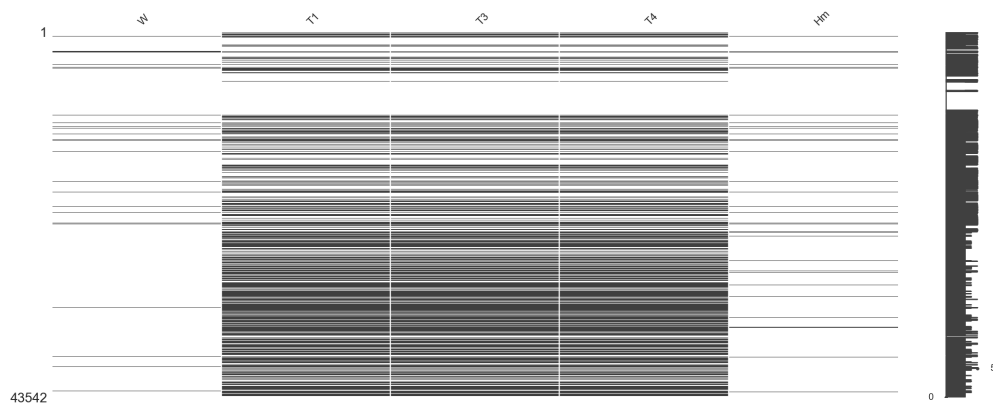
Table 2.4 provides a comparison of the two data sets, the first with a high data completeness and the second with a low data completeness. This satisfies the completeness investigation requirement DC1.1 in Table 2.3.

### 2.6.1.2 Accuracy

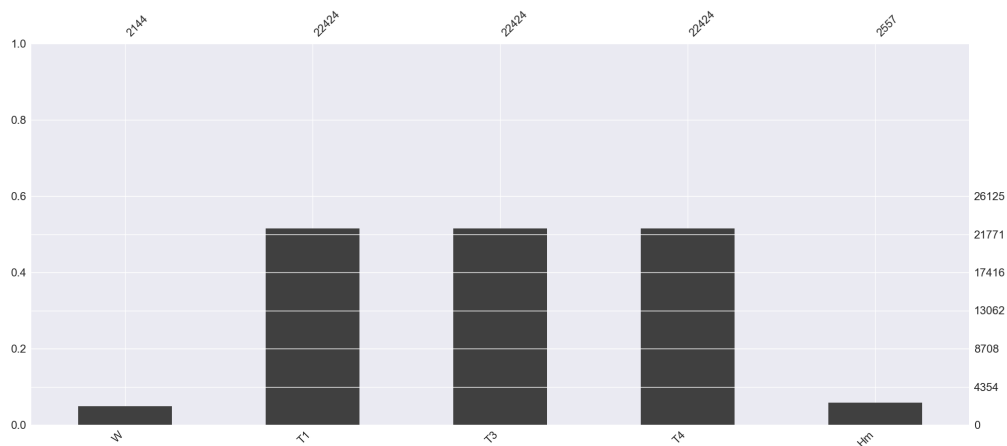
The accuracy of a data set directly influences the quality of the analysis of the data and the insight gained from that.

**Values:** In the context of preprocessing data that originates from a system that was deemed accurate, the main triggers for inaccurate data are seen as unexpected values, typically outliers. Considering the temperature data of an EWH, temperature values up to 60 °C are expected [50] and temperatures exceeding 80 °C are not expected and may indicate erroneous values.

Considering the frequency plot of the outlet, inlet and ambient temperatures, as seen in Fig. 2.7, a clear distribution can be seen for all below 80 °C. There are, however, outliers for all three of these sensors at 100 °C, which lie outside of the expected range



(a) Low data completeness chart.



(b) Low data completeness bar chart.

**Figure 2.6:** Low level of data completeness.

of values. Considering the raw data from the outlet sensor, a significant spike is evident around 2016-10-07, which is unlike any other recorded values. Data cleaning will need to effectively handle outliers such as this.

**Time Stamps:** Another aspect to consider is the time accuracy of the data. The SECs are designed to report in at a constant period of 1 minute; any deviation from this registers as an inaccurate period of sampling. Considering successive observations as shown in Fig. 2.9(a), a time drift was observed. The drift seems constant, with every observation being recorded about 1 min and 1-2 s after the previous observation. This drift could lead to regular missed observations, reducing the completeness of the data set. Fig. 2.9(b) shows the frequency distribution of the seconds value of the recorded time stamp, indicating no clear pattern to more frequent seconds values.

This satisfies the accuracy investigation requirement DC1.2 in Table 2.3.

### 2.6.1.3 Consistency

The provided data had a high level of consistency regarding the particular fields of interest. All the sensor measurements were stored as numerical data types with no erroneous data types present. The consistency was a concern due to the rapid development of the Geasy project and the expected evolution of the database. This contributed to the selection of an offline data set for development and analysis.

This satisfies the consistency investigation requirement DC1.3 in Table 2.3.

### 2.6.1.4 Reliability

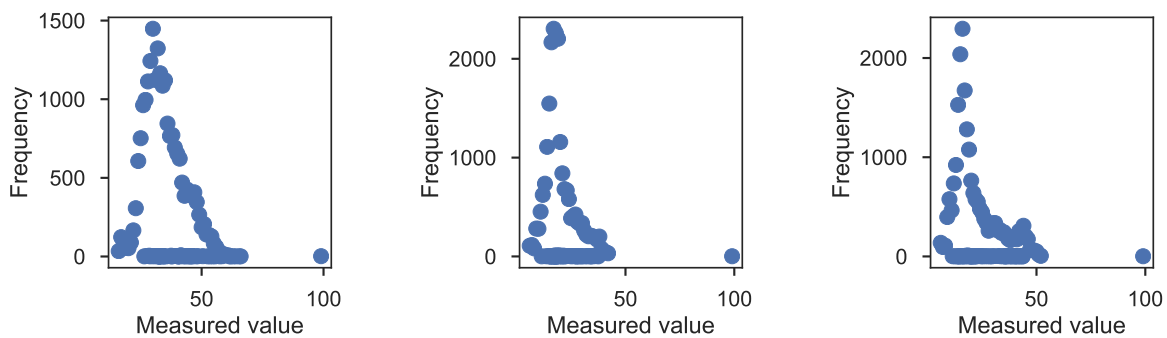
The provided data set is assumed to have a high level of reliability and no further alterations, besides improving the data quality, were made.

This satisfies the reliability investigation requirement DC1.4 in Table 2.3.

### 2.6.1.5 Relevance

The recorded values in the database primarily include all the required values to perform effective analysis of the data. Of particular importance are the sensor values. These sensors are discussed in 2.2.1.3. Of the six sensors, five are relevant to the analysis to be performed in this dissertation. The one sensor that is irrelevant for this study is the outlet far temperature sensor, which was previously used to estimate flow rates and event times. Furthermore, where available, the EWH volume and element ratings are also important metrics. The relevant parameters, along with their units are listed in Table 2.5.

This satisfies the relevance investigation requirement DC1.5 in Table 2.3.



(a) Outlet temperature.

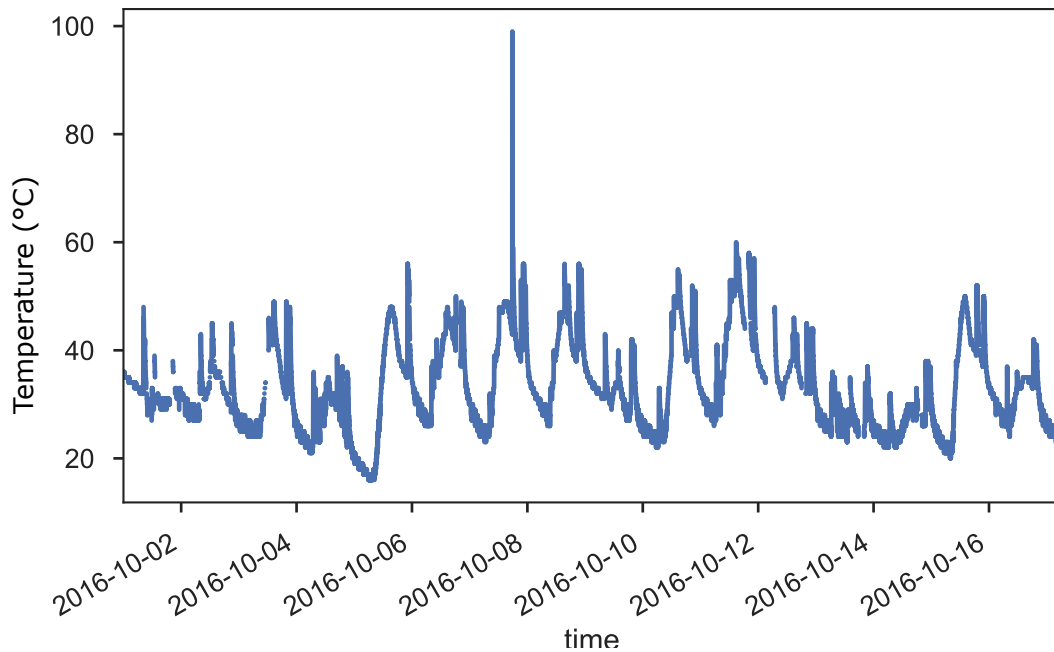
(b) Inlet temperature.

(c) Ambient Temperature.

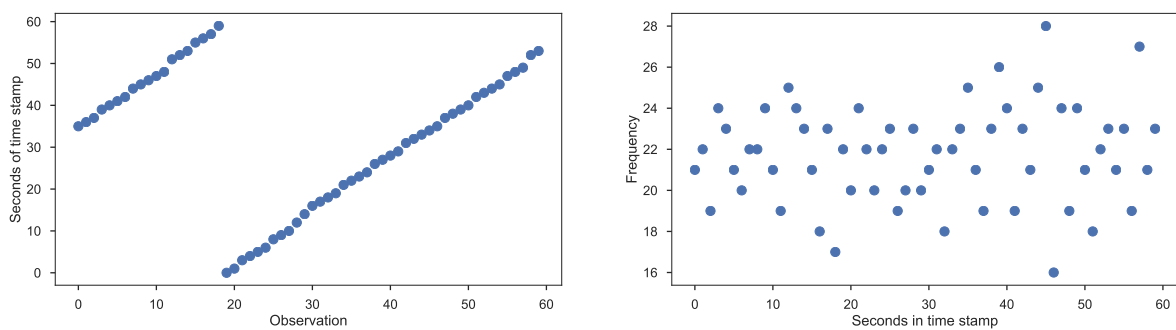
**Figure 2.7:** Investigation of the temperature outliers through frequency plots, suffered by the Geasy SECs.

**Table 2.5:** Parameter descriptions as used in the data set.

Label	Description	Unit
<b>W</b>	Power	kW
<b>T1</b>	Outlet temperature	C
<b>T3</b>	Inlet temperature	C
<b>T4</b>	Ambient temperature	C
<b>Hm</b>	Hot water flow	L



**Figure 2.8:** Raw temperature plot showing erroneous spike of 100 °C.



(a) Time stamp drift experienced for 60 consecutive observations, shown by the recorded seconds value of the time stamp. (b) Time stamp drift seconds frequency over the period of a day.

**Figure 2.9:** Investigation of the time stamp drift suffered by the Geasy SECs.

### 2.6.1.6 Summary

This section provided valuable insight into the current state of the data quality, satisfying requirement DC1 from Table 2.3.

## 2.6.2 Sampling Period Regularisation

The data was observed to have a time stamp drift, as seen in section 2.6.1.2. Data cleaning routines and methods used during further analysis of the data are typically sensitive to non-regular sample periods. For this data set, sampling period regularisation is achieved by the method as described in section 2.3.1. As a result of the time grouping, the seconds values are essentially truncated and recorded as 0 for all observations. This provides a regular sampling period of 1 minute. Due to the SEC sampling period drift, the resultant data structure with regular sampling period generates some observations with no recorded values.

This satisfies the sampling period regularisation requirement DC2.1 in Table 2.3.

## 2.6.3 Outlier Mitigation

Outlier mitigation is achieved through first identifying the outlier, then removing the erroneous value. This transforms the outlier mitigation problem into a data imputation problem. Fig. 2.8 shows an example of an outlier. The feature vector under consideration for outlier mitigation is assessed statistically to determine the most appropriate parameters regarding outlier detection. In the case of temperature features, the full feature vector is considered during statistical analysis. On the other hand, for the water and power usage features, only the non-default values (the non-zero) values are considered for the statistical analysis. Through experimentation it was found that outlier detection was highly successful using an offset rule identification approach. The rule considers any points found outside of 4 standard deviations from the mean as outliers.

This satisfies the outlier mitigation requirement DC3.1 in Table 2.3.

## 2.6.4 Missing Value Imputation

Systems containing aspects of data collection often have elements missing. The potential origins of missing data are numerous, especially with more complex infrastructures. Failure of specific nodes may manifest in the recorded data as missing observations with a specific pattern. The nature of these patterns may provide insight into better practices for estimating the missing values.

### 2.6.4.1 Potential Origins of Missing Values

With reference to section 2.2, there are numerous elements of the infrastructure working together to provide the intended service of the SEC. However, any of the nodes in the infrastructure may fail, causing data loss for as long as the node remains out of service. Table 2.6 provides a brief summary of potential failure nodes, the relative likelihood of failure and some possible reasons.

Some of the most common failure scenarios are briefly discussed below.

**Table 2.6:** Potential reasons for failure of various nodes.

Node	Root Cause	Likelihood	Reason
<b>Geasy</b>	Power loss	High	Consumer habits/mistrust of system/utility provider
	Airtime depletion	Moderate	Excessive data transmission/connection reestablishment
	Firmware bug	Low	Counter overflow
	Physical failure	Low	Numerous
<b>Cellular Network</b>	Power loss	Low	Numerous
	Insufficient bandwidth	Low	Available bandwidth not enough for new levels of traffic
	Poor connectivity	Moderate	Great distance to tower/excessive traffic on tower
<b>MQTT</b>	Broker failure	Low	Numerous
<b>DTL</b>	Script failure	Low	Numerous
<b>Database</b>	Server connectivity	Low	Numerous
	Meteor crash	Low	Numerous
	Server crash	Low	Numerous
	Run-length encoding	High	Design flaw
<b>Installation</b>	Failure to record	Moderate	Unaware or forgot

**Geasy** The Geasy SEC is potentially the most volatile node in the Geasy network. This node is at the end of the branch and interfaces with the EWH. The SEC can be affected by power loss due to multiple reasons, including users manually switching their EWHs off and DSM from the utility provider such as ripple control or load shedding. Furthermore, the GSM connection may fail if the airtime is depleted as communication is then inhibited by the cellular provider. On the SEC, the firmware may have some bug that prevents it from operating within design parameters. And as with any hardware system, a physical fault may occur which could necessitate a unit replacement.

**Cellular Network** The backbone of the communication infrastructure, the cellular network, is mostly reliable, but may fail at times. These failures may include total blackouts or poor connectivity. Furthermore, if the network is under stress, more data loss may be experienced.

**Host Server** A large majority of the infrastructure is dependent on software running on servers. This includes the MQTT broker, DTL, DB and the RLE. If the server goes down, all these nodes go down. Though unlikely, server crashes do happen.

**Run Length Encoding** RLE is employed in the Geasy infrastructure, as mentioned in section 2.2.2.3. Typically used when the data is known to be complete, the RLE in this case is applied prematurely to the incoming data stream. The volatility of the data stream may cause the SEC to report in correctly for a sampling period, followed by an outage of communication of an unknown amount of time. The RLE algorithm has no data

to compare for this period, resulting in no change being recorded in the database until communication is restored. As a result, the RLE may indicate that the SEC recorded values that remained constant for a significant amount of minutes, if not hours. Due to this, RLE would be better used on a cleaned data set in preparation for data warehousing.

**Installation** During the installation and registration of Geasy units, there are certain attributes which need to be recorded to enable accurate further analysis. These attributes range from user details, such as geographic location, to physical attributes of the EWH, such as volume, element rating and orientation. Units may be installed by the user or by a contracted person who may not record these parameters. This leaves the device attributes lacking during analysis.

#### 2.6.4.2 Persistence

For continuous data values which only transition between a maximum and a minimum value, such as a constant power rated device as with the case of an element in an EWH, an acceptable and straightforward method of filling exists. The approach to fill the missing values consists of persisting the last valid value. This approach enables gaps in the data to be filled with existing valid values which, given the nature of the data, are most likely the correct values. This will ensure that a few possible duplicates that were encoded as well as values that were not sent due to loss in connectivity will be assumed to have remained constant.

#### 2.6.4.3 Interpolation

For more variable data values, such as the temperature and water usage, a more involved approach is required. It would be inaccurate to assume that the previous values simply persist. To cater to the variable nature of the data, interpolation is used to fill the missing values. Due to the 1 minute resolution of the Geasy system, a linear interpolation should suffice for the temperature and water usage values.

#### 2.6.4.4 Imputation Design Decision

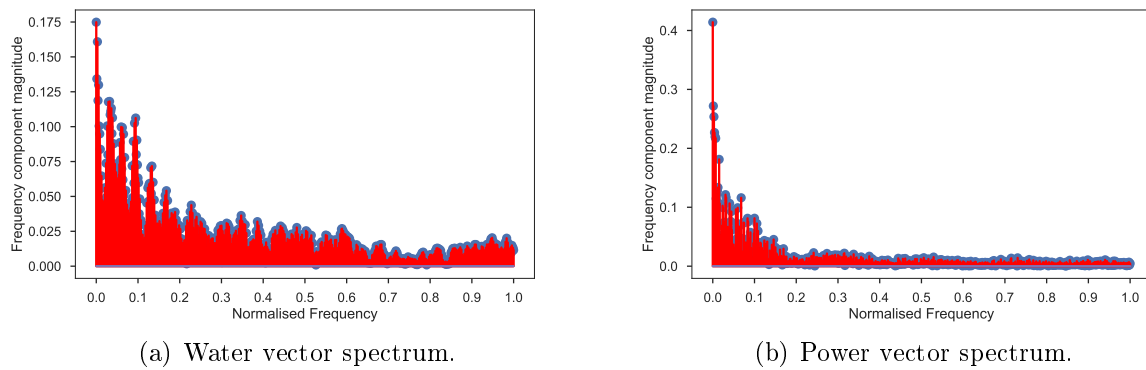
Due to the nature of expected usage events lasting a few minutes, a design decision was made regarding the imputation of values. The temperature values are expected to be continuous and, as such, no imputation limit is applied to the interpolation of the temperature features. The two remaining features, flow rate and power usage, both should only have finite periods in which the values are not the default 0. As a result, the decision was made to impute no more than 5 missing observations, after which the default value of 0 is assumed. This means that any missing data for the flow rate and power usage features lasting for more than 5 minutes will be assumed to be 0 until the next valid value is obtained.

This satisfies the complete data sets requirement DC4.1 in Table 2.3.

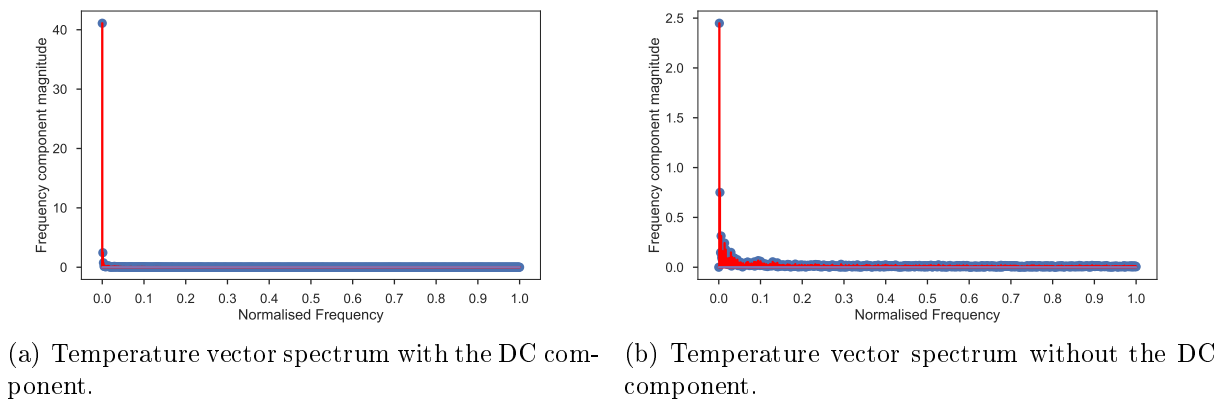
#### 2.6.5 Filter Design

Of the 5 feature vectors, only the temperature vectors were selected for filtering. The water vector has data which is highly stochastic and a lot of information is retained within the higher frequency bands, as seen in Fig. 2.10(a). The power vector has data





**Figure 2.10:** Energy spectra of vectors that do not qualify for filtering.



**Figure 2.11:** Energy spectra of outlet temperature vector with and without DC component.

which is prone to jump discontinuities upon power state transitions (on and off), which also manifests as high frequency energy as seen in Fig. 2.10(b). Filtering of a signal with jump discontinuities produces Gibbs phenomenon in the reconstructed signal, which is an undesirable side effect.

**Temperature Vectors:** Moving on to the filter design for the temperature, first the frequency spectra are investigated as seen in Fig. 2.11. The outlet temperature was found to have a rather high mean value, with the sample as seen in Fig. 2.11(a) having a mean value of around 40. Due to the high mean value, the energy spectrum is also investigated with the DC component removed as seen in Fig. 2.11(b). The intuition gained from this shows that the majority of the information is located in the lower frequency band.

**Essential Bandwidth:** This lower frequency band energy is confirmed by calculating the essential energy of the signal. For this sample, the energy is 1696.15. In order to retain 99.6 % of the signal energy, only 0.14 percent of the frequency spectrum is required, starting from the DC frequency. Considering the signal with the large DC component removed, the aim is to maintain at least 95 % of the information. In this case, the energy is 7.12 with only 1.39 % of the frequency spectrum being required, starting from the lowest remaining frequency. The conclusion drawn from this analysis is that the intuition about the majority of the information being contained in the lower frequency band is accurate.

**Table 2.7:** Filter design parameters.

Parameter	Value
Filter	Butterworth digital low-pass
Zero-phase	True
Order (N)	8
Cut-off (Wn)	0.2
Gain	1

**Filter Parameters:** Given the analysis of the temperature vectors, the design of the filter may proceed. The aim of the filter is to mitigate the quantization error generated by the lack of sensor resolution. For the design a Butterworth digital low-pass filter was selected. The filter is to be applied as a zero-phase filter in order to maintain the phase of the temperature vectors relative to the unfiltered vectors. Through experimentation filter parameters were determined which produced satisfactory results. The parameters are summarised in Table 2.7.

This satisfies the filtering requirement DC5.1 in Table 2.3.

### 2.6.6 Summary

The data cleaning framework applies various routines to the feature vectors in order to improve the data quality. Table 2.9 provides a summary of each feature vector and the routines that are used. The routines are represented by the characters A to E, as shown in Table 2.8.

## 2.7 Results

In this section, the results of the data cleaning framework are investigated. The investigation is done on a feature vector basis as each feature was cleaned in unique ways, as indicated in Table 2.9.

**Table 2.8:** Data cleaning routines summary.

Routine	Label	Section
Sampling Period Regularisation	A	2.6.2
Outlier Mitigation	B	2.6.3
Imputation - Persistence	C	2.6.4.2
Imputation - Interpolation	D	2.6.4.3
Filter	E	2.6.5

**Table 2.9:** Data cleaning summary of routines applied to each feature vector.

Feature Vector	A	B	C	D	E
Power	✓	✓	✓		
Water	✓			✓	
Temperature outlet	✓	✓		✓	✓
Temperature inlet	✓	✓		✓	✓
Temperature ambient	✓	✓		✓	✓

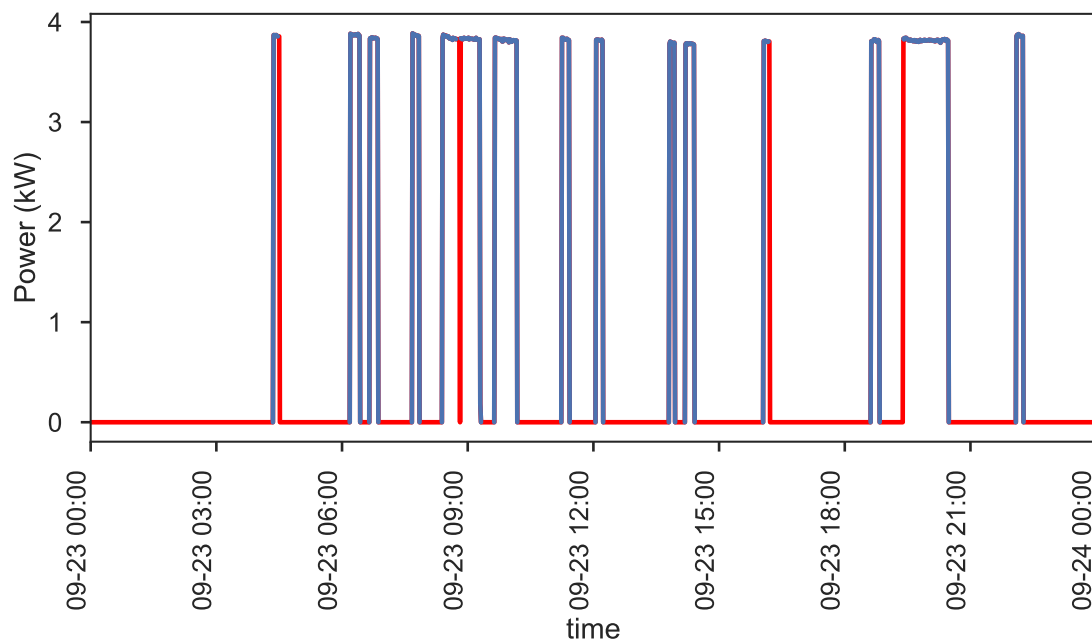
### 2.7.1 Power Data

Fig 2.12 shows a real world power data cleaning example for the period of a day. In this figure, blue represents the original data as read from the database, with the cleaned data shown in red. The cleaned data is displayed first, with the original data overlaid to clearly indicate in red where the original data needed improvement. With this result, it is evident that a lot of 0 values were missing. Additionally, at around 04h30 and 16h30, the power draw periods were brought to an end, something missing from the original data, which could lead to significant incorrect additional power usage data through a purely RLE decoding algorithm. The converse is also observed, where no clear starting point for a demand period was recorded, as seen at 19h30. At 09h00, a period exceeding 5 minutes was observed to have no data recorded. This stretch of missing data resulted in the algorithm producing the default value of 0, until the next valid data point was encountered again.

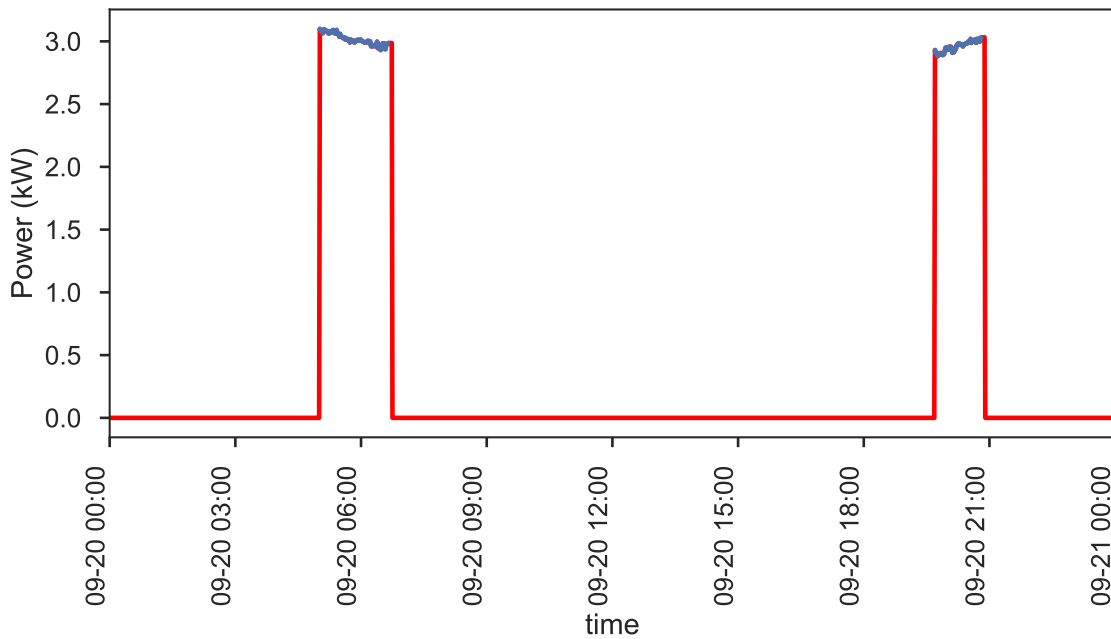
Considering Fig 2.13, another real world power data case is observed. In this figure, two periods of continuous power demand were recorded, but no starting or stopping points were recorded. Conventional RLE would indicate that there was power demand for the entire day, but experience with typical EWH power demand as well as more thoroughly recorded days from the same EWH indicate that this is an anomalous day. The red lines provide definite starting, stopping and default values for all the missing values.

### 2.7.2 Water Data

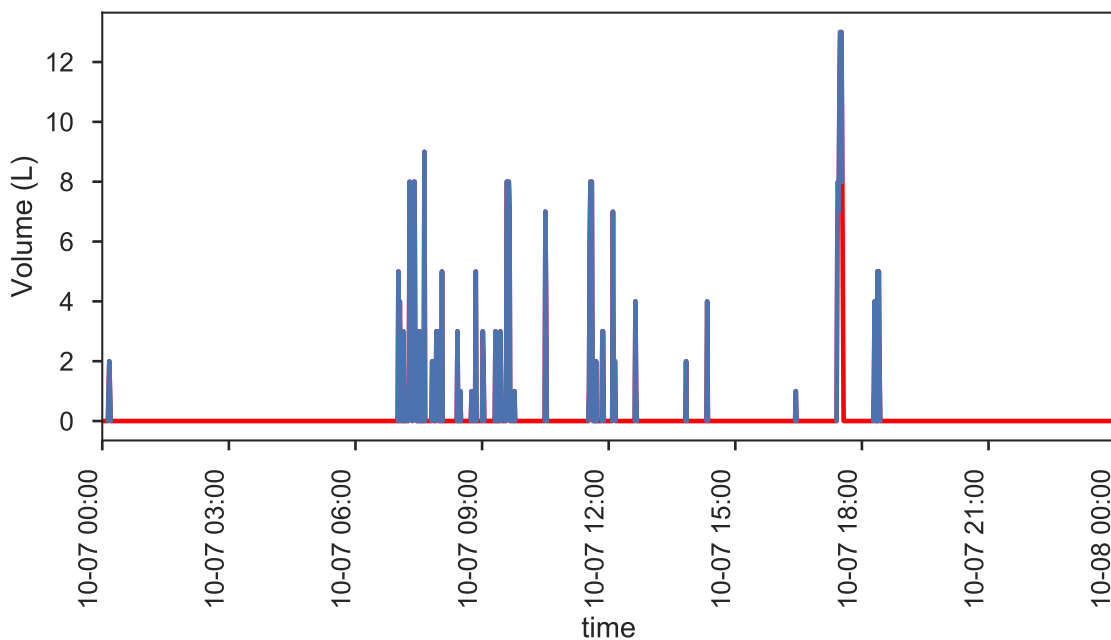
The water usage data for a typical day is shown in Fig 2.14. From this figure one of the largest concerns are observed; water usage that was not explicitly ended just before 18h00. Decoding this data with RLE would incorrectly indicate that the water flow rate



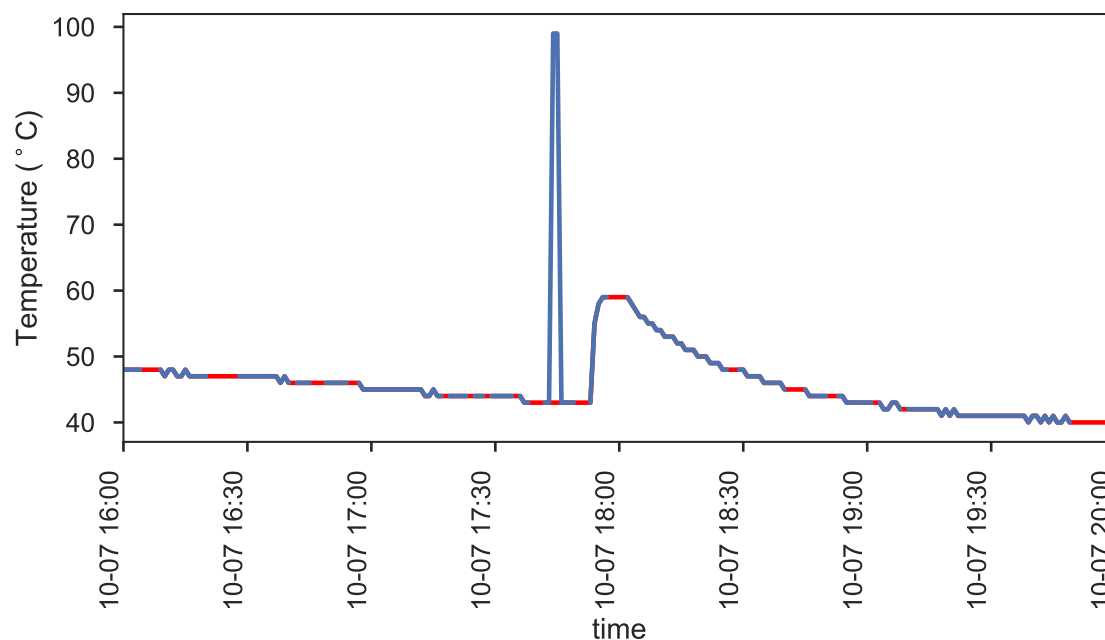
**Figure 2.12:** A typical day of an EWH's power demand cleaned. The original data is indicated in blue, with the filled values in red.



**Figure 2.13:** A less complete day of an EWH's power demand with no starting or stopping times recorded, cleaned. The original data is indicated in blue, with the filled values in red.



**Figure 2.14:** A typical day of an EWH's water usage cleaned. The original data is indicated in blue, with the filled values in red.



**Figure 2.15:** An excerpt of temperature indicating an anomalous spike in recorded value and the cleaned version. The original data is indicated in blue, with the filled values in red.

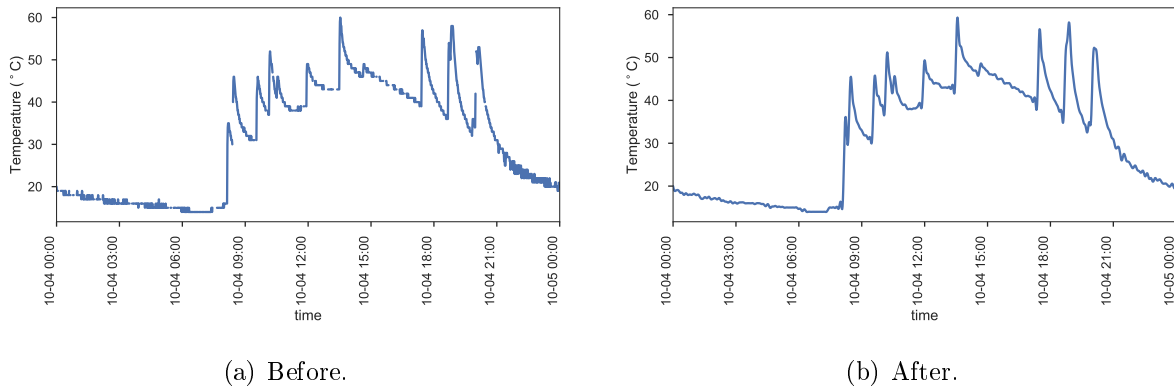
continued at a constant 8 L/min for almost another hour. Cleaning this data resulted in the water usage to strive toward 0, as it seemed to be ending before data loss occurred. Furthermore, the erratic nature of the recorded water usage is maintained instead of being filtered out. Finally, the default value of 0 is filled in between the ending and starting periods of water usage.

### 2.7.3 Temperature Data

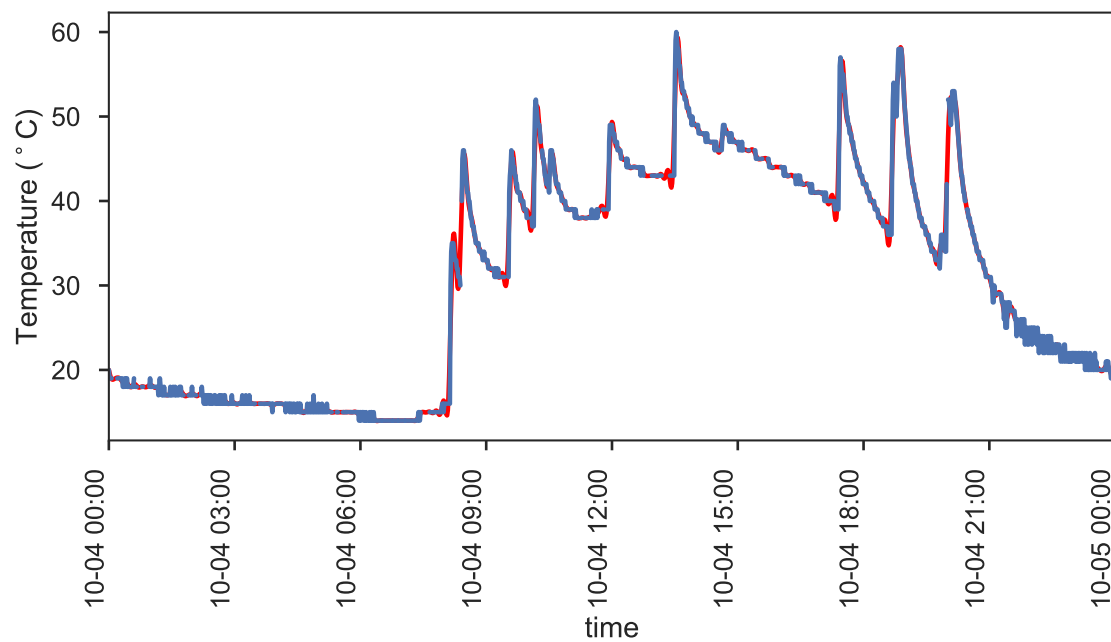
Besides missing data, anomalous recordings that were observed to be constrained to the temperature data sets are data spikes. One such spike is shown in Fig. 2.15. Here a spike takes the temperature from around 45 °C to 100 °C for a few observations. Cleaning the temperature data successfully removed this spike and filled in the missing data as shown in red in Fig. 2.15.

Besides data spikes, the temperature data was also filtered to produce a smoother signal, removing the noise created by the lower accuracy temperature sensors. Fig. 2.16(a) shows the original data and Fig. 2.16(b) shows a more complete and smoother cleaned signal. This smoother signal could potentially track the actual temperature much better due to higher resolution, but at the resolution of most of the sensors, this detail does not carry as much weight.

Fig. 2.17 shows the the signal obtained from the outlet of an EWH with the original data in blue, compared to the cleaned data in red. From this it is evident that even in this highly complete data set, some data was still missing and required imputation. The cleaned signal managed to track the original data very well with the only concern being the overshoot experienced at sharp discontinuous edges in the data, as observed around 18h30.



**Figure 2.16:** Before and after of a typical day's EWH outlet temperature. Noteworthy is how much smoother the cleaned signal is.



**Figure 2.17:** A typical day of an EWH's outlet temperature cleaned. The original data is indicated in blue, with the filled values in red.

# Chapter 3

## Data Analysis and Forecasting

The inherent power of data lies in the analysis thereof. Data analysis is the process of unlocking the potential knowledge contained within data typically originating from the measurement of some real-world phenomenon. Two primary goals are associated with data analysis: prediction and information [51]. Prediction aims to provide a forecast of what an output of a process will be given future input variables, whereas information aims to extract knowledge about how the underlying process works.

In the data analysis arena, there are two approaches that are typically used, namely data modelling and algorithmic modelling [51]. With data modelling a stochastic data model is assumed about the process which produced the output from the the given input data. This approach is typically founded in established statistical analysis and iterative evaluation of how well models fit the data. Algorithmic modelling attempts to find a function which operates on the input to produce the output. This leads to a black box approach where the underlying mechanics of the algorithm is typically abstracted from the user.

For the purposes of this study, a data modelling approach will be used on the hot water usage data.

### 3.1 Time Series Analysis

Time series analysis is specifically concerned with the analysis of data observed at various points in time. This type of data creates unique statistical modelling and inference problems [52]. Time series data is ubiquitous in modern science fields, including economics with stock market analysis, medical with blood pressure as a result of new drug as well as physical and environmental sciences, including engineering.

Time series data is not analysed by conventional statistical methods due to the correlation introduced by the time-adjacent points [52]. Many conventional statistical methods assume independent and identically distributed observations. Time series analysis consists of a systematic approach which aims to answer both mathematical and statistical questions typically based on time correlations. The primary objective of time series analysis is to develop mathematical models that provide plausible descriptions for sample data by identifying, modelling and extrapolating patterns found in historical data [53].

There are two main approaches used for time series analysis, a time domain and a frequency domain approach [52]. The time domain approach presumes a correlation between adjacent points in time where the current value is best explained by a dependence on past values. This approach is typically used to develop forecasting tools. The frequency

domain approach presumes the characteristics of interest are best described by periodic sinusoidal variations found naturally in a lot of data. Both approaches may produce similar results for long series; however, comparatively, the time domain performs better with shorter samples.

## 3.2 Methods Used

This section briefly covers the various methods employed during the data analysis and modelling process.

### 3.2.1 Time Series Decomposition

In order to analyse common attributes of a time series, the input data is often decomposed into various components, which together will form the original signal. These components are typically:

- **T**: the trend component
- **S**: the seasonality component
- **R**: the residual component

These components can be determined from either the additive or multiplicative models. The additive model as shown in (3.1), sums the components of the time series together to produce the original signal. This model is typically used where the seasonality and residual components do not change as the level of the trend rises or falls.

$$y(t) = T(t) + S(t) + R(t) \quad (3.1)$$

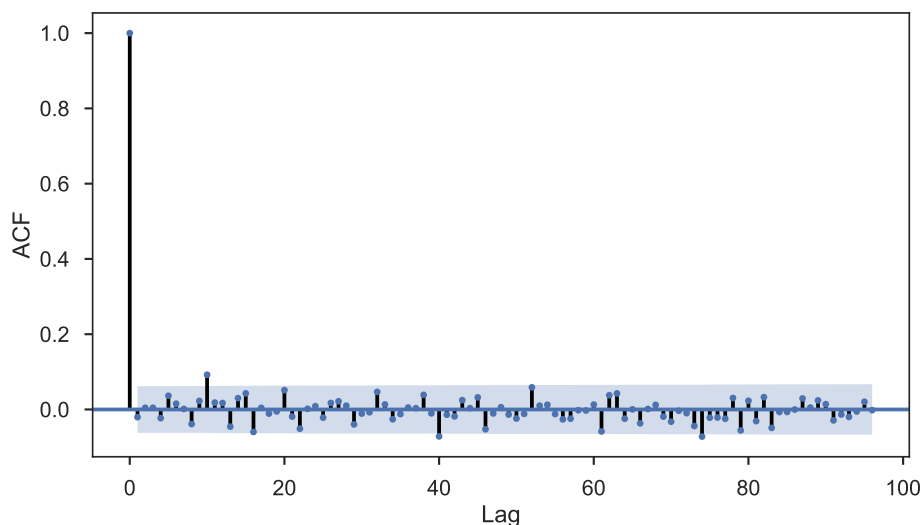
On the other hand, the multiplicative model as seen in (3.2), multiplies the components together to produce the original signal. This model is typically used when the seasonality and residual components appear to be proportional to the level of the time series.

$$y(t) = T(t) * S(t) * R(t) \quad (3.2)$$

### 3.2.2 Measure of Dependence: Correlation

In measuring the relationship between two data sets, the correlation is often determined to quantify the interdependence of the variables [46]. The correlation is obtained from the application of numerous statistical measures taken from the data sets and comparing the results. One of the first requirements is the mean as shown in (3.3), where  $X$  is the data set under examination;  $x_i$  are individual points in the data set and  $P_i$  is the probability of the points in the data set and  $N$  is the total number of points. The next requirement is the variance as calculated from (3.4), where  $\sigma_X^2$  represents the variance. From the variance, the standard deviation is obtained by taking the square root of the variance as shown in (3.5). Finally, a measure of correlation is achieved between two data sets through the covariance, depicted in (3.6). The covariance indicates a positive value if the variability of the two data sets are similar and a negative value if the opposite is the case. The magnitude of the covariance does not relay much information as it is not normalised. To compensate for this, the correlation function, shown in (3.7), normalises the covariance





**Figure 3.1:** Autocorrelation of a random data set.

with the product of the standard deviation of both data sets. This results in a value in the range of -1 to 1, which indicates perfect negative correlation to perfect positive correlation, and 0 which indicates no correlation at all. Correlation plots typically provide a confidence interval of 95 % which, if exceeded, indicates less than 5 % of a statistical fluke.

$$\bar{X} = \sum_{i=1}^N x_i P_i(x_i) \quad (3.3)$$

$$\sigma_X^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{X})^2 \quad (3.4)$$

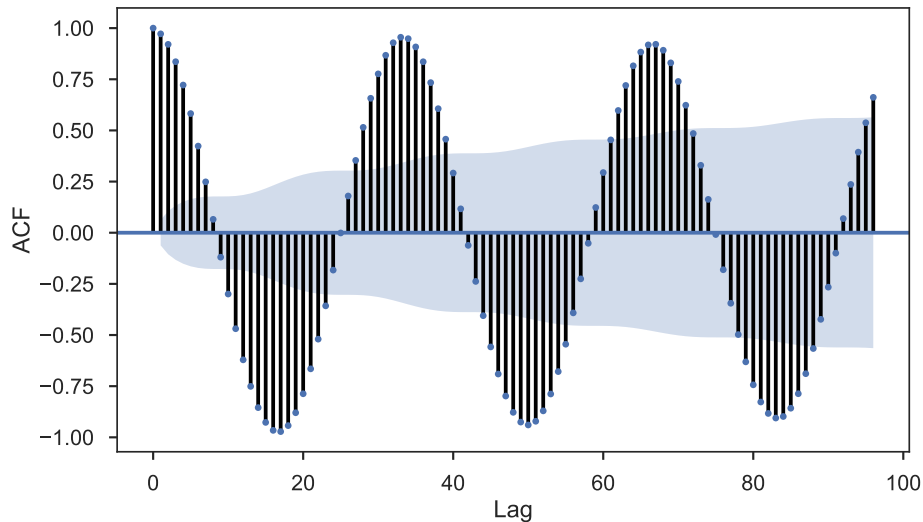
$$\sigma_X = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{X})^2} \quad (3.5)$$

$$\text{cov}(X, Y) = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{X})(y_i - \bar{Y}) \quad (3.6)$$

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (3.7)$$

### 3.2.2.1 Autocorrelation

Autocorrelation of time series signals measures the similarity of the signal with its own time displaced version. Periodic signals will correlate at time delays which are integer multiples of the period. Therefore, a typical use of autocorrelation is to detect non-randomness in data and to identify an appropriate time-series model. The autocorrelation of a signal is obtained from (3.8), where  $k$  indicates the desired lag value. The Autocorrelation Function (ACF) of a random (white noise) signal can be seen in Fig. 3.1. Besides the lag at 0, there is no clear correlation significantly exceeding the confidence interval.



**Figure 3.2:** Autocorrelation of a non-random data set.

Contrasting this with the ACF of a sinusoidal function in Fig. 3.2 which has strong autocorrelation, there is clear autocorrelation at numerous lags, with the significance greatly exceeding the confidence interval cone.

$$r_x = \frac{\sum_{i=1}^{N-k} (X_i - \bar{X})(X_{i+k} - \bar{X})}{\sum_{i=1}^N (X_i - \bar{X})^2} \quad (3.8)$$

### 3.2.2.2 Partial Autocorrelation

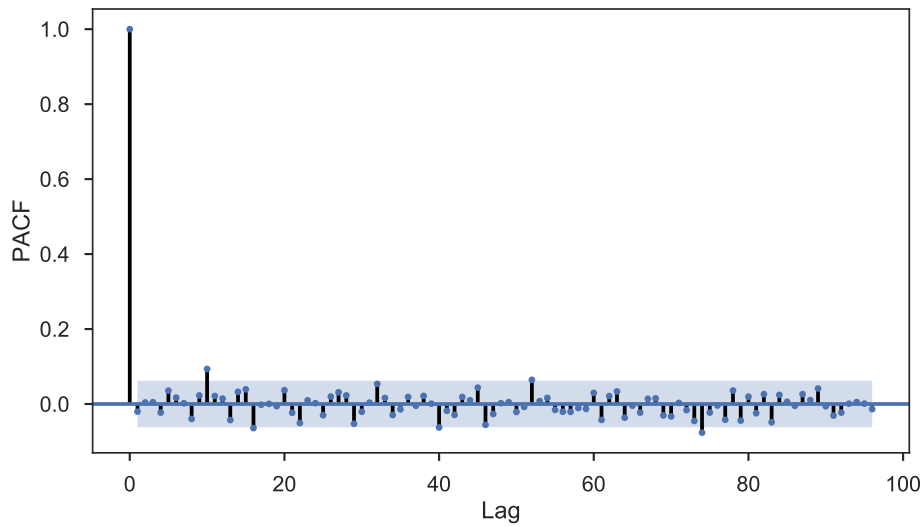
Another measure of correlation is obtained through the partial autocorrelation function (PACF). Similar to the ACF, the PACF produces a correlation of a data set with its own, time lagged values; however, it is adjusted for a common factor that may be affecting both data sets.

## 3.2.3 Stationary Time Series

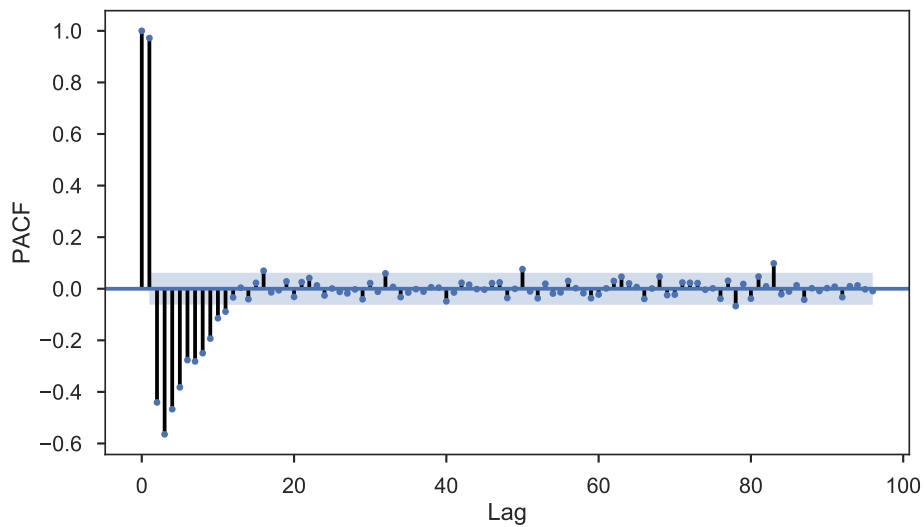
Stationarity is the notion of regularity in a time series. A stationary time series process has a time-invariant mean, variance and autocorrelation [52]. A key requirement of statistical time series modelling with the purpose of forecasting, is that the time series must be stationary. A strictly stationary time series requires the distribution of the variable to be time-invariant and all moments need to be independent of time, which is a very strong assumption in practice. Weakly stationary time series alleviates this strict assumption by only requiring a constant mean and that the autocovariance should only depend on the selected lag value.

### 3.2.3.1 Augmented Dickey-Fuller (ADF) Stationarity Test

The Augmented Dickey-Fuller (ADF) test, tests for stationarity in a time series. The procedure of the ADF test involves an autoregressive model which has lagged values subtracted from it (differenced), as shown in (3.9), to test for the presence of a unit root which means the series is non-stationary, the null hypothesis ( $H_0$ ) where  $\delta = 0$ . The



**Figure 3.3:** Partial autocorrelation of a random data set. (Lag 82 is shorter than that of ACF.)



**Figure 3.4:** Partial autocorrelation of a non-random data set.

alternate hypothesis ( $H_A$ ) is that there is no unit root which means the series is stationary,  $\delta < 0$ . Unlike the Dickey-Fuller test, the ADF reports a p-value and a t-statistic. A p-value of less than 0.05 indicates  $H_0$  may be rejected. Furthermore, a t-statistic below the respective critical values (typically for 10 %, 5 % or 1 %), may reject  $H_0$  with the respective margin of error. The more negative the t-statistic is, the stronger the evidence that the null hypothesis may be rejected.

$$\Delta Y_t = \delta Y_t + u_t \quad (3.9)$$

### 3.2.4 Residual Diagnostics

A residual in forecasting is the difference between the forecasted value and the observed value:  $e_i = y_i - \hat{y}_i$ . With time series forecasting, the forecast residual is based on one-step forecasts; where  $\hat{y}_t$  is the forecast of  $y_t$  based on past observations  $y_1, \dots, y_{t-1}$ .

White noise is defined as a time series generated from an uncorrelated random set of variables,  $w_t$ , with a mean of 0 and a finite variance  $\sigma_w^2$  [52]. The designation of "white" noise originates from an analogy with white light and indicates all possible oscillations are present with equal strength. Typically in applications, white noise is required to be independent and identically distributed (IID) random variables. A frequently used white noise series is Gaussian white noise:  $w_t \sim \text{IID } N(0, \sigma_w^2)$ .

When creating statistical models of time series, one measure of the suitability of a model to a particular series is frequently evaluated through residual diagnostics. A good model will yield residuals which possess the following properties:

- No correlation. If there is correlation between residuals, there is information still contained in the residuals which can be used to improve the model.
- Zero mean. If the residuals do not have a zero mean, the forecasts are biased.

A typical aim is to obtain Gaussian white noise residuals. Therefore, residual diagnostics is a method of ensuring all possible information is being extracted from the series being modelled.

#### 3.2.4.1 Residual Normality

The normality assumption places strict restrictions on the residuals which are typically violated with social science research. Fortunately, the consequences of non-normality range from quite mild to even non-existent. [54].

#### 3.2.4.2 Quantile-Quantile Plot

A normal quantile-quantile plot (Q-Q plot) is a visual diagnostic tool to assess the observed values against a theoretical distribution [54]. The residuals of a model are plotted against a theoretical normal distribution. The aim is to have all the points align on a straight line, which would indicate normality of the residuals. However, the degree to which the straight line is adhered to is to a certain point arbitrary.

### 3.2.4.3 Density Estimation

A kernel density estimation of the model residuals plotted against a theoretical normal distribution. Using this method, problems related to skew and kurtosis may be effectively diagnosed.

### 3.2.4.4 Correlogram

The correlogram for a time series is the ACF and is used to assess the correlation found in the residuals of a model. Using the established principles from section 3.4.2.1, any remaining information in the residual may be identified and incorporated into the model.

## 3.2.5 Forecast Models

Forecast models are selected for statistical time series by using a time domain approach. This approach was selected due to the observed and intuitive cyclical nature of hot water usage being largely determined by work schedules, i.e., regular morning and/or evening usage. In general, parsimony is to be exercised when selecting an appropriate model; that is, simple models are preferred to complex ones [53].

### 3.2.5.1 Auto-Regressive Integrated Moving Averages

Auto-Regressive Integrated Moving Averages (ARIMA) modelling is a method of using past values to forecast future values from a time series data set [52]. The ARIMA model is a generalisation of an Auto-Regressive Moving Average (ARMA) model, with an initial differencing step (the ‘integrated’ term) to deal with data that has been shown to be non-stationary. The full ARIMA model definition is shown in (3.11). The prime notation indicates the differencing of the data set. In short the model is written  $ARIMA(p, d, q)$ , with  $p$  representing the order of the autoregressive part;  $d$  representing the degree of first differencing involved; and  $q$  representing the order of the moving average part. For brevity, some notation is introduced to simplify the expression of the model. In (3.10), the  $L$  represents the lag operator, which indicates lag and  $d$  indicates the order of differencing.

$$L^d x_t = x_{t-d} \quad (3.10)$$

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) (1 - L)^d X_t = \left(1 + \sum_{i=1}^q \theta_i L^i\right) \epsilon_t \quad (3.11)$$

The AR terms relate an observation  $x$  at time  $t$  as a linear sum of the previous forecast values of the signal with some error, represented by  $\epsilon_t$  as white noise, as shown in (3.12).

$$X_t = c + \sum_{i=1}^p \phi_1 X_{t-i} + \epsilon_t \quad (3.12)$$

Complimentary to the AR approach of using past forecast values, the MA terms relate an observation  $x$  at time  $t$  as a linear sum of past forecast error values of the signal, as shown in (3.13). The

$$X_t = \epsilon_t + \sum_{i=1}^q \theta_1 \epsilon_{t-i} \quad (3.13)$$

**Table 3.1:** Identification of purely SARMA models from of ACF and PACF plots. Lags are integers,  $k = 1, 2, \dots$  and  $s$  indicates the seasonality of the lags. Adapted from [52].

Plot	$AR(P)_s$	$MA(Q)_s$	$ARMA(Q)_s$
ACF	Tails off at lags of $ks$	Cuts off after lag $Qs$	Tails off at lags of $ks$
PACF	Cuts off after lag $Ps$	Tails off at lags of $ks$	Tails off at lags of $ks$

ARIMA models are typically applied to non-seasonal data which have a trend. However, the ARIMA model has been adapted to deal with seasonal data, leading to the creation of the Seasonal ARIMA (SARIMA) model [52]. In addition to the  $p, d, q$  terms of the ARIMA model, the SARIMA model adds  $P, D, Q$  and  $s$  terms. In short, the full model is written  $SARIMA(p, d, q)(P, D, Q)_s$ . Using lag notation, (3.14) represents a SARIMA model. Specifically a SARIMA (1,1,1)(1,1,1)<sub>2</sub> model is shown. The terms with a lag term,  $L$ , without an exponent represent the original, non-seasonal ARIMA terms while the terms with lag terms raised to an exponent (in this case 2), represent the seasonal ARIMA terms.

$$(1 - \phi_1 L)(1 - \Phi_1 L^2)(1 - L)(1 - L^2)X_t = (1 + \theta_1 L)(1 + \Theta_1 L^2)\epsilon_t \quad (3.14)$$

SARIMA modelling has the advantage of being based on established statistical properties and has an effective modelling process. Software implementations of the model are also available in mainstream statistical software packages. SARIMA models are, however, linear models and can only extract linear relationships from the data. Furthermore, the data needs to be preprocessed to ensure that it is stationary before SARIMA modelling may be attempted.

### 3.2.5.2 Estimating a SARIMA Model (Box-Jenkins Method):

The Box-Jenkins approach for identifying a plausible autoregressive integrated moving average model is done in four distinct steps [52]. The first step is to ensure the time series is stationary, which involves successive differencing in the case where the series is non-stationary. The next step is model identification where the orders of the remaining AR and MA terms are identified using the ACF and PACF plots. This transitions into a model estimation which is where the ACF and PACF plots are investigated for any seasonality or significant components which may lead to multiple models being developed. Finally, during model validation, the performance of the selected models is evaluated typically through residual diagnostics. The general behaviour observed in ACF and PACF plots for purely SARMA models is summarised in Table 3.1.

### 3.2.6 Parameter Grid Search

After manually investigating parameters for a suitable SARIMA model, it is common to attempt to optimise the selected parameters by performing a grid search of combinations of parameters [55].

### 3.2.7 Performance Metrics

Some numerical performance metrics are employed to assess the performance of selected models, which aids in selecting the best model.

### 3.2.7.1 Akaike Information Criterion (AIC)

The Akaike Information Criterion (AIC) is a measure of the relative quality of a statistical model for a given data set based on in-sample performance [53]. This measure is to be minimised during parameter selection to obtain a higher quality model. The absolute value of the AIC result has little meaning; therefore, the AIC result is used to evaluate different models of the same data set, with the lowest AIC value representing the better model. The AIC function is shown in (3.15) with  $k$  representing the number of parameters in the model and  $\hat{L}$  representing the maximum likelihood function. The likelihood function of the model is shown by (3.16), where  $x$  represents the data,  $M$  represents the statistical model and  $\hat{\theta}$  the parameter values that maximise the likelihood function.

$$AIC = 2k - 2\ln(\hat{L}) \quad (3.15)$$

$$\hat{L} = P(x|\hat{\theta}, M) \quad (3.16)$$

### 3.2.7.2 Bayesian Information Criterion (BIC)

The Bayesian Information Criterion (BIC), also known as the Schwarz Information Criterion (SIC), is another measure of the relative quality of a statistical model for a given data set based on in-sample performance [53]. The BIC formula is shown in (3.17) which is similar to the AIC formula, except where the 2 before the first term is replaced by  $\ln(n)$ , where  $n$  represents the number of data points.

$$BIC = \ln(n)k - 2\ln(\hat{L}) \quad (3.17)$$

### 3.2.7.3 Mean Squared Error (MSE)

The mean squared error (MSE) is a measure of the quality of a forecast [53]. A non-zero MSE indicates a bias in the forecast. However, for practical models that have not been overfitted, a non-zero MSE is expected and is used to evaluate the relative of a model's out of sample forecast performance where a lower score indicates less bias. The MSE is defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (3.18)$$

Where  $\hat{Y}_i$  is the prediction at time step  $i$  and  $Y_i$  is the observed value of the data set.

## 3.2.8 Statistical Diagnostic Tests

To evaluate the suitability of the selected model, statistical diagnostic tests are run on the model residuals.

### 3.2.8.1 Ljung-Box

The Ljung-Box test is a general goodness-of-fit test used to test if any of the autocorrelation results of a time series vary from zero [53]. When used with ARIMA models, the test is executed on the model residuals. A p-value less than 0.05 indicates strong evidence that the residuals are white noise. The Q value is calculated with (3.19) with  $n$

as the sample size,  $m$  the number of lags being tested,  $k$  is lag number and  $\hat{r}_k$  is sample autocorrelation at lag  $k$ . The null hypothesis is rejected when (3.20) is satisfied, where  $\chi_{1-\alpha,h}^2$  is the chi-square distribution table value with  $h$  (typically 2) degrees of freedom and significance level  $\alpha$  (typically 5 %).

$$Q = n(n+2) \sum_{k=1}^m \frac{\hat{r}_k^2}{n-k} \quad (3.19)$$

$$Q > \chi_{1-\alpha,h}^2 \quad (3.20)$$

### 3.2.8.2 Heteroskedasticity

Heteroskedasticity refers to the variability of a dependent variable based on a range of values of an independent variable that predicts it. Variance typically increases with linear models and is often used as a test to determine the suitability of a selected model. The aim of a model is to have a residual that is homoscedastic, as a heteroskedastic residual would indicate the model is unable to consistently predict the dependent variable from the independent variable.

### 3.2.8.3 Jarque-Bera

The Jarque-Bera (JB) test is used to assess the normality of residuals [56]. It does this by assessing the skew and kurtosis coefficients of the residuals. The skew is a measure of asymmetry and is expected to have a value of 0. It is defined in (3.21), where  $\mu_3$  is the third central moment and  $\sigma$  is the standard deviation. The kurtosis measures both the peakedness and tail heaviness of a distribution relative to that of a normal distribution. It is expected to have a value of 3 and is defined in (3.22) where  $\mu_4$  is the fourth central moment. Finally, the JB test is shown in (3.23) where  $n$  represents the number of observations,  $S$  is the sample skew and  $K$  is the kurtosis. A JB value of 0 indicates the data is normally distributed.

$$S = \frac{\mu_3}{\sigma^3} \quad (3.21)$$

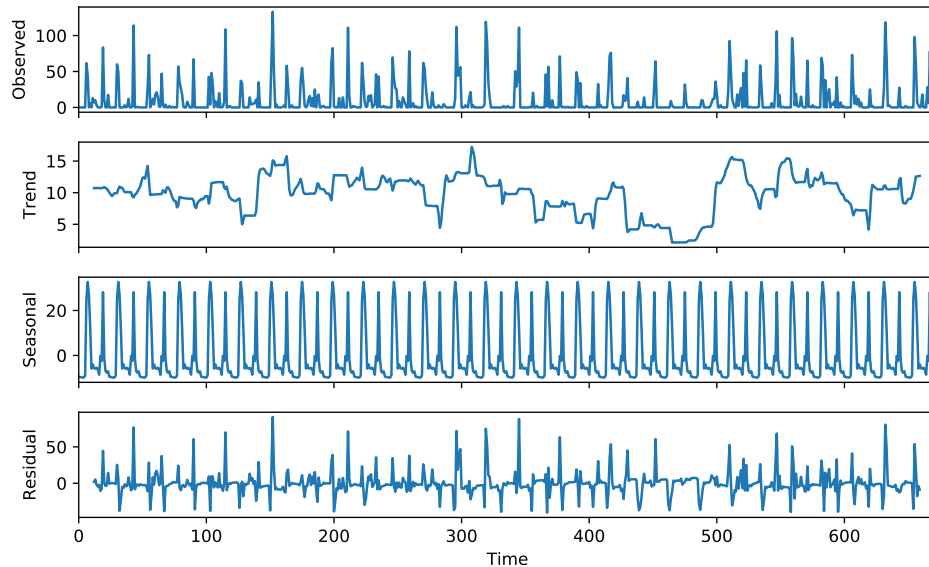
$$K = \frac{\mu_4}{\sigma^4} \quad (3.22)$$

$$JB = \frac{n}{6} \left( S^2 + \frac{(K-3)^2}{4} \right) \quad (3.23)$$

### 3.2.8.4 Visual Inspection

Throughout the modelling process, visual inspection is of fundamental importance for identification of patterns and evaluation of modelling performance [53]. Visual inspection starts with the raw series, identifying anomalous data, evolving into time series decomposition to analyse the components of the series. From here, model parameter selection and final forecasting performance are done with visual inspection along with numerical tests to evaluate the fitness of a model's forecasting performance.





**Figure 3.5:** Time series decomposition of a single EWH's water usage over the period of one month.

### 3.3 Statistical Analysis Tools Used

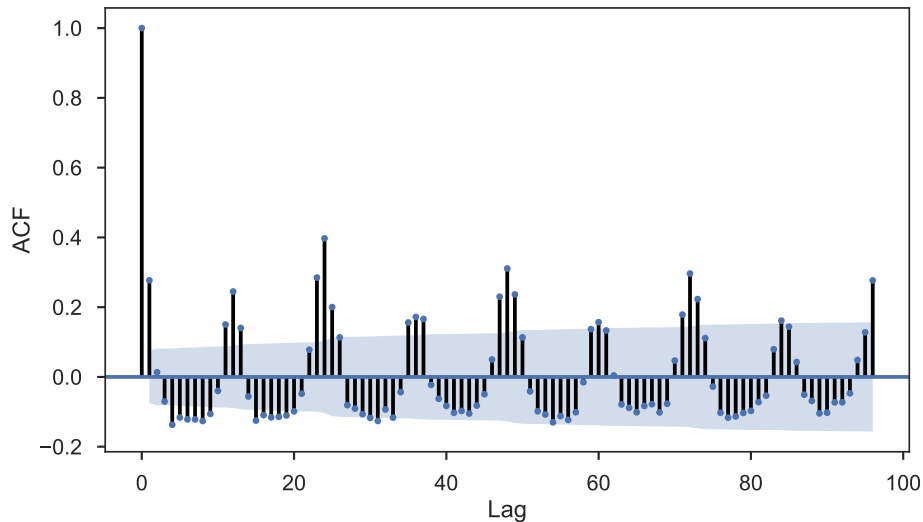
Statsmodels is a Python module that provides classes and functions for the estimation of various different statistical models as well as for conducting statistical tests and statistical data exploration. The module primarily focusses on providing methods for estimating statistical models.

### 3.4 Usage Data Exploration

For the purposes of this study, the focus of data exploration will be on EWH hot water usage data. Due to the time scale of the data, the data has been adjusted to a sampling period of every 60 minutes, down from the original 1 minute resolution. The data was summed in order to represent hourly blocks of water usage. The reason for this period adjustment is two-fold: the 60 minute resolution better captures the periodic structure of the data, which is of primary concern in this study; and the fewer data points enable more rapid model investigation which, due to the averaging nature of reducing the sampling period, tend to better capture a generalised structure in the data as it is more robust against outliers in the data. The data is first explored through time series decomposition, then through correlation, both of which assist with modelling the data.

#### 3.4.1 Water Usage Decomposition

One of the first time series investigations executed on the hot water usage data was the time series decomposition. Considering Fig. 3.5, the decomposition of the data indicates the observed data, trend, seasonal and residual components of the data. Due to the non-monotonic trend of the data, an additive model was selected for this purpose. No transforms were performed based on the data due to the promising results from this decomposition. The observed data shows frequent, regular spikes of varying amplitude that may contain useful time series information.



**Figure 3.6:** Autocorrelation of the water data set.

The trend windows indicate that there is a fluctuating trend in the data, with the data around element 480 showing a dip in the region of 0. This indicates mean is not stationary and corrective action in the form of differencing needs to be applied as part of modelling.

In the seasonal window, there is a clear seasonal component to the data; a large, broader peak followed by a slightly smaller but narrower peak in a repeating fashion, one for each of the 28 days explored. The amplitude of the seasonal component peaks are only around 25 % of the observed peaks which is due to the stochastic nature of the hot water usage. Conventionally, with data sets spanning years with a resolution of months or weeks, the seasonal component corresponded to the literal seasons. However, due to the month long period with hourly resolution, the seasonal component has been identified as repeating every 24 hours, which intuitively corresponds to the 24 hour day whose structure is determined by the work schedule. This indicates promising forecasting possibilities.

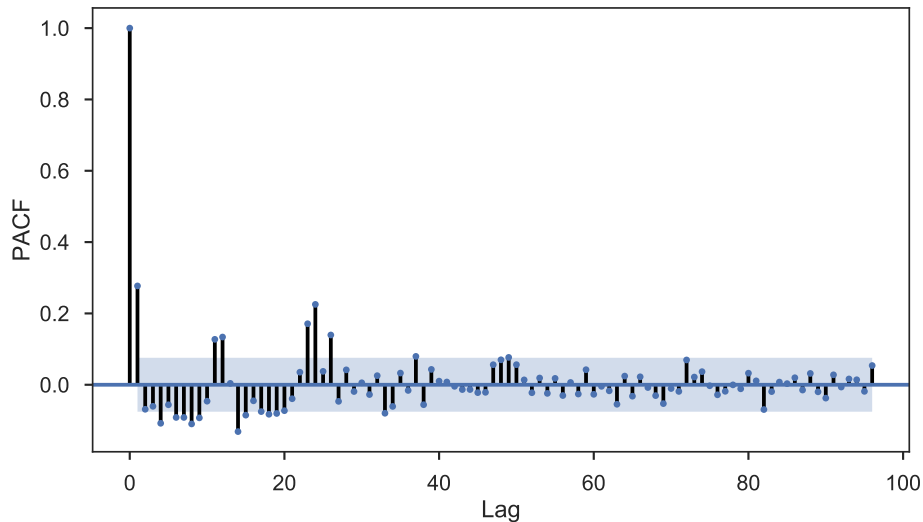
The residual component of the decomposition contains severe peaks, both positive and negative. This indicates that a lot of information remains in this residual component which may be extracted by a more suitable modelling approach, which in turn will improve the forecasting performance.

### 3.4.2 Water Usage Correlation

A more focused investigation into the temporal structure in the hot water usage data is undertaken using the visual correlation tools.

#### 3.4.2.1 Autocorrelation

From the ACF plot in Fig. 3.6 a strong periodic correlation is observed at multiple lags with a steady envelope of decline, each exceeding the 95 % confidence interval. Two evident periodic components are found at lags 12 and 24, representing lags of 12 and 24 hours respectively. The 12 hour lag is less significant than the 24 hour lag, and after 4 days of lag, just barely manages to exceed the confidence interval. As a result, the 24 hour



**Figure 3.7:** Partial autocorrelation of the water data set.

lag is the most interesting component as it indicates significant correlation is observed every 24 hours, or a single day. Intuitively this ties in with the typical work schedule which, at least on week days, have a regular daily schedule but may vary by an hour. As a result, the ACF indicates strong 24 hour lag seasonality.

It should be noted that the first significant lag is found at lag 1, which intuitively makes sense as a 1 hour shift in a schedule of 24 hours represents a change of 4.2 %, which is within the normal time deviation of the observed usage. As a result, this component is not as interesting and is therefore omitted from further analysis.

### 3.4.2.2 Partial Autocorrelation

Additionally, the PACF plot in Fig. 3.7 shows a rapidly declining envelope of correlation. In this plot, there is less obvious temporal structure in the data; however, the four peaks at lags 1, 12, 14 and 24 are of interest. Peaks at lags 1 and 24 are the most significant, however the component at lag 1 will be omitted from further analysis due to the reason stated in section 3.4.2.1. Considering the component at lag 24, again an indication of a 24 hour seasonal component is confirmed.

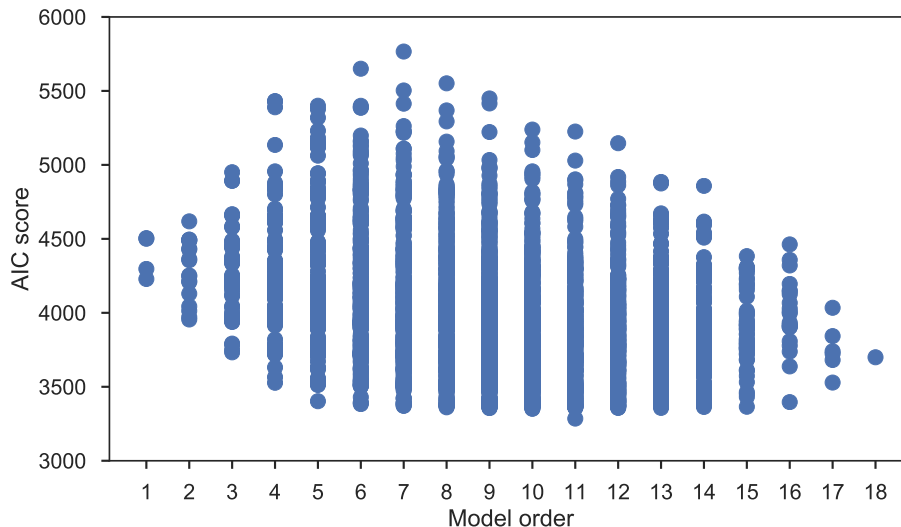
From this, the seasonality is confirmed as 24 hours and will be used during modelling.

## 3.5 SARIMA Modelling

This section presents the SARIMA modelling results used to obtain the linear model used to forecast the data.

### 3.5.1 Selecting Parameters

Parameter selection was done in 2 stages. Firstly, the Box-Jenkins model identification procedure was followed to establish a baseline, followed by a grid search for parameters with the aim of parameter optimisation.



**Figure 3.8:** AIC results vs total number of parameters in the model.

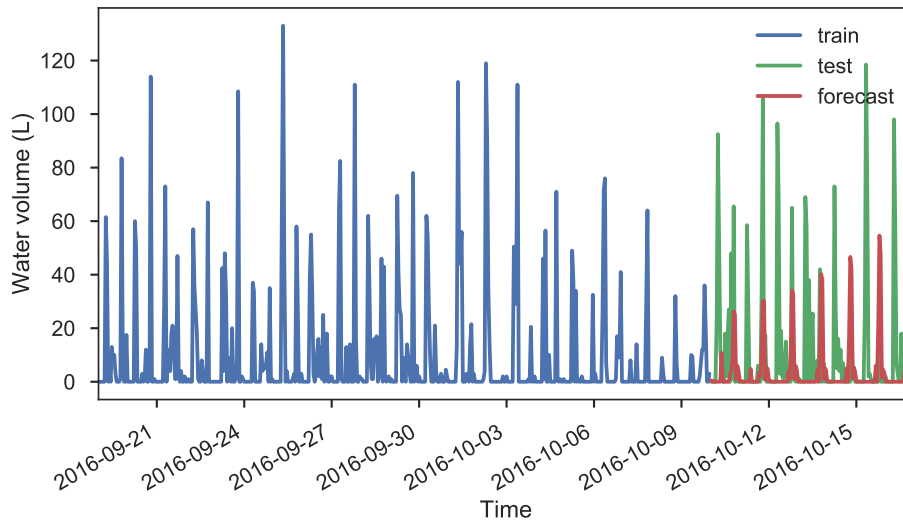
### 3.5.1.1 Box-Jenkins Model Identification

Using the Box-Jenkins model identification approach as discussed in section 3.2.5.2, a model was estimated iteratively to fit the data. Using this approach, a SARIMA model with parameters  $(0, 0, 0) \times (2, 1, 3, 24)$  was selected. This resulted in an AIC score of 3506 with a model order of 6 and MSE of 364.5. The selected SARIMA model consists purely of seasonal terms as during estimation it was found that additional ARIMA terms provided negligible AIC score improvements at the expense of greatly increased model training time. Additionally, models with added ARIMA terms often displayed decreased visual performance, such as increasing or decreasing the trend observed in the forecast peaks.

### 3.5.1.2 Grid Search

In order to optimise the selected model obtained in section 3.5.1.1, a grid search was performed with parameter values in the range of those obtained through model identification. Given that the highest parameter obtained, besides the seasonal component of 24, was 3, the range of parameter values was selected as 0, 1, 2, 3 for all 6 parameters. This gives rise to a significant amount of models to be investigated,  $4^6 - 1 = 4095$ , where parameters  $(0,0,0)(0,0,0)$  are not a valid model as there are no model terms. Grid searching took 46 hours on an i7 4770K at 3.9 GHz with 16 GB of DDR3 RAM at 1866 MHz.

Figure 3.8 shows the AIC scores for the 4095 models trained during the grid search. The model orders range from 1 to 18 and the AIC scores from 3285 to 5766. Selecting the most optimal model based on the AIC score of 3285 results in a model with parameters SARIMA(0, 0, 2) × (3, 3, 3, 24), model order 11 and MSE of 662.3. This is a decrease in AIC score of 281, but an increase in model order of 5. Figure 3.9 shows the forecast obtained from this model. It is immediately evident that the forecast has a linearly increasing peak trend, an undesirable attribute. Rectifying this increasing peak by adding AR terms produces a similar forecast as obtained in section 3.5.1.1 with slightly lower AIC scores but greatly increased model orders. Compared to model SARIMA  $(0, 0, 0) \times (2, 1, 3, 24)$  with an AIC score of 3506 and order 6, the improved grid search model SARIMA(1, 0,



**Figure 3.9:** Time series forecast of a single EWH's water usage over the period of one month using only lowest AIC score as selector.

$2) \times (3, 3, 3, 24)$  has an AIC score of 3391 with model order 12 and model  $SARIMA(3, 0, 2) \times (3, 3, 3, 24)$  has an AIC score of 3412 with model order 14. The increased model orders and significantly increased model training times do not justify the small amount of improved AIC score. As stated in section 3.2.5, lower model orders are generally preferred.

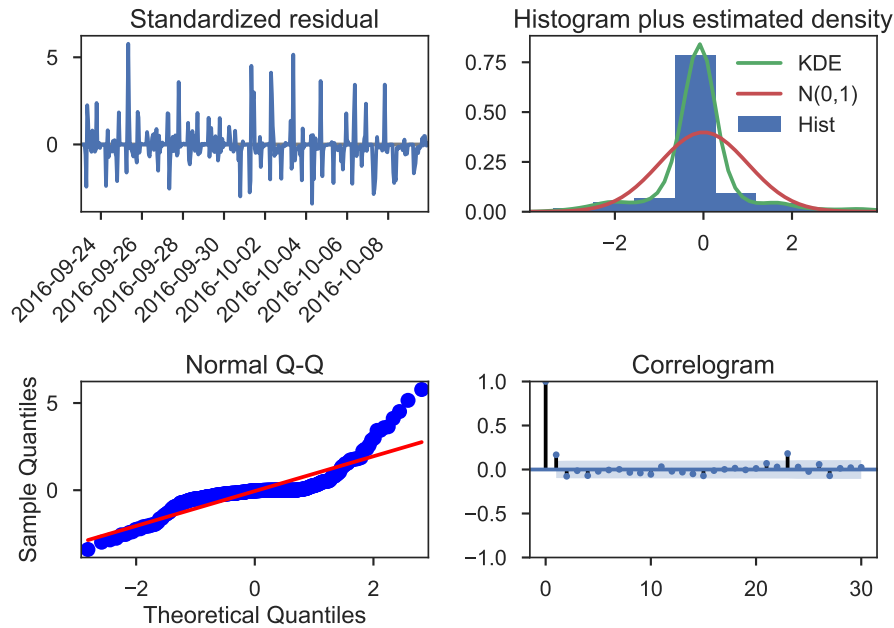
Due to the lower complexity, relatively good AIC score and significantly lower MSE obtained through the Box-Jenkins method, further analysis will be done with the model  $SARIMA(0, 0, 0) \times (2, 1, 3, 24)$ .

### 3.5.2 Model Diagnostics

Figure 3.10 shows the residual diagnostics obtained from the model identified in section 3.5.1.1. Considering the residual diagnostics in Fig. 3.10, it is evident that just like the original data, the residual data is not normally distributed, which violates the original assumption of normality of residual diagnostics. The kernel density estimate of the residual indicates much less spread, with the majority of values indicated around 0. Furthermore, the normal Q-Q plot shows the correlation of a normal distribution and the residual have trouble aligning along the red line, with the upper tail proving rather heavy (deviating from the red line). Fortunately, the effects of these normality-based tests have been confirmed to have mild to non-existent consequences. The correlogram, on the other hand, indicates significant reduction in correlating terms in the data as much of the time series information containing structure has been successfully extracted by the selected model. Furthermore, the residual plot has significantly improved as compared to the decomposition residual in Fig. 3.5, with peaks of around 6 L vs peaks of 100 L as with the decomposition.

## 3.6 Results

This section presents the results of modelling the hot water usage data and forecasting from the trained model.



**Figure 3.10:** Time series model diagnostics of a single EWH's water usage over the period of one month.

**Table 3.2:** Summary of selected SARIMA model attributes.

Attribute	Value
SARIMA parameters	$(0, 0, 0) \times (2, 1, 3, 24)$
AIC	3506
MSE	348.4
Scaling coefficient	1.1338
Training data	75 %
Testing data	25 %
Forecast period	7 days
Model resolution	1 hour

### 3.6.1 SARIMA Forecast

Figure 3.11 shows the forecasting result obtained by using the model as specified in Table 3.2. In the figure, the blue line represents the training data, the green line represents the testing data and the red represents the forecast data. The forecast data is observed to correctly capture the seasonal component of the data, as observed in Fig. 3.5. This seasonal component does not, however, fully capture the amplitude of the observed test data, with the test data showing peaks in excess of 3 times the forecast value as observed in Fig. 3.11.

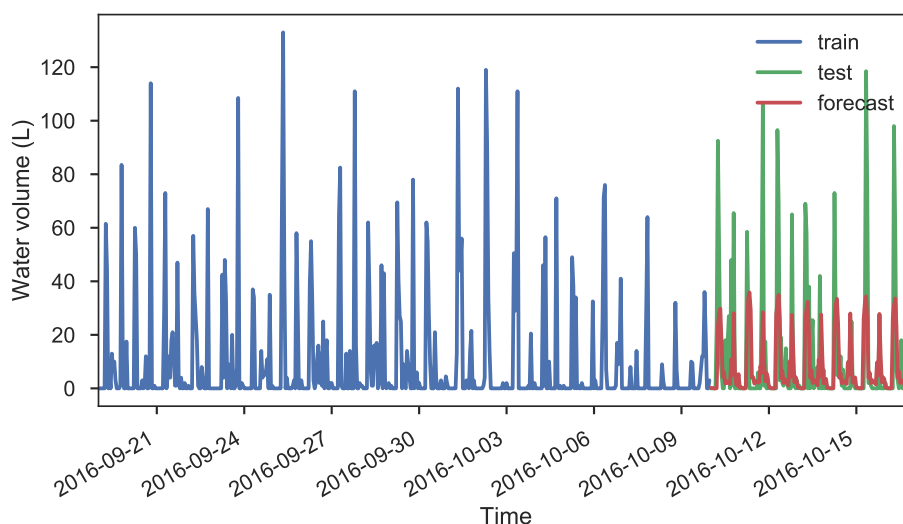
### 3.6.2 Volume-Corrected SARIMA Forecast

A possible method of correcting this scaling issue, which would still preserve volumetric accuracy, is to adjust the total volume consumed, normalised to a day, of the forecast to the observed value in the training data. The training data indicates an average daily volume of 272 L in contrast to the 225 L observed from the forecast data. This results

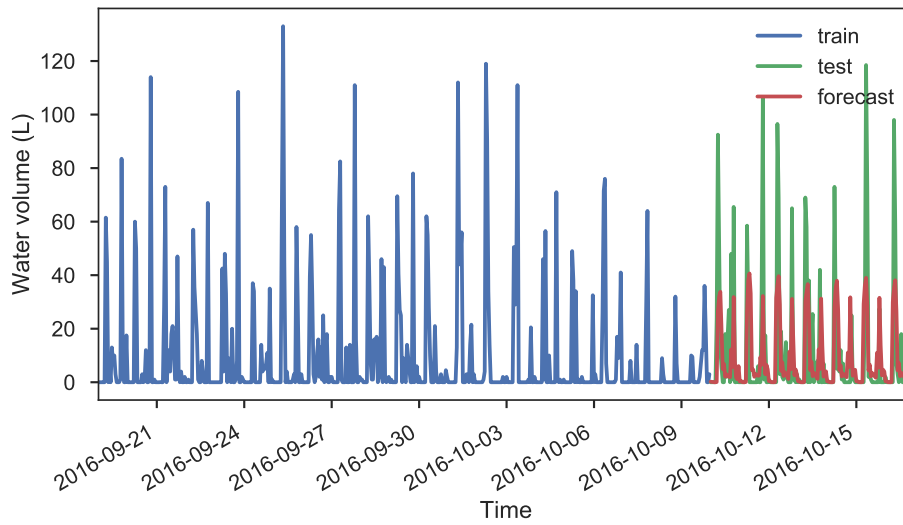
in a training to forecast ratio of 1.1338. Scaling the forecast data with this value, to ensure the average volume of hot water used per day is equivalent to the training data, results in a forecast as shown in Fig. 3.12. The scaled forecast data now indicates a 13.4 % increase in peaks, not nearly enough to meet the peaks indicated by the test data. However, considering the average volume used per day, the scaled forecast standing at 272 L and the test data, coincidentally, the same as the training data at 272 L, shows a strong correlation. Additionally, the MSE improved from 364.5 to 348.4.

Fig. 3.13 shows the test data in green, with the forecast data in red for the full week of forecasting performed. From this it is evident that the test data has clear regular components to it, though some of these regularities are malformed as observed just after the first significant peak. Nonetheless, the forecast data effectively captured the seasonal nature of the hot water usage, clearly indicating that this data set has a predisposition for using the bulk of hot water in the mornings on a flexible schedule of about 2-3 hours. In the evenings, besides the anomalous cases, the usage schedule of the generally less usage is more regular with about 1 hour of deviation, if any. As previously mentioned, the amplitude scaling of the forecast data is unable to indicate the true peaks observed in the test data; however, the average volume consumed per day is accurate at 272 L.

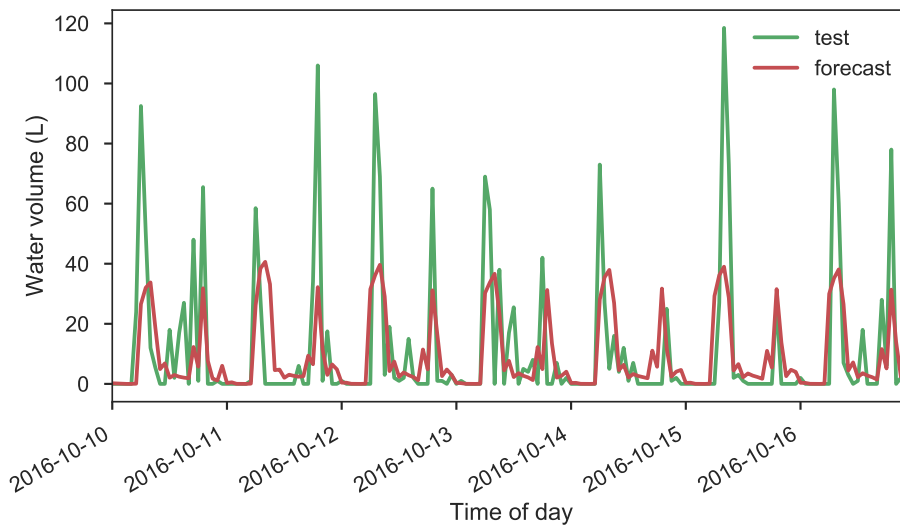
**Verdict:** From these results, the linear SARIMA model was able to effectively capture the seasonal component of the hot water usage data and with scaling was able to accurately capture the average volume used per day. These results indicate a promising application for the SARIMA linear model as a forecasting tool. The linear statistical analysis of the data provided valuable insight that may drive future data modelling attempts by providing a solid baseline to work from. The non-linear irregularities in the data would, however, be better modelled by a non-linear model. Furthermore, given the stochastic nature of the generating process, the manual creation of the linear SARIMA model is not feasible as it does not scale well. Using available modern computing power and machine learning, more accurate, higher resolution forecasts that scale much better could be obtained without the labour intensive model estimation as required with statistical analysis.



**Figure 3.11:** Time series forecast of a single EWH's water usage over the period of one month using 75:25 data split.



**Figure 3.12:** Time series forecast of a single EWH's water usage with scaling coefficient of 1.1338, over the period of one month using 75:25 data split.



**Figure 3.13:** Time series forecast of a single EWH's water usage with scaling coefficient of 1.1338, versus the measured data.



## Chapter 4

# Data Application: Demand-Side Management

This chapter presents the design and analysis of the algorithms that will be used to develop a data-driven DSM application as stated by Objective 3. Firstly, existing EWH models are revised to enhance performance. Next, the main drivers of EWH time management is investigated, followed by an investigation of the relevant weights of each of the drivers. From these parameters a cost function is derived to determine individual EWH heating priority in a MG of EWHs. Finally, the developed cost function is used as part of a heuristic function to optimise power demand utilisation to remain within a hard power limit while maintaining consumer comfort. The algorithm is evaluated based on a few selected performance metrics to assess the feasibility of this algorithm from multiple perspectives. The resultant application is an overseer for real-time adaptation of consumer energy levels in EWHs, or Oracle for short.

The Oracle is evaluated by utilising data from the Geasy project as presented in chapter 2 and the simulator Rev3 as developed in section 4.5.

### 4.1 Smart Grid Demand Side Management

DSM aims to to change the shape of the demand load curve through peak shaving, valley filling and profile shifting through activities that intentionally modify consumer usage patterns [11]. One of the popular management techniques is achieved through load scheduling. This section gives a brief overview of popular scheduling techniques employed.

#### 4.1.1 Schedule Optimisation Techniques

Recently, there has been an increased interest in load scheduling techniques in literature with the aim of energy management. These techniques are available in a multitude of forms. Some provide consumer-based scheduling, others are aimed at MGs [13]. Throughout all the scheduling approaches, there are two main techniques that receive attention and are briefly discussed below.

##### 4.1.1.1 Linear Programming

Linear programming (LP) is a pure mathematical approach to scheduling. Typically this involves multiple parameters that model the physical parameters of the load, including constraints placed on the optimisation problem, which are taken into account when

optimising the schedule. LP guarantees an optimal solution, bounded by the provided constraints, provided the solution is obtainable. LP techniques are typically cumbersome since they require all input data at time of optimisation and require the data to be deterministic. This hampers the ability of LP algorithms to adapt to dynamic situations where rescheduling has to occur frequently. Furthermore, in [57], it was shown that peak demand reduction using LP techniques is an NP-hard problem which has a time order complexity of at least  $O(n^k)$ . NP-hard problems also tend to have a solution space that is typically orders of magnitude out of reach of even high-end modern computers.

#### 4.1.1.2 Heuristics

Heuristics is a dynamic approach used to ‘find’ a solution through minimal modelling, only using a few key parameters. This method is referred to as heuristic optimisation and provides a solution that is bounded by all the constraints, while significantly reducing both the space and time order complexity of the problem to be optimised. Schedules are obtained by completing each of the three phases (transition, evaluation and determination) for each time step of simulation. By cycling over these three phases, the algorithm converges on a schedule that is better than rule-based scheduling. This form of scheduling is typically more suitable in dynamic situations where schedules may be changed frequently based on unforeseeable requirement adjustments. Due to the more linear path finding approach of heuristics, the solutions are obtainable on less powerful hardware in much less time than required by LP. Heuristics algorithms have been used in applications where the complexity of finding an optimal solution has been too expensive and a suboptimal solution is acceptable [58].

### 4.1.2 Grid-Centric Scheduling

The rise of microprocessor technology and the various hardware and software technology improvements have led to great interest in large-scale parallel and distributed systems. These systems are aimed to have numerous processes executed on them both concurrently and sequentially. One of the largest difficulties with these kinds of systems is the development of effective distribution, or scheduling, the processes amongst the numerous processing elements to achieve various performance goals. These goals include minimising execution time, minimising communication delay and either minimising or maximising resource utilisation. As a result, scheduling has been seen as a resource management problem [59].

It is for this reason that SG DSM scheduling techniques have evolved, to optimise the resource utilisation based on the available supply and demand [13].

#### 4.1.2.1 Static Priority

With static priority-based scheduling, the assignment of priorities to various tasks are done before execution of scheduling. Higher priority tasks are less likely to experience a halting of operation compared to lower priority tasks. The major advantage of static-priority scheduling is that all the overhead of the scheduling process is incurred before program execution, resulting in more efficient scheduling execution time [59]. A disadvantage of static-priority scheduling is that the scheduling program is unable to adapt to changes in the system and needs to be manually reconfigured to account for these changes.

### 4.1.2.2 Dynamic Priority

With dynamic-priority scheduling, the realistic assumption is made that very little *a priori* knowledge is available about the resource requirements of the tasks to be scheduled. Priorities are assigned to tasks as the scheduling is executed based on some predefined algorithm. The advantage of this approach is the adaptability of the scheduling to account for changing circumstances [59]. A typical example of this is in real-time systems where the upcoming tasks need to be scheduled as they are required.

### 4.1.3 User-Centric Scheduling

In contrast to grid-centric scheduling, user-centric scheduling typically aims to ensure the consumer has priority during scheduling such that the discomfort of the consumer is mitigated. This type of scheduling tries to address the possible decline in tolerance from the consumer's perspective which may result from the little regard their comfort receives from typical DLC [60].

#### 4.1.3.1 EWH as Energy Storage Medium

A combination of DLC and set point control for EWHs was investigated with the aim of reducing consumer comfort in [60]. The proposed control was proved by using a single EWH for which the objective of minimising consumer discomfort was achieved by regulating EWH set point temperature to a higher value, effectively storing thermal energy in the EWH in preparation for a DLC event. This proved that significant reduction in consumer thermal discomfort is possible through intelligent scheduling at the expense of a reasonable increase in electricity demand. The simulation relied on estimated flow rates and volume values of typical water usage events instead of actual measured flow rates and volume values.

#### 4.1.3.2 Minimising Average Deviation of Household Temperature During DR Event

A heating, ventilation and air conditioning (HVAC) control strategy which takes the consumer's comfort into consideration along with weather forecasts was proposed in [12]. The aggregation strategy was based on mixed integer LP (MILP), which aimed to minimise the average deviation of the household temperature values during a DR event while satisfying a load reduction target. This scheduling was done for one day in advance at a resolution of 5 minutes. The study successfully simulated an improved consumer comfort value while adhering to a load reduction target.

## 4.2 EWH Modelling Overview

This section provides an overview of EWH modelling techniques which allows the most suitable EWH model to be selected for the DSM application.

### 4.2.1 Thermal Principles of Nodal Models

This section provides an overview of the thermal principles of nodal EWH models in preparation for EWH model vectorisation. The work contained in this section is largely

based on the work done by Cloete [24] and Nel [20]. This forms the basis of the employed software model of the EWH used in this chapter.

**Water enthalpy:** Water enthalpy refers to the internal energy contained in a body of water and the work done on it as a product of the pressure and the volume as seen in:

$$H = E + PV \quad (4.1)$$

Where  $H$  indicates enthalpy,  $E$  indicates internal energy contained in the water,  $P$  indicates the pressure and  $V$  indicates the volume.

Assuming a uniform temperature distribution and pressure on the body of water, the change of enthalpy of the water contained in an EWH equals the heat gained or lost and can be calculated as:

$$\Delta H = q = \Delta U + P\Delta V \quad (4.2)$$

With:

$$U_{water} = c\rho V_{EWH}T_{water} \quad (4.3)$$

With  $U_{water}$  representing the internal energy contained within the water inside the EWH in joule (J);  $c$  is the specific heat capacity of water, 4184 J/(kg °C);  $\rho$  is the density of water, 1000 kg/m<sup>3</sup>;  $V_{EWH}$  is the volume of the EWH cylinder in m<sup>3</sup> and  $T_{water}$  is the temperature of the water contained within the EWH in °C.

The available energy to be transferred is measured from a reference point to which the environment will strive. This reference point is set to the inlet water temperature as seen in (4.4).

$$E_{water} = \Delta U = c\rho V_{EWH}(T_{hot} - T_{inlet}) \quad (4.4)$$

Where  $E_{water}$  represents the total thermal energy contained within the water in joule (J);  $c$  is the specific heat capacity of water, 4184 J/(kg °C);  $\rho$  is the density of water, 1000 kg/m<sup>3</sup>;  $V_{EWH}$  is the volume of the EWH cylinder in m<sup>3</sup>;  $T_{hot}$  represents the temperature of the water within the EWH in °C and  $T_{inlet}$  represents the temperature of the inlet water in °C, which will replace the extracted hot water. In this case  $T_{inlet}$  is used as the reference point for zero thermal energy contained within the water.

**Recovered thermal energy:** The energy recovered from the EWH during hot water usage is in the form of thermal energy, which is a fraction of the total thermal energy contained within the water and is calculated as:

$$E_{use} = c\rho V_{rate}t(T_{hot} - T_{inlet}) \quad (4.5)$$

Where  $E_{use}$  represents the total recovered energy in joule (J);  $c$  is the specific heat capacity of water, 4184 J/(kg °C);  $\rho$  is the density of water, 1000 kg/m<sup>3</sup>;  $V_{use}$  is the volume of water extracted from the EWH cylinder in m<sup>3</sup>;  $T_{hot}$  represents the temperature of the water within the EWH in °C and  $T_{inlet}$  represents the temperature of the inlet water in °C, which will replace the extracted hot water. And  $t$  is the duration of the extraction period in seconds (s).

**Table 4.1:** Parameters of sample EWH used for cost function design.

Parameter	Symbol	Unit
Specific heat capacity of water	$c$	J/(kg °C)
Density of water	$\rho$	kg/m <sup>3</sup>
EWH volume	$V_{EWH}$	L
EWH element power	$P_{EWH}$	kW
EWH outlet temperature	$T_{hot}$	°C
EWH inlet temperature	$T_{inlet}$	°C
EWH ambient temperature	$T_{amb}$	°C
EWH thermal resistance	$R$	°C/W
Cooling constant	$k$	1/μs

**Input energy:** The thermal energy contained within the water of the EWH is raised by the element which heats the water. This transfer of energy is calculated as:

$$E_{input} = P_{EWH}t \quad (4.6)$$

Where  $E_{input}$  represents the added energy in joule (J);  $P_{EWH}$  is the power rating of the element in watt (W) and  $t$  is the duration of the input period in seconds (s).

**Thermal radiation:** Thermal radiation is the flow of thermal energy from a high potential (a hot mass), to a lower potential (a cool mass) until an equilibrium is reached. In the context of an EWH, thermal radiation occurs between the hot water inside the EWH and the environment surrounding the EWH, through the thermal insulation. This is an undesirable loss of energy. The rate of thermal radiation is proportional to the temperature difference between the two masses and is expressed by Newton's exponential law of cooling. This is modelled by a first order differential equation:

$$\frac{\delta}{\delta t}T_{EWH}(t) = -k(T_{EWH}(t) - T_{amb}(t)) \quad (4.7)$$

Where  $T_{hot}$  represents the temperature of the water within the EWH in °C;  $T_{amb}$  represents the ambient temperature of the environment in °C and  $k$  is the cooling constant.

Substituting (4.3) into (4.7) and solving for the energy lost, the following formulation is obtained:

$$E_{loss} = c\rho V_{EWH}(T_{hot} - T_{amb})(1 - e^{-kt}) \quad (4.8)$$

With:

$$k = \frac{1}{c\rho V_{EWH}R} \quad (4.9)$$

Where  $E_{loss}$  represents the energy lost to the environment through thermal radiation in joule (J) and  $R$  is the thermal resistance of the EWH in °C/W. A summary of all the parameters and their units is given in Table 4.1.

## 4.2.2 EWH Modelling

Virtual assessment of EWH response is commonly achieved through simulation. Simulation of an EWH requires the EWH to be modelled based on the desired research to be

done. This section gives an overview of the various modelling techniques used to achieve EWH modelling.

#### 4.2.2.1 EWH Modelling Techniques

EWHs can be broadly classified into two main categories: an aggregate model for numerous EWHs and an individual model for a single EWH [61]. The aggregate model is typically used in applications where the objective is to improve the overall electrical network efficiency and stability. This macro context usage comes at the expense of individual consumer satisfaction. On the other hand, the individual model takes individual consumer satisfaction into account with the drawback of increased computational complexity.

#### 4.2.2.2 Individual EWH Modelling Techniques

The modelling of individual EWHs is done using three main approaches [62]:

**Data-driven** modelling can be seen as a black box modelling approach. Experiments are performed through which data is collected and relationships are found between the input and output variables using mathematical techniques. This approach requires little knowledge of the system and underlying processes being modelled. Due to the nature of relating output variables to input variables, this approach achieves high accuracy on the training data but may suffer from generalisation beyond the training domain.

**Physics-based** modelling can be seen as a white box modelling approach. Models are derived using the governing laws of physics which require detailed knowledge of the system and the underlying processes. This approach has good generalisation capability but can suffer from poor accuracy.

**Grey box** modelling is a hybrid of the data-driven and physics-based approaches. The basic structure of the model is developed using physics-based methods and the specific parameters of the model are refined by using parameter estimation algorithms on measured data. This approach combines the advantages of the previously mentioned approaches and results in a model that has good generalisation capability with better accuracy than the pure physics-based model.

### 4.2.3 EWH Nodal Models

One example of a physics-based approach to individual EWH modelling was developed by Nel [20]. Using energy flow equations, the approach balances the energy input and energy output of an EWH. The energy input consists of thermal energy added to the water by heating and the energy output consists of extracted hot water and heat lost to the environment. With both the models developed, the inlet water is assumed to be the zero energy baseline for further energy calculations.

#### 4.2.3.1 One-Node Model

Nel [20] developed a model based on an assumed state of the water contained within the EWH. In the case of the one-node model, the body of water is assumed to have a uniform temperature distribution. In this model, water extracted from the EWH is assumed to

have the average temperature of the water contained within the EWH. Furthermore, the inlet water entering the EWH to replace the extracted water is assumed to instantaneously mix with the water contained in the EWH, adjusting the average temperature of the body of water in the EWH.

#### 4.2.3.2 Two-Node Model

An extension of the one-node model discussed in 4.2.3.1 was developed by Nel [20] whereby an EWH would operate in the one-node state until a significant volume of water is extracted from the EWH. Once an extraction occurs which exceeds a set threshold volume over a short duration, the EWH state transitions to a two-node state. This two-node state is developed to mimic the natural stratification that occurs in the EWH where the upper, warmer, less dense node consists of the remaining warm water in the EWH and the lower, colder, more dense node consists of the inlet water replacing the volume of water extracted from the EWH.

The EWH remains in the two-node state until the element sufficiently heats the lower node to the same temperature as the upper node, allowing the nodes to merge, upon which the model transitions back into the one-node state.

## 4.3 Methods Used

This section provides an overview of the related methods used in the development of the DSM application.

### 4.3.1 Vectorisation

Vectorisation is the process of rewriting explicit instruction loops whereby data is processed per element of an array, to have the instruction be executed on all the elements of the array simultaneously as shown in Fig. 4.1. In computer programming this is known as array programming and due to its nature it is a high-level programming model. This has numerous benefits, of which the most common goal is improved execution times (lower is better). Furthermore, code vectorisation typically leads to more readable code, enabling better documentation of code as broad concepts on data manipulation can be expressed more concisely. Conditional evaluation is no longer a branch method, as both the true and false conditions are evaluated based on the individual cases respectively selected by masking, leading to the final result. Modern hardware has specific instruction sets to deal with vectorisation due to the numerous benefits it provides.

#### 4.3.1.1 Single Instruction, Multiple Data

Data processing is done by executing instructions on the data values. The trivial method of achieving this is by executing a single instruction on a single data value, known as Single Instruction stream, Single Data stream (SISD). With the SISD paradigm, each incoming data value is processed individually by having a single instruction executed on it. This is suitable for low data volume applications. However, if the data volume is significant and the incoming data is of an embarrassingly parallel nature, this may no longer be a feasible approach, as the execution time would become cumbersome. To facilitate improved performance, modern computing platforms support Single Instruction stream, Multiple Data stream (SIMD) capability. This is achieved with instruction set

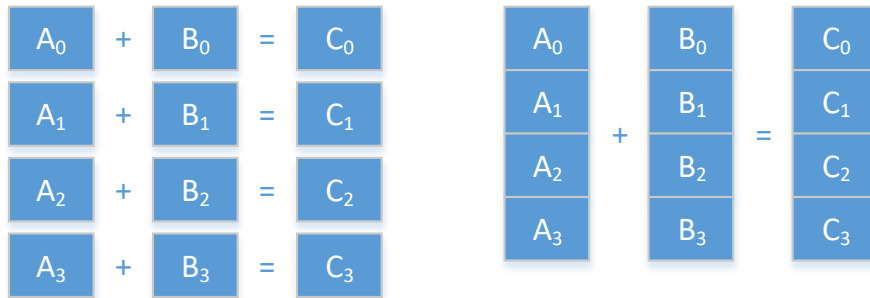


Figure 4.1: Scalar instruction vs vectorised instruction.

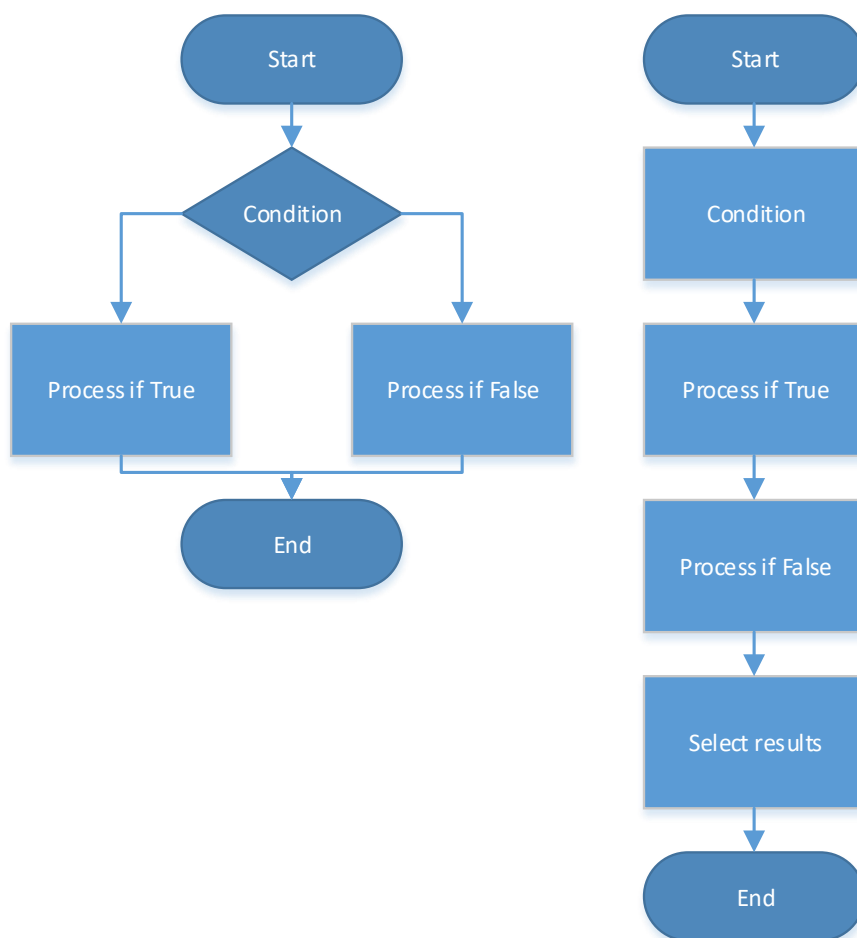
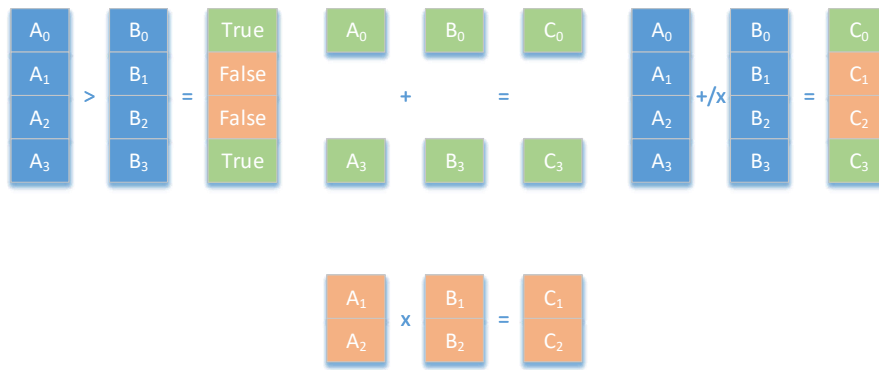
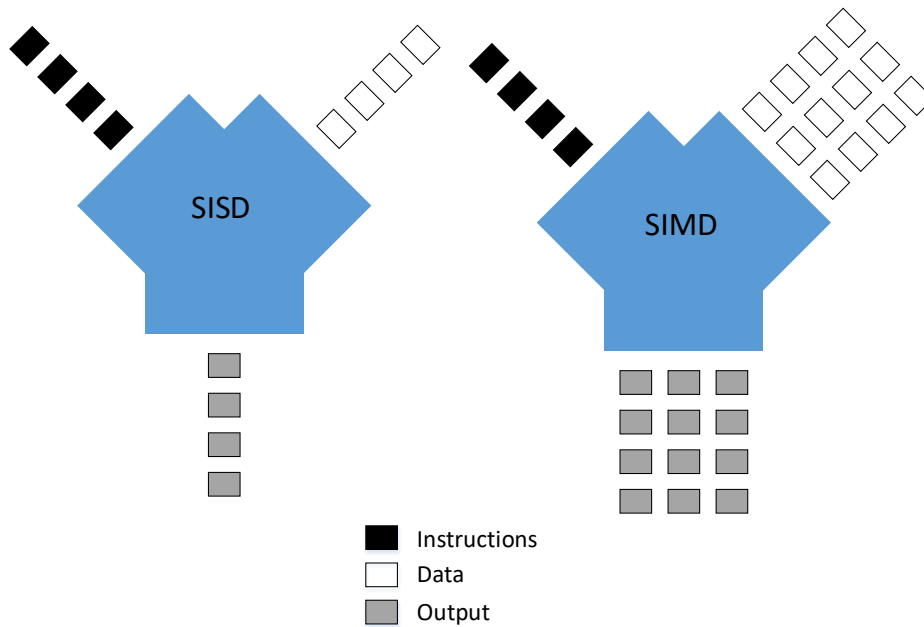


Figure 4.2: Scalar conditional flow vs vectorised conditional flow.





**Figure 4.3:** Conditional evaluation of vector with subsequent instruction execution and result.



**Figure 4.4:** SISD and SIMD stream representations.

extensions, started by the MMX instruction set and later the SSE instruction set. SIMD allows a single instruction to be performed on a large number of data inputs in the same cycle, greatly reducing the execution time in comparison to a SISD approach. A graphical representation of the instruction, data and output streams for both SISD and SIMD is given in Fig. 4.4.

### 4.3.2 Cost Function

A cost function is a mathematical formula used conventionally to predict the cost of a certain action based on varying inputs. This assists with evaluation of decisions based on changing variables. Typically the cost function is customised according to the specific application and can range from a simple linear formula to a more complex one with a

large number of inputs.

A simple cost function can take the form of a linear equation:

$$f_x = mx + c \quad (4.10)$$

Where  $f(x)$  is the cost function being defined,  $x$  is the input variable under investigation,  $m$  is the variable's cost per unit and  $c$  is the constant cost. A linear cost function could expand to accommodate multiple input variables with unique  $m$  values.

A cost function provides the ability to compare the outcomes of various inputs directly, enabling better evaluation of the options. Setting up a cost function based on the relative importance of each input provides a simple method of aiding decision making. It is this ability of the cost function to be defined as a function of various input parameters and their importance that allows the cost function to be a generic tool that can be effectively applied to numerous situations provided an accurate cost function is defined for the specific situation.

## 4.4 Developed Tools

During development of the Oracle, the vectorised nature of the intended calculations necessitated some tools to be developed to more manipulate the data into vectorised formats and to process the vectorised data effectively. This section describes the tools that were developed to achieve this.

### 4.4.1 Aggregate Data Container

The data wrangling techniques used in chapter 2 are primarily focused on processing the data of a single SEC at a time. In order to prepare the data for the SIMD vectorisation as discussed in section 4.3.1.1, the data from all selected SECs need to be aggregated. Considering the independent nature of the data received from each SEC, a container which groups measured data together by sensor is required to take advantage of the performance improvement.

The data container was developed to use the data interface developed in chapter 2. This allows data of individual SECs to be loaded and preprocessed before aggregating. For each SEC, the data is extracted, preprocessed and then split by sensor into tabular data containers aligned on the time stamp. As a result, this data container has 5 sensor data containers (power, water, outlet temperature, inlet temperature and ambient temperature). Two additional attribute data containers (EWH element rating and volume) are also included. The columns of these tabular structures correspond to a single SEC's observations and attributes.

To reduce the frequency of DB requests, a design decision was made which would load a large, predefined period of data for all selected SECs. Once this data has been loaded, subsequent shorter period analysis from within this longer period is supported through simple time slicing. This greatly improves performance of multi-period data analysis and processing.

### 4.4.2 Event Detection

A key requirement for analysis and forecasting is to detect and identify events. In the context of EWH usage, an event is defined as a volume of water withdrawal over a time

period. Both the volume and time period are unspecified, as an event may constitute anything from a small to a large volume of water consumed over anything from a short to a long period of time. The Geasy infrastructure would record a volume upwards of 0.5 L, as covered in section 2.2.1.3, in a period upwards of 1 minute, as covered in 2.2.1.2. Therefore, if a volume of 0.5 L is exceeded within a few seconds, the volume would be recorded as having occurred in the span of 1 minute.

#### 4.4.2.1 Event Attributes

There are two primary attributes to be determined from the measured water usage; first is the starting time of the event and the second is the total volume of water consumed during the event. These two attributes provide valuable insight into usage habits, a requirement for forecasting. The duration of the event is of less importance for the purpose of forecasting, but is implicitly used during event volume calculation.

In detecting events from data, both a single and a consecutive number of non-zero water measurements constitutes an event. The starting time of the event is defined as the time of the first non-zero water flow rate, with the end time of an event being the time of the next zero flow rate. Using this definition, the two primary attributes can be extracted from the hot water usage data.

The total volume of the event is determined by summing the consecutive water volume consumed during the event time period, as shown in (4.11)

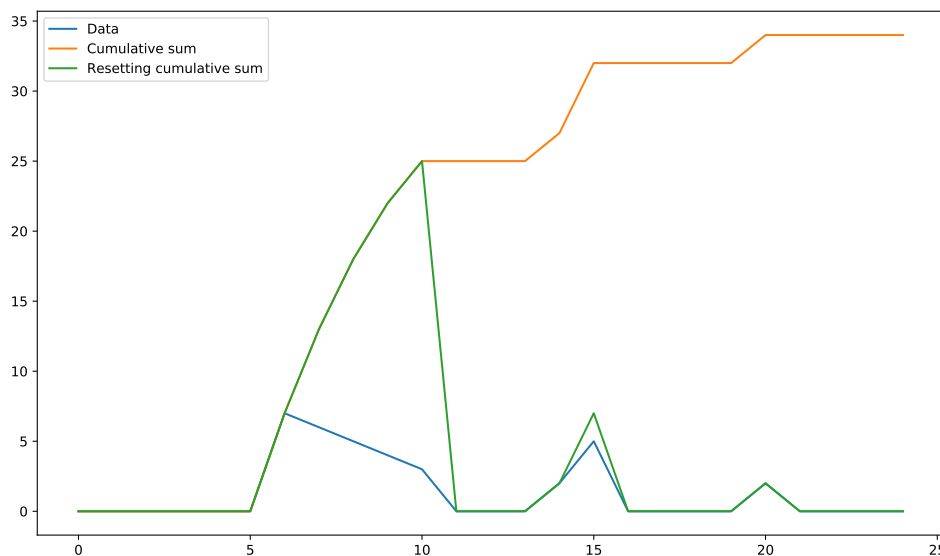
$$V_{event} = \sum_{i=t}^{t+m} v_i \quad (4.11)$$

Where  $V_{event}$  represents the total volume of the event in L,  $v_i$  represents the volume per observation in L,  $t$  represents the starting time of the event and  $m$  represents the duration of the event.

#### 4.4.2.2 Conditional Reset Cumulative Sum

An event detection algorithm was developed which works for a single EWH as well as an MG of EWHs. This algorithm adapts to the variability of both the water usage volume and the duration of the event. This adaptability required the development of a conditionally resetting cumulative sum, a cumulative sum which resets to zero as soon as the input data point is a zero, as shown in Fig. 4.5. Due to the requirement of extracting the starting time of an event, the process is executed on a time-inverted version of the hot water usage data. Using the resetting cumulative sum on the time-inverted data will ensure that both the starting time and total volume of the events may be extracted from a single observation.

The conditionally resetting cumulative sum is achieved in a few steps, described here with reference to water usage events. Firstly, the cumulative sum of the data is obtained, which translates to the total volume used for each EWH at various times. To remove the plateaus (the record of total volume used which persists between events), effectively shifting the periods with no flow rates down to a volume of zero, the preceding plateau's value is subtracted from the volume obtained during the event. The essence of the resetting cumulative sum is to ensure that only volumes recorded during an event are non-zero. The result of this can be seen in Fig. 4.5, where the blue line represents the original data, the orange line represents the cumulative sum of the data and the green line represents the resetting cumulative sum of the data.



**Figure 4.5:** Example of a resetting cumulative sum.

In the process of event detection, firstly the original data is time-inverted, such that the first observation becomes the last and vice versa. The resetting cumulative sum of this time-inverted data is then obtained. Due to the time-inverted data, the apex of the events on the resetting cumulative sum occur at the starting time of the event and indicate the total volume consumed during that event. This produces a wave-like pattern, which can be seen in Fig. 4.5. The next step is to remove all but the apex values from each event, to ensure only the time and volume information of each event is retained. Finally, this time-inverted result is again time-inverted to produce a result which indicates the events by means of the starting time and total volume consumed during that event. The effect of this process is illustrated in Fig. 4.6, where the blue line represents the original water usage data and the green line represents the output from the event detection algorithm. The important points to notice are, firstly, that the starting time of the event coincides with the first non-zero flow rate of the water usage data and that the magnitude of the event represents the total volume of the water consumed during the event.

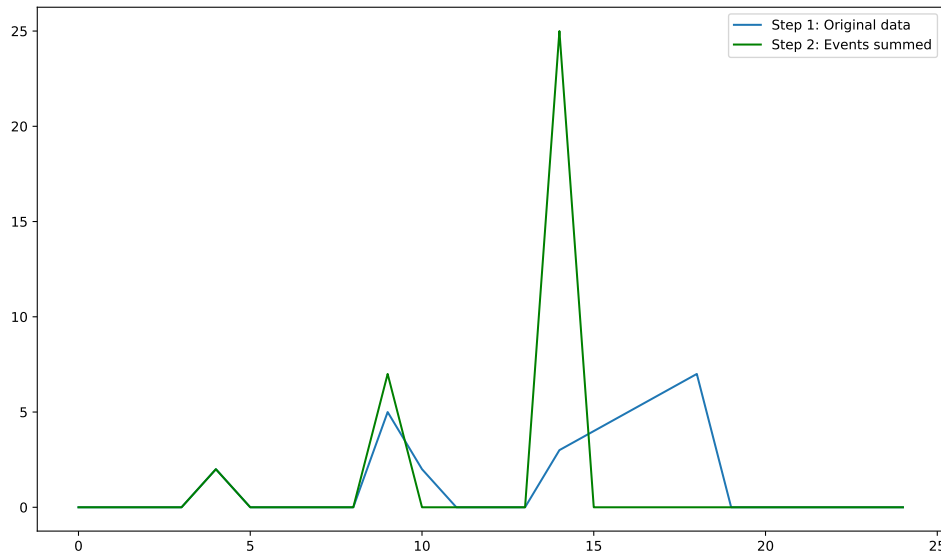
### 4.4.3 Mean Event Temperature Calculation

A measure of the temperature of an event is obtained by using the mean event temperature. The weighted mean was selected to provide the measure of water temperature the consumer will experience during usage. The weighted mean is obtained from:

$$T_{event} = \frac{\sum_{u=1}^U T_u v_u}{\sum_{u=1}^U v_u} \quad (4.12)$$

Where  $T_{event}$  is the weighted mean event temperature in °C;  $T_u$  is the per minute water temperature in °C;  $v_u$  is the per minute water volume in L; and  $U$  is the duration of the event in minutes.

The procedure of efficiently calculating the weighted mean event temperature for a single EWH or a MG of EWHs is done in three steps. The first step is to multiply the outlet temperature data with the water usage data, which provides the weighted product as seen in (4.12). The next step uses the exact same procedure used for event



**Figure 4.6:** Example of detected events from water flow rate data.

detection, as discussed in section 4.4.2. This procedure time-inverts the data, calculates the resetting cumulative sum and time-inverts the result to obtain event magnitudes located at the starting time of the events. In the case of the mean event temperature, the event magnitudes are the sum of the weighted products. The final step is to calculate the weighted mean by dividing the sum of the weighted products by the total volume of the events, as obtained from the event detection algorithm in section 4.4.2.

## 4.5 Simulator Design

The EWH simulator used in conjunction with the EWH model as developed in [20] was designed to achieve the requirements as laid out in Table 4.2. This was done in preparation for the requirements of the Oracle. The design of the EWH model was achieved by utilising the OOP nature of Python. EWH model classes were created for both the one and two node models, with the class attributes and methods reflecting the relevant constants and algorithms required during the operation of each. The class design was abstracted in such a way that the usage of either model was as similar as possible, besides the initialisation of the model object.

The simulator modifies the values stored in the models by first evaluating whether the internal temperature has exceeded the thermostat threshold. Next, the schedule of the EWH is assessed to determine whether the EWH is allowed to draw power. Based on this, the input energy for the time step is determined. Next, the energy used due to water usage is determined. And finally, the energy lost in the time step is determined from the current temperature values. All of these energies are used to calculate the change in energy and, finally, the change in internal temperature of the EWH.

### 4.5.1 Simulator Rev0

The first implementation of the model and simulator followed the algorithm as discussed in section 4.2.3. This implementation took the form of a scalar model and simulator, written in standard Python, which was capable of simulating the energy flow and temperature

**Table 4.2:** Simulator requirements.

ID	Requirement
1	Simulator must implement the one node EWH model
2	Simulator must implement the two node EWH model
3	Simulator must be able to operate at a preselected interval period
4	Simulator must be able to concurrently simulate multiple EWHs
5	Simulator must be able to manage a variety of different EWHs concurrently
6	Simulator must use the measured values as model parameters
7	Simulator must be able to adhere to a prescribed heating schedule
8	Simulator must be able to concurrently simulate multiple power limit scenarios
9	Oracle interval must be able to exceed the simulator interval
10	Simulator must be computationally efficient

response of a single EWH for a variable amount of time. Both the one and two node variants were successfully implemented. The data layout of the time series is shown in 4.13.

$$A_m = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \quad (4.13)$$

This version of the simulator satisfied requirements 1, 2 and 6 of the simulator, as stated in Table 4.2.

### 4.5.2 Simulator Rev1

The second version of simulator remained scalar in nature, as did the models. However, the main improvement made over Rev0 was the correction of the sequential energy and temperature calculations. The original algorithm for each interval of the one-node model is shown below:

1. if usage event: calculate  $E_{use}$  and update  $T_{inside}$
2. if standing losses: calculate  $E_{loss}$  and update  $T_{inside}$
3. if energy input from element: calculate  $E_{input}$ ,  $\Delta T_{inside}$ , thermostat duty cycle and  $T_{inside}$

The improved algorithm for the one-node model was implemented using the energy balance equation for each interval, as shown below:

1. if usage event: calculate  $E_{use}$
2. if standing losses: calculate  $E_{loss}$
3. if energy input from element: calculate  $E_{input}$
4. calculate change in temperature:  $\Delta T_{inside} = E_{input} - E_{use} - E_{loss}$
5. calculate new internal temperature:  $T_{inside} = T_{inside} + \Delta T_{inside}$

This effectively removes the erroneous temperature updating that was previously done, which resulted in an unbalanced energy equation. This version mainly corrected the energy equation used with the EWH models and has no direct requirement outcome from Table 4.2. Each EWH still required its own simulation and grid synchronisation was still lacking.

### 4.5.3 Simulator Rev2

The third version of the model and simulator had a focus on concurrency. Due to the relative slow execution of scalar simulations for multiple data sets and the difficulty with synchronising the effects each would have on a combined grid, the shift was made to concurrently simulate multiple EWHs. Due to the SIMD nature of the data, where some instructions are executed sequentially for multiple independent data sets, the shift to vectorisation was made. The resultant data structure of the aggregate time series is shown in 4.14. To facilitate the initial implementation, the Pandas library and the versatile DataFrame construct were used. This allowed a natural representation of the data to be stored in tabular format, where each observation occupied a single row and each column corresponded to a single EWH. This enabled vectorised iterations over the data elements during simulation, greatly improving the execution speed of the simulations. This also enabled synchronising the grid of EWHs for each interval, allowing a grid power limit to be implemented.

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \quad (4.14)$$

However, due to the versatility of the DataFrame, the iterations were very slow and would take numerous minutes to execute for only a few days worth of intervals. Still improving on the performance of the pure scalar implementation, but not close to expected performance figures.

**Verdict:** This version of the model and simulator satisfied requirements 3, 4, 5 and 7, enabling concurrent simulation of unique EWHs while abiding to a prescribed heating schedule for each EWH.

### 4.5.4 Simulator Rev3

The fourth version of the model and simulator further improved on the concurrency of Rev2 while greatly improving the computational efficiency. The shift from the versatile DataFrame was made to the NumPy array, a less versatile but much more computationally optimised data container. This alone provided the largest performance improvement. Keeping with the tabular paradigm, each row in an array represented an observation and each column an EWH. The next change to be applied was to incorporate multiple power limits into a single simulation, forming an  $M \times N \times P$  matrix, in order to reduce any unnecessary serial execution of the simulator. This was achieved by concatenating copies of the input data sets for each power limit into a third dimension, creating a 3D input data structure for all the input data sets. This data structure is shown in 4.7.

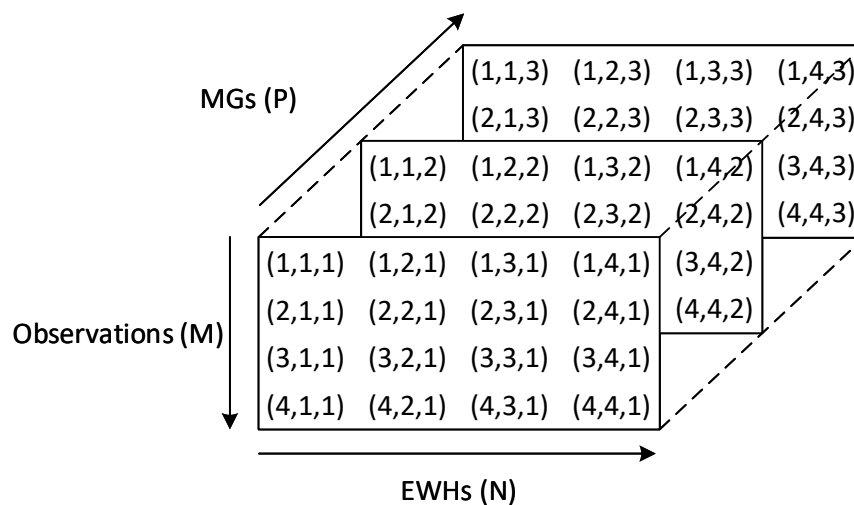


Figure 4.7: 3D array structure.

The evaluation of the model at each time step is then done by evaluating each observation as a  $1 \times N \times P$  matrix, where each of the  $P$  MG sub-matrices are independent from the other MGs. This enables the concurrent simulation of the same MG at various power limits.

**Verdict:** This version of the model and simulator satisfied requirements 8 and 9.

**Final Verdict:** Throughout the refinement of the simulator, low computational complexity was kept in mind and successfully implemented, which satisfied requirement 10. All the requirements were satisfied in the final Rev3 simulator, indicating a successful design.

## 4.6 Numerical Processing Tools Used

Numpy is a scientific computing package used with Python. It provides powerful vectorisation capabilities along with code execution in Cython to speed up processing time. Numpy is a mature library that is packaged in the SciPy stack which is the scientific computing stack of Python.

## 4.7 Performance Metrics for Oracle

The performance of the algorithm is determined by the relative performance in a set of selected metrics. These range from MG perspective, consumer perspective and simulation perspective.

### 4.7.1 Peak Demand and Demand Profile

One of the primary goals of the algorithm is to reduce the peak demand of the MG of EWHs, and is one of the direct constraints applied during simulation. As such, this is one



of the key metrics to be evaluated, to ensure that the peak demand does not exceed the specified peak limit as stated in (4.17). As a result of the peak demand metric, the overall demand profile of the MG will potentially differ from the measured reference value due to the demand shifting that occurs as the EWHs are utilised as thermal energy storage devices. The resultant demand profile provides insight into how the demand has been smoothed by the algorithm.

### 4.7.2 Mean Event Temperature

From the consumer's perspective, the resultant temperature of the hot water during an event is typically the highest priority. If the experienced temperature is too low, especially during winter months, it will diminish the incentive for the consumer to participate in the peak reduction strategy. This metric is used as one of the consumer comfort indicators.

### 4.7.3 Cold Events

The other consumer comfort indicators are found in the cold events experienced. A cold event is defined as an event with a mean temperature below that of the selected threshold. Due to the stochastic nature of event sizes, this adds complexity to the algorithm which strives to maintain user comfort during the simulation process.

#### 4.7.3.1 Cold Event Count

The primary indicator of cold events is the explicit amount of cold events experienced over the course of the Oracle scheduling. Due to the willingness of consumers to participate in DR being largely determined by their own comfort, this indicator provides a direct number to compare various scheduling approaches to one another.

#### 4.7.3.2 Cold Event Temperatures

A secondary indicator of cold events to augment the number of cold events indicator, is the extent of how much comfort is violated during cold events. The figures of interest are the minimum event temperatures for a worst case scenario perspective and the mean cold event temperature for a sense of how cold the experienced cold events are.

### 4.7.4 Total Energy Consumption

As one of the goals of SGs is to enable wiser control of energy consumption, it follows that more efficient use of the consumed energy is of high priority. Though not a focal point in the development of the algorithm, it is included as a metric to establish better overall analysis of the algorithm performance. The total consumed energy for the various peak limits are compared to the measured reference to provide a baseline for energy consumption performance.

### 4.7.5 Simulation Performance

Of paramount concern is the computational complexity of the algorithm. The aim is to keep a linear computational complexity which would allow the algorithm to scale well with an increase of data and simulation period.

**Table 4.3:** Oracle requirements.

ID	Requirement
1	Oracle must be able to handle multiple EWHs
2	Oracle must be able to handle multiple power limits
3	Oracle must be able to adhere to a prescribed heating schedule
4	Oracle must ensure EWH grid remains within set power limit
5	Oracle must ensure user comfort to the best of its ability
6	Oracle must be computationally efficient
7	Oracle must be modular to handle a multitude of load management scenarios

## 4.8 Oracle Development

The Oracle algorithm was designed in Python to facilitate a rapid development cycle. Implementing the algorithm in Python allows seamless integration with the data interfacing package developed in chapter 2. The Oracle was developed with a set of requirements in mind, these are listed in Table 4.3.

### 4.8.1 EWH Prioritisation

The allocation of priority to individual EWHs is typically based on MG parameters, such as reserve margin and expected demand. For MG and consumer aware priority allocation as introduced in this dissertation, the stochastic nature of consumer behaviour and consumer comfort is considered during priority allocation. The resulting priorities are then considered along with the MG reserve margin to establish a DR which strives to satisfy both MG constraints and consumer comfort levels.

The MG constraint is satisfied by ensuring that the total demand of the EWH MG remains within a specified limit. Moreover, to maintain consumer comfort, the goal is to ensure that the temperature of the EWH water is sufficiently high in order to facilitate comfortable temperatures during the next hot water usage event. These two factors are accounted for in two steps; the first step evaluates the EWH priority within the MG based on user comfort during usage events and the second step evaluates, based on the first step, which EWHs can have power allocated to them while remaining within the power constraint.

#### 4.8.1.1 Parameter Selection

The conventional EWH scheduling wisdom sees users allocating power to their EWH to heat in preparation for later expected hot water usage. The amount of time required for the EWH to sufficiently heat the water is rarely known and is based on an estimate from previous experience, typically ranging from one to two hours.

To improve on this arbitrary time estimate, an investigation into the main drivers of EWH energy flow is considered to identify the drivers with the highest weights. These drivers will then be used to develop a method to establish the relative urgency of EWHs.

From the conventional wisdom of EWH scheduling, the main goal is to have hot water available for an expected hot water usage event. The components that directly influence the heating schedule include:

- Time remaining until hot water is required.
- Current water temperature.
- Water usage.
- Heating rate (energy input rate).
- Inlet water temperature.
- Ambient temperature.

The time parameter ties all these components together. To identify the weights of each of these components, the energy and rate of energy transfer per unit of time are investigated.

Using the above established theory, the relative contribution of each aspect is weighed against the total thermal energy contained within the water of the EWH. This will provide insight into how taxing or beneficial the components are, which directly impacts the heating schedule of an EWH.

Using sample EWH parameters, as shown in Table 4.4, along with the aforementioned equations, the theoretical energy magnitude expended or gained as a result of usage, loss and heating was calculated over the period of one minute. These values are then normalised to a percentage of the total EWH energy. The resulting values are summarised in Table 4.5. The sample EWH parameters were selected based on the mean EWH parameters from the field study.

From Table 4.5 it is evident that energy recovery through hot water usage takes place at a much higher rate than both energy input through heating and heat loss through thermal radiation. Even at a low flow rate of 5 L/min, typically associated with washing hands, a magnitude of 244 W h energy can theoretically be recovered, 3.5 % of the total thermal energy contained in the EWH. The energy recovered at a high flow rate of 20 L/min, typically associated with a shower or drawing a bath, clocks in at 977 W h, 14 % of the total thermal energy contained in the EWH. Considering the energy input due to heating, the 3 kW element manages to input 50 W h of energy into the water, only 0.72 % of the total thermal energy in the EWH. Contrasting this with the heat loss, at 0.924 W h, or 0.01 % of the total thermal energy, the main concern is clearly hot water usage.

**Table 4.4:** Parameters of sample EWH used for cost function design.

Parameter	Symbol	Value	Unit
<b>Specific heat capacity of water</b>	$c$	1.1628	W h/(kg °C)
<b>Density of water</b>	$\rho$	1000	kg/m <sup>3</sup>
<b>EWH volume</b>	$V_{EWH}$	150	L
<b>EWH element power</b>	$P_{EWH}$	3	kW
<b>EWH outlet temperature</b>	$T_{hot}$	60	°C
<b>EWH inlet temperature</b>	$T_{inlet}$	18	°C
<b>EWH ambient temperature</b>	$T_{amb}$	20	°C
<b>EWH thermal resistance</b>	$R$	0.7	°C/W
<b>Cooling constant</b>	$k$	2.207	1/μs
<b>Low flow usage</b>	$V_{rate\_low}$	5	L/min
<b>Medium flow usage</b>	$V_{rate\_med}$	10	L/min
<b>High flow usage</b>	$V_{rate\_high}$	20	L/min

**Table 4.5:** Parameter energy values for a 1 minute period of the sample EWH.

Parameter	$V_{rate}$ (L/min)	$E$ (W h)	% of Total
<b>Low flow usage</b>	5	244.183	3.50
<b>Medium flow usage</b>	10	488.367	7.00
<b>High flow usage</b>	20	976.733	14.00
<b>EWH heating</b>	N/A	50.000	0.72
<b>EWH heat loss</b>	N/A	0.924	0.01
<b>EWH total energy</b>	N/A	6976.667	100.00

As a result, the largest consideration towards creating a schedule is the expected volume of hot water to be extracted, followed by the element rating of the EWH. A balance between these two components needs to be found in terms of time trade-off. Since the energy recovery is typically at a rate in the range of 4.9 - 19.5 times that of the energy input, the energy input must be allowed up to 19.5 times longer to sufficiently heat the water. Due to the stochastic volumes of hot water used, the expected volume usage contributes less information than the expected time of the hot water usage event, in terms of ensuring consumer comfort.

As minimal heating in preparation of a small event may be sufficient for the small event, but greatly detrimental to a larger event, the expected volume usage is deemed to contribute less information required for maintaining user comfort.

**First parameter:** From this, the first parameter is obtained, the expected time of the hot water usage event.

The second largest consideration towards creating a schedule is the current level of thermal energy contained within the EWH, i.e., how hot the EWH water is. If there is sufficient thermal energy available, heating the water is not a high priority as the hot water usage may occur and it will maintain consumer comfort. This is contrary to a low level of thermal energy, i.e. cold water, whereby the consumer comfort will not be maintained and may require a lengthy heating period prior to the usage event to ensure consumer comfort.

**Second parameter:** From this, the second parameter is obtained, the current temperature of the EWH water.

#### 4.8.1.2 Cost Function

In order to establish the relative urgency of the EWH power requests, the EWH MG is evaluated based on the above selected parameters. These parameters enable the evaluation of energy required at some point in the future versus the amount of time available to deliver the required energy.

In order to assign priority scores to the EWHs, each is evaluated by the selected parameters as described above. To gain better insight into the urgency

$$k_n = -\alpha t_{event} + \beta t_{heat} \quad (4.15)$$

---

**Algorithm 1:** Oracle priority allocation algorithm.

---

```

1 receive thermostat result  $P_{thermostat} = \{P_1, \dots, P_N\}$ ;
2  $cost = cost\_function(alpha, time\_delta, beta, temp\_delta)$ 
3  $priority = cost.argsort()[::-1]$ 
4  $P_{cumulative} = cumsum(P_{thermostat}[priority])$ 
5  $P_{selected} = P_{cumulative} \leq P_{limit}$ 
6  $P_{remainder} = P_{limit} - P_{selected}$ 
7  $P_{optimise} = P_{notselected} \leq P_{remainder}$ 
8  $prioritised = P_{selected} | P_{optimise}$ 
9  $prioritised[priority] = prioritised$ 

```

---

### 4.8.2 MG Power Limit Enforcement

Once the cost function has been evaluated, the resultant cost vector is used to allocate power to the MG of EWHs. The pseudo code of the algorithm can be seen in Algorithm 1. This algorithm works in two phases, the first allocates power to the majority of EWHs and the second searches for a remaining EWH that may fill the remaining surplus capacity. The goal of this phase is to ensure the MG power demand remains within the power limit while maximising the number of EWHs that have power allocated to them. The objective is shown in (4.16) and the constraint is shown in (4.17).

$$P_{grid} = \sum_{n=1}^N P_n \quad (4.16)$$

With the constraint that:

$$P_{grid} \leq P_{limit} \quad (4.17)$$

Procedure:

1. Sort thermostat result by EWH priority, as determined by the cost function.
2. Cumulative sum the resultant arrays, by the P power limits
3. Determine number of EWHs able to turn on while remaining within power limit, for P power limits
4. Determine surplus capacity remaining of the P power limits
5. Determine number of EWHs able to turn on within surplus capacity remaining of the power limits (optimisation step)

#### 4.8.2.1 Phase One

Phase one of the power allocation assigns power to the majority of eligible EWHs that fall within the power limit. This is done by determining the number of EWHs with highest priority that may be allocated power.

The input to this phase is the result of an intersect (AND) between the thermostat control, which determines based on set-point control which EWHs should heat the water, and schedule control, which is a preallocated schedule that was set for each individual EWH. Therefore, it may be that some EWHs have a power request of zero, instead of

their element value. At an algorithm level, this result is achieved by the intersect of the thermostat and schedule results.

The other input to this phase is the priority assignment of each individual EWH, based on the prior developed cost function. This is used to establish which EWHs have the highest priority and should be considered for power allocation first. At an algorithm level, this is used to sort the thermostat and schedule result by descending order of priority.

Using this priority-sorted power request vector, the number of EWHs allowed to have power allocated to them is determined. At an algorithm level, this was achieved by cumulative summing the power request vector and selecting all elements that are below the index of the element that results in a cumulative sum larger than the power limit. A mathematical representation of a cumulative sum of a vector is shown in (4.18). This ensures that all the selected EWHs will remain within the specified power limit.

An important note, the Oracle must be evaluated along with the thermostat and schedule. Since the power request vector may contain zeros, the Oracle could assign a large number of EWHs, which at time of Oracle evaluation were within the power limit, but at subsequent evaluation of the thermostat and schedule control, more non-zero power requests may be present in the Oracle allocated priority list.

$$y_i = \sum_{j=1}^i x_j \quad (4.18)$$

#### 4.8.2.2 Phase Two

Phase one allocated power to the majority of EWHs which allows the MG to remain within the specified power limit. However, due to variable EWH element ratings, this may not provide an optimal solution. If, during the cumulative sum, the element that causes the power request vector to overshoot the power limit is large, say 4 kW, this method will terminate the search for more EWHs as the cumulative sum will indicate that the power limit will be exceeded beyond this point. It may be the case that directly following the large 4 kW element, a smaller 2 kW element may be requesting power that would still fit within the remaining power limit.

This necessitates another evaluation of the power request vector, to establish whether there may be a suitable element which could fill in the remaining power limit to maximally utilise the specified power limit. To achieve this, the elements not considered for power allocation during phase one are measured against the remaining power limit surplus. The element with the highest priority is added to the list of EWHs that are to have power allocated to them, while remaining within the power limit constraint.

Due to the typical EWH element ratings, as shown in Table 4.6, only a single EWH may be selected from the remaining EWHs in order to preserve a balance between optimally utilising the power limit and preserving the inexpensive nature of the Oracle.

## 4.9 Results

Using the defined metrics from section 4.7, the Oracle performance is evaluated in this section.

Table 4.6: Parameters used during Oracle simulation.

Parameter	Symbol	Value	Unit
Specific heat capacity of water	$c$	1.1628	W h/(kg °C)
Density of water	$\rho$	1000	kg/m <sup>3</sup>
EWH volume	$V_{EWH}$	100, 150, 200	L
EWH element power	$P_{EWH}$	2, 3, 4	kW
EWH outlet temperature	$T_{hot}$	Simulated	°C
EWH inlet temperature	$T_{inlet}$	Measured	°C
EWH ambient temperature	$T_{amb}$	Measured	°C
EWH thermal resistance	$R$	0.7	°C/W
Number of EWHs	$N$	34	
Oracle actuation period	$\Delta t_{oracle}$	10	minutes
Hot temperature threshold	$T_{hot}$	45	°C
Peak limit	$P_{limit}$	20, 30, 40, 50	kW
Set point temperature	$T_{limit}$	50, 60, 70	°C
Ripple control time start		18:00	
Ripple control time end		20:00	
Day period start		2016-10-13	
Day period end		2016-10-14	
Month period start		2016-09-19	
Month period end		2016-10-17	
UTC		+2	hours

#### 4.9.1 Evaluation Parameters

For reference, all the metrics are evaluated against the measured field data as well as a simulated ripple control scheme. For all simulations, no individual scheduling is applied by the consumer, i.e. the EWH is constantly supplied with power. The Oracle and ripple control scheduling is applied through SEC control. The Oracle parameters used during this experiment are summarised in Table 4.6.

#### 4.9.2 Peak Demand and Demand Profile

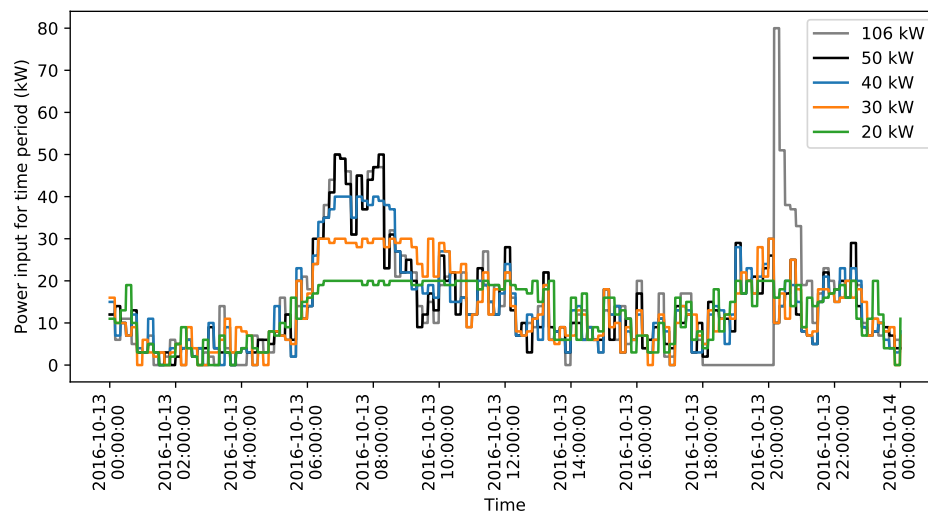
The peak demand outcome is summarised in Table 4.7. Regardless of the specified power or set temperature limit, the Oracle ensured the power limit is never exceeded. The highest selected power limit of 50 kW managed to just exceed the measured peak demand of 47 kW during the full month period for all set limits. During this same period, the ripple control managed a peak demand of 106 kW, more than double that of the measured and Oracle peak demand.

**Verdict:** This satisfies the peak demand metric.

The demand profile outcome is best described with a visual representation of a typical day's demand profile. Such a view is shown in Fig. 4.10. This shows a representation of the load profiles of the Oracle at various power limits with the ripple control as reference. Here the above mentioned peak demands are clearly illustrated to adhere to the set power limits. The lower power limits, such as the 20 kW limit, results in a much flatter profile overall where the typical peak demand period is spread out over a longer period, as indicated during the morning peak. The higher power limits, such as the 50 kW limit,

**Table 4.7:** Peak demand in kW experienced for various combinations of temperature and power limits compared against the measured values.

Set Limit Power Limit	50		60		70	
	Day	Month	Day	Month	Day	Month
20	20	20	20	20	20	20
30	30	30	30	30	30	30
40	40	40	40	40	40	40
50	47	50	50	50	50	50
106	80	87	80	106	99	106
Measured	38	47	38	47	38	47

**Figure 4.8:** Grid power comparison for temperature limit of 60 °C at various power limits for a day period.

notably has a less flat profile and a shorter peak demand period that is on par with the reference ripple control morning peak demand. During the evening peak demand, the ripple control managed to result in a significant peak directly succeeding the ripple period. This peak was noted as 80 kW for the selected day at a set limit of 60 °C. This is a clear indication of how ripple control is an archaic control scheme that could lead to large, unexpected peak demand spikes.

**Verdict:** This satisfies the demand profile metric.

### 4.9.3 Mean Event Temperature

The mean event temperatures are summarised in Table 4.8. This metric evaluates the level of consumer comfort and, as a result, is of paramount importance. From Table 4.8 it is observed that none of the synthetic test measurements dipped below the measured temperature of 46 °C for the selected day, nor the full month period. The lowest set limit of 50 °C provided the most uniform measurements of 48 °C which is 2 °C above the measured temperature. This is an encouraging result as the consumer's comfort is strongly catered to. Increasing the set limit to 60 °C and 70 °C sees the consumer comfort



**Table 4.8:** Mean event temperature in °C experienced for various combinations of temperature and power limits compared against the measured values.

Set Limit Power Limit	50		60		70	
	Day	Month	Day	Month	Day	Month
20	47	48	54	55	61	62
30	48	49	56	57	64	56
40	48	49	57	57	65	66
50	48	49	57	58	65	66
106	48	48	57	57	65	66
Measured	46	46	46	46	46	46

increased by 11 °C and 20 °C respectively on average from the measured temperature. This indicates that the Oracle places high emphasis on maintaining consumer comfort.

**Verdict:** This satisfies the mean event temperature metric.

#### 4.9.4 Cold Events

Cold events are a clear indication of the amount of discomfort endured by the consumer. This is to be mitigated by the Oracle to improve the consumer's willingness to participate in this form of DR.

##### 4.9.4.1 Cold Event Count

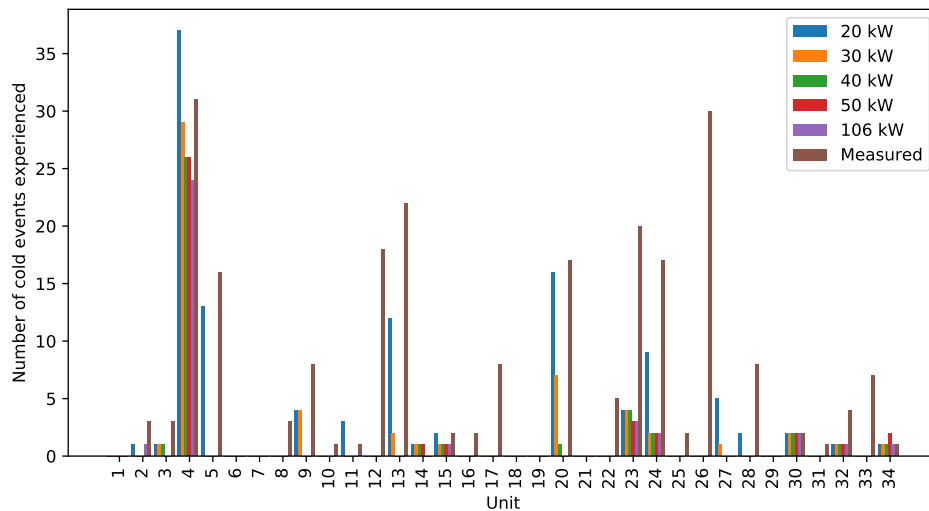
The simplest metric to evaluate a consumer's comfort is the number of cold events experienced as summarised in Table 4.9. Regardless of the selected set and power limits, the Oracle manages to significantly reduce the amount to experienced cold events for both the day and month periods. With the selected day indicating 232 cold events, the Oracle with a set limit of 50°C manages to reduce the amount to 71 with a moderate power limit of 40 kW. Interestingly, the ripple control allowed more cold events for this day, with 74 recorded cold events. This same trend is evident and more obvious looking at the month period results. The measured events report 5376 cold events compared to the lowest selected power limit of 20 kW with 2562 cold events, less than half the measured amount of cold events. Again, increasing the power limit further improves on these figures, with a 50 kW power limit only resulting in 1513 cold events, over 1000 less than in the 20 kW power limit case.

Stepping up the set limit to 60 °C and 70 °C both incur even more impressive reductions in cold events. At 50 kW power limit, the 60 °C set limit manages to reduce the cold events to only 578 for the month period and the 70 °C set limit manages only 402 cold events. With this metric it is again evident how ripple control is not a good control strategy as it always manages more cold events than the moderately selected 50 kW power limit. This is especially evident with the 60 °C set limit where ripple control manages 675 cold events, almost 100 more cold events than the 50 kW power limit Oracle controlled case.

Figure 4.9 shows the distribution of cold events experienced in the selected day period at a set limit of 60 °C. From this figure it is evident that there is an EWH that is being used over its expected capacity and is potentially skewing the greater impression of the Oracle algorithm. Though as evident, the results still indicate strong performance from the algorithm.

**Table 4.9:** Number of cold events experienced for various combinations of temperature and power limits compared against the measured values.

Set Limit Power Limit	50		60		70	
	Day	Month	Day	Month	Day	Month
20	134	2562	114	1967	101	1934
30	86	1693	56	894	52	866
40	71	1524	40	632	35	493
50	71	1513	38	578	28	402
106	74	1756	35	675	29	416
Measured	232	5376	232	5376	232	5376

**Figure 4.9:** Number of cold events for temperature limit of 60 °C at various power limits for a day period.

**Verdict:** This satisfies the number of cold events metric.

#### 4.9.4.2 Cold Event Temperature

Along with the number of cold events, the mean cold event temperature is evaluated to measure the extent of the discomfort which the consumer will experience. Table 4.10 summarises the findings. It is notable how all the Oracle controlled EWHs manage to have fewer cold events as well as warmer cold events than the measured and ripple controlled benchmarks. The Oracle controlled results are typically over 40 °C compared to the measured 37 °C. This indicates that the cold events that do occur are not as unbearable as what the consumers have been experiencing thus far.

**Verdict:** This satisfies the cold event temperature metric.

#### 4.9.5 Total Energy Consumption

The final metric of the Oracle algorithm was not a direct driver during the algorithm development but is a typical metric used in evaluating SG control algorithms, the total energy consumption. Table 4.11 provides a summary of the obtained energy values for the

**Table 4.10:** Mean event temperature of cold events in °C experienced for various combinations of temperature and power limits compared against the measured values.

Set Limit Power Limit	50		60		70	
	Day	Month	Day	Month	Day	Month
20	40	40	40	40	40	40
30	41	42	41	41	41	38
40	42	42	42	42	41	39
50	42	42	41	43	39	39
106	41	42	39	42	38	41
Measured	37	37	37	37	37	37

**Table 4.11:** Energy consumption in kWh experienced for various combinations of temperature and power limits compared against the measured values.

Set Limit Power Limit	50		60		70	
	Day	Month	Day	Month	Day	Month
20	246	6317	306	7873	362	9332
30	260	6527	326	8317	385	9962
40	266	6554	333	8429	403	10233
50	266	6559	336	8442	408	10286
106	264	6497	333	8366	406	10200
Measured	319	7422	319	7422	319	7422

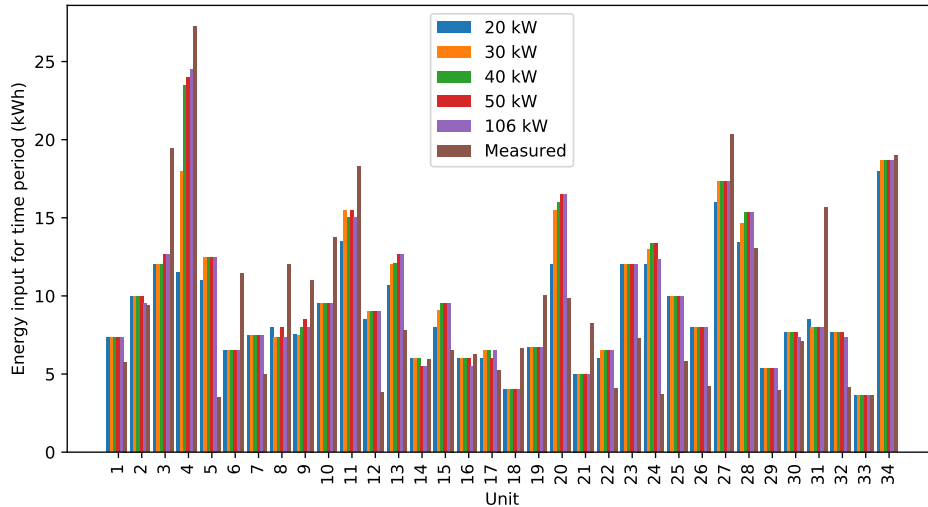
various tests. The measured energy was found to be 319 kWh and 7422 kWh for the day and month periods respectively. This exceeds the amount of energy consumed by all the 50 °C set limit experiments, of which the highest energy consumption was experienced by both the 40 kW and 50 kW power limits at 266 kWh.

Increasing the set limit to 60 °C puts the energy consumption of the lowest power limit of 20 kW at 306 kWh and 7873 kWh, close to the measured values. Increasing the power limit has the obvious result of increased energy consumption, with the 50 kW limit consuming the largest amount of energy at 336 kWh and 8442 kWh respectively. Notably, the ripple control consumed only slightly less energy than the Oracle controlled tests with the closest match being the 40 kW power limit.

**Verdict:** This metric succeeds in putting the energy consumption of the algorithm in context along the measured values and a simulated ripple control, though the total consumed energy is higher.

#### 4.9.6 Simulation Performance

The simulator was refined to be less computationally expensive and the Oracle was developed with low computational complexity in mind. As such, a key benchmark is the algorithm performance in terms of execution time. The original model and simulation times from [25] are used as reference. The original design is described as 13000 times faster than the existing PDE models and 150 times faster than the existing two-node models. The comparison is made that a single EWH, simulated for a 10 day period, takes about 100 milliseconds on a desktop computer. Scaling this linearly to 10000 EWHs simulated for a 10 day period on 100 desktop computers would take only 10 seconds [25]. This



**Figure 4.10:** Total energy consumption for temperature limit of 60 °C at various power limits for a day period.

**Table 4.12:** Simulation execution time in seconds for various numbers of EWHs considered.

<b>EWHs</b>	<b>Day</b>	<b>Week</b>	<b>Month</b>
34	0.58	1.71	5.43
136	0.65	2.02	6.35
340	0.78	2.53	8.07
1360	1.35	4.58	15.38

benchmark is stated for the two node model, which is not linear and, consequently, not the most accurate comparison. Regardless, the refined one node model with the Oracle algorithm is evaluated against this benchmark as it is the only available execution time.

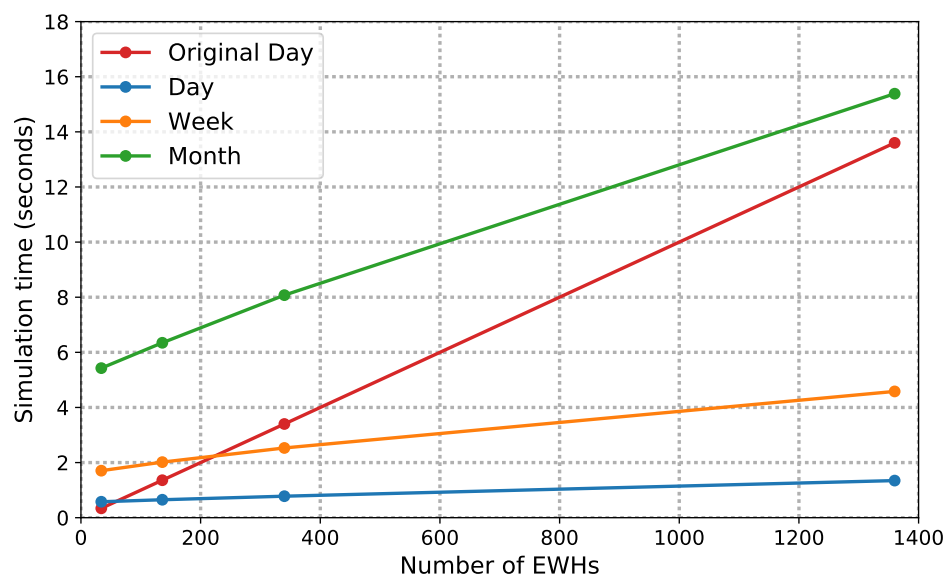
The simulation performance for the refined model and Oracle algorithm was evaluated on a laptop with an i7 4712MQ CPU and 16 GB of DDR3 RAM.

Due to the vectorisation employed, there is a constant time offset present in the execution times of the refined model which makes the execution time non-linear initially, but after this initial offset the trend becomes linear as shown in Fig. 4.11. The original model's performance for simulating a single day is shown in red which has the refined equivalent in blue. The vectorised model scales much better in terms of execution time. To more accurately compare the initial developed model with the refined model, a normalised figure is calculated from the execution times, as shown in Table 4.13. From these values it is evident that only the case where 34 EWHs are simulated for a single day executes slower than the original model. Any increase in EWHs and/or time period drastically reduces the normalised time taken to execute the simulations. For the case of 1360 EWHs simulated over a month period, the normalised execution time per EWH per day is reduced to 0.4 milliseconds compared to the original 10 milliseconds, 25 times faster execution.

**Verdict:** The simulation performance was significantly improved and enables concurrent evaluation over the original scalar implementation.

**Table 4.13:** Normalised simulation time in milliseconds for 1 EWH over a period of 1 day.

<b>EWHs</b>	<b>Day</b>	<b>Week</b>	<b>Month</b>
34	17.01	7.17	5.70
136	4.79	2.12	1.67
340	2.29	1.06	0.85
1360	0.99	0.48	0.40
1 original	10	10	10

**Figure 4.11:** Oracle simulation times for various amounts of EWHs and time periods compared to the original simulator time for a period of a single day.

# Chapter 5

## Conclusion

Utilities face growing demand and challenges in supplying energy. Current unidirectional DSM, such as ripple control, is detrimental to consumer comfort. Modern information technologies enable SGs to deliver power in more efficient ways by responding to wide ranging conditions and events. This information flow generates a lot of data, which must be processed to enable the automated capabilities of the SG.

Using the data collected from the Geasy project, a data cleaning framework was developed to improve the data quality in preparation for further analysis. This data was then statistically analysed to extract useful temporal information which led to the development of linear statistical models for usage forecasting. The forecasting model proved to accurately capture the seasonal structure of the temporal data. Additionally, the Geasy data was used along with a verified EWH model to develop a novel SG DSM application. This novel SG DSM application could be used in near real-time to provide DSM for EWHs which considers the needs of both the utility and the consumer while controlling individual EWHs. This enables the DSM application to ensure minimal consumer discomfort while maintaining peak demand levels below a specified maximum value. This DSM application was evaluated using metrics established from the perspectives of the utility and the consumer.

### 5.1 Evaluation of Work

This section provides a summary of the work done and the results achieved in Chapters 2, 3 and 4 with reference to the dissertation objectives stated in Chapter 1.

#### **Objective 1: Assess and improve current data quality.**

**1(a) Investigate Geasy data path for causes of data quality reduction:** The various nodes in the communication infrastructure of the Geasy platform were successfully investigated and potential reasons for failure were identified for each node. The likelihood of the failures are also estimated to give an overview of how the nodes may (if feasible) be improved to reduce data loss in future. Of primary concern are the power loss of the Geasy SEC, airtime depletion, poor connectivity and the eventual RLE that contribute to the largest amount of data loss.

**1(b) Assess current Geasy data quality:** The data quality of numerous Geasy SECs was successfully investigated and the largest concern was identified as incomplete data,

where some reported completion of as high as 93 %; others reported completion of only 51.5 %. The data accuracy was found to be high, with only a few obvious data spikes observed in the temperature data sets; however, a time accuracy concern was noted where the sampling period has an added drift from the specified 60 s resolution which made the effective sampling period 61-62 s. The data had a high level of consistency throughout the considered fields of interest. Reliability was assumed to be high as no further alterations besides data quality improvements have been made. The selected fields are all relevant for effective further analysis of the data.

**1(c) Develop numerical data cleaning framework:** A numerical data cleaning framework was successfully developed which employed cleaning routines based on the requirements for the specific data fields. All data fields had sampling period regularisation performed to ensure a constant sampling rate of 60 s, an important aspect with further data analysis. All fields except the hot water usage required outlier mitigation to remove erroneous outliers. The power field responded best to persistence imputation for imputing missing values and both the temperature fields and the water usage field responded best to linear imputation. The power and water fields were limited to a maximum imputation of five samples where the temperature fields were not limited to a maximum imputation sample length. Additionally, the temperature fields were filtered to reduce sensor noise and produce smoother, more natural temperature curves.

**Objective 2: Analyse and model temporal structure of EWH usage.**

**2(a) Investigate temporal structure of EWH hot water usage:** Additive time series decomposition successfully indicated there is a seasonal component (for the provided time frame) to be extracted from the hot water usage data. Due to the stochastic nature of human behaviour, a prominent residual component remained.

**2(b) Model EWH hot water usage using linear statistical model:** The hot water usage was successfully modelled using SARIMA models. The Box-Jenkins method of identifying model parameters effectively produced a suitable model which captured the seasonality of the data. Grid searching proved to be ineffective in this case for optimising the parameters. Due to the evident non-linear nature of the hot water usage data, purely relying on the AIC metric for model selection was shown to be misleading; however, utilising the MSE along with visual inspection proved effective.

**2(c) Evaluate feasibility of linear model for EWH hot water usage forecasting:** The linear SARIMA models effectively modelled the seasonal component of the hot water usage data and produced a realistic seasonal forecast. The models were, however, unable to model the amplitude of the actual usage as the seasonal component was found to only be a fraction of the complete time series. Besides the forecasting performance, the modelling procedure was found to be labour intensive. The residual diagnostics proved ineffective with the hot water usage data as the assumption of normality was violated in all cases, allowing little diagnostic information to be gained. Accurate, mass modelling of the hot water usage data for multiple EWHs using linear statistical modelling techniques is therefore not feasible.

**Objective 3: Develop a smart grid demand-side management application.****3(a) Revise nodal EWH model by Nel [25] for increased scale of simulation:**

Utilising the concepts brought forward by Nel [25], the scalar model was successfully vectorised. This significantly improved the simulation performance when simulating multiple EWHs concurrently. Along with the model vectorisation, EWH data handling software was created to effectively extract, clean and handle the large, diverse data sets.

**3(b) Investigate main drivers considered during EWH scheduling:**

By investigating the energy flow associated with normal EWH usage, two key drivers were identified which directly impact EWH scheduling. The first is expected time of the next hot water usage event, which provides a target for when hot water is required. The second is the current temperature of the EWH, which provides insight into the amount of energy currently contained in the EWH.

**3(c) Develop DSM application for balancing consumer and utility needs:**

A two step algorithm was developed which first assesses the relative priority of the various EWHs, and, secondly, evaluates from this rank which of the EWHs can be turned on while remaining within the predefined power limit. This algorithm was evaluated by numerous metrics which represent requirements from both the utility and the consumer perspective. For the utility, the peak demand was successfully kept within the predefined power limit of up to 50 kW, compared to a simulated ripple control which reached 106 kW, while providing some demand profile smoothing depending on the selected power and temperature limits. For the consumer, the mean event temperatures were improved from a measured 46 °C to an average of 57 °C for the 60 °C temperature limit. This also greatly reduced the number of cold events experienced, with the measured 5376 cold events brought down to only 578 for the full month period for the 60 °C temperature limit. Finally, the algorithm and model vectorisation was proven to scale really well; where the original implementation by Nel [25] achieved an average simulation time of 10 ms per EWH per day, the revised implementation which also implements DSM control was able to reduce that to 0.4 ms per EWH per day, a 25 times faster simulation.

## 5.2 Recommendations

From the results of the dissertation objectives along with the proposed scope, the following recommendations are made for future work.

### 5.2.1 Data Wrangling

**Implement text-based data cleaning:** To enhance further analysis of EWH usage, metadata for each of the EWHs could be used to provide further insight into the specific EWH conditions such as installation orientation and installation location. This metadata may involve a lot of manual data entry and text-based data cleaning may improve the quality of the recordings.

**Implement data warehousing preparation:** In preparation for data warehousing, the developed numerical data cleaning framework along with a text-based data cleaning framework may be used to improve the data quality before permanent data storage.



**Implement data redundancy:** To decrease the risk of data loss, preventative measures may greatly reduce the reliance on data cleaning for more valuable data. Battery backup systems for smart controllers along with local message caching and message acknowledgment protocols may significantly reduce the amount of data loss experienced in a developing infrastructure.

### 5.2.2 Data Analysis and Forecasting

**Investigate non-linear statistical analysis and modelling:** The linear time series decomposition and the eventual linear statistical models indicated prominent residual components which could indicate non-linearities. Therefore, the data may be better described by non-linear decomposition, such as robust seasonal and trend decomposition using Loess (STL) and non-linear time series modelling.

**Investigate impact of time resolution on modelling:** To better group the data and significantly reduce modelling time, the EWH data was grouped in periods of 60 minutes. This showed good performance with linear modelling. This does, however, reduce the time resolution of the forecasting. Therefore, the impact of higher time resolution needs to be investigated as it may lead to higher accuracy forecasts.

**Investigate machine learning applications:** This study focussed on the statistical modelling of the EWH data. However, machine learning techniques could provide a more automated black box approach to data modelling and forecasting. Machine learning is typically also non-linear in nature and may therefore produce a more feasible modelling and forecasting approach, given strong evaluation guidelines.

### 5.2.3 Data Application: Demand-Side Management

**Investigate versatility of the cost function as a DSM tool:** The developed DSM application focussed on the management of EWH power supply in a SG. However, due to the nature of the algorithm, the versatility may enable the development of other DSM applications. This may target EWHs but with different objectives to be met such as TOU, pool pumps, other forms of HVAC or any devices that can afford to suffer power loss in the short term in order to meet some set objective.

**Investigate integration performance:** Due to the near real-time execution of the DSM application, a field test based on participating EWHs could provide valuable insight as to how the algorithm performs and which additional objective may be added. More or less aggressive objectives may be implemented to evaluate the relative performance from both utility and consumer perspectives.

**Investigate grid feedback:** In the true sense of SG, a feedback system from the SG to the SEC control platform providing useful information such as the current power margin available for HVAC could enable dynamic near real-time control. This would enable EWHs to be utilised as thermal storage while mitigating cold events experience by the consumer. Combining this with predictive scheduling and a truly smart utility network may be realised.

## List of References

- [1] M. Roux, N. H. Naude, M. J. Booyesen, and A. Barnard, “Electric Water Heaters In SmartGrids: Individual Savings Versus Network Peak Load Management,” in *SAUPEC. In stampa*, 2017. [Online]. Available: [https://www.researchgate.net/profile/Mj\\_thinus\\_Booyesen/publication/312093779\\_Electric\\_water\\_heaters\\_in\\_smartgrids\\_Individual\\_savings\\_versus\\_network\\_peak\\_load\\_management/links/586f66bf08ae8fce491dc16c/Electric-water-heaters-in-smartgrids-Individual-savings-versus-network-peak-load-management.pdf](https://www.researchgate.net/profile/Mj_thinus_Booyesen/publication/312093779_Electric_water_heaters_in_smartgrids_Individual_savings_versus_network_peak_load_management/links/586f66bf08ae8fce491dc16c/Electric-water-heaters-in-smartgrids-Individual-savings-versus-network-peak-load-management.pdf)
- [2] M. Roux and M. J. Booyesen, “Use of smart grid technology to compare regions and days of the week in household water heating,” in *2017 International Conference on the Domestic Use of Energy (DUE)*, Apr. 2017, pp. 276–283.
- [3] United Nations, *Water and energy*. Paris: Unesco, 2014. [Online]. Available: <http://unesdoc.unesco.org/images/0022/002257/225741e.pdf>
- [4] M. E. Bildirici, T. Bakirtas, and F. Kayikci, “Economic growth and electricity consumption: Auto regressive distributed lag analysis,” *Journal of Energy in Southern Africa*, vol. 23, no. 4, pp. 29–45, 2012.
- [5] N. M. Odhiambo, “Electricity consumption and economic growth in South Africa: A trivariate causality test,” *Energy Economics*, vol. 31, no. 5, pp. 635–640, Sep. 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0140988309000115>
- [6] Eskom Holdings SOC, “Eskom Home,” 2017. [Online]. Available: <http://www.eskom.co.za/Pages/Landing.aspx>
- [7] P. Nel, M. Booyesen, and B. van der Merwe, “Energy perceptions in South Africa: An analysis of behaviour and understanding of electric water heaters,” *Energy for Sustainable Development*, vol. 32, pp. 62–70, Jun. 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0973082615301666>
- [8] Department of Energy, “Department of Energy, South Africa Annual report,” 2017. [Online]. Available: <http://www.energy.gov.za/files/Annual%20Reports/DoE-Annual-Report-2016-17.pdf>
- [9] “News24 water investigation: Warnings and a turning point.” [Online]. Available: <http://www.news24.com/SouthAfrica/News/news24-water-investigation-warnings-and-a-turning-point-20171016>
- [10] M. Albadi and E. El-Saadany, “A summary of demand response in electricity markets,” *Electric Power Systems Research*, vol. 78, no. 11, pp. 1989–1996, Nov. 2008. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0378779608001272>

- [11] K. Kostková, . Omelina, P. Kyčina, and P. Jamrich, “An introduction to load management,” *Electric Power Systems Research*, vol. 95, pp. 184–191, Feb. 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S037877961200288X>
- [12] O. Erdinc, A. Tascikaraoglu, N. G. Paterakis, Y. Eren, and J. P. S. Catalao, “End-user comfort oriented day-ahead planning for responsive residential hvac demand aggregation considering weather forecasts,” *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 362–372, Jan. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7457313/>
- [13] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart Grid; The New and Improved Power Grid: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/6099519/>
- [14] S. Haller, S. Karnouskos, and C. Schroth, “The internet of things in an enterprise context,” in *Future Internet Symposium*. Springer, 2008, pp. 14–28.
- [15] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [16] The Economist, “The world’s most valuable resource is no longer oil, but data - Regulating the internet giants,” May 2017. [Online]. Available: <https://www.economist.com/news/leaders/21721656-data-economy-demands-new-approach-antitrust-rules-worlds-most-valuable-resource>
- [17] Bridgiot. (2017) Consumer Bridgiot. [Online]. Available: <https://www.bridgiot.co.za/consumer/>
- [18] M. J. Booyesen, J. Engelbrecht, and A. Molinaro, “Proof of concept : Large-scale monitor and control of household water heating in near real-time,” Pretoria, South Africa, Jul. 2013. [Online]. Available: <http://hdl.handle.net/10019.1/85478>
- [19] J. W. K. Brown, “Design, and implementation of an intelligent water heater control module for feedback demand side management,” Master’s thesis, Stellenbosch: Stellenbosch University, 2016. [Online]. Available: <http://scholar.sun.ac.za/handle/10019.1/98441>
- [20] P. J. C. Nel, “Rethinking electrical water heaters,” Master’s thesis, Stellenbosch: Stellenbosch University, 2015. [Online]. Available: <http://scholar.sun.ac.za/handle/10019.1/98076>
- [21] M. Booyesen and A. Cloete, “Sustainability through Intelligent Scheduling of Electric Water Heaters in a Smart Grid.” *IEEE*, Aug. 2016, pp. 848–855. [Online]. Available: <http://ieeexplore.ieee.org/document/7588943/>
- [22] Water Research Commission, “WRC Home,” 2017. [Online]. Available: <http://www.wrc.org.za/>
- [23] Mkhondo Local Municipality, “Mkhondo Local Municipality,” 2015. [Online]. Available: <http://www.mkhondo.gov.za/>

- [24] A. H. Cloete, “A domestic electric water heater application for Smart Grid.” Master’s thesis, Stellenbosch: Stellenbosch University, 2017. [Online]. Available: <http://scholar.sun.ac.za/handle/10019.1/100886>
- [25] P. Nel, M. J. Booysen, and A. B. van der Merwe, “A computationally inexpensive energy model for horizontal electrical water heaters with scheduling,” *IEEE Transactions on Smart Grid*, pp. 1–1, 2016. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7438883>
- [26] J. Han and M. Kamber, *Data mining: concepts and techniques*, 2nd ed., ser. The Morgan Kaufmann series in data management systems. Amsterdam ; Boston : San Francisco, CA: Elsevier ; Morgan Kaufmann, 2006.
- [27] X. Cheng, L. Fang, L. Yang, and S. Cui, “Mobile Big Data: The Fuel for Data-Driven Wireless,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1489–1516, Oct. 2017.
- [28] Y. Chu, S. Yang, and C. Yang, “Enhancing data quality through attribute-based metadata and cost evaluation in data warehouse environments,” *Journal of the Chinese Institute of Engineers*, vol. 24, no. 4, pp. 497–507, Jun. 2001. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/02533839.2001.9670646>
- [29] W. Kim, B.-J. Choi, E.-K. Hong, S.-K. Kim, and D. Lee, “A taxonomy of dirty data,” *Data mining and knowledge discovery*, vol. 7, no. 1, pp. 81–99, 2003. [Online]. Available: <https://link.springer.com/content/pdf/10.1023/A:1021564703268.pdf>
- [30] M. L. Lee, T. W. Ling, and W. L. Low, “IntelliClean: A knowledge-based intelligent data cleaner.” ACM Press, 2000, pp. 290–294. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=347090.347154>
- [31] L. Li, “Data Quality and Data Cleaning in Database Applications,” Ph.D. dissertation, Edinburgh Napier University, 2012.
- [32] S. Ahuja, M. Roth, R. Gangadharaiah, P. Schwarz, and R. Bastidas, “Using Machine Learning to Accelerate Data Wrangling,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 343–349.
- [33] R. Wang, V. Storey, and C. Firth, “A framework for analysis of data quality research,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 4, pp. 623–640, Aug. 1995. [Online]. Available: <http://ieeexplore.ieee.org/document/404034/>
- [34] K. Orr, “Data quality and systems theory,” *Communications of the ACM*, vol. 41, no. 2, pp. 66–71, Feb. 1998. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=269012.269023>
- [35] J. Hipp, U. Güntzer, and U. Grimmer, “Data Quality Mining Making a Virtue of Necessity,” Santa Barbara, CA, May 2001. [Online]. Available: [http://www.cs.cornell.edu/johannes/papers/dmkd2001-papers/p5\\_hipp.pdf](http://www.cs.cornell.edu/johannes/papers/dmkd2001-papers/p5_hipp.pdf)
- [36] R. Y. Wang and D. M. Strong, “Beyond Accuracy: What Data Quality Means to Data Consumers,” *Journal of Management Information Systems*, vol. 12, no. 4, pp. 5–33, Mar. 1996. [Online]. Available: <http://www.tandfonline.com/doi/full/10.1080/07421222.1996.11518099>

- [37] C. Fox, A. Levitin, and T. Redman, “The notion of data and its quality dimensions,” *Information processing & management*, vol. 30, no. 1, pp. 9–19, 1994. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0306457394900205>
- [38] T. C. Redman, “The impact of poor data quality on the typical enterprise,” *Communications of the ACM*, vol. 41, no. 2, pp. 79–82, Feb. 1998. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=269012.269025>
- [39] H. Müller and J.-C. Freytag, *Problems, methods, and challenges in comprehensive data cleansing*. Professoren des Inst. Für Informatik, 2005. [Online]. Available: [http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/techreports/2003-hub\\_ib\\_164-mueller.pdf](http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/techreports/2003-hub_ib_164-mueller.pdf)
- [40] M. A. Hernández and S. J. Stolfo, “Real-world data is dirty: Data cleansing and the merge/purge problem,” *Data mining and knowledge discovery*, vol. 2, no. 1, pp. 9–37, 1998. [Online]. Available: <http://www.springerlink.com/index/u63557414136mh1t.pdf>
- [41] Bridgiot. (2017) Bridgiot. [Online]. Available: <https://www.bridgiot.co.za/>
- [42] Build Apps with JavaScript | Meteor. [Online]. Available: <https://www.meteor.com/>
- [43] Node.js. [Online]. Available: <https://nodejs.org/en/>
- [44] MongoDB for GIANT Ideas | MongoDB. [Online]. Available: <https://www.mongodb.com/>
- [45] D. E. Robertson and J. J. Dowling, “Design and responses of Butterworth and critically damped digital filters,” *Journal of Electromyography and Kinesiology*, vol. 13, no. 6, pp. 569–573, Dec. 2003. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1050641103000804>
- [46] B. P. Lathi and Z. Ding, *Modern digital and analog communication systems*, international 4th ed ed., ser. The Oxford series in electrical and computer engineering. Oxford: Oxford Univ. Press, 2010, oCLC: 745577403.
- [47] S. Cass, “The 2017 top programming languages,” Jul 2017. [Online]. Available: <http://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>
- [48] E. DeBill, “Module counts,” Jul 2017. [Online]. Available: <http://www.modulecounts.com/>
- [49] A. B. M. Moniruzzaman and S. A. Hossain, “Nosql database: New era of databases for big data analytics-classification, characteristics and comparison,” *International Journal of Database Theory and Application*, vol. 6, no. 4, 2013. [Online]. Available: <https://arxiv.org/abs/1307.0191>
- [50] B. Lévesque, M. Lavoie, and J. Joly, “Residential water heater temperature: 49 or 60 degrees Celsius?” *Canadian Journal of Infectious Diseases and Medical Microbiology*, vol. 15, no. 1, pp. 11–12, 2004.
- [51] L. Breiman, “Statistical modeling: The two cultures (with comments and a rejoinder by the author),” *Statistical science*, vol. 16, no. 3, pp. 199–231, 2001.

- [52] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*, ser. Springer Texts in Statistics. New York, NY: Springer New York, 2011, doi: 10.1007/978-1-4419-7865-3. [Online]. Available: <http://link.springer.com/10.1007/978-1-4419-7865-3>
- [53] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*, ser. Wiley series in probability and statistics. Hoboken, N.J: Wiley-Interscience, 2008.
- [54] H. Best and C. Wolf, *The SAGE handbook of regression analysis and causal inference*. London: SAGE Publications, 2015.
- [55] A. Shabri, “Least square support vector machines as an alternative method in seasonal time series forecasting,” *Applied Mathematical Sciences*, vol. 9, pp. 6207–6216, 2015. [Online]. Available: <http://www.m-hikari.com/ams/ams-2015/ams-121-124-2015/58525.html>
- [56] G. Brys, M. Hubert, and A. Struyf, “A robustification of the Jarque-Bera test of normality,” in *COMPSTAT 2004 Symposium, Section: Robustness*, 2004.
- [57] S. Tang, Q. Huang, X.-Y. Li, and D. Wu, “Smoothing the energy consumption: Peak demand reduction in smart grid,” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1133–1141. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/6566904/>
- [58] V. M. Lo, “Heuristic algorithms for task assignment in distributed systems,” *IEEE Transactions on Computers*, vol. 37, no. 11, pp. 1384–1397, Nov. 1988.
- [59] T. L. Casavant and J. G. Kuhl, “A taxonomy of scheduling in general-purpose distributed computing systems,” *IEEE Transactions on Software Engineering*, vol. 14, no. 2, pp. 141–154, Feb. 1988.
- [60] A. Belov, A. Vasenev, P. J. M. Havinga, N. Meratnia, and B. J. v. d. Zwaag, “Reducing user discomfort in direct load control of domestic water heaters,” in *2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, Nov. 2015.
- [61] M. Shaad, A. Momeni, C. P. Diduch, M. Kaye, and L. Chang, “Parameter identification of thermal models for domestic electric water heaters in a direct load control program,” in *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Apr. 2012, pp. 1–5.
- [62] A. Afram and F. Janabi-Sharifi, “Review of modeling methods for HVAC systems,” *Applied Thermal Engineering*, vol. 67, no. 1-2, pp. 507–519, Jun. 2014. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1359431114002348>