Design of an autonomous energy management agent for a hybrid microgrid.

Prepared by:
Julian Banks (BNKJUL001)

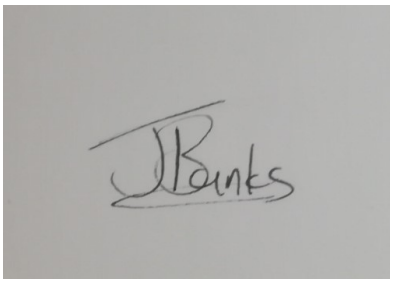
Prepared for:
Professor Sunetra Chowdhury
Department of Electrical Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Mechatronics Engineering

October 30, 2023

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.



October 30, 2023

Julian Banks

Date

Acknowledgements

> Why was the electrical engineer calm during the power outage? Because he knew how to conduct himself.
>
> —*unknown*

This research would not have been possible without the guidance and support of my supervisor, Professor Chowdhury. A special thanks to Mr Braune, who readily made data available for this study. To Patrick, thank you for your words of advice as I tentatively learned to navigate the world of statistics.

My journey over the last few years at UCT has been memorable, I would not have been infused with the passion for engineering that I now have without the many good times shared amongst friends. Furthermore, I would like to thank UCT for providing me with 4 years of excellent tuition.

To my wonderful Bells, your support has been unwavering throughout the last year. I am privileged to be able to experience your kindness, humour, intelligence and dedication.

Closer to home, thank you to my mum, you have always been the epitome of hard work and resilience while being able to find joy in the world. In the same vein, thank you to my two sisters who are both role models to me in the many aspects of life that they flourish. Many moments along the way, wanting to live up to you Dad, has been the motivation to tackle learning with enthusiasm.

Abstract

In light of the current global and local energy crisis, there has been significant exploration into renewable energy. Microgrids serve as a piece of the solution by integrating renewable generation on a modular scale. The term given to microgrids with both AC and DC subgrids is a hybrid microgrid.

This report explores the possibility of using an autonomous agent to perform energy management for such a microgrid, with the aim of reducing energy costs and increasing energy security.

The study first identifies suitable data to model a feasible hybrid microgrid and outlines the ensuing design process in HOMER Pro. The resulting hybrid microgrid incorporates photovoltaic (PV) and wind generation as well as featuring a diesel generator and an energy storage system (ESS) to counter the ongoing load shedding issues in South Africa.

Day-ahead forecasts are crucial for enabling the autonomous agent to plan and schedule energy use and purchases effectively. The load data for the hybrid microgrid was used to train three machine learning algorithms to perform these forecasts in 24 hourly time steps. The best-performing algorithm, a Seasonal Auto Regressive Integrated Moving Average (SARIMA) model, yielded a per time-step (over 24 steps) Mean Absolute Percentage Error (MAPE) of 16.83% for the AC load and 23.47% for the DC load.

These steps culminated in the design of an Reinforcement Learning (RL) environment that mirrors the parameters of the designed hybrid microgrid. This provided a platform to train an agent in energy management. The RL environment simulated the microgrid and trained the agent by rewarding actions that reduced energy costs. The training was a success, reducing energy costs by 5.38% on a stable utility grid. Remarkably, the agent performed even better on an unstable grid experiencing load shedding, reducing total energy costs by 6.12% and eliminating unmet demand.

Overall, the utilisation of autonomous agents for energy management succeeded in reducing energy costs and improving energy security. While this process has yet to be tested in a real-world scenario, the results provide a successful framework for further development and implementation.

Contents

Abbreviations **ix**

1 Introduction **1**
1.1 Background to the study . . . 1
1.2 Objectives of this study . . . 1
1.2.1 Tasks to be investigated . . . 1
1.2.2 Purpose of the study . . . 2
1.3 Scope and limitations . . . 2
1.4 Plan of development . . . 3

2 Literature review **4**
2.1 Hybrid microgrids . . . 4
2.2 Energy management techniques . . . 6
2.2.1 Local control . . . 7
2.2.2 Supervisory control . . . 7
2.2.3 Energy scheduling . . . 8
2.3 Application of machine learning to energy management . . . 9
2.3.1 Machine learning as a tool for prediction . . . 9
2.3.2 Machine learning as a tool for scheduling . . . 10
2.4 Conclusion . . . 11

3 Theory **12**
3.1 Machine learning theory for forecasting . . . 12
3.1.1 Analysis of potential regression techniques . . . 12
3.1.2 Auto-regressive models . . . 13
3.1.3 Neural Networks . . . 14
3.1.4 Long Short Term Memory Networks . . . 15
3.2 Reinforcement learning theory . . . 15
3.3 Machine learning software . . . 17
3.3.1 MATLAB . . . 17
3.3.2 Python . . . 17

4 Data collection **18**
4.1 AC and DC load data . . . 18
4.1.1 Cleaning load data . . . 19
4.2 Time of use rates . . . 19
4.3 Load shedding 2022 schedule . . . 21

5 Microgrid design **22**
5.1 Hybrid microgrid design . . . 22
5.1.1 Homer PRO Input . . . 23
5.1.2 Electrical inputs . . . 23
5.1.3 Economic factors . . . 23
5.1.4 Homer PRO output . . . 24
5.2 Final grid parameters . . . 25
5.3 PV and wind generation data . . . 26
5.3.1 Simulated generation data . . . 26
5.3.2 Generation forecasting . . . 26

6 Load forecasting **28**
6.1 Exploratory data analysis . . . 28
6.1.1 Feature Selection . . . 30
6.2 Fitting LSTM model . . . 31
6.2.1 Data Preparation . . . 32
6.2.2 Normalisation . . . 32
6.2.3 LSTM configuration and training . . . 32
6.2.4 Forecasting method . . . 33
6.3 Fitting ARIMA model . . . 35
6.3.1 Data Preparation . . . 35
6.3.2 Determining Model Parameters . . . 35
6.3.3 Training the model . . . 36
6.3.4 Forecasting method . . . 36
6.4 Fitting SARIMA model . . . 36
6.4.1 Determining seasonal parameters . . . 37
6.4.2 Training Model . . . 38
6.4.3 Forecasting method . . . 38
6.5 Fitting ANN model . . . 39
6.5.1 Data preparation . . . 39
6.5.2 Training model . . . 39
6.5.3 Forecasting method . . . 39

7 Load forecasting results and discussion **41**
7.1 Results . . . 41
7.1.1 Metrics of evaluation . . . 41
7.1.2 Forecasting procedure . . . 41
7.1.3 Evaluation of the LSTM model . . . 42
7.1.4 Evaluation of the SARIMA model . . . 43
7.1.5 Evaluation of the Neural Net model . . . 44
7.2 Load forecasting discussion . . . 44
7.2.1 Model performance . . . 44
7.2.2 AC and DC load variablity . . . 45
7.2.3 Forecast stability . . . 45

7.2.4 Final load forecasting model . . . 46

8 Energy management agent **47**
8.1 Reinforcement learning applied to EMS . . . 47
8.2 Simplified environment . . . 48
8.2.1 Environment . . . 48
8.2.2 Results of EMSv0.1 . . . 51
8.3 Development of the environment and agent . . . 51
8.3.1 Implementation of a hybrid grid and load shedding . . . 52
8.3.2 Implementation of a demand charge and continuous action space. . . . 53
8.4 Final RL implementation . . . 53
8.4.1 Environment . . . 53
8.4.2 Observation space . . . 54
8.4.3 Action space . . . 55
8.4.4 Reward . . . 55
8.4.5 Step function . . . 56
8.4.6 Training method . . . 57

9 Results & Discussion **58**
9.1 Evaluation metrics . . . 58
9.2 Results . . . 58
9.2.1 Performance of the agent with no load shedding . . . 59
9.2.2 Performance of the agent with load shedding . . . 59
9.3 Discussion . . . 61
9.3.1 Hybrid microgrid design . . . 61
9.3.2 Load forecasts . . . 61
9.3.3 Integration into an RL environment . . . 62
9.3.4 Efficacy of the final EMS agent . . . 62
9.3.5 Behavioural insights . . . 62
9.3.6 Brief exploration of generation forecasts . . . 64

10 Conclusions and Recommendations **65**
10.1 Conclusion . . . 65
10.2 Recommendations . . . 66
10.2.1 Data collection . . . 66
10.2.2 Generation forecasts . . . 66
10.2.3 Further training & tuning . . . 66
10.2.4 RL environmental complexity . . . 66
10.2.5 Integration . . . 67

Bibliography **68**

A Pricing for design of hybrid microgrid **72**

B Additional environment information **75**

C Ethics Forms **78**

Abbreviations

ACF Auto Correlation Function

ANN Artificial Neural Network

AR Auto Regressive

ARIMA Auto Regressive Integrated Moving Average

AutoML Automated Machine Learning

CEEMDAN Complete Ensemble Mode Decomposition with Adaptive Noise

CNN Convolutional Neural Network

DDPG Deep Deterministic Policy Gradients

EMS Energy Management System

ESS energy storage system

GRU Gated Recurrent Unit

HESS Hybrid Energy Storage System

IMF Intrinsic Mode Functions

KPSS Kwiatkowski–Phillips–Schmidt–Shin

LSTM Long Short Term Memory

MAPE Mean Absolute Percentage Error

MDP Markov Decision Process

ML machine learning

MPPT maximum power point tracking

NPC Net Present Cost

PACF Partial Auto Correlation Function

PER Prioritized Experience Replay

PID Proportional, Integral and Derivative

PPO Proximal Policy Optimisation

PV photovoltaic

ReLe Rectified Linear Unit

RL Reinforcement Learning

RMSE Root Mean Square Error

RNN Recurrent Neural Network

SAC Soft Actor Critic

SARIMA Seasonal Auto Regressive Integrated Moving Average

SCADA Supervisory Control and Data Acquisition

TOU Time of Use

WNN Wavelet Neural Network

Chapter 1

Introduction

1.1 Background to the study

This study integrates two critical domains: renewable energy, particularly focusing on hybrid microgrids, and machine learning, especially in the realm of autonomous agents. The rising climate crisis has accelerated the adoption of renewable energy sources. A key element in this integration is the concept of microgrids. Typically comprising various forms of distributed generation, one or more loads, and energy storage system (ESS), microgrids have the flexibility to operate in either islanded or grid-connected modes. An Energy Management System (EMS) is generally used to regulated the power flow among these components and the grid.

Optimising the EMS could lead to more sustainable energy penetration, enhanced power quality, and cost savings. This is where machine learning and, more specifically, Reinforcement Learning (RL) come into play. Advances in machine learning have made autonomous agents increasingly capable. By combining the advances in machine learning to forecast future values of time series, it is possible that an RL agent could learn to take predictive actions within a complex environment.

In the context of this study and energy management at large, if an autonomous agent could accurately predict future load values and learn to manage energy effectively based on these predictions, the benefits for the microgrid could be substantial.

1.2 Objectives of this study

1.2.1 Tasks to be investigated

This study sets out to determine if an autonomous agent can be trained to make load predictions and then use these predictions to perform energy management. The success of the agent will be measured by its ability to reduce operational costs and increase energy security.

The study has four key tasks which were derived from insights gained during the literature review. Each key task is detailed in its own chapter.
The framework is structured as follows.

1. **Data collection:** This crucial step outlines the process of finding the necessary data to model the hybrid microgrid.

2. **Microgrid design method:** Once the data is collected, this section details how HOMER Pro

will be utilised to design the hybrid microgrid.

3. **Load forecasting approach:** This section delves into machine learning theory, particularly focusing on time series forecasting, to inform the approach to load forecasting. The goal is to explore and discern the most suitable models for accurate load forecasting.

4. **Autonomous energy management system:** The culmination of the methodology is determining the best reinforcement learning approach to train an autonomous energy management agent. This process will make use of the designed microgrid, the load and generation forecasts.

1.2.2 Purpose of the study

Amid the prevailing energy crisis, there's a pressing need for the widespread adoption of renewable energy solutions. This study aims to explore the potential benefits of incorporating an autonomous energy management agent, equipped with machine learning generated load forecasts, into a hybrid microgrid.

The primary objective of this study is to test whether such an agent can reduce energy costs and minimise unmet demand.

Achieving this objective will underscore the tangible potential for such systems to cut operational expenses, thereby rendering renewable energy microgrids more economically attractive.

1.3 Scope and limitations

In order to understand the scope of the study as a whole, it is necessary to understand the scope of each task:

1. **Data collection:** The data was set to be collected from a single building on UCT upper campus. Any necessary data that was not available would be simulated or sourced from an open-source database. The data's temporal span and resolution were set to cover one year, sampled at hourly intervals.

2. **Microgrid design:** The hybrid microgrid's design must meet the project brief. Specifically, it must include an AC and DC load, and AC and DC form of generation, the total capacity must be less than 1MW and it must be able to operate in islanded mode in the event of grid instability.

3. **Load forecasting:** Load forecasting must be accomplished using machine learning techniques and should provide forecasts up to 24 time steps ahead, sufficient for day-ahead planning.

4. **Generation forecasting:** While methods for generation forecasting should be outlined, training and testing an additional set of machine learning algorithms for this purpose would extend the project's scope excessively.

5. **Autonomous energy management:** This scope includes the design and creation of a reinforcement learning environment matching the designed hybrid microgrid's parameters. It also involves accounting for grid factors like load shedding, Time of Use (TOU) rates, demand charges, as well as the training and testing of an autonomous energy management agent.

1.4 Plan of development

The report begins by developing an understanding of microgrids and EMS's through a literature review. It then goes on to develop theory for the necessary machine learning techniques that will be employed.

The report begins with a literature review, laying the foundation by discussing microgrids and EMSs. Following this, the theory underlying the machine learning techniques to be used is explored.

Insights gained in the literature review, informing the required data, will guide the data collection and cleaning process. This data will, in turn, inform the design of the hybrid microgrid. After finalising the necessary parameters and data for the microgrid, various machine learning models will be trained using the associated load data. Additionally, the HOMER Pro model of the hybrid microgrid will be used to simulate generation values that can be used in the RL environment. The most effective forecasting model will then be chosen to provide the autonomous agent with day-ahead forecasts.

The final phase involves designing the RL environment to match the parameters of the hybrid microgrid. Given the multifaceted nature of simulating the hybrid microgrid as an RL environment, this process will be undertaken iteratively. That is, each iteration of the RL environment will incorporate an additional feature of the designed microgrid until it mirrors the required complexity. Finally, an autonomous agent will be trained and tested within the RL environment.

Chapter 2

Literature review

There has been extensive research into hybrid microgrids, particularly focusing on their configuration, optimisation and interconnectivity with utility grids.

This literature review focuses on three key topics which, when combined, show the potential to form a novel avenue for the advancement of hybrid microgrid technology.

First, it explores the configurations of hybrid microgrids, in the context of this paper referring to microgrids that have both AC and DC subgrids[1]. Second, it examines contemporary energy management techniques that have been applied to such microgrids, delving into the different levels of energy management. Third, it investigates the application of machine learning for 24-step day-ahead load and generation forecasting, a critical aspect in the effective management of microgrids. As well as exploring machine learning methods used to perform energy scheduling.

2.1 Hybrid microgrids

Microgrids have gained traction as new forms of renewable energy generation have been developed and political pressure to decarbonise energy has increased. Microgrids are typically low-voltage networks employing distributed generation to meet local demand. Typical forms of generation include solar photovoltaic (PV), wind and diesel. Microgrids are usually equipped to operate in islanded or grid-tied modes. If they are required to reliably provide power to a load during islanded modes it is also necessary to include energy storage systems in the design.

Microgrids often use diesel generators to ensure energy security. This is due to the intermittent nature of popular renewables such as solar and wind. Additionally, it is costly to install an energy management system that is large enough to ensure energy security in the event of the microgrid operating in islanded mode. Diesel is a potent fossil fuel and an unattractive option in an energy system for the future. Alternatives such as bioenergy that implement a waste-to-energy cycle using biomass or bio-waste have been proposed. These have been excluded from this study as the composition of the hybrid microgrid is not the main focus, however, it would be interesting to explore their integration into microgrids in the future[2].

The increasing number of DC loads, driven by consumer electronics and electric vehicles, combined with the inherent DC characteristics of renewables like PV, presents a compelling case for including DC subgrids into hybrid microgrids. Other advantages of DC subgrids include being free from frequency, power factor and phase sequence issues[3].

A hybrid microgrid, comprising both AC and DC subgrids, offers a platform for reducing the number of conversion stages when both AC and DC generation sources are integrated. Naturally, this is a result of the AC generation predominantly catering for the AC load and the DC generation primarily meeting the DC load. In turn, the reduction in power conversion steps leads to a reduction in power conversion losses[4].

Bidirectional converters can be used to control the flow of active power between the AC and DC subgrid. This results in the behaviour that one subgrid can become a load or a source of generation while the other subgrid becomes the inverse. The result is that only the power imbalance between the subgrids needs to pass through conversion stages[5].

Within the literature, there are various configurations and topologies of hybrid microgrids. The essence of the two most common types are as follows:

- **Coupled AC hybrid microgrids:** In this configuration, the AC subgrid is connected to the utility grid by transformers. The DC subgrid is consequently connected to the AC subgrid via power converters[6].
- **Decoupled AC hybrid microgrids:** Contrarily, in this setup, the DC subgrid is connected directly to the utility grid via power converters. The AC subgrid then links to the DC subgrid[6].

The main advantage of a couple AC hybrid microgrid is that there are fewer power conversion components as only a transformer is required to connect the subgrid to the utility grid. Additionally, when the AC network is operating in grid-connected mode, the voltage and frequency are tied to that of the grid. This decreases the need for vigorous frequency control etc.

The main advantage of the decoupled AC configuration comes from the inherent control provided by the power converters between the utility grid and the microgrid. They allow for accurate tracking of power flow and could make fault detection easier[6]. However, the converse side of this is that they have been found to require more complex circuitry in order to integrate with an AC grid [7].

It is found that coupled AC configurations are more common, as they can form a convenient and reliable system with comparative ease[7]. An example of a coupled AC subgrid can be seen in figure 2.1. Notably, the transformer provides the microgrid with galvanic isolation. However, the transformer may be subjected to power surges if both the AC and the DC subgrids experience a deficit in power generation.

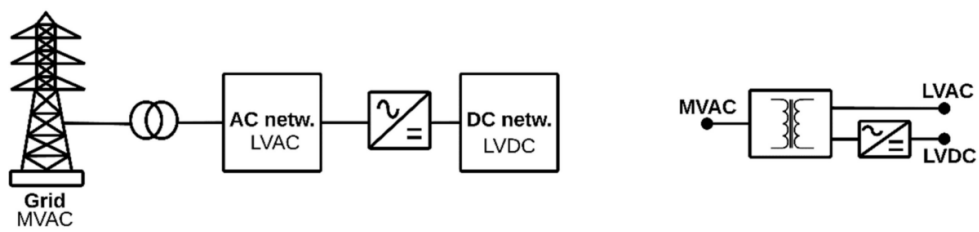


Figure 2.1: Coupled AC microgrid in completely isolated configuration. (source: [8])

None of these configurations have explicitly included an ESS. Renewables are intermittent by nature

and, in a South African context, load shedding is an eventuality. Due to this an ESS is considered essential to secure a microgrid's energy security. An ESS is responsible for storing energy during periods of excess generation, allowing it to supplement the energy supply during periods of generation shortage. This typically requires a bi-directional DC-DC converter to facilitate energy flow in the DC subgrid [9]. The development of ESS is currently a very active space in research, however, the accepted industry standard is currently the lithium-ion battery due to its ability to withstand a high number of cycles, fast charging and discharging rates and high depth of discharge[10].

It is worth noting that the composition and design of microgrids are based on too many factors for a universal approach to be viable. Each design needs to be tuned for its environment and application-related parameters. In [11], this is handled by employing a 2 layered multi-criteria decision-making model. This first calculates the weights of each criterion and then uses ranking methods to determine the optimum compromise. HOMER Pro software was employed to test simulations of various configurations.

From the available literature, the usage of HOMER Pro to design microgrids has been widely accepted. Common sources of DC generation such as solar PV can be used to meet DC loads and to charge DC ESS such as lithium-ion batteries. Wind turbines can be configured to provide AC or DC power depending on their design and construction, making them a flexible option for the design of a hybrid microgrid. Diesel generators are commonly used to provide power security and produce AC power. While there are various topologies available for hybrid microgrids, the most general one appears to be the coupled AC configuration.

2.2 Energy management techniques

A crucial element of any microgrid is its control and EMS. The energy flow between the distributed generation and the loads, the flow of energy between the AC and the DC subgrids, the management of energy storage systems and the switching between islanded and grid-tied operations are just some of the necessary control systems. Additionally, the implemented control system needs to be flexible and able to handle any number of operating points due to the unreliable nature of renewables[12].

In [9] the control system is split into three parts, the energy scheduling system, the supervisory control and the local control.

- The energy scheduling system uses a day-ahead schedule to organise power generation. This comes from a prediction of weather patterns and load profiles.
- The supervisory control manages the discrepancy between the scheduled power and the actual power used and generated.
- The local control system manages the voltage, current, and frequency of the microgrid.

The results showed that the combined control system was effective at reducing costs and increasing the reliability of the hybrid microgrid.

These three segments can be used as a framework to loosely classify various methodologies and applications of energy management. This section delves into each category, aiming to inform the best approach for energy management in the context of this study.

2.2.1 Local control

The proposed system in [9] includes solar PV and hydrogen fuel cells, the local control of each of these systems is performed by a Proportional, Integral and Derivative (PID) controller. The fuel cell setup uses a reference voltage and increases the generation whenever the voltage at the terminals falls below the reference voltage and vice versa. The solar PV system uses maximum power point tracking (MPPT) to optimise generation from the solar PV panels and the output of the MPPT is used as the input to the PID loop.

Droop control is a well-established form of control in AC microgrids. In [5] its use was extended to control a hybrid microgrid with both AC and DC subgrids. This was achieved by adjusting existing equations to fit the parameters of a DC grid. It is worth noting that droop control is a low-level form of control that focuses on sharing load proportionally among forms of generation in real-time, rather than on planned energy management.

2.2.2 Supervisory control

In [13], the energy management system is implemented logically by using a set of conditions. The conditions include production from the microgrid, the price of grid electricity, the load and the state of charge of the energy storage system. It is assumed that all of the loads and generation values are known at all times. This is a high-level form of control where the main priority is to keep the flow of power balanced between the loads and the sources. A similar rule-based control scheme from another paper,[9], can be visualised using a flow chart, an example is given in figure 2.2. The scheme was successful in balancing the loads and improving the performance of the system theoretically. However, it does not address how the actual control will be implemented, the high level of information that all of the conditions require implies that the system will require many sensors and a centralised control unit. These are factors that greatly affect the cost and reliability of the system.

This issue of reducing the number of sensors required to monitor load and generation metrics is tackled in [14]. The approach focuses on the voltage of the DC link. Control is performed by measuring the voltage of the DC link and using PI controllers to keep the voltage at a constant reference voltage. This is done by injecting power or drawing more power (selling it to the grid or storing it). A rule-based control scheme is still implemented at this stage. However, the only inputs required from sensors are current through the DC-link (derived from the voltage), and the battery state of charge. Thus, the number of required sensors is reduced. In [14], supercapacitors are included to help smooth the transients and reduce the stress on the batteries. An important consideration that [14] brings up is the operation of the hybrid in off-MPPT mode. Typically MPPT is used to draw the maximum power from a PV system. However, if generation exceeds the load, the energy storage system is fully charged and the microgrid is in islanded mode then there is nowhere for the excess power to go and the PV system should not track the maximum power point.

Similar to a rule-based scheme, [15] uses states to implement supervisory control. In this case, every combination of loads and generation is assigned to a state. The controller then decides what state to put the system into. The states have entry conditions that are based on the available power of the sources and the required powers of the loads. The states have been chosen to maximise the renewable penetration and a diesel generator is only used to prevent load shedding if the wind and PV generation

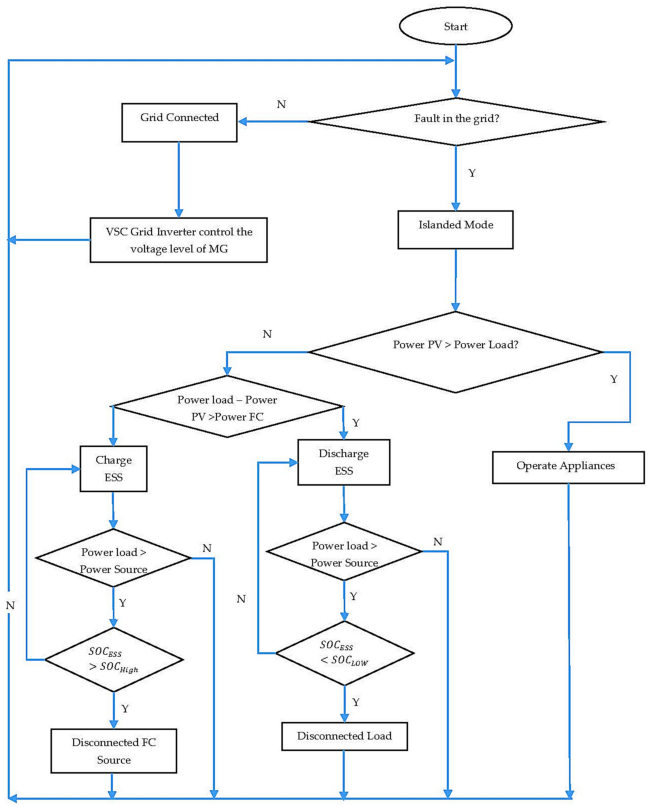


Figure 2.2: A simple rule based scheme for energy management [9].

cannot meet the demand and the batteries are fully discharged. The supervisory control is shown to operate as expected and successfully change states to match loads. However, the performance is not compared to a benchmark so it is difficult to judge how effective the energy management system truly was. An important practical note that [15] brings up, is the need to ensure impedance and voltage matching of converters. For example, if the DC subgrid is connected to the PV panels via a DC/DC converter and to the AC subgrid via an AC/DC converter then any mismatch between these converters will negatively affect power quality.

2.2.3 Energy scheduling

Matching controllable loads to peaks in power generation was explored in [16]. This was done by using predictions of the upcoming wind patterns and moving a controllable load to be connected during periods of peak production. An example of this could be scheduling the ESS to be charged at a point of peak power production based on a weather prediction. The study found that the method showed an increase in the state of charge of the energy storage system as well as a decrease in the usage of the supplementary diesel generator. However, this method relies on accurate forecasting of weather.

The EMS that [17] uses is for a simple microgrid. The researchers approach the problem from an angle of optimisation and use integer linear programming to determine the optimal method of control. The microgrid in question contains a PV system, a wind turbine and an ESS of lead-acid batteries. The EMS uses weather and load profile predictions along with grid pricing of electricity to schedule the state of charge of the batteries and to minimise the objective function. The objective function is designed to minimise operating costs. For cases when the load was met purely from the microgrid's

generation, it was found that using MPPT controllers was slightly more effective than the optimised EMS. However, in cases when the microgrid was required to buy electricity from the grid the optimised EMS outperformed the MPPT controllers as it was able to schedule the purchase of electricity during off-peak periods at a lower price.

In summarising this section, it is critical to examine the literature within the framework of this study, particularly with regard to the application of machine learning to energy management.
Machine learning excels in areas such as high-level decision-making, understanding complex relationships, and making accurate predictions. These capabilities distinguish it from low-level control techniques such as droop control, thereby shifting the focus toward supervisory and scheduling control methods. As shown in multiple studies ([13],[15], [17]) employing high-level supervisory energy management is beneficial for both reducing energy costs and increasing energy security. This aligns with the objective of this study.

When machine learning techniques used for prediction and forecasting are coupled with techniques for training autonomous agents, the result could provide a potent combined form of supervisory and scheduling control. This potential is further explored in the subsequent section.

2.3 Application of machine learning to energy management

Drawing on the insights gained from the previous section, this section delves deeper into two primary areas. Firstly, the use of machine learning in generating forecasts is explored. Secondly, the application of machine learning in training autonomous energy management agents is examined.

2.3.1 Machine learning as a tool for prediction

Renewable energy sources like wind and PV systems, are heavily influenced by weather conditions. This makes the ability to accurately forecast both generation capacity and future load values key for efficient energy management. Machine learning techniques have emerged as a promising tool for enhancing the precision of these forecasts, representing a growing field of study [18].

In [19], researchers use weather data in the form of temperature, humidity, air pressure, and air density collected at the site of renewable energy sources. It is interesting that they have chosen to measure air pressure and density rather than air speed. The collected data was used by the Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) algorithms to predict weather patterns. After the method was refined and an attention mechanism was added to the network the best result achieved was 17.99% Mean Absolute Percentage Error (MAPE) for a day ahead prediction.

The work done in [19] goes on to use a combination of algorithms. A Complete Ensemble Mode Decomposition with Adaptive Noise (CEEMDAN) and a GRU were trained to predict the day-ahead electrical load. An interesting property of the CEEMDAN network is that it can break the data into Intrinsic Mode Functions (IMF)s. This allows the model to identify the slower-moving and dominant IMFs. These are easier to predict and help to reduce the prediction error. As a result, the MAPE error was lowered to only 2.88%. The paper is unclear, but this appears to be the error when predicting one time step ahead rather than a full 24-step day-ahead prediction.

In [20] a similar combination of CNN and GRU was used to forecast price, wind speed and solar irradiation. However, much more accurate results were reported with MAPE values as low as 3.6% for wind speed and 5.47% for solar irradiance. However the researchers have not been clear about the forecast period, the forecast seems to only be one hour ahead which would explain the low MAPE values.

A feature that machine learning has shown promise in is the prediction of load profiles. If it is possible to accurately perform next day load prediction it will be possible to implement an informed EMS. In [21], this idea is explored and showed promising results. Various combinations of Artificial Neural Network (ANN), Wavelet Neural Network (WNN) and Kalman Filters are used to test the accuracy of load forecasting using neural networks. It was found that using a WNN and then feeding the results into an ANN in a hybrid neural network produced the best results. Given the necessity for an intelligent grid to be able to accurately schedule production, accurate predictions of loads will be incredibly valuable.

An important argument is raised by [22], supervised learning can provide a high level of accuracy and can be tempting when research is carried out with a defined model and set of data. However, this will create models that are case-specific and not easily adaptable to the real world, unless of course a huge amount of data can be accessed. In [22] the EMS is split into two main tasks, prediction of generation and load and generation scheduling. The day ahead prediction of the energy generation and load is performed by Automated Machine Learning (AutoML) combined with Prioritized Experience Replay (PER). The results boast a high level of accuracy with a MAPE of 8% for the PV prediction, 7% for the wind prediction and 1.8% for the load prediction. The paper also goes on to use linear programming to find the optimal scheduling scheme for the system once the predictions have been made.

2.3.2 Machine learning as a tool for scheduling

In [23] the energy management problem is simplified to finding the best combination of energy sources that should meet the current load. The scheduling predictions were simplified further to be based on 4 inputs, namely: the hourly load demand, the temperature and two binary values for the availability of sun and wind. This accounts for varying loads but does not account for varying levels of wind or solar irradance as both factors have been simplified to a boolean. The output of the neural network was divided into classes. Where each class represented a different configuration of active energy sources. Four neural net algorithms were tested. All of the algorithms were found to perform well however the best one in terms of accuracy and performance was the Decision Tree algorithm.

The scheduling challenge is optimised in [24] by transforming the problem into a Markov Decision Process (MDP). A Q-learning algorithm is then chosen due to its suitability for handling MDP problems to tackle the issue of energy storage management. The paper found varying results with the six variations of the algorithm that it tested. Unfortunately, none of them could match the theoretical limit that was calculated by the mixed integer linear programming method. However, they were able to converge on their result more quickly and were thus computationally cheaper.

In [25], RL techniques were used to perform energy management on a grid-tied microgrid with PV generation and a battery. A simulated commercial load profile and PV profile were used to provide a

24-hour ahead forecast. By giving the EMS agent the ability to purchase and store energy it was able to decrease operational costs by 4.033% during summer seasons and 2.199% during winter seasons. Notably, these operational costs incorporated battery degradation as well as energy costs.

Similarly, in [26], an agent is trained through RL to reduce power consumption from the grid. Three-step-ahead Q-learning is used to improve battery scheduling with a dynamic load and solar PV capacity. The agent is provided with forecasts by introducing uncertainty into the solar and load data.

An isolated microgrid with hybrid generation explores RL control in [27]. But focuses on using the control for optimal sizing rather than for energy management. In another paper[28], a hybrid microgrid employs RL methods to perform energy management on a Hybrid Energy Storage System (HESS), but it does not focus on reducing energy costs but rather improving power quality and system performance by regulating charging and discharging.

From this exploration, it becomes evident that machine learning offers a diverse and effective array of approaches for both load and generation forecasting. RL stands out as a particularly promising avenue for incorporating the real-world complexities often faced in energy management as seen in [25]. For example, RL can overcome the simplifications that are sometimes made by classification or regression models, such as reducing solar availability to a boolean, as seen in [23]. However, a gap in the existing literature appears to be the lack of integration of machine learning forecasting techniques into scheduling tasks in studies where RL is employed. Moreover, there are relatively few examples where RL environments have been specifically applied to hybrid microgrids for energy management.

2.4 Conclusion

The topic of hybrid microgrids has been produced in many research papers with varying approaches. The configuration of each microgrid will vary depending on the environment meaning that solutions have to be adaptable and fitted on a use-case basis. However, there are some universal trends, such as the need for energy storage and technologies like wind and solar generation.

Moreover, energy management in microgrids provides an opportunity to reduce the operating cost of a microgrid while increasing energy security [16]. There are many contemporary methods that work at varying levels, from high-level supervisory control, [9, 13], to low-level control methods such as droop control[5].

With this context, machine learning has been widely used to provide EMS's with load and generation forecasts[19, 20]. RL, a subset of machine learning, has shown promise in training autonomous agents to manage energy efficiently, consequently reducing energy costs[25].

However, to the best of the researcher's knowledge, an unexplored avenue exists in the integration of machine learning-based methods for load and generation forecasting into a RL environment. This is particularly true when considering an environment specifically designed for hybrid microgrids. Such integration could potentially lead to more realistic autonomous EMS's.

Chapter 3

Theory

This chapter delves into two domains of machine learning, informed by insights gleaned from the literature review. The first domain concerns the utilisation of machine learning for forecasting time series data, while the second focuses on leveraging RL to simulate and train an agent, enabling it to adeptly navigate intricate environments.

Subsequently, the chapter covers some fundamental concepts related to data processing and introduces the software tools employed throughout the research.

3.1 Machine learning theory for forecasting

From the reviewed literature, generating accurate load forecasts emerged as a significant challenge in energy management. At the heart of this challenge lies the task of predicting future values within a time series. Such time series analyses and predictions commonly employ regression-based learners.

3.1.1 Analysis of potential regression techniques

Regression models employ a supervised learning approach, using historical data pairs, consisting of predictors and corresponding targets, to train the model. The essence of supervised learning is to discern a mapping from input features to a target variable, aiming to learn relationships that enable predictions on new, unseen data.

In essence, regression models can be designed to predict continuous numerical outcomes. In continuous models, this prediction manifests as a line or curve that best fits the data. The efficacy of a trained regression model is typically assessed through metrics such as variance, bias, and error. A high variance may indicate overfitting, capturing noise rather than underlying data patterns, while a high bias may indicate underfitting, where the model's predictions are consistently wrong [29].

There are many variant forms of regression such as linear, polynomial, multiple and multivariate regression.
For data that doesn't fit a linear trend, polynomial regression is employed. It plots curves to capture nonlinear trends in data. It's imperative to tune the degree of the polynomial to avoid overfitting.
Multiple regression uses several predictors to determine a single outcome while multivariate regression facilitates the prediction of multiple outputs.

3.1.2 Auto-regressive models

Auto Regressive (AR) models are prevalent time series forecasting models that work by capturing and fitting different aspects of temporal data patterns. The key feature of a pure AR model is that the future values of a time series depend only on its past values (known as lags).

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + e \tag{3.1}$$

In equation 3.1, Y_{t-1} is the first lag, β_1 denotes the coefficient of the first lag term. The model's intercept term is α and e is the error term. The parameter 'p' establishes the order of the AR model, indicating the number of lagged values used as predictors (how many past values contribute to the future value) [30].

AR models serve as building blocks for more detailed models such as Auto Regressive Integrated Moving Average (ARIMA) and Seasonal Auto Regressive Integrated Moving Average (SARIMA).

3.1.2.1 ARIMA

ARIMA enriches the AR model by integrating two pivotal features. First, it employs differencing to render the time series stationary. Differencing involves computing the difference between consecutive data points, effectively eliminating trends and ensuring constant mean over time. Second, ARIMA incorporates a moving average (MA) component, making the model responsive to past forecasting errors. Consequently, ARIMA excels in modelling non-seasonal time series that exhibit discernible patterns.

An ARIMA model combines the equations of AR and MA as seen in equation 3.2:

$$Y_t = \alpha + +\beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + e_t + \phi_1 e_{t-1} + \phi_2 e_{t-2} + ... + \phi_q e_{t-q} \tag{3.2}$$

Equation 3.2 extends equation 3.1 by adding terms with $\phi_i e_{t-i}$ to characterise the MA component. Each error term, e_{t-i}, corresponds to the error from its respective lag term, and ϕ signifies the coefficient for that term. The parameter 'q' dictates the number of error terms incorporated.

For achieving stationarity, a time series might need to be differenced. It's important to apply the minimal differencing to ensure near-stationarity, characterised by a constant mean and an Auto Correlation Function (ACF) plot that approaches zero within a reasonable number of lags.
If many of the lag terms show a positive autocorrelation then the series should be differenced further. Conversely, if the first lag shows a negative autocorrelation then the series may be over-differenced [31].

The order 'p' for the AR component can be ascertained from the Partial Auto Correlation Function (PACF). A standard initial estimate sets 'p' equal to the number of lags surpassing the significance threshold in the PACF plot. Analogously, the MA order 'q' can be gleaned from the ACF, setting it equivalent to the significant lags in the ACF [32].

After initial parameterisation, it's often necessary to refine these terms. Techniques such as a grid search are commonly employed to ensure that the best parameters have been found.

To summarise, an ARIMA model, tailored for non-seasonal time series forecasting, is delineated by three parameters:

- 'p': Order of the AR component.
- 'q': Order of the MA component.
- 'd': Number of differencing iterations to achieve stationarity.

3.1.2.2 SARIMA

One shortcoming of the ARIMA model is its inability to account for inherent seasonality. The SARIMA model accounts for this by including another set of seasonal parameters into the equation [33].

- P (Seasonal Auto Regression Order): Like its non-seasonal counterpart, the seasonal order P can be estimated from the PACF. However, instead of looking at the immediate lags, the lags at the seasonal frequency are inspected. For instance, for monthly data with yearly seasonality, one would inspect the PACF value of lag 12.
- Q (Seasonal Moving Average Order): Analogous to the non-seasonal q order, Q can be estimated from the ACF, except that the AFC values are examined at the seasonal lags.
- D (Seasonal Differencing Order): This is the number of seasonal differences required to make the series stationary. This entails differencing a data point with a point from a previous season rather than an immediately preceding term.
- Seasonality: This denotes how many steps make up a season. Typically for monthly data, it would be every 12 steps, while for hourly data it may be every 24 steps to represent the daily seasonality.

3.1.3 Neural Networks

Neural networks are computational algorithms loosely inspired by the structure of the human brain. They consist of a layered structure of interconnected nodes. A node will receive inputs from the previous layer of nodes (or the input layer), multiply each input by the node's weights, add the bias term, and pass the summation through an activation function to an output [34].

This process can be visualised in figure 3.2 where $f()$ is the activation function. The summation has a bias term which is essentially a constant that is added to the summation of the weights and inputs. The activation function is most easily visualised by the Rectified Linear Unit (ReLe) function as seen in figure 3.1. When using the ReLu function, if the input is less than zero the output is zero, else the output is simply the input.

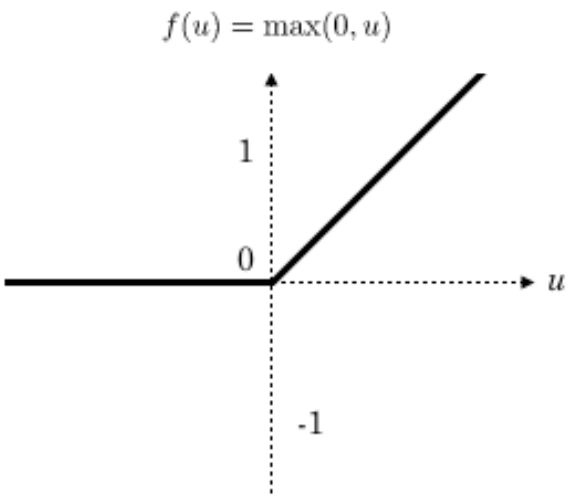


Figure 3.1: Relu activation function [35]

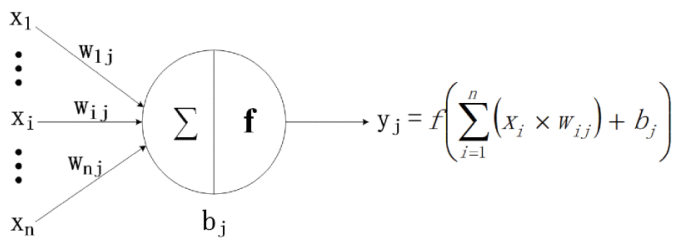


Figure 3.2: Representation of a node in a neural network[34].

During training the weights and biases of each node are gradually changed as it learns the mapping from input to output. The number of neurons per layer and the number of layers is entirely configurable, as is the number of connections between layers. This basic design forms the basis of many other networks.

3.1.4 Long Short Term Memory Networks

Long Short Term Memory (LSTM) networks, a variant of Recurrent Neural Network (RNN), excel at time series analysis due to their ability to capture long-range temporal dependencies [36]. Like other neural networks, they consist of input layers, hidden layers with neurons, and output layers.

What sets LSTM networks apart is their memory block, which uses its cell state to store information over extended sequences. This memory block's functioning is overseen by a system of gates [37]:

- **Forget Gate**: Using the current input and the previous block's output, it determines how much of the past information (from the cell state) is kept in the current cell state.
- **Input Gate**: Decides which values in the cell state need updating and then computes the new values that should be added to the state.
- **Output Gate**: Based on the current cell state and the input, it determines the information to be sent as output to the next cell.

In this manner, the LSTM adjusts weights and biases not only to predict data but also to learn how to propagate data [38]. This structure ensures that LSTM networks can learn to remember and forget information, making them highly effective for tasks where contextual information spanning long sequences is essential.

3.2 Reinforcement learning theory

Reinforcement learning has several notable differences compared to other machine learning algorithms. In supervised learning methods, the data and the labelled targets are fed into the training model in order to learn some mapping. Where as in reinforcement learning the agent interacts with an

environment and gathers its own data through an iterative process.

The core idea entails defining an environment for the agent to act in. The agent then makes an observation of the environment and can choose to take an action. The action is then applied to the environment. If it is a favourable action the agent can earn a reward, if it is not then it could be penalised. Such an interaction is visualised in figure 3.3. This interaction is repeated until the end of an episode. An episode can be ended by the agent reaching a goal(succeeding), meeting a termination requirement (failing), or reaching the maximum number of time steps (truncated).

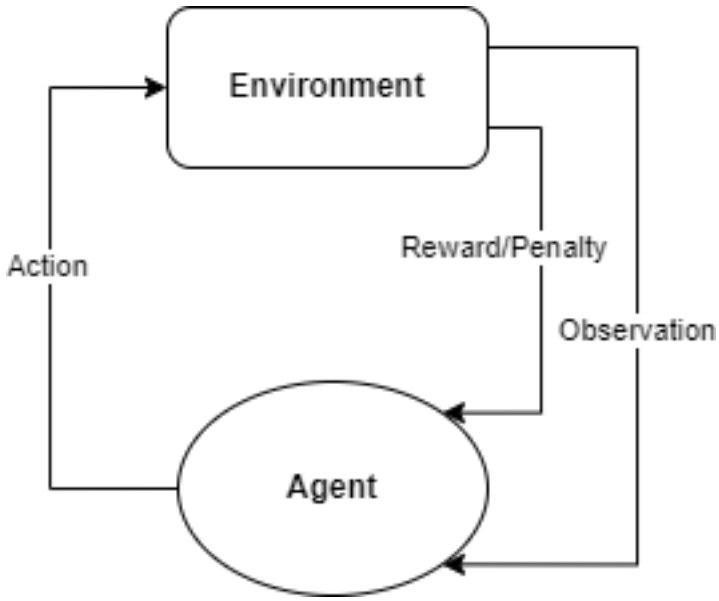


Figure 3.3: Simple visualisation of an agent interacting with an environment.

The agent operates based on a policy in reinforcement learning. A policy defines the strategy or behaviour of an agent, like its guidebook. The policy contains a learnt mapping from observed states of the environment to the best actions to choose. If a policy is deterministic it defines a specific action for each state, if it is stochastic, probabilities are assigned to each action.

Several popular algorithms guide the learning of these policies, including Deep Deterministic Policy Gradients (DDPG), Proximal Policy Optimisation (PPO) and Soft Actor Critic (SAC) among others. Each algorithm comes with its own set of hyperparameters that, when tuned appropriately, can significantly influence the learning process's success.

Reward shaping is crucial in training an agent. The earned reward provides feedback to the agent about its actions, guiding its learning. The design of the reward function requires careful consideration and typically requires an in-depth knowledge of the environment and task that the agent is attempting to learn.

3.3 Machine learning software

3.3.1 MATLAB

MATLAB, developed by MathWorks, includes a host of machine learning (ML) capabilities. Notable extensions include the Deep Network Designer, Classification Learner, and the Experiment Manager apps. Due to its comprehensive documentation, learning content and active community, MATLAB provides a user-friendly platform.

Given its extensive use in engineering and powerful time series analysis toolboxes, this study will utilise MATLAB for the training and evaluation of the load forecast.

3.3.2 Python

Python stands as an immensely popular, open-source language for machine learning. Its expansive user base provides support and extensive documentation. A multitude of libraries, dedicated to different facets of machine learning, make Python a powerful choice for machine learning.

One such library is stable-baselines3. It offers high-quality implementations of various RL algorithms, such as PPO, DDPG, and SAC. Importantly, it provides a platform for the integration of custom environments[39].

Weights and Biases is a machine learning experiment tracking platform. It offers a range of tools to log, visualise and analyse experiment metrics. It can easily be integrated into Python projects and offers support for environments created in stable-baselines3.

Chapter 4

Data collection

This chapter outlines the procedures undertaken for data collection. Drawing from the insights of the literature review and the project brief, it was identified that modelling a hybrid microgrid necessitates data in the following key areas:

1. AC and DC load data
2. Time of use rates
3. Load shedding occurrences

To enhance the relevance and applicability of the data, a building on the UCT upper campus was chosen as the primary source for data collection. This choice was driven by the availability of data combined with the fact that the building's energy usage is appropriate for a small microgrid. To maintain uniformity and facilitate analysis, all data was collated in hourly intervals. The designated time frame for this data collection was the entirety of 2022.

4.1 AC and DC load data

The load data serves several essential roles in this study:

1. It serves as the platform for the modelling and sizing of the hybrid microgrid.
2. It forms the primary dataset for training a model to generate 24-hour load forecasts.
3. It is used in the simulation of the reinforcement learning environment.

Permission was requested to utilise the load data from an academic building on UCT's upper campus. Upon approval, this building's load profile served as the AC load data source. While the UCT building data was an apt fit for the hybrid microgrid's AC load, it lacked a DC load counterpart. To address this, a dataset with an hourly load profile was identified on Mendeley Data, licensed under Creative Commons [40]. A middle school's load profile was selected due to its analogous trends and distributions to the UCT building data. To ensure consistency, the dataset was adjusted to commence on the same day of the year. As the data originated from the northern hemisphere, adjustments were made to align the winter seasons between both hemispheres.

Subsequently, the AC load data from UCT and the DC load data from Mendeley Data were consolidated into a single table. Table 4.1 details the relevant headings.

Table 4.1: Table headings for combined load data

Heading	Description
DC_load	The DC load profile was appended to the dataset (kW)
AC_load	UCT building system true power, in kilo-Watts (kW)
time	Time in 24 hour format
date	Date in yyyy/mm/dd format
tou	Time of Use eg. Off-peak, Standard, Peak
tou_id	Time of Use ID, a numerical representation of tou

4.1.1 Cleaning load data

The AC load data contained 6 rows with null entries. Given the importance of maintaining the time series integrity, these null entries were not removed as this would create inconsistencies in the time steps. Instead, each null entry was filled using its immediately preceding value.

Upon analyzing the AC load data for potential outliers, several instances of near-zero power readings were observed, as illustrated in figure 4.1. Overlaying the days when load shedding occurred in 2022 onto the UCT building's electrical load plot confirmed that these anomalies were attributable to load shedding events. Given the anomalies' potential to hinder a model's predictive accuracy, it was deemed necessary to clean the data. The lower 4.7 percentile of data, which contained the majority of the anomalous readings, was removed. The empty entries were then filled using linear interpolation. The entire cleaning process can be visualised in figure 4.2.

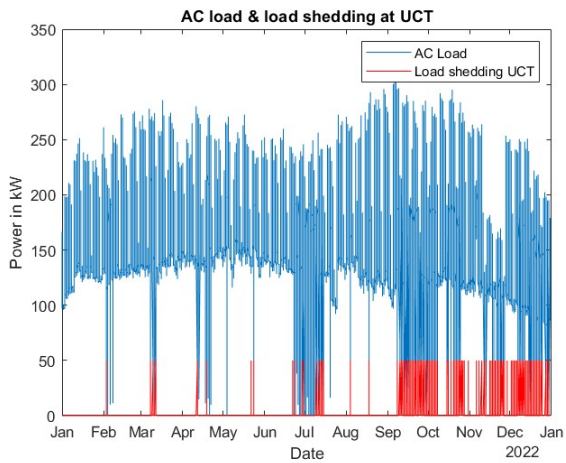


Figure 4.1: UCT building electrical load, overlayed with loadshedding at UCT.

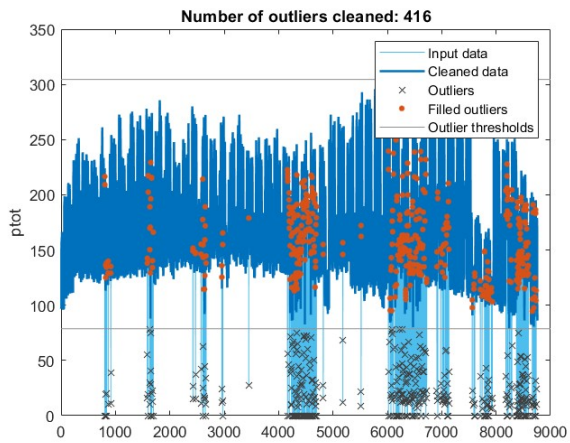


Figure 4.2: AC load and outlier replacement

The final AC and DC load data is aptly suited for training a model that can generate accurate 24-hour load forecasts. Furthermore, it is appropriate for its key role in the design of the hybrid microgrid.

4.2 Time of use rates

Obtaining accurate TOU rates is important, as they represent a primary avenue for cost reduction through energy management. Therefore, they are a key component in the design and simulation of the

RL environment used to train an autonomous agent.

A detailed examination of the City of Cape Town's rates tariffs was undertaken to find the time of use rates specific to large power users.

As outlined in the City of Cape Town's Annual Tariff Policies [1]. An electricity tariff is defined based on the category of service provided. The categorisation of a user is based on the type of service, level of service consumption, type of connection, time of use and any other important factors. Given the nature of the academic building, it is categorised as a commercial/industrial power user, distinctly separate from lifeline or residential users. With a maximum recorded power consumption of 300kW, the academic building aligns with the definition of a Large Power User operating under LV tariffs[41].

A comprehensive tariff structure, detailing the TOU tariff rates specific to large power users with a low voltage connection, is available in the City of Cape Town's electricity consumptive tariffs[2]. This structure is replicated in table 4.2 for ease of reference.

Table 4.2: Time of use rates for a large power user in Cape Town [42]

Services	**Unit**	**Remarks**	**R** (reg)	**R** (unreg)	**R** (total)	**R** incl. VAT
High Demand Jun-Aug		Winter Months				
Peak	c/kWh		483.99	31.42	515.41	592.72
Standard	c/kWh		150.58	31.42	182.00	209.30
Off Peak	c/kWh		84.39	31.42	115.81	133.18
Low Demand Sept-May		Summer Months				
Peak	c/kWh		161.71	31.42	193.13	222.10
Standard	c/kWh		113.10	31.42	144.52	166.20
Off Peak	c/kWh		73.82	31.42	105.24	121.03
Service Charge	R/day		133.69	0.00	133.69	153.74
Demand Charge	R/kWh		219.93	0.00	219.93	252.92

The City of Cape Town defines the periods for different TOU charges as:

- **Peak time of use:** Weekdays from 07:00 to 10:00 and 18:00 to 20:00.
- **Standard:** Weekdays from 06:00 to 07:00, 10:00 to 18:00, and 20:00 to 22:00.
- **Off Peak:** All other times.

Leveraging this information, an hourly dataset was established, mapping each hour to its associated TOU fee for the entire year. This dataset was used in the design phase of the hybrid microgrid, serving as an input to HOMER Pro. Furthermore, it was employed in the simulation of the RL environment, assisting in the computation of energy costs for each time step.

[1] https://resource.capetown.gov.za/documentcentre/Documents/Financial%20documents/Annexure%207%20-%20Tariff%20Policies.pdf

[2] https://resource.capetown.gov.za/documentcentre/Documents/Financial%20documents/Electricity%20Consumptive%20Tariffs.pdf

4.3 Load shedding 2022 schedule

Given South Africa's ongoing energy crisis, load shedding has become a prevalent challenge, continuing for the foreseeable future. This has led to an increase in polluting and costly diesel generators used to bridge the energy gap. The situation necessitates an autonomous energy management agent with the ability to navigate the eventuality of load shedding.

The City of Cape Town's records provided a comprehensive account of load shedding events[43]. This extensive dataset details load shedding across all zones in the greater Cape Town region, spanning from 2018 to September 2023. For the purposes of this study, events from the year 2022 corresponding to zone 15 were extracted, given that this zone encompasses UCT.

Figure 4.3 illustrates the load shedding events for 2022. A noticeable uptick in load shedding towards the year's end is evident. Meanwhile, figure 4.4 provides insights into the distribution of these events based on the time of day. Notably, the distribution curve suggests a higher frequency of load shedding during nighttime as opposed to working hours, with a local peak during the middle of the day.

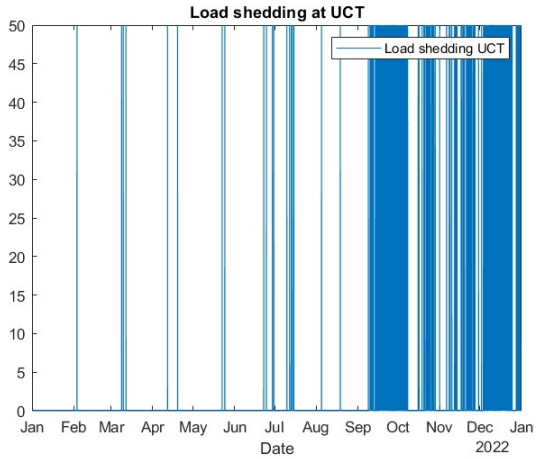


Figure 4.3: Load shedding events throughout 2022

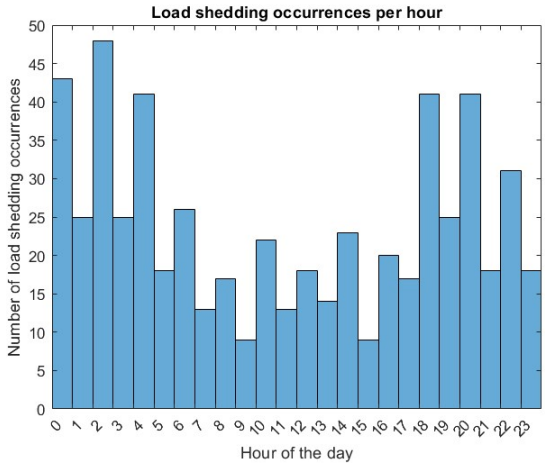


Figure 4.4: Distribution of load shedding events based on time of day

Having collated the essential data, the load, TOU, and load shedding datasets will serve as foundational inputs for the hybrid microgrid's design. The load data will be pivotal in training a model for 24-hour load forecasts. Collectively, the datasets will be instrumental in simulating the final RL environment used to train an autonomous energy management agent.

Chapter 5

Microgrid design

The design of the hybrid microgrid is a core aspect of the study as it dictates the environment in which the autonomous energy management agent will operate. The design should incorporate an AC and a DC load, ensure adequate generation to meet demand without excessive surplus generation and remain economically feasible.

HOMER Pro is a powerful software tool that is widely used for the optimisation and design of microgrids in renewable energy ventures. It facilitates the modelling, simulation and analysis of energy systems, aiming to achieve optimal performance, cost-effectiveness and sustainability.

Users can fully configure inputs related to loads, energy resources, equipment specifications and economic parameters. HOMER Pro then executes an optimisation algorithm to find the configuration and sizing with the lowest Net Present Cost (NPC). To clarify NPC is the total cost incurred over the project's lifetime adjusted to a value in present time, accounting for both the discount rate and inflation.

In terms of component choices, wind turbines were selected to serve the AC load, while a PV array was chosen for the DC load. A lithium-ion battery bank will be used for the ESS. These selections stemmed from their prevalent use in microgrids and the abundant availability of wind and solar power.

The sizing of the wind turbine, PV array, ESS and diesel generator will be accomplished with HOMER Pro, following this procedure:

1. Input data and parameters.
2. Evaluate proposed hybrid microgrids.
3. Adjust parameters if needed.
4. Finalise hybrid microgrid.

5.1 Hybrid microgrid design

As can be seen in figure 5.1, the design will be a coupled AC hybrid microgrid, this decision was informed by the literature review. The AC load along with wind turbines, a diesel generator and a connection to the utility grid make up the AC subgrid shown in blue. The DC subgrid is made up of the DC load, a solar PV array and an ESS shown in orange. A 2-way converter is used to transfer energy between the subgrids.

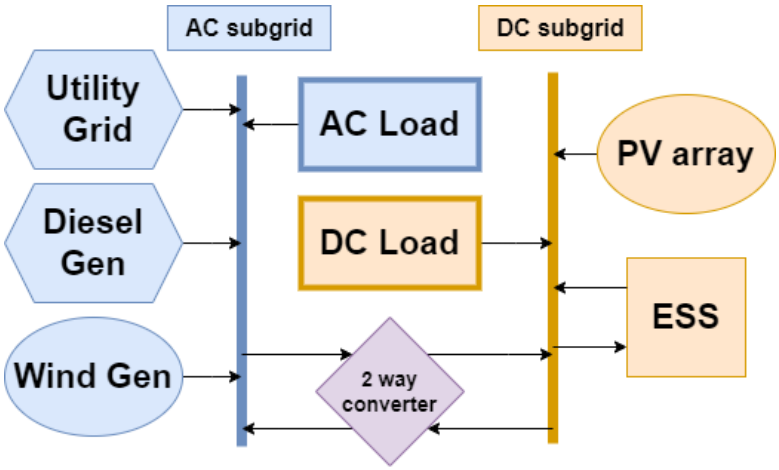


Figure 5.1: Input grid layout to Homer Pro.

5.1.1 Homer PRO Input

The following section details all of the key input parameters that were chosen for the design of the hybrid microgrid in HOMER Pro.

5.1.2 Electrical inputs

1. **Load profiles:** The data that was collected from UCT and cleaned in chapter 4 was used to model the AC load. The data collected from Mendeley data [40] in chapter 4 was used to model the DC load.

2. **Load shedding:** In order to model the effects of load shedding on the microgrid, the load shedding schedule that was found in chapter 4 was used to set the grid outages in Homer Pro. The aim being to see how load shedding and an unstable grid would affect the sizing of microgrid components.

5.1.3 Economic factors

1. **Component pricing:** Homer PRO requires pricing input for each component in order to find an economically viable solution. In order to find approximations of component prices, multiple purchase prices were found and averaged. A summary of the prices and final inputs is shown below in table 5.1. To find the full list of references and purchase links please view the table in Appendix A, table A.1.

2. **Capacity shortage:** The economic penalty for a capacity shortage was set to R5/kWh. This was used to incentivise the optimiser to reduce unmet demand and find a system that was capable of meeting the load with an unreliable grid. It is important to note that the value of R5/kWh is not based on a real number as the monetary value for unmet demand depends on the demand's use case. For example in a factory producing valuable products, unmet demand would have a large monetary cost associated with it. Since the building in question is an academic building there is no easy way to directly relate unmet demand to a monetary value. The final value of R5/kWh was set to be slightly higher than the cost of purchasing electricity to indicate to the optimiser that a capacity shortage is undesirable.

3. **Inflation:** Conservatively, inflation was set to 6% which is the estimated headline inflation for 2023 taken from a statement made by the South African Reserve bank [1]. It is worth noting that inflation of electricity prices has been much higher, however for simplicity the headline inflation was used.

4. **Energy tariffs:** The TOU rates that were collected in the chapter 4 were used to define the microgrid's tariff structure.

5. **Feed-in tariff:** The City of Cape Town offers a feed-in Tariff. This is the price that it is willing to purchase energy from a non-residential small-scale embedded generator. However, there is a condition applied that the user must purchase more energy than they sell to the grid. It was decided that the feed-in tariff would be excluded from this study to limit the scope.

Table 5.1: Component prices input to Homer PRO

Component	Size	Price (R)
Solar PV	1 kW	4 396.00
Wind turbine	10kW	48 598.00
Diesel generator	1kW	2 919.00
Battery li-ion	100kWh	646 963.00

5.1.4 Homer PRO output

HOMER Pro uses NPC to optimise microgrid sizing. As previously mentioned, the NPC represents the present value of the entire cost associated with a project over its lifespan, including initial investments, operational expenses, maintenance and potential replacement costs. It is used as the optimisation metric since economic feasibility is key to the success of any project.

Initially, the simulation was run to find the lowest NPC. However, the solution that HOMER Pro proposed was vastly oversized and impractical. As can be seen in table 5.3, the microgrid consists of 3.53MW generation capacity. This is far above the project brief of 1MW and the space required to install 1634 PV panels and 190 wind turbines would be extremely vast.

Table 5.3: Component summary of HOMER Pro optimised system.

Component	Quantity	Specification	Total Capacity
Generic flat plate PV	1634	1kW	1634kW
Generic wind turbine	190	10kW	1900kW
Diesel generator	0	0	0kw
Li-ion battery	0	100kWh	0kWh

Additionally, the system also produces a huge amount of excess electricity. As shown in table 5.4, the grid produces 318% of the total load in excess electricity. While the grid has a low NPC and a low unmet demand a more practical alternative would need to be found. It was decided that the maximum

[1] https://www.resbank.co.za/en/home/publications/publication-detail-pages/statements/monetary-policy-statements/2023/Statement-of-the-Monetary-Policy-Committee-July-2023

Table 5.4: Electrical summary of HOMER Pro optimised system.

Component	Energy (kWh/yr)	Percent total load (%)
Consumption summary.		
AC load	1,512,922	66.3
DC load	767,845	33.7
Total	2,280,767	100
Excess and unmet load.		
Excess electricity	7,251,238	317.93
Unmet electric load	9,076	0.40

generation capacity of the generic flat plate PV array would be limited to 500kw and similarly the generation capacity of the generic wind turbine would be limited to 500kw. These limits were guided by the project brief which states that the microgrid should not have a capacity greater than 1MW. The HOMER Pro optimiser was run again to ensure feasibility. As expected the PV array and the wind turbines were set to their respective upper limits. However, the generator was sized to a max capacity of 560kW. This is grossly oversized and is only used to generate a total of 2 895kWh of power per year. In order to achieve more reasonable results, the generator was limited to 100kW capacity.

5.2 Final grid parameters

The composition of the final microgrid can be seen in table 5.5. It consists of a PV array with capacity 500kW, wind turbines with total capacity 500kW, a diesel generator with 100kW capacity and a lithium-ion battery with 500kWh storage capacity.

Table 5.5: Component summary of 1MW generation capacity microgrid.

Component	Quantity	Specification	Total Capacity
PV units	500	1kW	500kW
Wind units	50	10kW	500kW
Diesel generator	1	100kW	100kW
Li-Ion battery	5	100kWh	500kWh

In table 5.6 The amount of excess electricity has been significantly reduced to 42.8% of the total load while the unmet demand is reduced to 9.78kW, practically nothing.

Table 5.6: Electrical summary of final microgrid design.

Component	Energy (kWh/yr)	Percent of total load
Consumption Summary.		
AC load	1,518,997	66.3
DC load	770,836	33.7
Total	2,289,273	100
Excess and unmet load		
Excess electricity	980,269	42.82
Unmet electric load	9.78	0.00

This configuration meets the outline that was set in the project brief. It has been chosen as it sets a balance between the NPC, unmet load, excess electricity and the total generation capacity. The

final parameters of the hybrid microgrid will dictate the characteristics of the RL environment used to train the energy management agent. Furthermore, they were used to simulate the solar PV and wind turbine generation data.

5.3 PV and wind generation data

To simulate the RL environment, an hourly dataset detailing the power generation of both the PV array and wind turbines is needed. Given that the sizing of these components is tailored to the microgrid design of this study, generation data had to be simulated post finalisation of the microgrid.

5.3.1 Simulated generation data

Homer PRO was used to generate simulated PV and wind output with an hourly resolution with a window spanning one year. The PV system was a 500kW array and the simulated generation uses historical metodata that is extrapolated from satellite data. The wind system was comprised of 50 generic 10kW wind turbines.

The wind data can be found here: Simulated Wind Generation.
The solar data can be found here: Simulated PV Generation.

A distribution of the energy generated according to the hour of the day is shown in figure 5.2 and 5.3 for the PV array and the wind turbines respectively.

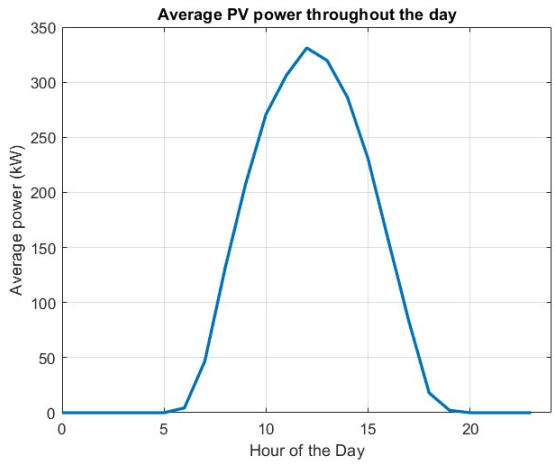


Figure 5.2: Daily PV array power generation profile.

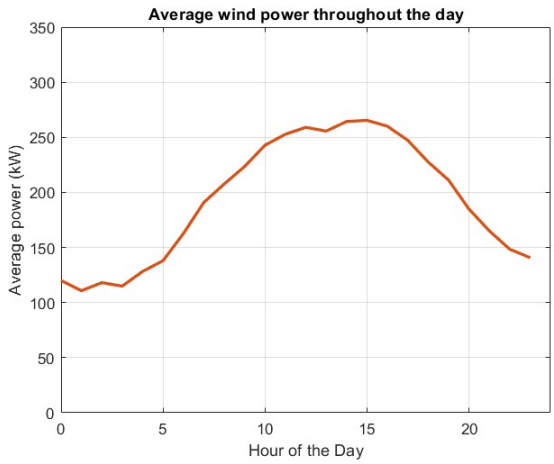


Figure 5.3: Daily wind turbine power generation profile.

The simulated generation dataset was used to provide the RL environment with hourly generation values.

5.3.2 Generation forecasting

A key section of energy management is to have a prediction of the generation capacity that a microgrid will have available during the scheduling window.

There was not sufficient scope or time to properly undertake the task of training a model to perform generation forecasts. This was an unfortunate but necessary decision that was made to keep the study within a reasonable scope. With more time it would be interesting to test the use of machine learning and historical data to make generation forecasts.

In chapter 8, the implementation of generation forecasting in the RL environment is explained in more detail. In summary, the agent receives the actual values for the upcoming 24 hourly steps, essentially giving it perfect forecasts. This is definitely an area for improvement through further research.

As a side note, sophisticated programs exist that provide generation forecasts, for example, Solargis Forecast. However, these complete package programs can be expensive. As an alternative, it is also possible to use APIs for weather services to retrieve forecasts of the necessary weather conditions. These forecasts can then be used to calculate an estimate of the generation capacity during the forecast window.

Given the existence of these generation forecasting methods, it would be interesting to explore their potential as an alternative to machine learning for generation forecasting.

Chapter 6

Load forecasting

This chapter dives into training models that are able to perform time series forecasts. In the context of the study, it uses the AC & DC load data collected in chapter 4 to train a machine learning model. The model is then used in the energy management process to provide the agent with load forecasts.

The final models that were selected to be used for load forecasting were informed by the machine learning theory and the literature review. Three models were selected the LSTM, ARIMA, SARIMA and an ANN. The following steps were taken for each model:

1. Preprocess data into correct formats
2. Fit model parameters
3. Determine forecasting method

The machine learning theory regarding time series analysis is covered in Section 3.1. A brief overview of the theory behind each of these models is covered in subsection 3.1.4, 3.1.2 and 3.1.3 respectively.

The model that produces the best load forecasts will then be used in the final energy management system.

6.1 Exploratory data analysis

Before fitting the models it is necessary to perform an exploratory data analysis in order to further understand the data. The aim is to unearth valuable insights, patterns and trends that may have initially been hidden. The focus will be on the load profile and consumption patterns in an attempt to inform our approach when using the data for predictions.

When observing a histogram of the distribution for the AC load data is distinctly bi-modal as can be seen in figure 6.1. This was expected and can largely be attributed to differences in consumption during the day versus consumption during the night.

Furthermore, it is interesting to observe that the AC load during the day also has two distinct peaks. This was further explored and it was determined that it arose from different usage patterns during working days and during weekend days. The breakdown goes further to show that weekends also have two distinct distributions, one for Saturday and another for Sunday as can be seen in figure 6.2.

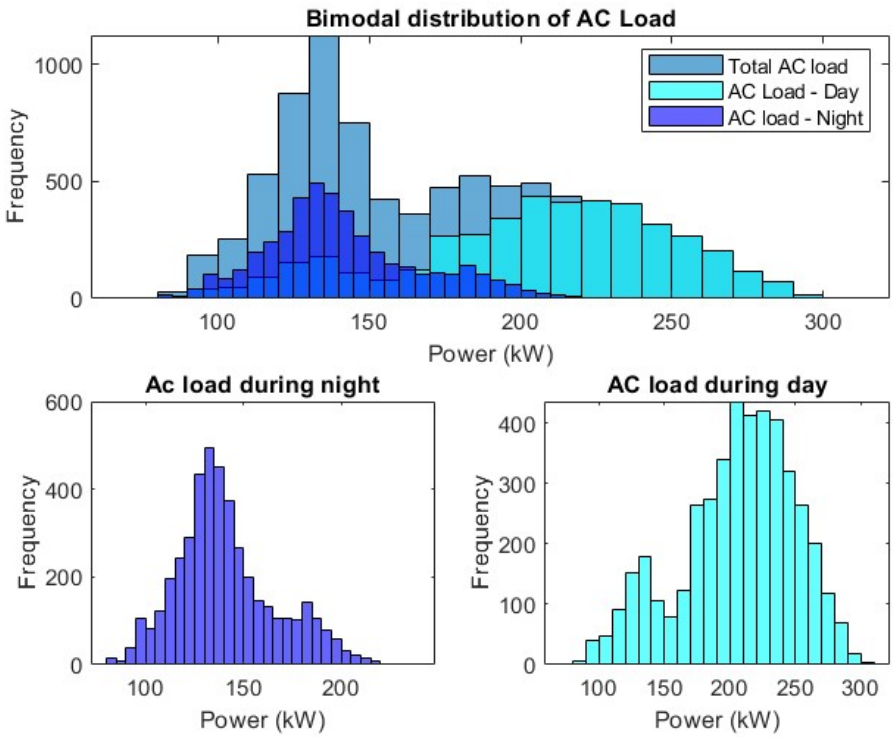


Figure 6.1: The bimodal distribution of the AC load.

The average hourly load profiles follow a very similar trend, in line with what was expected. This can be seen in figure 6.5.

The seasonality of the data can be seen through a boxplot of the power values grouped by month. There is a slight uptick in power usage during the winter months although it is not as prominent as was expected. The large number of outliers shown in the boxplots of the DC load was also unexpected. These outliers arise from the heavy skewness of the data with many more readings falling outside of working hours and on weekends compared to during working hours, leading to the median of the data being a comparatively low value.

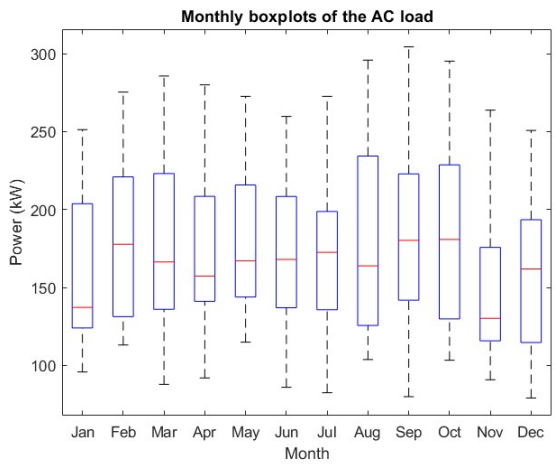


Figure 6.3: Boxplot of the AC load per month

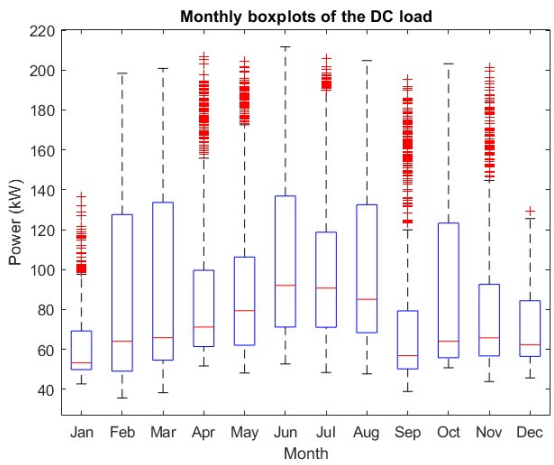


Figure 6.4: Boxplot of the DC load per month

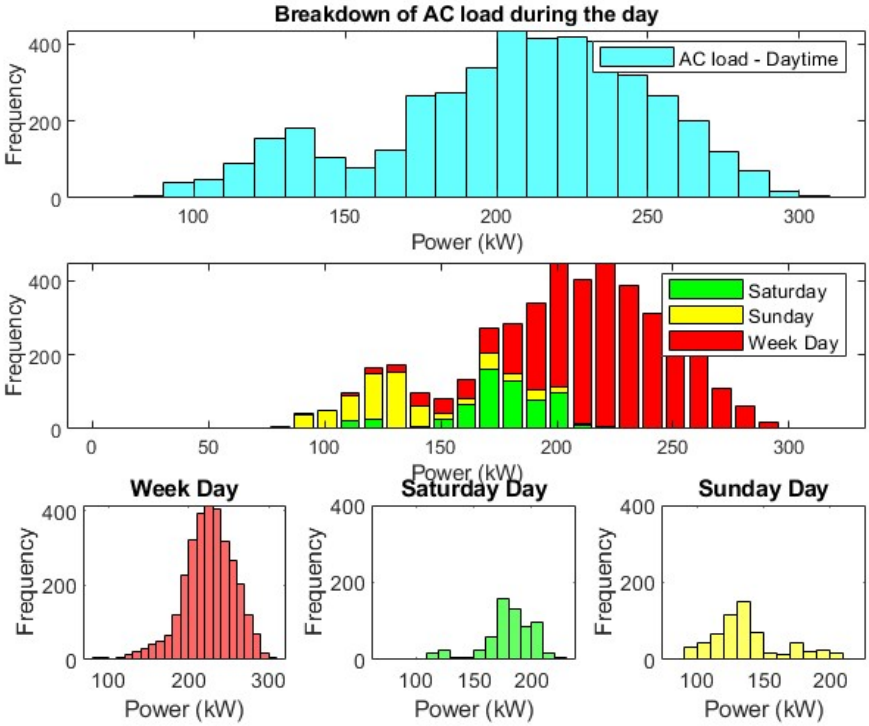


Figure 6.2: Breakdown of the AC load distribution during the day

6.1.1 Feature Selection

During the feature selection process, indicator variables were added to the dataset. These variables were chosen to be known values that could be used to aid in forecasting the unknown load at a future time. For example, whether the time step falls into working hours or not. These indicators can be seen in table 6.1

One-hot encoding is a technique used to transform categorical variables, such as the 'time of use id' (tou_id), into a format that improves machine learning effectiveness. This technique is beneficial as many algorithms operate under the assumption that the input variables have an ordinal relationship, meaning that the order of the categories would matter. For instance, the tou_id variable is recorded as 1,2, and 3 to signify 'off-peak', 'standard', and 'peak' times respectively, the model may incorrectly assume that the 'peak' category is three times as significant as the 'off-peak' category. One-hot encoding mitigates this issue by converting each category into a separate boolean feature, thereby avoiding any unintended ordinal interpretation.

There are caveats to this approach. One-hot encoding can rapidly increase the dimensionality of the dataset which can lead to increased computing cost and training time. Additionally, it can introduce collinearity which refers to a situation where two or more predictor variables in a regression model are highly correlated. Collinearity can lead to reduced generalisation and unstable training.

A heatmap of the correlation matrix for the dataset is displayed in figure 6.6. This visualisation helps identify which indicator variables might enhance the model's predictive performance. Specifically, the variables of interest are the AC and DC columns. The heatmap reveals that the AC variable is strongly positively correlated with the 'wrkHr' indicator, which takes on a value of 1 during working hours and

Table 6.1: Table header showing potential features for machine learning.

tstamp	AC	tou_id	DC	weekDay	Sat	Sun	wrkHr	dayTime	Off-Peak	Peak	Standard
01-Jan-2022 00:00:00	96.855	1	52.2	0	1	0	0	0	1	0	0
01-Jan-2022 01:00:00	97.053	1	52.0	0	1	0	0	0	1	0	0
01-Jan-2022 02:00:00	97.229	1	52.2	0	1	0	0	0	1	0	0
01-Jan-2022 03:00:00	96.859	1	52.2	0	1	0	0	0	1	0	0
01-Jan-2022 04:00:00	98.018	1	52.2	0	1	0	0	0	1	0	0

0 otherwise.

The AC variable also exhibits strong correlations with the 'tou_id' indicator and two of its one-hot encoded columns. Notably, the 'off-peak' indicator shows a strong negative correlation, while the 'standard' indicator displays a strong positive correlation. The 'peak' indicator has a weaker correlation. Concerning potential issues of collinearity, the 'standard' and 'off-peak' indicators are strongly negatively correlated, which could pose challenges during model training.

Similarly, the DC variable displays a correlation pattern akin to that of the AC variable, but all of the indicators have slightly weaker correlations with the DC variable. This discrepancy might be attributable to the simulated nature of the DC values, in contrast to the real measurements for the AC values.

In summary, the 'wrkHr', 'off-peak', and 'standard' indicators appear promising for predicting both AC and DC variables. However, the strong negative correlation between 'off-peak' and 'standard' could introduce collinearity issues. The 'tou_id' variable might be a good alternative if issues do arise, although it could introduce biases toward specific categories.

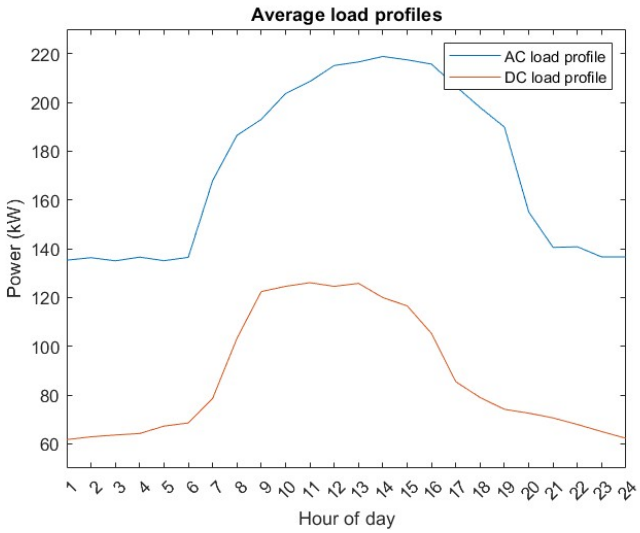


Figure 6.5: The average hourly AC & DC load profiles.

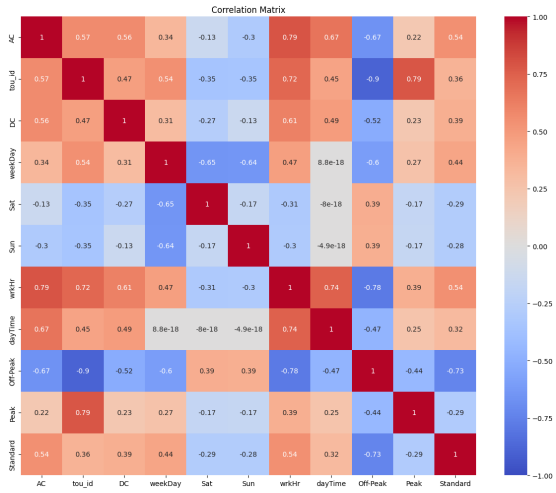


Figure 6.6: Heatmap of the correlation matrix.

6.2 Fitting LSTM model

In this step, the hourly electrical load data is prepared for an LSTM network, which excels at handling multivariate time-series forecasting. The data is first loaded and organised into a format suitable for training. Subsequently, it is partitioned into a training and testing data set and normalised. The model is then trained and a forecasting method is developed to generate a 24-step-ahead forecast.

6.2.1 Data Preparation

The electrical load data that was prepared in the previous section was imported into the workspace. The data was re-organised from the imported table into a cell array. Since LSTM models are capable of multivariate predictions, multiple predictors can be used. The indicators that were found to have strong correlations with the load data were kept while all the other values were removed.

6.2.1.1 Data partitioning

The array was grouped into batches to improve training efficiency. The number of time steps in each batch should be enough to provide information about the time series but not so much as to slow the training down extensively. Different size batches were tested and it was found that a batch size between 14 and 35 days provided the best results.

The dataset is split into a 70-30 ratio for training and testing purposes, ensuring that the temporal sequence is preserved to maintain the time-series integrity. A target matrix is also created. To forecast the values of future time steps, the targets are specified as the training sequences with values shifted by one time step. The training values are stored in the matrix XTrain and the training targets are stored in TTrain. The same idea is applied to the testing data with the test inputs being stored in XTest and the test targets being stored in TTest.

6.2.2 Normalisation

The mean and standard deviation for the training dataset were calculated. This was used to normalise both the testing and training data set by using equation 6.1, where μ denotes the mean and σ the standard deviation.

$$\hat{X} = \frac{X - \mu}{\sigma} \tag{6.1}$$

It is important to note that the testing partition was normalised using the mean and standard deviation calculated for the training data set.

The values for the mean and the standard deviation were stored so that the predictions could be transformed back to the original scale.

6.2.3 LSTM configuration and training

A basic LSTM configuration was first tested with an input layer, LSTM layer, fully connected layer, and a regression layer. The size of the input layer corresponds to the size of the input matrix. Both the number of LSTM layers and the number of nodes per layer are fully configurable. Generally, models with a higher number of nodes and layers are able to learn more complex relationships but take longer to train and may overfit the data. The final LSTM layer then feeds into the fully connected layer which feeds into the regression layer to estimate the next value. This can be see in listing 6.1

Different size LSTM layers were tested and it was found that the AC load was predicted best by a much

smaller layer and that the DC load benefited greatly from having larger layers. The final configuration for the DC load LSTM model can be seen in 6.1

```
numChannels = size(dataCell{1},1);
layers = [
    sequenceInputLayer(numChannels)
    lstmLayer(128)
    lstmLayer(128)
    lstmLayer(128)
    fullyConnectedLayer(numChannels)
    regressionLayer];
```

Listing 6.1: LSTM configuration.

The Adam optimiser was used to train the network, given its well-documented efficacy in similar tasks [44]. Convergence was reached at approximately 2,000 epochs; further training led to overfitting and poorer generalisation of the test data. To handle sequences with varying lengths, left padding was applied to the input sequences. This step was crucial to prevent the model from attempting to predict 'empty' or 'null' values at the ends of sequences with differing numbers of observations. These details can be seen in listing 6.2

```
    options = trainingOptions("adam", ...
    MaxEpochs=2000, ...
    SequencePaddingDirection="left", ...
    Plots="training-progress", ...
    Verbose=0);
```

Listing 6.2: LSTM training parameters.

6.2.4 Forecasting method

Closed-loop forecasting is employed to make predictions for a series of future time steps, specifically from **t** through to **t+k** based on observations recorded for previous steps **1** through to **t-1**.

1. **Initialisation:** The network is initialised to with previous values up to **t-1**.
2. **First prediction:** A forecast is made for the immediate next step, **t**.
3. **Update:** The network is updated with the forecasted value **t** as if it were an actual value. This effectively 'closes the loop,' allowing the model to use its own prediction for the subsequent forecast.
4. **Further predictions:** A new prediction is made for the next time step **t+1**, using the updated state.
5. **Iteration:** Steps 3 and 4 are repeated until the end of the forecast window, **t+k**.

After generating predictions for a full 24-hour period, the forecasted values for all 24 time steps are stored in an output array. An illustrative example of what a day-ahead forecast generated through this method looks like can be seen in figure 6.7

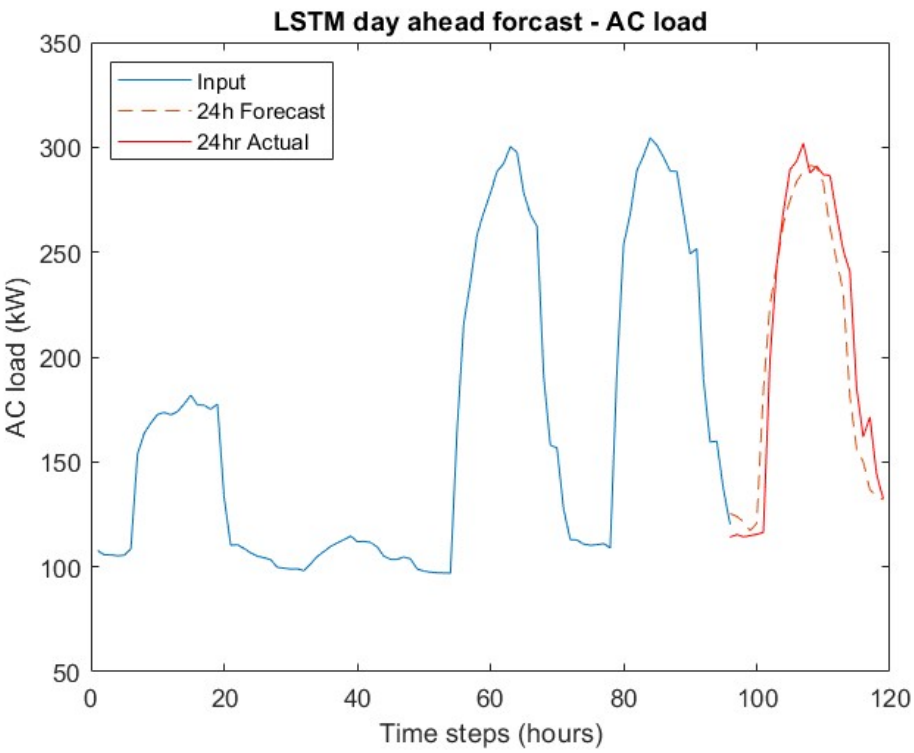


Figure 6.7: Day ahead forecast of the AC load using LSTM.

As can be observed from figure 6.7, the network performs well, matching the actual load closely. The code that was used to implement the fitting and forecasting of the LSTM network can be found here.

6.3 Fitting ARIMA model

This section delves into the fitting of an ARIMA model, a popular algorithm in the field of time-series forecasting for several decades. As explained in Section 3.1.2, fitting an ARIMA model necessitates the identification of three parameters: the auto-regressive order (p), the moving average order (q), and the differencing order (d). This section outlines the process undertaken to determine suitable values for these parameters, fit the model and conduct the forecasting.

6.3.1 Data Preparation

An ARIMA model only uses a single time series input, thus the data was streamlined to the core time series. That is a timestamp, the AC power, and the DC power.

6.3.2 Determining Model Parameters

Determining the value of d: A Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test was performed to determine if the data was stationary or not. The KPSS test returned that it was not stationary, in line with this result, the ACF showed that the series had many significant lags.
The data was then differenced and the same tests were repeated. The KPSS test returned that the data was stationary and the AFC showed fewer significant lags. Although there were still some significant correlations as can be seen in figure 6.8. Additionally, there is a clear spike in the AFC around the 24th lag. This indicates that there is a solid seasonal component at the 24th lag. This is a good reason to attempt a SARIMA model.

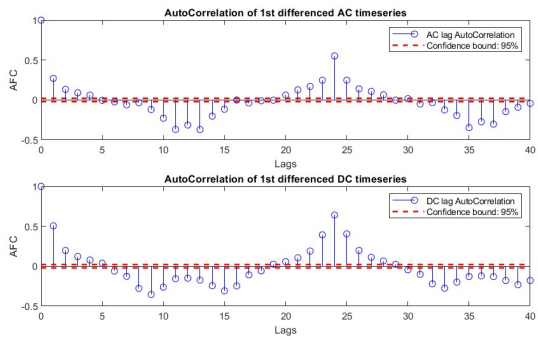


Figure 6.8: AFC of the 1st order differenced timeseries.

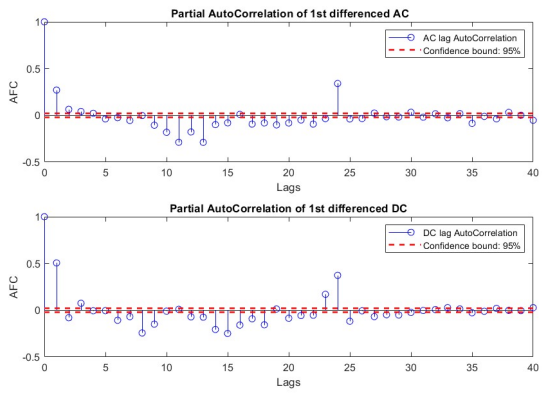


Figure 6.9: PAFC of the 1st order differenced timeseries.

A second differencing was attempted but the lag components quickly became highly negative. Therefore the simpler model with first-order differencing was selected for the initial tests. Thus the order of differencing is expected to be 1, thus setting 'd' to 1.

Determining the value of p: In order to determine the value of 'p' the PACF of the first differenced sequence was taken as seen in figure 6.9. This showed that only 1 lag was significantly over the confidence bound in both the AC and the DC time series. The AC time series did have a 2nd and 3rd lag that were marginally over the confidence bound.
It was decided that a value of 2 would be used for 'p' in initial tests.

Determining the value of q: The value of 'q' can be determined from the ACF of the first-order differenced series, shown in figure 6.8. Both the AC and DC series have 1 lag significantly over the confidence boundary. They also have 2-3 and 2-4 lags respectively that are marginally over the confidence boundary.
The initial value of 'q' was set to 2 to keep the model simple.

6.3.3 Training the model

The econometrics toolbox in MATLAB was used to try variations of the expected model inputs determined above. It was found that a model with p = 2, d = 1, and q = 2, produced the most accurate forecasts.
The code used to create and train the model is shown below in listing 6.3.3.

```
ARIMA_AC = arima('Constant',NaN,'ARLags',1:2,'D',1,'MALags',1:2,'Distribution','Gaussian');
ARIMA_AC = estimate(ARIMA_AC,X.AC,'Display','off');
```

Listing 6.3: Training of ARIMA model

6.3.4 Forecasting method

ARIMA models in MATLAB have a built-in closed-loop forecast function. This was used to forecast 24 steps and produce day-ahead predictions of the load as can be seen in figure 6.10.

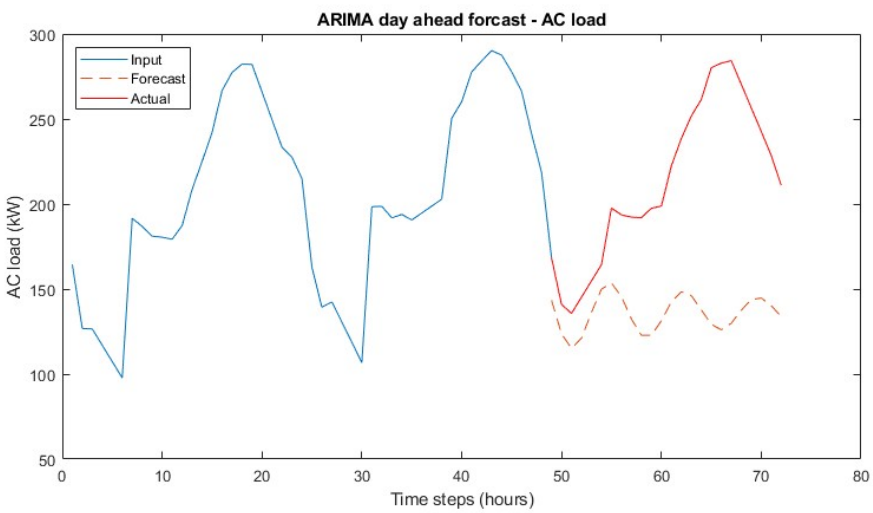


Figure 6.10: Day ahead forecast of the AC load using ARIMA model.

This did not prove to be a very accurate model and it was clear that the seasonal component of the series needed to be taken into account. It was therefore decided to fit a SARIMA network.

6.4 Fitting SARIMA model

In light of the strong positive correlation of the 24th lag in the ACF, it was decided that a seasonal model would be explored. SARIMA models have additional parameters that need to be determined.

Specifically, these are the seasonal auto-regressive order (P), the seasonal moving average order (Q), the seasonal differencing order (D), and the seasonality, which indicates the number of time steps over which the seasonal pattern recurs.

6.4.1 Determining seasonal parameters

Determining the value of D: In order to test the hypothesis that the time series required seasonal differencing, the first differenced time series was further differenced at the 24 component. This choice of seasonality is based on the fact that daily patterns occur every 24 hours.
To illustrate the effects, consider that on Day 1 at 08:00, the AC load was 230 kW. On Day 2 at 0:800, it was recorded at 250 kW. In this case, the seasonal difference for Day 2 at 08:00 would be calculated as the load on Day 2 minus the load on Day 1, resulting in a seasonal difference of 20 kW.

After applying seasonal differencing the ACF was re-examined. The AC time series now displayed substantially reduced auto-correlation in the first few lags. Although the 24th lag showed a negative correlation. The DC series still showed slightly more correlation in the first few lags, although the overall pattern was improved, as can be seen in figure 6.11.
To investigate whether the DC time series required additional differencing, a further round of differencing was conducted. However, the lags quickly turned negative, leading to the conclusion that further differencing would not yield any benefit. Thus it can be concluded that the value of 'D' can be set to 0 or 1 with a seasonality of 24, given that the 24th seasonal lag is negative.

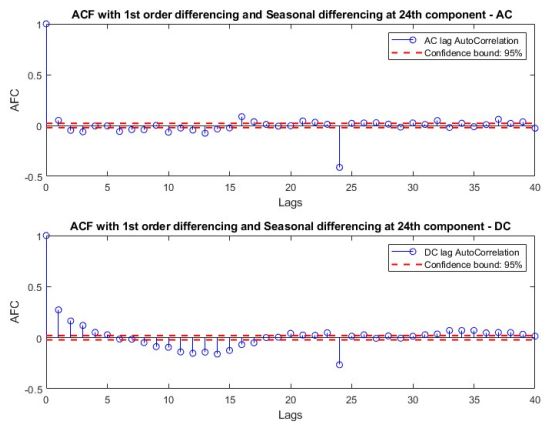


Figure 6.11: The ACF of the AC & DC time series with 1st order differencing and seasonal differencing at the 24th Component.

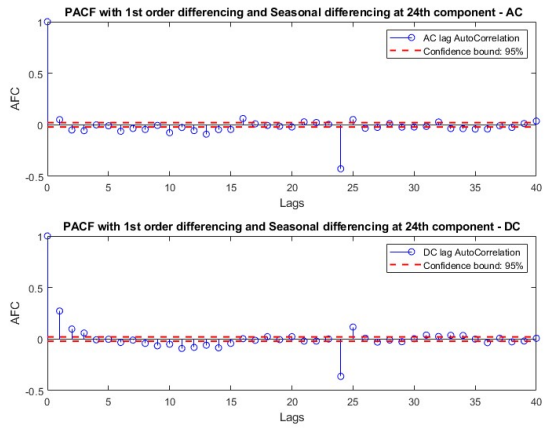


Figure 6.12: The PACF of the AC & DC time series with 1st order differencing and seasonal differencing at the 24th Component.

Determining the value of P: The PACF was also examined for the seasonally difference time series, as shown in figure 6.12. For the AC time series, the plot revealed that some lags were marginally significant, suggesting that the value of 'P' could be either 0 or 1. On the other hand, the DC time series displayed slightly more noteworthy lags, indicating that experimentation with higher values of 'P' may be warranted.
Determining the value of Q: In an analogous manner the ACF was examined for the seasonally differenced time series, as shown in 6.11. Similarly, the AC time series displayed no significant lags suggesting that the value of 'Q' could be 0 or 1, while the DC time series may benefit from a higher

value of 'Q'.

6.4.2 Training Model

In order to find the best combination of input parameters. A grid search was performed across all of the parameters. The limits of the grid search were set using the information gathered in the above subsection 6.4.1. They can be seen in table 6.2. Grid searches are a common way to find the optimal parameters for ARIMA models and their variants.

Table 6.2: SARIMA grid search parameters.

p values	d values	q values	P values	D values	Q values	S (seasonality)
[0, 1, 2]	[0, 1]	[0, 1, 2]	[0, 1]	[0, 1]	[0, 1]	24

The grid search iteratively fits a model, validates the forecasts over a set of validation data and records the best parameters.
The best parameters that were found differed slightly from what was expected with values of (p=0,d=0,q=2)(P=1,D=0,Q=1)[24]. Most interestingly it was found that the model performed best without any differencing. This indicates that the data is more stationary than expected.

6.4.3 Forecasting method

Forecasting was performed in the same manner as for the ARIMA model. This time the results looked much more promising as can be seen in figure 6.13

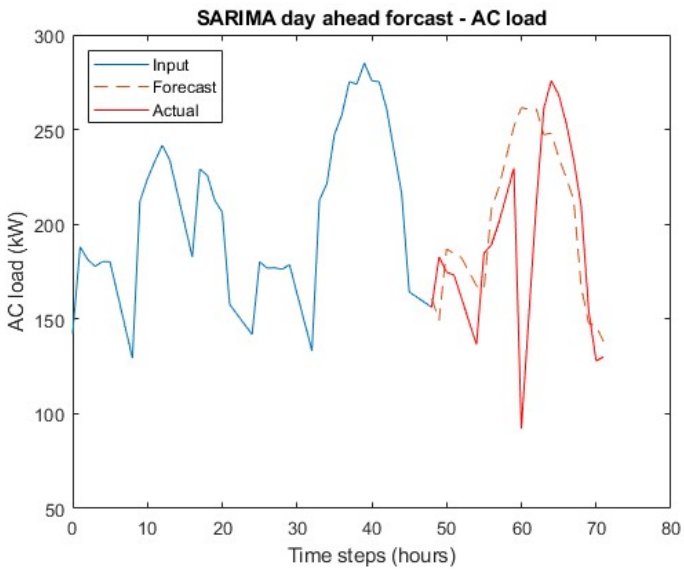


Figure 6.13: Day ahead forecast of the AC load using SARIMA model.

The code used to fit the SARIMA model can be found here.

6.5 Fitting ANN model

6.5.1 Data preparation

For a basic regression network without any exogenous factors, the data was once again streamlined to just the core time series.
A 70/30 split was made with the training and testing data. To provide the model with targets an additional table was made with all of the observations shifted by one time step.

6.5.2 Training model

In order to prevent overfitting and to optimise the neural network's performance, various training options were set, as shown in listing 6.5.2. One of the key configurations was an upper limit on the number of training evaluations. This prevents the model from over-training on the dataset, which reduces its ability to generalise to unseen data.
Importantly, hyperparameter tuning was performed using the 'OptimizeHyperparameters' function. The following parameters were defined for optimisation in the 'params' variable:

1. **Number of layers:** The model explored configurations ranging from 1 to 5 layers to evaluate the trade-off between model complexity and performance.
2. **Size of each layer:** The number of nodes in each layer was set to be between 1 and 300.
3. **Layer weights initialisation:** Finds the best option between 'glorot' and 'he' initialisation methods.
4. **Layer biases initialisation:** The weight biases were optimised to initial values of zero or one.

The activation function for the hidden layers was set to be the ReLe function, known for its effectiveness. And the built-in standardisation was set to be active. The training structure can be seen in listing 6.5.2

```
params = ...
opt = struct("ShowPlots",true,"MaxObjectiveEvaluations",100);
net = fitrnet(X_train, T_train,"OptimizeHyperparameters",params,"
    ↪ HyperparameterOptimizationOptions", opt);
```

Listing 6.4: Training parameters for regression net.

The results of the tuning can be seen in table 6.3

Table 6.3: Model parameters after optimisation

	Num layers	Layer 1	Layer 2	Weights init.	Biases init.
AC	2	1	57	he	zeros
DC	2	1	24	he	ones

6.5.3 Forecasting method

In a similar manner to the closed loop forecasting that took place with the LSTM net, it was necessary to write code that forecasted one time step into the future, updated the model with that prediction and then used it to predict another time step into the future.

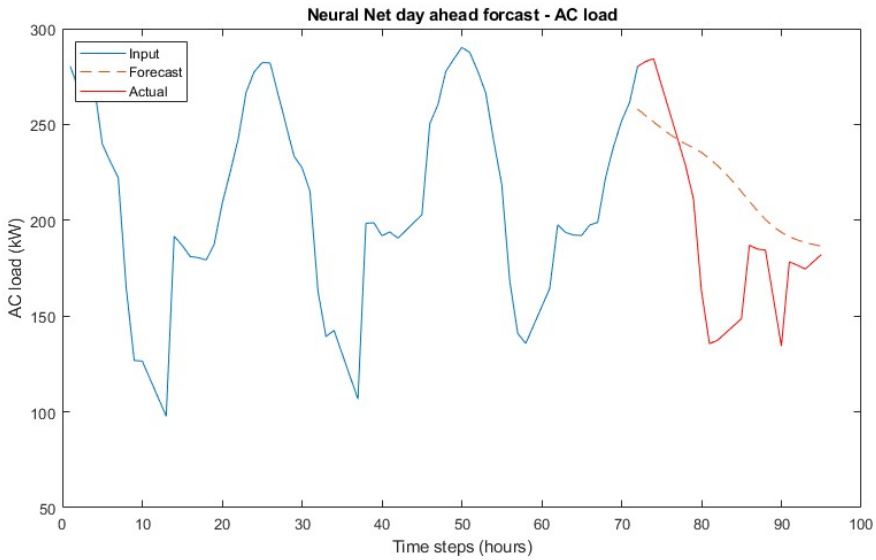


Figure 6.14: Day ahead forecast of the AC load using a regression neural net model.

In a manner similar to the closed-loop forecasting conducted with the LSTM network, it was necessary to develop code to perform closed-loop forecasting. This involved forecasting one time step into the future, updating the model with that forecast and then using the updated model to make subsequent forecasts.

```
    input = X_test(1:offset);  %Fetch the load values up to this step
    first_forecast = predict(net, input); %Peform the first forecast
    Yin = first_forecast(end:end);  %store the forecasted value
    Y = zeros(24,1);        %initialise a variable to store the 24 steps
    Y(1) = Yin;
    for i=2:24     %loop through the remaining steps
        input = [input(2:end)' Y(i-1)'];   %Append the forecast to the nets input
        input = input';  %transpose it
        YNext = predict(net, input); %predict the next step,, using the previous forecasts
        Y(i)  = [YNext(end:end)];   %append the forecast to the forecast array
    end
```

Listing 6.5: Forecasting with neural net.

The forecasting showed some promise but is significantly worse than the LSTM and SARIMA models. An example of what the day ahead forecast looks like can be seen in figure 6.14
The code used to fit and forecast with the neural net can be found here.

Chapter 7

Load forecasting results and discussion

7.1 Results

In the realm of energy management, accurate 24-hour ahead forecasts are critical for performing energy scheduling. To identify the most suitable forecasting model, the three models that were trained in section 6 were evaluated based on their predictive performance. The best-performing model was used to provide the energy management agent with load forecasts in the reinforcement learning environment.

7.1.1 Metrics of evaluation

To compare the forecasting performance of the tested models, two primary metrics were employed.

- Root Mean Square Error (RMSE): Since this measures the average magnitude of the errors between predicted and observed values, it is a useful metric for gauging what the error is in the original units. For example, an RMSE of 43.2 would indicate that on average there is an error of 43.2 kW in the load forecast.
- MAPE: This provides a relative measure of the error in terms of a percentage, making it easier to interpret and compare across examples and scales.

It is expected that the accuracy of forecasts will drop as they progress further into the future. For this reason, the RMSE and MAPE were evaluated at every point in the forecast to observe the trend across the forecasting window.

7.1.2 Forecasting procedure

It was necessary to obtain many forecasts for each model in order to achieve an accurate idea of how the model performs. In order to do this the following steps were taken:

1. Make a 24-hour ahead forecast for the current time step
2. Record the forecasted load values (all 24 values)
3. Store the actual values corresponding to those predictions (24 values)
4. Advance the time series by one step
5. Repeat steps 1-4 until a 24-hour-ahead forecast has been made for every value in the testing dataset.

The output is a cell array that stores the associated forecast and actual value for every time step. This array can then be used to calculate the RMSE and MAPE.

When attempting to determine the mean MAPE and RMSE values over the 24-time step window, an interesting challenge presented itself. Due to the large gradients present in some time steps (typically

morning and evening as power consumption changes rapidly), if the model predicts this change one time step too early or too late, as illustrated in the figure 7.1 the corresponding time steps absolute percentage error would be extremely large as can be seen at time step 5 in figure 7.2. This large spike will contribute heavily to the mean value of the MAPE and RMSE over the prediction period.

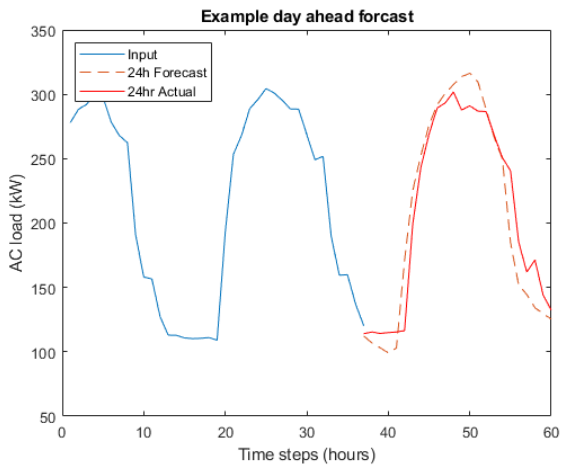


Figure 7.1: Example 24 hour ahead forecast

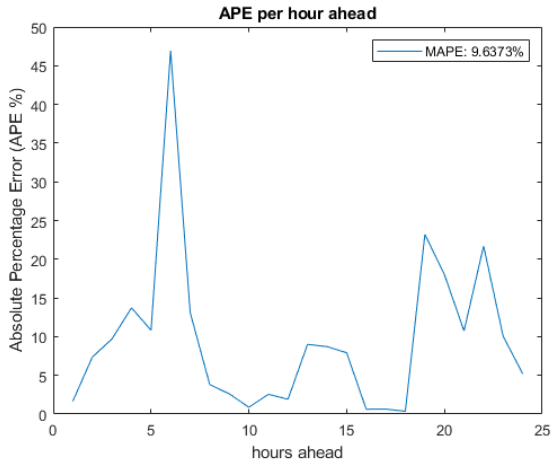


Figure 7.2: Large spike in error due to rapidly changing load.

In order to get a more accurate idea of the total errors, equations 7.1 and 7.2 integrate actual and predicted values over the forecast window to estimate total MAPE and RMSE. This mitigates the impact of the large per-step error spikes. While useful, this approach has limitations: it may underrepresent errors if the shapes of the actual and predicted curves differ. However, from visual inspections of the forecasts it is known that their shapes are similar. Therefore, these equations offer additional but not standalone insight into forecast errors.

$$total\ forecast\ MAPE = \frac{|\int_1^{24} Actual - \int_1^{24} Predictions|}{\int_1^{24} Actual} * 100 \tag{7.1}$$

$$total\ forecast\ RMSE = \sqrt{(mean\{[\int_1^{24} Actual - \int_1^{24} Predictions]^2\}} \tag{7.2}$$

7.1.3 Evaluation of the LSTM model

The MAPE for both AC and DC loads starts below 10% and is shown in figure 7.3. The AC MAPE flattens around the 4th time step, whereas the DC MAPE does so around the 10th.

Figure 7.4 shows that the initial RMSE for AC is higher than for DC. The AC RMSE's rate of increase changes at the 4th time step, while the DC RMSE rises more steeply initially but plateaus sooner.

The key metrics are summarised in table 7.1. The row 'Mean step' is simply the average value of all the per-step RMSE and MAPE values. The total column is the average over the forecasted window calculated using equations 7.1 and 7.2.

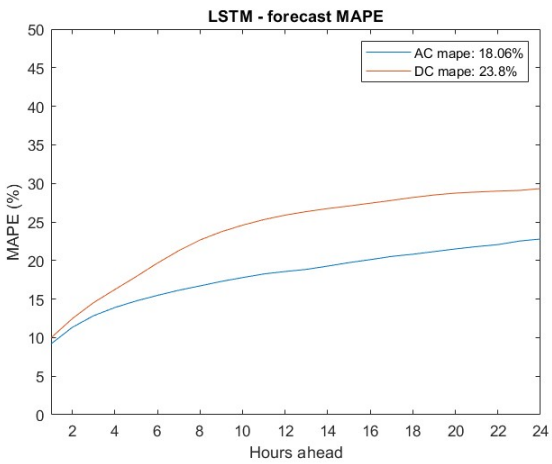


Figure 7.3: The MAPE per time step of the day ahead LSTM forecasts

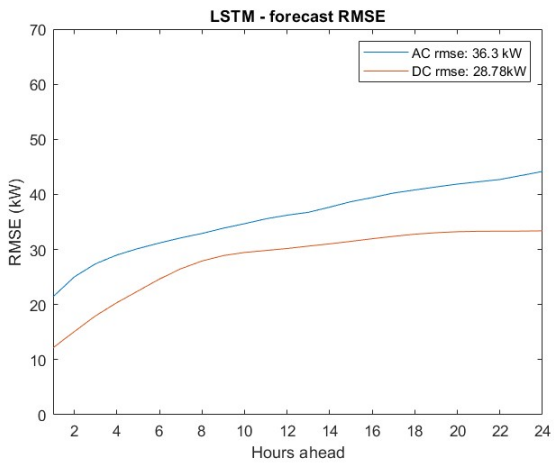


Figure 7.4: The RMSE per time step of the day ahead LSTM forecasts

Table 7.1: Summary of results for the LSTM network.

	AC: **MAPE (%)**	DC: **MAPE (%)**	AC: **RMSE (kW)**	DC: **RMSE (kW)**
1 step ahead	9.18	9.97	21.42	12.15
Mean	18.06	23.80	36.30	28.78
Total	11.42	18.67	525.55	436.41

7.1.4 Evaluation of the SARIMA model

Figure 7.5 shows the MAPE curves for both AC and DC loads. Both start low and spike until the 4th time step, then plateau. The AC load's MAPE is 8.74%, with a per-step MAPE of 16.82%. Using Equation 7.1, the total error is around 11.05%, likely due to the well-matched shape of the SARIMA forecast shown in Figure 6.13. In contrast, the DC load starts with a 10.27% error and has higher per-step and total errors — nearly double the AC total error.

Figure 7.6 reveals a similar trend in RMSE, with an initial steep rise and subsequent plateau. The AC RMSE is higher than the DC RMSE due to its higher mean, making numerical comparisons less straightforward.

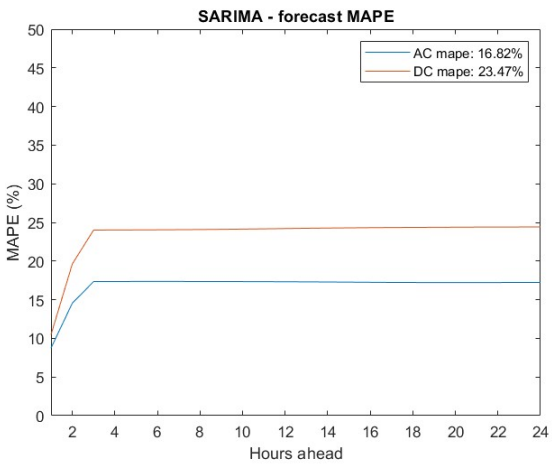


Figure 7.5: The MAPE per time step of the day ahead SARIMA forecasts

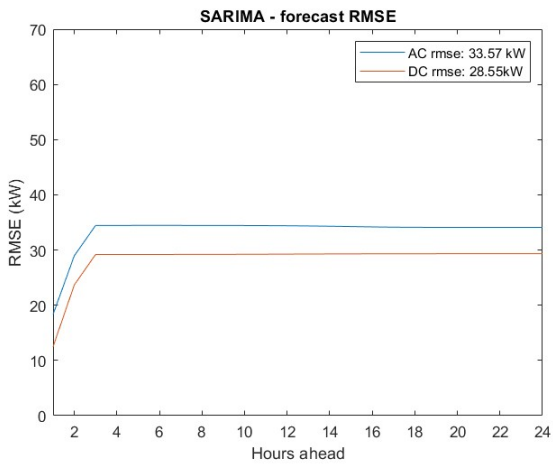


Figure 7.6: The RMSE per time step of the day ahead SARIMA forecasts

Table 7.2: Summary of results for the SARIMA network.

	AC: **MAPE (%)**	DC: **MAPE (%)**	AC: **RMSE (kW)**	DC: **RMSE (kW)**
1 step ahead	8.74	10.47	18.31	12.46
Mean step	16.82	23.47	33.57	28.55
Total	11.05	22.00	525.548	459.14

7.1.5 Evaluation of the Neural Net model

Figure 7.7 displays the MAPE for the neural net regression model, starting at approximately 7.5% for both AC and DC loads. The error peaks at the 12th time step, with AC peaking at around 30% and DC at 45%.

Figure 7.8 shows the RMSE, following a similar pattern to the MAPE. The AC RMSE is higher due to scale differences. Key metrics are summarised in Table 7.3

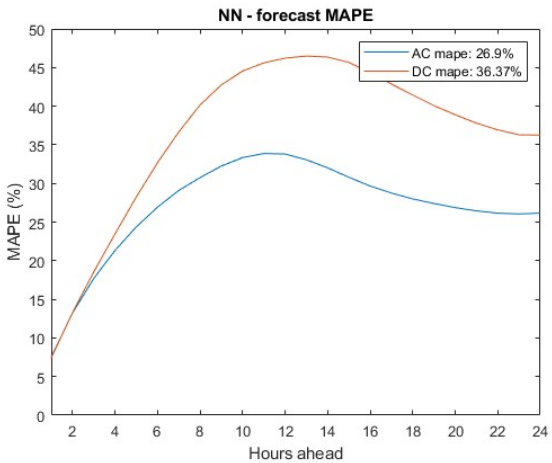


Figure 7.7: The MAPE per time step of the day ahead NN forecasts

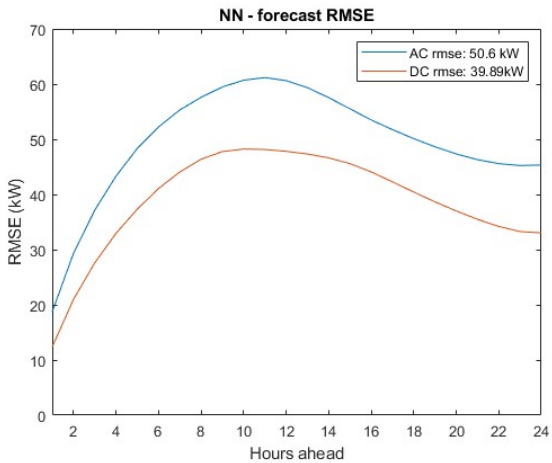


Figure 7.8: The RMSE per time step of the day ahead NN forecasts

Table 7.3: Summary of results for the NN network.

	AC: **MAPE (%)**	DC: **MAPE (%)**	AC: **RMSE (kW)**	DC: **RMSE (kW)**
1 hour ahead	7.52	7.40	18.75	12.37
Mean step	26.90	36.37	50.60	39.89
Total	14.69	25.46	637.20	564.45

7.2 Load forecasting discussion

7.2.1 Model performance

The SARIMA model excelled in most key metrics, except for 1-step ahead predictions. Its forecast errors were also the most stable, yielding the best overall results.
While the neural network performed well in 1-step ahead predictions with a 7.5% MAPE for both AC and DC loads, it struggled with capturing the 24-hour seasonal pattern. This limitation was evident at the 12-hour ahead step, where errors peaked.

The LSTM network better captured the seasonal relationship, but SARIMA stood out for its accuracy in modelling seasonal load profiles, particularly in later time steps.

7.2.2 AC and DC load variablity

Across all examined models, the DC forecast consistently underperformed in comparison to the AC forecast. This observation prompted an investigation into the variability of both load profiles. Day-to-day and step-to-step variabilities are two metrics that can be used to examine variation in the collected load data.

The day-to-day variability is calculated by first determining the mean power usage for each day in the dataset, denoted as μ_{daily}. Subsequently, the difference between consecutive daily mean values, $\Delta\mu_{\text{daily}}$, is calculated as:

$$\Delta\mu_{\text{daily}} = \mu_{\text{daily},i} - \mu_{\text{daily},i-1} \tag{7.3}$$

The standard deviation of these differences, $\sigma_{\text{day-to-day}}$, is then found and converted into a percentage by dividing it by the overall mean of the dataset, μ_{overall}:

$$\text{Day-to-Day Variability (\%)} = \left(\frac{\sigma_{\text{day-to-day}}}{\mu_{\text{overall}}}\right) \times 100 \tag{7.4}$$

Similarly, the step-to-step variability is calculated by finding the difference in values between consecutive time steps, ΔP:

$$\Delta P = P_i - P_{i-1} \tag{7.5}$$

The standard deviation of these differences, $\sigma_{\text{step-to-step}}$, is computed and converted to a percentage by dividing it by μ_{overall}:

$$\text{Step-to-Step Variability (\%)} = \left(\frac{\sigma_{\text{step-to-step}}}{\mu_{\text{overall}}}\right) \times 100 \tag{7.6}$$

The data in Table 7.4 indicates that the collected DC load data has substantially higher variability in both day-to-day and step-to-step measures. This increased variability likely contributes to the consistently poorer performance of models in forecasting the DC load compared to the AC load. The step-to-step variability is significantly lower relative to the day-to-day variability. This may shed light on why certain models, particularly those lacking a built-in understanding of seasonal patterns—such as regression neural networks—could accurately predict one-step-ahead loads but struggled to forecast loads for entire days without incurring significant errors.

Table 7.4: Step-to-step and day-to-day variability of the loads.

Load	Step-to-step variability (%)	Day-to-day variability (%)
AC load	10.536	14.7
DC load	15.21	30.017

7.2.3 Forecast stability

Forecast stability refers to the consistency in predicted values for a specific future time step, even as new forecasts are made at closer intervals to that time step. Essentially, if the forecasted value for a

future time remains relatively unchanged despite newer predictions, it's deemed stable.
To visualise the stability of forecasts as a specific time step is approached, a heatmap shown in figure 7.9 was constructed. In this visualisation, lighter shades correspond to higher power generation, and darker shades indicate the opposite. The y-axis represents the forecasted time steps ahead, while the x-axis tracks the progression of time. A close examination of the heatmap reveals that forecasts maintain remarkable stability up until just a few time steps ahead. This does not indicate that the forecasts for the first few time steps ahead have a higher error, just a higher instability. This transition point around 4 steps ahead aligns with the sharp flattening of the MAPE in figure 7.5.

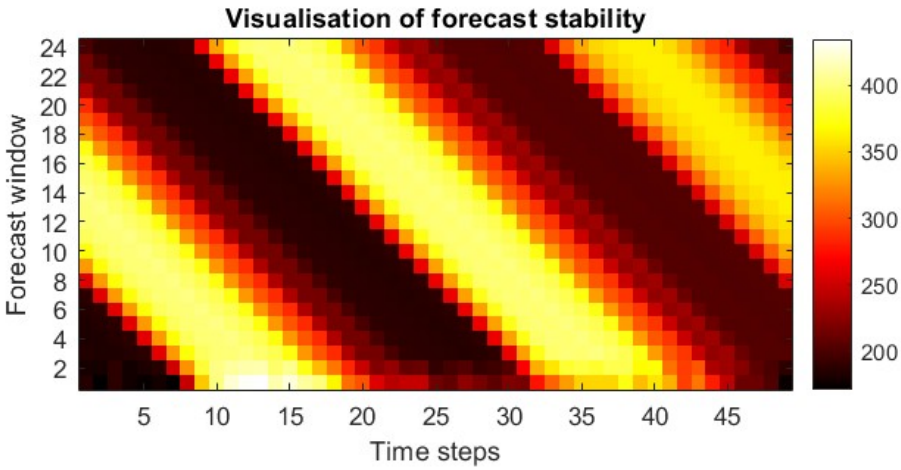


Figure 7.9: Visualisation of the forecast stability over multiple time steps

7.2.4 Final load forecasting model

From these evaluations, the SARIMA model notably surpassed the LSTM and Neural Net models, evidenced by its lower Mean-per-step and total MAPE and RMSE values. Moreover, the SARIMA model consistently yielded stable forecasts. Given these merits, it was chosen as the candidate for integration with the broader system. Consequently, forecasts were generated for every time step within the dataset and subsequently supplied to the energy management agent within the reinforcement learning environment

Chapter 8

Energy management agent

This chapter delves into the application of RL to create an autonomous EMS. Building upon the hybrid microgrid that was designed in chapter 5, the microgrid was transformed into an RL simulation environment. This simulated microgrid could then be run over a span of a year, with hourly time steps, employing the data accumulated as described in chapter 4. A crucial aspect of this system was the agent's access to load forecasts, which have been previously established in chapter 6. Before delving into the actual implementation, we will briefly discuss the application of RL theory to the energy management problem, setting the stage for the subsequent steps.

8.1 Reinforcement learning applied to EMS

Building upon the discussion in the RL theory section 3.2, it is clear that RL offers a promising tool for training agents to navigate complex environments. In the context of this study, the agent represents the EMS, while the microgrid serves as the environment. Through iterative training, the agent should learn to master energy management, aiming to reduce penalties for inefficient actions. A visualisation of this application can be seen in figure 8.1.

The environment's configuration aligns closely with the parameters delineated in the hybrid microgrid design chapter, with some simplification such as ignoring electrical losses. This includes the AC and DC loads, PV array, wind turbines, diesel generator, battery system, and converters.

For the agent to make informed actions, it was furnished with current and forecasted observations from the environment. Immediate data would include current loads, generation metrics, and battery levels. Meanwhile, predictive observations comprise load forecasts, generation forecasts, upcoming energy pricing, and the load shedding schedule.

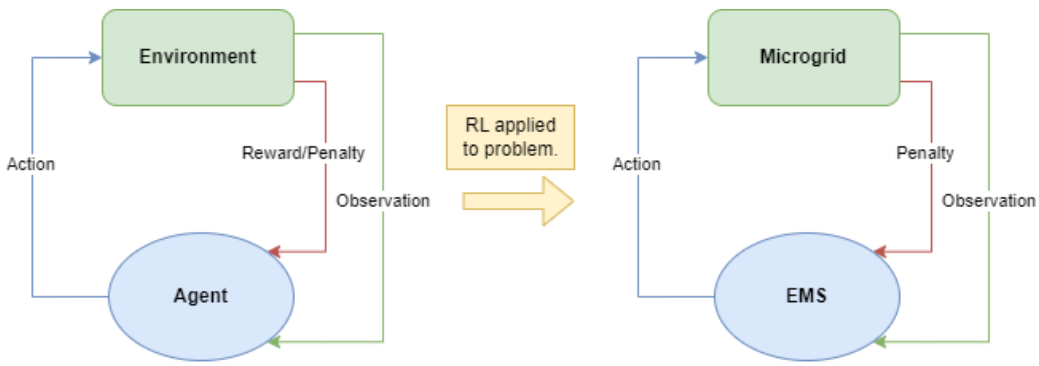


Figure 8.1: RL applied to the challenge of energy management.

Implementation of reinforcement learning
The implementation process evolved iteratively, with each iteration adding complexity to the RL simulation. Initially, the environment featured a singular load, one source of generation, and basic TOU rates. As updates were rolled out, new features were introduced, such as load shedding, hybrid loads and generation. The final layer of complexity was added with the integration of a demand charge.
Development steps

1. Initiate with a basic environment.
2. Train the agent.
3. Test the agent's performance.
4. Enhance the environment by adding a new feature.
5. Repeat steps 2-4, refining until the desired environment complexity is achieved.

To clarify, the load and generation are assumed to be constant for the duration of the time step. Due to the hourly resolution of the data, to convert from kW to kWh the load and generation values can simply be multiplied by 1h.

8.2 Simplified environment

To clarify the development of the RL approach there are some key terms used in the stable-baselines3 library that should be elaborated on.
The 'environment' is the term that refers to the class that initialises the space that the agent will explore and learn to navigate. It also defines key functions that are used to simulate the behaviour of the environment and facilitate interaction between the agent and the environment.
A brief summary of them is as follows:

- **Observation space**: Represents the data that the agent receives about the environment. This is used for the agent to make a decision on its course of action.
- **Action space**: This defines the set of actions that the agent can take.
- **Step function**: Invoked after the agent selects an action. It administers the selected action, updates the environment, and updates the agent's observation space accordingly.
- **Reset** This function resets the environment and readies it for a fresh training iteration or an evaluation run.

Other standard functions that can be integrated into the library include the render and the close function, neither of which were necessary for this project. The upcoming subsections detail the step-by-step process that was followed to develop a simplified environment in alignment with the guidelines provided by the stable-baselines3 documentation.

8.2.1 Environment

While drafting the initial version, the emphasis was on simplicity while retaining the fundamental attributes of a microgrid and the EMS
To streamline the design:

- Only one load was incorporated, opting for the AC load data, while the DC load was omitted.
- In terms of generation, only the output from the PV array was factored in, ignoring the contributions from the wind turbines.
- System intricacies, such as converter losses, load shedding, and demand charges were set aside.

The primary objective for this rudimentary EMS agent revolves around proving its ability to reduce energy costs incurred by grid purchases. It serves as a proof of concept, showing the agent's potential to recognise the financial advantages of purchasing and storing energy during off-peak times to avoid peak prices.
These simplifications paved the way for an initial test environment. Below is an outline detailing the steps undertaken to establish the environment. For an in-depth exploration, the complete code can be accessed here.

8.2.1.1 Observation space

The observation space defines the information that the agent can perceive about its environment. It can be imagined as the agent's eyes into the world that it exists in.
The initial observation space was defined in a dictionary format (Dict), where each entry has a key and an associated expected value range, shape, and data type. It contained the following entries:

- **Load forecast:** The upcoming 24 values of the AC load.
- **Generation forecast:** The upcoming 24 values for PV generation.
- **TOU rate forecast:** The current and upcoming 24 TOU rates.
- **Battery level:** The battery's current state of charge.
- **Current load:** The current value of the AC load.
- **Current generation:** The current value of the PV generation.

It's crucial to emphasise that in these initial versions, the load forecasts are not based on predictive models. Instead, they directly use the upcoming 24 actual values for the load and generation, giving the model perfect foresight in this phase. Furthermore, the TOU rate forecast doesn't currently provide the monetary value of energy cost. It instead offers the time-of-use ID corresponding to each hour, distinguishing between off-peak, standard, and peak periods.

8.2.1.2 Action space

The action space defines the actions that the agent can take within the environment. In this initial version, two actions are available to the agent: **standby** or to **fully charge its batteries**.

Standby: Opting for the standby action allows the agent to rely on a pre-established, rule-based system to manage the grid. This is a brief summary of how the standby action unfolds:

- Renewable generation attempts to meet the load demand.
- Excess generation is used to charge the battery.
- The battery is used to meet any shortfall in generation.
- If the battery level falls below its minimum threshold, the remaining shortfall is purchased from the grid.

To avoid confusion, the agent is **not** tasked with orchestrating the process listed above. By choosing the standby action, the agent merely signals the microgrid environment to implement this set of predetermined rules. Refer to figure B.1 in Appendix B for a visual breakdown of the set of rules that were used in the final environment.

Fully charge batteries: This action is designed to be used if the agent forecasts impending power shortages during peak times and allows it to preemptively purchase and store electricity during off-peak times. When this action is chosen:

- Using equation 8.1 the total energy requirement is calculated as the sum of the current load and the energy necessary to fully charge the battery.

$$E_{total} = E_{load} + E_{ESS_charge_deficit} \tag{8.1}$$

- The net energy to be purchased from the grid is calculated by subtracting the current renewable generation from the total energy requirement as seen in equation 8.2.

$$E_{purchased} = max(E_{total} - E_{renewable_gen}, 0) \tag{8.2}$$

As a result, the battery will always be recharged to its full capacity after this action has been selected. Note that load shedding is not present in the environment at this stage so the grid is assumed to always be available.

8.2.1.3 Reward

The design of the reward function, shown in equation 8.3, serves to encourage the agent to minimise energy purchase costs. Specifically, the reward for an action is computed as the product of the energy purchased from the grid during that step and the corresponding TOU multiplier. To reflect the objective of minimising energy costs, the reward is negated. Thus, in the context of the agent's learning, it is perceived as a penalty to be minimised rather than a reward to be maximised.

$$\text{Reward} = -(E_{purchased} \cdot TOU) \tag{8.3}$$

As mentioned the TOU rates at this stage did not relate to a monetary value but were designated as follows:

- Off-peak = 1
- Standard = 2
- Peak = 3

8.2.1.4 Step function

Each step progresses the agent one time step through the environment (one hour). After an agent chooses an action, the action is applied to the current step. The new observation state is calculated along with the reward function. Both are then fed back to the agent for it to decide on its next action. The step function is a core mechanism within the RL framework. Every time it is invoked, it advances the agent by one time unit in the environment, which in this context, equates to one hour.

- It implements the 'standby' or 'fully charge battery' function depending on the agent's selection.
- The new observation for the agent is determined
- Concurrently, the reward for the selected action is calculated.
- The new observation space and the reward are relayed back to the agent for it to select its next action.

8.2.2 Results of EMSv0.1

This initial version, despite its simplicity, yielded promising outcomes. For training, 70% of the available data was allocated to the environment. The agent was trained using various algorithms, such as the Proximal Policy Optimiser and the Actor Agent Critic, over 1.5 million time steps.

Subsequently, the agent was tested in a separate environment, utilising the remaining 30% of the dataset. To establish a baseline, a reward was computed by simulating an entire episode of the environment in standby mode. This baseline reward approximates the agent's performance if it opted solely to purchase energy as needed with no planning.

A simulation was then run with the trained agent making decisions. The agent showcased its ability to increase its reward, effectively reducing the penalty associated with energy costs by 8%. Although this improvement cannot be equated to monetary savings in this phase, it undeniably indicates the agent's proficiency in making preemptive electricity purchases, thereby avoiding peak prices at a later time step.

This promising behaviour is evident upon reviewing figure 8.3 In this figure, the green bars represent the trained agent's data, while the red bars indicate the higher penalties incurred without the agent. Notably, the agent has reduced penalties during both standard and peak times, while increasing the energy during off-peak times. Consequently, the agent has successfully achieved a reduced overall penalty.

Figure 8.2: Reward accumulated without an agent vs with a trained agent.

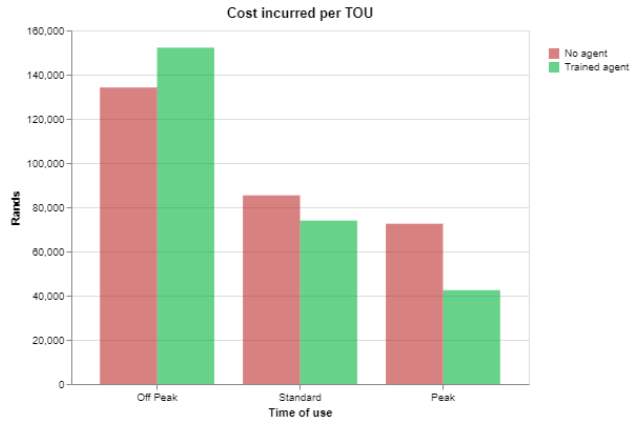


Figure 8.3: The penalty incurred by the agent per time of use category.

8.3 Development of the environment and agent

The development of the RL microgrid environment followed an iterative approach. With each iteration, the system was tested against a base case to confirm its ability to energy cost savings. While it's beyond the scope of this section to delve into the details of every iteration, a summary of the iterations and additions can be found in table B.1 in Appendix B.
Several milestones punctuated the development journey. These include the incorporation of load shedding, the handling of both AC and DC loads, and generation - central to the hybrid nature of the microgrid as envisioned in the project scope. Subsequently, tangible monetary costs were introduced, followed by the integration of a demand charge.

8.3.1 Implementation of a hybrid grid and load shedding

This version incorporated the AC & DC loads, Wind & PV generation, and UCT's 2022 load shedding schedule. A diesel generator, limited to 100kw, was introduced in accordance with the hybrid microgrid design. The rule-based structure was adjusted to account for these changes. Most notably if there is an energy shortage and the utility grid is unavailable, the generator activates. Any additional shortage is recorded as unmet load. Details of the final rule scheme are in Appendix B, figure B.1.

The reward function was updated to include a penalty for unmet demand and the cost of diesel generation.
The agent adapted well to this environment, surpassing baseline rewards as can be seen in figure 8.4 and 8.5. The agent was able to reduce operating costs by 10%.

Figure 8.4: Reward accumulated without an agent vs with a trained agent in environment v1.2.

Figure 8.5: Total cost incurred by the agent per time of use category in environment v1.2.

With the implementation of load shedding, the agent proactively purchases more energy in anticipation of grid unavailability. Although this increases the total energy purchases, savings in the resulting reduction of diesel generator use offset this. Notably, the agent's strategy results in a 75% reduction in unmet load, which is substantial. Figure 8.6 visualises the increase in energy purchases across the TOU rates, and figure 8.7 illustrates the large reduction in generator usage and unmet demand.

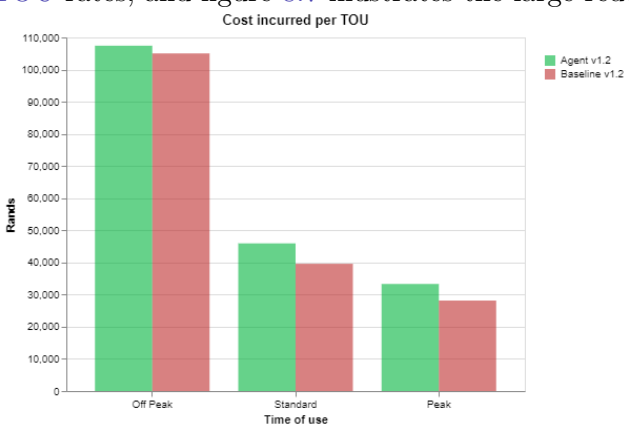


Figure 8.6: The cost of purchases by the agent per time of use category in environment v1.2.

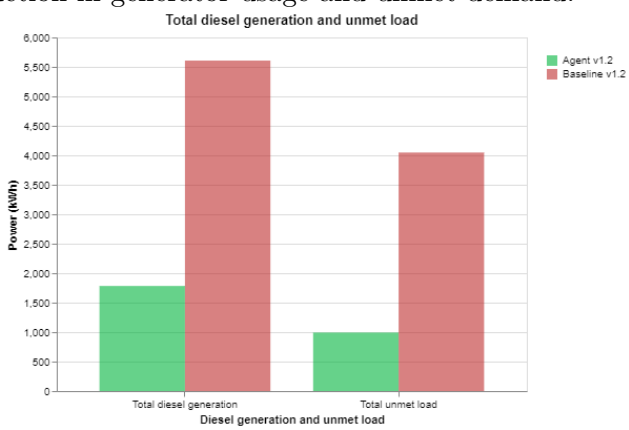


Figure 8.7: Total diesel generation and unmet load in environment v1.2.

8.3.2 Implementation of a demand charge and continuous action space.

A demand charge is a monthly fee determined by the highest power draw of the month, multiplied by a specified demand rate. This implies that if the agent purchases a large quantity of electricity within one time step, it would face a hefty demand charge.

In the study, implementing a demand charge posed a challenge, as the initial RL environment only allowed the agent to make large energy purchases in one time step. To address this, a continuous action space was introduced, but the agent initially struggled to adapt. Modifications from the stable-baselines3 library were employed to improve the agent's learning, including normalising values and transitioning the observation space to a 1D vector.

Upon removing the demand charge to test the agent's learning, it successfully managed to minimise energy expenses in the continuous action space. When the demand charge was reinstated, further attempts to use a demand penalty failed to prevent large energy purchases. A direct solution was ultimately applied: a cap was placed on the amount of energy the agent could request, preventing it from exacerbating the maximum demand while still meeting microgrid energy needs.

By this phase of development, the RL environment was judged to sufficiently align with the study's complexity scope. The subsequent section offers an in-depth overview of the environment's final version.

8.4 Final RL implementation

This section details the final design of the environment in which the agent learns to manage energy. Additionally, it provides an overview of the training methods of the agent.

8.4.1 Environment

The culminating environment incorporates both AC and DC loads, solar PV, and wind generation capabilities, and is equipped with a 500kWh battery and a diesel generator. Furthermore, it integrates load shedding, TOU rates, and a demand charge.
All input data spans an hourly interval over a year. As detailed in previous chapters, data pertaining to loads, generation, time of use rates, and load shedding history was meticulously amassed. The initial training strategy aimed to utilise the initial 70% of this data as a foundation for the training environment and reserved the subsequent 30% for evaluation purposes. This approach soon presented a challenge.
Figures 8.9 and 8.8 make clear the disparities inherent in this data split. The winter TOU rates, which are considerably higher than their summer counterparts, are wholly confined to the training environment. Contrarily, the load shedding data presents an imbalance with 19 instances during training, compared to a staggering 189 occurrences in the evaluation phase.

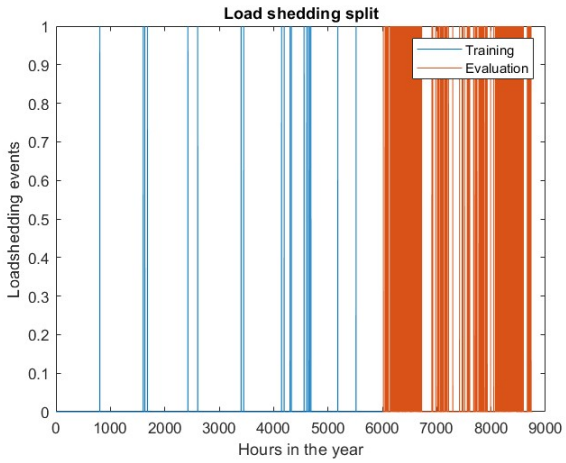


Figure 8.8: Load shedding events at UCT in 2022.

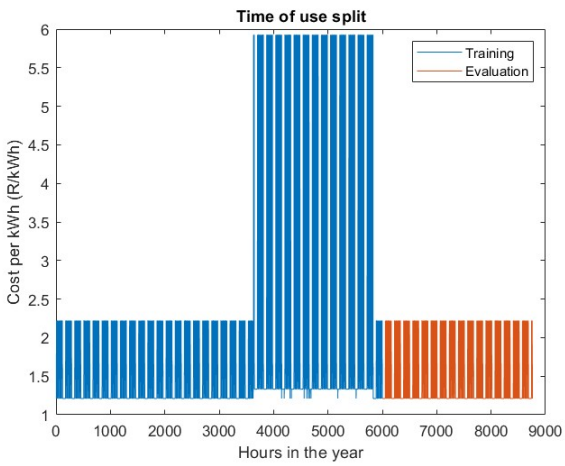


Figure 8.9: UCT's time of use price schedule for 2022.

To rectify the issue, the training environment was populated with the first half of each dataset (load, generation, and TOU), while the evaluation environment incorporated the latter half. This strategy aimed to balance the seasonal effects, ensuring the agent was trained on both summer and winter data and subsequently evaluated on similar variations. However, this approach means that only half the data was utilised for training purposes.

The treatment of the load shedding data deviated slightly from the above method. Given that the first half only recorded 19 load shedding incidents, it was deemed unsuitable for training. Instead, the more event-rich second half was employed in both the training and evaluation environments.

8.4.2 Observation space

The observation space was streamlined while preserving essential information. This design choice aims to aid the agent by simplifying relationships wherever possible, thereby reducing the number of relationships it needs to learn.
To this end, the load and generation values were all simplified into a single 'power balance' variable. The resulting final observation space was as follows:

- **Current power balance:** The microgrid's current power balance, where a negative power balance denotes that the load exceeds the demand.
- **Power balance forecast:** The upcoming 24 power balance values.
- **TOU forecast:** Incorporates the actual TOU tariffs relevant to the current time step and its forecast window.
- **Load shedding forecast:** Serves as an indicator for load shedding over the forecast window. A value of 1 signifies the occurrence of load shedding, whereas a 0 denotes the utility grid's availability.
- **Battery level:** The current battery level.

A key technical change that resulted in improved performance was the implementation of linear scalers. Before sending the observation space to the agent, all values that were measured in kWh were scaled using the same linear transform to be between the values of -1 and 1. Additionally, the TOU forecast

was scaled to fall between 0 and 1.

8.4.2.1 SARIMA load forecasts

The load forecasts in the final RL environment are sourced from the SARIMA forecasting model. This model was extensively developed in chapter 6 and evaluated in chapter 7.

Forecasts were structured into a 2D array. Each row of the array corresponds to a time step in the environment (one for every hour), and the 24 columns represent forecast values for each of the subsequent 24 hours. A MATLAB script containing the SARIMA model was used to populate this array for every time step. Subsequently, the forecasts were imported into Python for use in the RL environment where they were incorporated into the 'power balance' calculations.

8.4.3 Action space

The finalised action space was set to be a continuous range between -1 and 1. The value of -1 to 1 was selected since the stable-baselines3 library uses a Gaussian distribution to calculate the probabilities of taking an action. Scaling the action space to this range is recommended in their documentation[39].

When the agent provides an action, it is scaled up, spanning from 0kWh to 300kWh — denoting the maximum energy the agent can request to purchase.

This upper limit was informed by knowledge of the microgrid and an understanding of the required purchasing power that would be impactful without increasing peak demand. It's essential to note that this action is not an absolute command. It represents the agent's purchase request. If during the step function, it is calculated that more energy is required to maintain a positive energy balance the microgrid will follow its rule-based scheme and purchase additional energy. In this manner, all actions remain constrained by the microgrid rules explained in the step function and depicted in figure B.1.

In operational terms, the agent's action enables strategic energy purchases, allowing for battery charging during off-peak times. Enabling the agent to prepare ahead of impending load shedding events or generation shortages.

8.4.4 Reward

The final reward structure was designed around three components: purchased energy, diesel generation and unmet load. Efforts to incorporate a demand penalty aimed at reducing the monthly demand charge into the reward were unsuccessful. Despite various approaches, such as a rolling max demand penalty or an exponential demand penalty, none were able to improve the agent's performance.

Therefore, the components of the final reward structure are detailed as follows:

Purchased energy: The cost of purchasing energy per time step is calculated from the amount of

energy purchased multiplied by the TOU rate for that step. As seen in equation 8.4:

$$C_{purchased} = E_{purchased} \cdot TOU \tag{8.4}$$

Where:

- $C_{purchased}$ is the energy cost incurred via purchases for that step.
- $E_{purchased}$ is the energy purchased in kWh.
- TOU is the cost per kWh in rands for that step.

Diesel generation: The cost of operation for a diesel generator is primarily influenced by the amount of diesel consumed and its price. This relationship is expressed in equation 8.5:

$$C_{diesel} = E_{diesel_gen} \cdot l \cdot r \tag{8.5}$$

Where:

- C_{diesel} is the diesel cost incurred for that step.
- E_{diesel_gen} is the energy produced by the diesel generator in kWh.
- l denotes the litre/kWh, which was set to 0.4l/kWh [45].
- r is the cost per litre of diesel in rands, set to R23[1]

Unmet load: This component quantifies the effect of unmet load in a given step. To successfully motivate the agent to reduce unmet demand, the penalty was raised from the initial R5/kWh used in the HOMER Pro design to R10/kWh. Its design aims to add a penalty that reflects any service disruption.

The total monetary cost per step, shown in equation 8.6, is then the summation of energy purchase costs and diesel generator expenses. Where C_{tot} denotes the total monetary cost incurred for that time step.

$$C_{tot} = C_{purchased} + C_{diesel} \tag{8.6}$$

The final reward was then calculated as the summation of the monetary cost and the unmet load penalty, is detailed in equation 8.7.

$$R = -(C_{tot} + E_{unmet} \cdot 10) \tag{8.7}$$

Where R denotes the reward for that step and E_{unmet} the unmet load in kWh.

8.4.5 Step function

An important function in the final version of the environment was the function that managed energy flow in the grid. It followed a rule-based system and implemented the agent's actions. As discussed under the action space the agent could request to purchase any amount of energy within the range of 0kWh up to an upper limit set to 300kWh. It manages the power distribution of the AC and DC grid,

[1] https://www.globalpetrolprices.com/South-Africa/diesel_prices/

the grid connection, and the diesel generator. Below is a high-level overview of the method, a more detailed flow chart is seen in figure B.1 in Appendix B.

1. Initial setup:
   - Fetch the AC and DC power balances.
   - Determine the current battery level.
   - Un-normalise the agent's purchase request.
2. Purchase power:
   - Check if the grid is available and purchase the agent-request energy.
   - If the grid is not available no energy may be purchased.
3. Power balance calculation:
   - Compute the total grid power balance using the AC and DC power balance and the additional power purchased.
4. Power allocation:
   - If there is a surplus of energy then the battery is charged with the surplus.
   - If the battery reaches its capacity the remaining surplus is recorded as excess generation.
   - If there is a power deficit then available energy from the battery is used.
   - If the battery reaches its low charge threshold the agent will attempt to purchase electricity from the grid to meet the deficit.
   - If the grid is not available it will use the diesel generator.
   - If the diesel generator does not have sufficient capacity to meet the deficit the remaining deficit will be recorded as unmet demand.

The final code that was used to implement the RL environment can be found here

8.4.6 Training method

The PPO algorithm was chosen for training the agent, primarily due to its early-stage training advantages compared to other models, as illustrated in Figure 8.10. The agent underwent training for a total of 1,500,000 simulation time steps. Further training was found to negatively impact the model's ability to generalise to the testing dataset.

A learning schedule was implemented to adjust the algorithm's learning rate, which started at 0.01 and gradually decreased to 0 as a function of the training episode's time steps. This technique was designed to promote exploration in the agent's early training phases while gradually reducing the exploration rate to improve performance consistency.

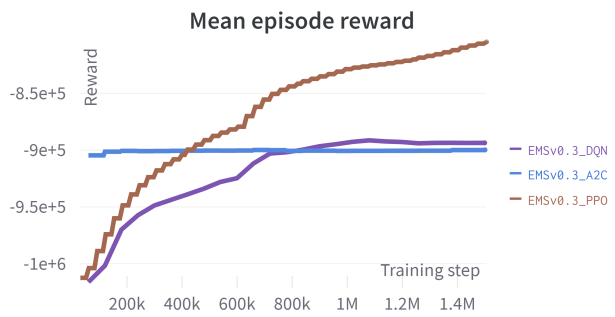


Figure 8.10: Mean episode reward of PPO, A2C, and DQN algorithms

Chapter 9

Results & Discussion

9.1 Evaluation metrics

The metrics that were used to evaluate the agent's performance were the total monetary cost, the unmet demand, and the amount of diesel generation used. The excess energy generated is also examined although this metric was not directly included in the design of the reward structure.

In order to measure the agent's performance, a baseline test was performed. The baseline test entailed simulating the microgrid without requesting to purchase any electricity, meaning that the microgrid would be governed by the rules that were summarised in section 8.4.5 and expanded on in the flow chart found in Appendix B, figure B.1.

9.2 Results

This section outlines the key results that the final energy management agent was able to achieve. Both cases with load shedding and without are evaluated in order to understand the agent's ability under varied conditions.
A summary of the important metrics is shown in table 9.1. These values are examined in closer detail in the remainder of this section.

Table 9.1: Summary of final results

	Total Cost (R)	Unmet demand (kWh)	Diesel Generation (kWh)	Excess Generation (kWh)
No load shedding				
Baseline	982392.25	0	0	437520.82
Agent	929576.61	0	0	441827.28
Percent decrease	5.38%	NA	NA	-0.98%
With load shedding				
Baseline	1023396.93	4217.12	5947.40	437520.82
Agent	960732.18	0	83.18	454411.43
Percent decrease	6.12%	100%	98.60%	-3.86%

9.2.1 Performance of the agent with no load shedding

The agent successfully learned to minimise the total cost of purchasing energy in an environment with a stable grid (no load shedding events), as seen in figure 9.1. The green bars show the performance of the agent while the red bars represent the baseline case.
As seen in figure 9.2 the agent was able to reduce the overall expenditure by purchasing electricity during off-peak times, resulting in overall monetary savings of 5.38%.

Figure 9.1: Total money spent by the agent vs the baseline with a stable grid.

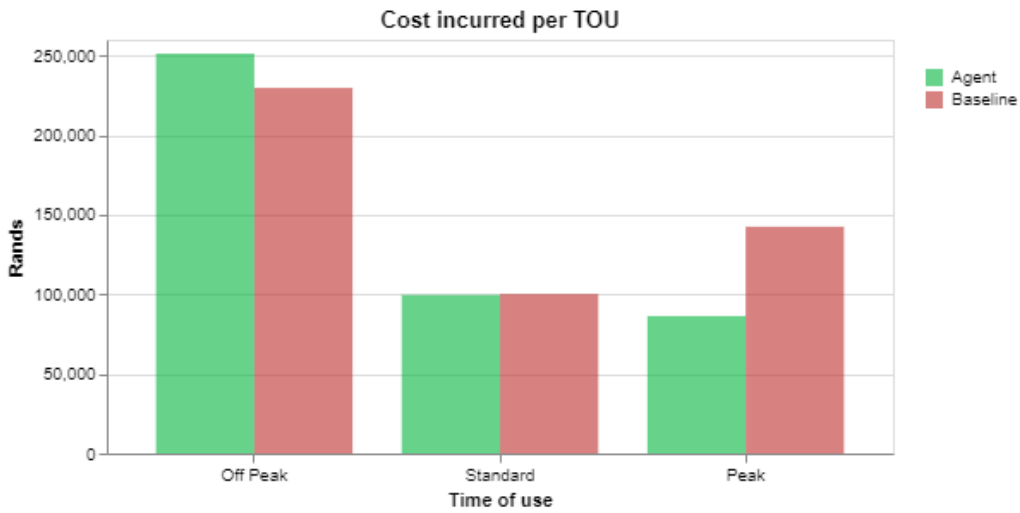


Figure 9.2: Money spent per TOU category.

9.2.2 Performance of the agent with load shedding

In environments that experience an unstable grid and load shedding, the agent was able to perform even better. It was able to reduce the total monetary expenditure by 6.12% as seen in figure 9.3.
As seen in figure 9.5 and 9.4, this reduction in cost was achieved by reducing the amount of money spent during peak times as well as drastically reducing the amount of money spent on diesel generation.

The diesel generation is mitigated to an almost negligible 83.18 kWh.

Figure 9.3: The total money spent by the agent vs the baseline case with load shedding.

The unmet demand was completely eliminated by the agent as seen in the second bar of figure 9.4. The agent was purchasing more energy to avoid using the diesel generator or risking unmet demand. Yet the agent was able to find significant savings by purchasing during the off-peak and even the standard TOU categories to avoid purchasing during peak times.

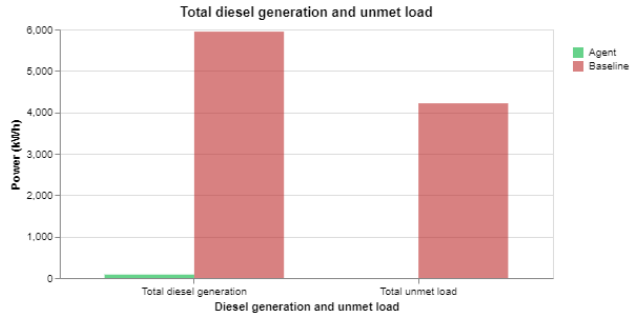


Figure 9.4: Maximum sizing metrics for the hybrid microgrid with load shedding.

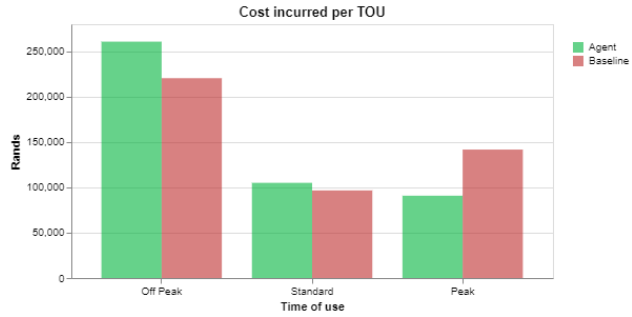


Figure 9.5: The cost of electricity purchased per TOU category with load shedding.

As expected the total purchased energy was increased by the agent. This is due to the agent purchasing energy before load shedding events when the grid would be unavailable. This is reflected in the increase in excess generation. The third column in figure 9.6 shows how much energy was requested by the agent. It is slightly less than the actual purchased amount indicating that the agent preferred to under-buy rather than over-buy.

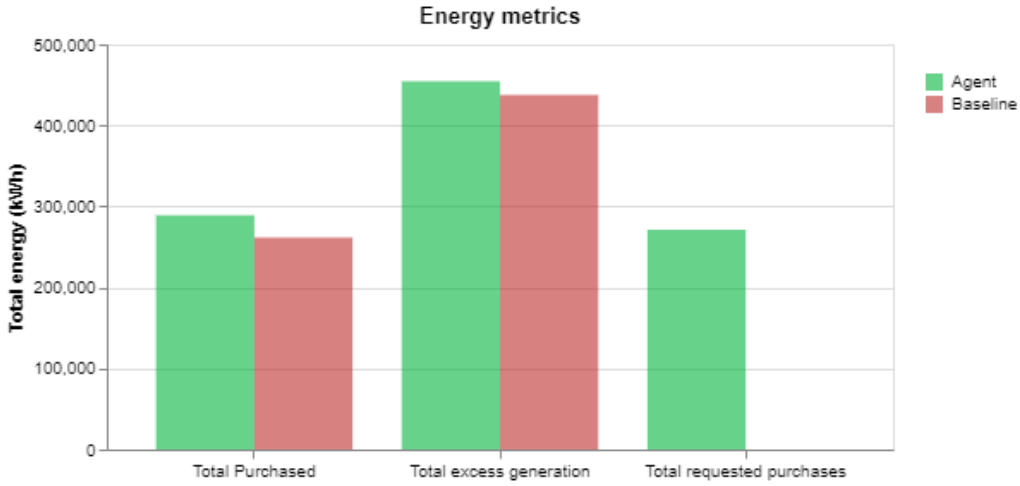


Figure 9.6: The cost of electricity purchased per time of use category with load shedding.

9.3 Discussion

This section examines and evaluates the extent to which each of the problem-solving tasks that were set out in the project scope was accomplished. Additionally, it goes on to explore the behaviour of the autonomous agent in more depth.

9.3.1 Hybrid microgrid design

This task functioned as a preliminary requirement for the two core engineering challenges, load forecasting and autonomous energy management agent. The most important aspect of the task was to design a feasible hybrid microgrid that could be used to model the RL environment. To this end the task was successful, providing a solid base for the next sections. It was enlightening to examine in detail the high energy costs that large buildings incur, and how these can be reduced with the installation of embedded generation and an ESS.
It is also worth mentioning that through effective energy management, the need for the diesel generator was almost completely eliminated, exposing a potentially over-designed microgrid. HOMER Pro allows for MATLAB code to be integrated with its optimiser, perhaps this is an avenue for the development of a more accurate sizing algorithm. One that incorporates more advanced energy management.

9.3.2 Load forecasts

A significant contributor to the agent's success was the accuracy of the load forecasts. With precise load forecasting, the agent was able to make informed decisions, anticipating future energy requirements and optimising its actions accordingly. The task can be said to have successfully met its goals.
However, for a more comprehensive analysis, extending the forecasting to include generation forecasts is desirable. In retrospect, at a minimum, introducing stochasticity into the generation forecasts provided to the agent would have been beneficial to simulate the inherent uncertainties in such forecasts.

9.3.3 Integration into an RL environment

Integrating the hybrid microgrid into an RL environment posed the greatest challenge of the study, with the environment going through 15 iterations. The environment successfully modelled power flow between the AC and DC subgrids, as well as the utility grid, while incorporating hybrid forms of generation and an ESS. Despite these advances, several complexities—such as battery degradation, sell-in tariffs, conversion losses, and battery charge and discharge rates—remain unaddressed.
Load forecasts were generated separately using MATLAB, and the forecasted values were integrated into the RL environment. While the final SARIMA model could technically be incorporated into the Python-based RL environment, this step was avoided to conserve computational time, with no compromise on the end results.
In summary, the study made substantial progress in modelling a hybrid microgrid within an RL environment. It successfully leveraged machine-learning-based load forecasts to achieve effective energy management, meeting its core objectives.

9.3.4 Efficacy of the final EMS agent

The final model implemented in this study demonstrated robust performance in all testing conditions. When operating with a stable grid and no load shedding it was able to save R52 895.23 over the span of half a year. Effectively making use of the load forecasts to preemptively purchase power during off-peak times.
Its performance was most notable when subjected to load shedding events. Over a span of half a year, the model led to substantial cost savings, amounting to R62 664.75. This result underscores the model's practical viability, as optimising energy costs is a universal challenge faced by commercial buildings.
Additionally, the model's ability to eliminate unmet demand (while saving money) is extremely useful in industrial settings, where any unmet demand can lead to huge losses if production is halted.

9.3.5 Behavioural insights

While the total metrics used in the results section demonstrate the agent's capability to effectively reduce cost, they offer limited insights into the agent's behaviour.
For a more profound understanding of the model's actions, one can examine figure 9.7. The top blue graphs show the power balance, which, being identical, confirms that both the agent and the baseline example experience equivalent conditions. The middle pink graphs portray the battery levels for the agent (left) and the baseline (right). The bottom tier shows the energy purchased, shaded in green, orange, and red to represent off-peak, standard, and peak TOU rates, respectively.

Evidently, the agent strategically charges its battery by purchasing electricity during off-peak periods, thus minimising purchases during peak times — a behaviour that aligns with expectations. Interestingly, the agent doesn't maximise battery charge and seems to underutilise off-peak rates. However, just before the TOU rate change (time step 15 to 18), the agent consistently purchases 300kWh, hitting its purchase request ceiling. This could suggest a potential advantage in raising its purchase limit. On the other hand, this behaviour validates the implementation of the upper limit, ensuring the agent does not inadvertently inflate the demand charge.

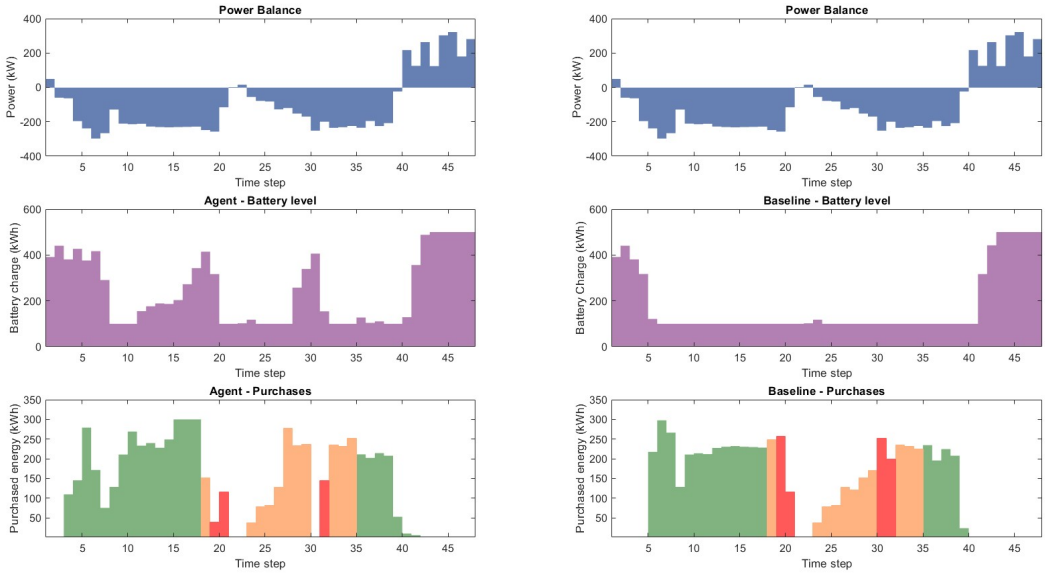


Figure 9.7: Purchasing behaviour of the agent.

In order to understand how the autonomous agent prepares for load shedding, one can refer to figure 9.8. In the graph, load shedding events are indicated by a red overlay on the power balance plots. Shortly before the 25th time step, the agent is observed to charge its batteries in anticipation of the impending load shedding. Contrarily, in the baseline case, the batteries are depleted, resulting in unmet demand.

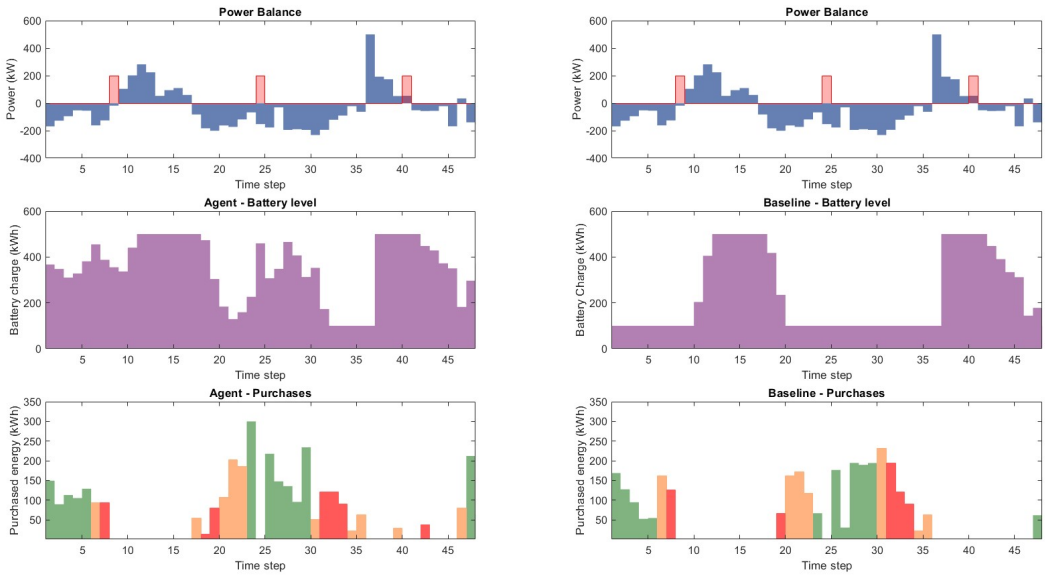


Figure 9.8: Purchasing behaviour of the agent with load shedding.

9.3.6 Brief exploration of generation forecasts

While the scope of this study did not permit the full development of a sophisticated generation forecasting model, for curiosity's sake, a basic SARIMA grid search was employed for model training. The stability of the forecasts is illustrated in figure 9.9. These forecasts demonstrate noticeably greater volatility when compared to the load forecasts generated through more meticulous methods. To examine the implications of employing imperfect generation forecasts, these were integrated into the RL environment as an experimental measure.

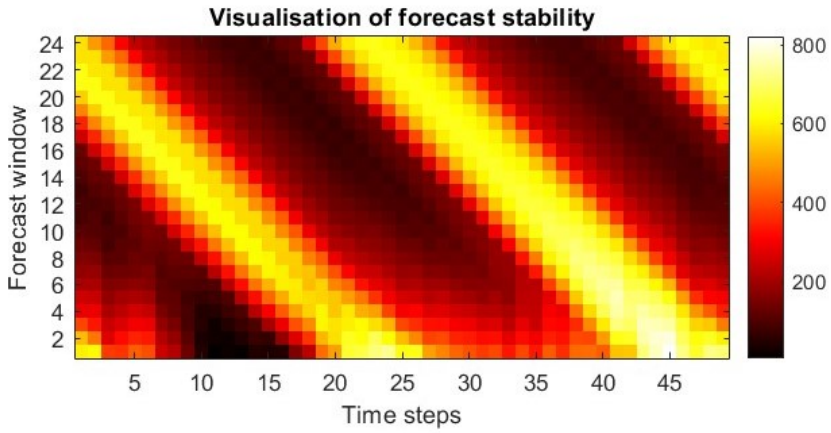


Figure 9.9: Generation forecasts.

The outcomes of this experiment are presented in table 9.3. Despite the inherent limitations in the forecasting model, the energy management agent was still capable of marginally reducing energy costs and entirely eliminating unmet demand. This example serves as an introductory glimpse into an RL environment that utilises machine learning for both load and generation forecasting. The preliminary results are promising, suggesting the need for further research.

Table 9.3: Results of generation forecast experimentation

	Total Cost (R)	Unmet demand (kWh)	Diesel Generation (kWh)	Excess Generation (kWh)
No load shedding				
Baseline	982392.25	0	0	437520.82
Agent	970742.85	0	0	455165.52
Percent decrease	1.25%	NA	NA	-4.03%
With load shedding				
Baseline	1023396.93	4217.12	5947.40	437520.82
Agent	1001224.28	0	77.54	469122.32
Percent decrease	2.17 %	100%	98.70%	-7.22%

Chapter 10

Conclusions and Recommendations

10.1 Conclusion

This study aimed to achieve several objectives: designing a robust hybrid microgrid, training a machine learning model for day-ahead forecasts, developing an RL environment that accurately simulates said hybrid microgrid, and training an autonomous agent for efficient energy management.

Focusing on key design aspects, the hybrid microgrid was successfully designed to serve as a reliable template for the RL environment. Notably, the performance of the energy management agent suggests that a diesel generator might not have been essential for the microgrid, as it was able to reduce diesel generation to only 48kWh over the span of half a year.
A machine learning model, specifically a SARIMA network, was trained to provide day-ahead forecasts with a mean-per-step MAPE of 16.82% for the AC load and 23.47% for the DC load. Additionally, the forecasts were very stable over the prediction window. The success of the energy management agent in later sections is a testament to the success of the load forecasting model.
The RL environment was iteratively designed to a high standard. It encompassed all critical criteria, including physical components and reliability factors such as load shedding. Additionally, it incorporated economic factors like TOU rates. As a result, the environment fulfilled the outlined project requirements, confirming its ability to serve as a training ground for the autonomous agent.

The agent successfully achieved its primary objectives: reducing energy costs and ensuring power security. Under stable grid conditions, it reduced costs by 5.38%, and under unstable conditions, the reduction was 6.12%. Furthermore, the agent was able to eliminate any unmet load, thereby ensuring power security. However, a key area for improvement lies in its inability to decrease the monthly demand charge.

In summary, the study demonstrates that implementing an autonomous agent, equipped with machine-learning-generated load forecasts, within a hybrid microgrid offers a viable method for reducing energy costs and unmet demand. The findings warrant further research in this direction to explore additional optimisations and practical implementations.

10.2 Recommendations

During the process of this study, some key areas were identified that would benefit from further work.

10.2.1 Data collection

Acquiring real-world data for DC, PV, and wind power would significantly enhance the validity of the study. Current simulations lack the credibility that actual data would provide, limiting the practical applicability of the results.
Additionally, the collection of data over a longer time frame would be useful, it is unrealistic to assume that the patterns learnt from half of a year would generalise well to future years.

10.2.2 Generation forecasts

The study would benefit from incorporating uncertain generation forecasts to simulate real-world conditions. Currently, the model assumes perfect foresight for energy generation, which is not realistic and could affect the study's ability to generalise.
While it would be interesting to explore machine learning to produce generation forecasts, it would also be interesting to incorporate alternative methods. Particularly the use of weather forecasts combined with calculations based on the microgrid's generation parameters.

10.2.3 Further training & tuning

Due to the focus placed on the development of the RL environment, there was limited scope to focus on training alternative algorithms and tuning hyperparameters. These practices can greatly increase the learning ability of the agent and could improve the results significantly.
While the model exhibited commendable performance, there remains room for enhancement. The learning process, in particular, has not been optimised. Unfortunately, the time frame and scope of the report did not allow for an in-depth analysis of the model's hyperparameters and an exploration into hyperparameter tuning.

10.2.4 RL environmental complexity

A notable limitation of the current RL environment is its inability to capitalise on reducing the demand charge. While the agent and environment were successful in preventing an increase in this charge, they fell short of actually reducing it. Potential advancements in this aspect may arise from refined training techniques. However, a more promising avenue is likely to be the development of a more sophisticated reward structure. Exploring a combined approach that involves purchase limits and an exponential demand penalty mechanism to penalise large energy purchases could offer intriguing insights.
Further improvements in the RL simulation environment could include adding real-world factors like converter losses and battery charge/discharge rates.
Incorporating a deferable load into the system could provide the agent with an additional control action, thereby allowing for more nuanced energy management strategies.
Furthermore, enhancing the temporal resolution of the environment could offer a more realistic

representation of energy consumption and generation patterns, thereby making the model more directly applicable to real-world scenarios.

10.2.5 Integration

The underlying premise of this study is its potential to offer a tangible solution for energy management. To push this concept towards real-world implementation, it is recommended to explore various Supervisory Control and Data Acquisition (SCADA) software options. The objective would be to evaluate the feasibility of integrating the autonomous agent with industry-standard software platforms. Software like Ignition, which supports simulated readings, could serve as an appropriate testbed for this endeavour.

Bibliography

[1] H. Lotfi and A. Khodaei, "Hybrid ac/dc microgrid planning," *Energy*, vol. 118, pp. 37–46, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544216318175

[2] A. K. Barik, S. Jaiswal, and D. C. Das, "Recent trends and development in hybrid microgrid: a review on energy resource planning and control," *International Journal of Sustainable Energy*, vol. 41, no. 4, pp. 308–322, 2022. [Online]. Available: https://doi.org/10.1080/14786451.2021.1910698

[3] R. A. Kaushik and N. M. Pindoriya, "A hybrid ac-dc microgrid: Opportunities & key issues in implementation," in *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, 2014, pp. 1–6.

[4] S. Moradi, G. Zizzo, S. Favuzza, and F. Massaro, "A stochastic approach for self-healing capability evaluation in active islanded ac/dc hybrid microgrids," *Sustainable Energy, Grids and Networks*, vol. 33, p. 100982, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352467722002272

[5] P. C. Loh, D. Li, Y. K. Chai, and F. Blaabjerg, "Autonomous operation of hybrid microgrid with ac and dc subgrids," *IEEE Transactions on Power Electronics*, vol. 28, no. 5, pp. 2214–2223, 2013.

[6] E. Unamuno and J. A. Barrena, "Hybrid ac/dc microgrids—part i: Review and classification of topologies," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 1251–1259, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032115008412

[7] P. H. Shaikh, A. Shaikh, Z. A. Memon, A. A. Lashari, and Z. H. Leghari, "Microgrids: A review on optimal hybrid technologies, configurations, and applications," *International Journal of Energy Research*, vol. 45, no. 9, pp. 12 564–12 597, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/er.6666

[8] K. Cabana-Jiménez, J. E. Candelo-Becerra, and V. Sousa Santos, "Comprehensive analysis of microgrids configurations and topologies," *Sustainability*, vol. 14, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/3/1056

[9] B. Naji Alhasnawi, B. H. Jasim, and M. D. Esteban, "A new robust energy management and control strategy for a hybrid microgrid system based on green energy," *Sustainability*, vol. 12, no. 14, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/14/5724

[10] R. Georgious, R. Refaat, J. Garcia, and A. A. Daoud, "Review on energy storage systems in microgrids," *Electronics*, vol. 10, no. 17, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/17/2134

[11] Z. Ullah, M. Elkadeem, K. M. Kotb, I. B. Taha, and S. Wang, "Multi-criteria decision-making model for optimal planning of on/off grid hybrid solar, wind, hydro, biomass clean electricity supply," *Renewable Energy*, vol. 179, pp. 885–910, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148121010673

[12] Z. Ullah, S. Wang, J. Lai, M. Azam, F. Badshah, G. Wu, and M. R. Elkadeem, "Implementation of various control methods for the efficient energy management in hybrid microgrid system," *Ain Shams Engineering Journal*, vol. 14, no. 5, p. 101961, 2023. [Online]. Available:

https://www.sciencedirect.com/science/article/pii/S2090447922002726

[13] N. Attou, S.-A. Zidi, M. Khatir, and S. Hadjeri, "Energy management system for hybrid microgrids." *Electrotehnica, Electronica, Automatica*, vol. 69, no. 2, 2021.

[14] S. Kotra and M. K. Mishra, "A supervisory power management system for a hybrid microgrid with hess," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 3640–3649, 2017.

[15] M. Hosseinzadeh and F. R. Salmasi, "Power management of an isolated hybrid ac/dc micro-grid with fuzzy control of battery banks," *IET Renewable Power Generation*, vol. 9, no. 5, pp. 484–493, 2015. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-rpg.2014.0271

[16] J. M. Lujano-Rojas, C. Monteiro, R. Dufo-López, and J. L. Bernal-Agustín, "Optimum load management strategy for wind/diesel/battery hybrid power systems," *Renewable Energy*, vol. 44, pp. 288–295, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148112001243

[17] A. C. Luna, N. L. Diaz, M. Graells, J. C. Vasquez, and J. M. Guerrero, "Mixed-integer-linear-programming-based energy management system for hybrid pv-wind-battery microgrids: Modeling, design, and experimental verification," *IEEE Transactions on Power Electronics*, vol. 32, no. 4, pp. 2769–2783, 2017.

[18] A. Almalaq and G. Edwards, "A review of deep learning methods applied on load forecasting," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 511–516.

[19] L. Lv, Z. Wu, L. Zhang, B. B. Gupta, and Z. Tian, "An edge-ai based forecasting approach for improving smart microgrid efficiency," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7946–7954, 2022.

[20] M. Afrasiabi, M. Mohammadi, M. Rastegar, and A. Kargarian, "Multi-agent microgrid energy management based on deep learning forecaster," *Energy*, vol. 186, p. 115873, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544219315452

[21] H. H. Aly, "A proposed intelligent short-term load forecasting hybrid models of ann, wnn and kf based on clustering techniques for smart grid," *Electric Power Systems Research*, vol. 182, p. 106191, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779619305103

[22] Y. Li, R. Wang, and Z. Yang, "Optimal scheduling of isolated microgrids using automated reinforcement learning-based multi-period forecasting," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 1, pp. 159–169, 2022.

[23] H. Musbah, H. H. Aly, and T. A. Little, "Energy management of hybrid energy system sources based on machine learning classification algorithms," *Electric Power Systems Research*, vol. 199, p. 107436, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037877962100417X

[24] K. Zhou, K. Zhou, and S. Yang, "Reinforcement learning-based scheduling strategy for energy storage in microgrid," *Journal of Energy Storage*, vol. 51, p. 104379, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X22004030

[25] G. Muriithi and S. Chowdhury, "Optimal energy management of a grid-tied solar pv-battery microgrid: A reinforcement learning approach," *Energies*, vol. 14, no. 9, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/9/2700

[26] R. Leo, R. S. Milton, and S. Sibi, "Reinforcement learning for optimal energy management of a solar microgrid," in *2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS)*, 2014, pp. 183–188.

[27] B. C. Phan and Y.-C. Lai, "Control strategy of a hybrid renewable energy system based on reinforcement learning approach for an isolated microgrid," *Applied Sciences*, vol. 9, no. 19, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/19/4001

[28] J. Duan, Z. Yi, D. Shi, C. Lin, X. Lu, and Z. Wang, "Reinforcement-learning-based optimal control of hybrid energy storage systems in hybrid ac–dc microgrids," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5355–5364, 2019.

[29] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, "A high-bias, low-variance introduction to machine learning for physicists," *Physics Reports*, vol. 810, pp. 1–124, 2019, a high-bias, low-variance introduction to Machine Learning for physicists. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370157319300766

[30] O. Barndorff-Nielsen and G. Schou, "On the parametrization of autoregressive models by partial autocorrelations," *Journal of Multivariate Analysis*, vol. 3, no. 4, pp. 408–419, 1973. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0047259X73900304

[31] M. A. Quddus, "Time series count data models: An empirical application to traffic accidents," *Accident Analysis & Prevention*, vol. 40, no. 5, pp. 1732–1741, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457508001048

[32] A. Kasyok, "Simple steps for fitting arima model to time series data for forecasting using r," *International Journal of Science and Research (IJSR), ISSN (Online)*, pp. 2319–7064, 2013.

[33] E. H. Etuk, "The fitting of a sarima model to monthly naira-euro exchange rates," *Mathematical Theory and Modeling*, vol. 3, no. 1, pp. 17–26, 2013.

[34] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022.

[35] L. Pauly, H. Peel, S. Luo, D. Hogg, and R. Fuentes, "Deeper networks for pavement crack detection," 07 2017.

[36] S. Muzaffar and A. Afshari, "Short-term load forecasts using lstm networks," *Energy Procedia*, vol. 158, pp. 2922–2927, 2019, innovative Solutions for Energy Transitions. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876610219310008

[37] N. Maitanova, J.-S. Telle, B. Hanke, M. Grottke, T. Schmidt, K. von Maydell, and C. Agert, "A machine learning approach to low-cost photovoltaic power prediction based on publicly available weather reports," *Energies*, vol. 13, p. 735, 02 2020.

[38] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 114–119, 2019.

[39] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[40] F. Angizeh, A. Ghofrani, and M. A. Jafari, "Dataset on hourly load profiles for a set of 24 facilities from industrial, commercial, and residential end-use sectors," 2020.

[41] C. of Cape Town, "City of cape town tarrif policies," 2018. [Online]. Available: https://resource.capetown.gov.za/documentcentre/Documents/Financial%20documents/Annexure%207%20-%20Tariff%20Policies.pdf

[42] ——, "City of cape town - tarrif, fees and charges book," 2023. [Online]. Avail-

able: https://resource.capetown.gov.za/documentcentre/Documents/Financial%20documents/Electricity%20Consumptive%20Tariffs.pdf

[43] C. o. Cape Town, "Loadshedding schedule 2018-2023," Sep 2023. [Online]. Available: https://odp-cctegis.opendata.arcgis.com/documents/42551ffb9c494932ac16c4e9cd7dc28e/about

[44] Z. Chang, Y. Zhang, and W. Chen, "Effective adam-optimized lstm neural network for electricity price forecasting," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, 2018, pp. 245–248.

[45] P. Díaz, C. Arias, R. Peña, and D. Sandoval, "Far from the grid: A rural electrification field study," *Renewable Energy*, vol. 35, no. 12, pp. 2829–2834, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148110002235

Appendix A

Pricing for design of hybrid microgrid

Table A.1: Prices of Solar PV units.

Name of unit	**Num units**	**Capacity (W)**	**Price (R)**	**Price Per Kw**	**Link**
Jinko Solar Panel Tiger MonoFacial	1	470	1895	4031.91	Link
Jinko Solar Panel Tiger Mono-Facial Pallet	31	555	67275	3910.20	link
JA Solar 144 cell Module	1	465	2192	4713.98	Link
Canadian Solar 555W Super High Power Mono - Pallet	31	555	79143	4600.00	Link
Canadian Solar 545W panel	1	545	2575	4724.77	Link

Table A.3: Prices of wind turbine units.

Name of unit	**Num units**	**Capacity (W)**	**Price (R)**	**Price Per Kw**	**Link**
MAGNUM 5kW	1	5000	30420	60840.00	link
ATLAS 7 KW Vertical	1	7000	31850	45500.00	link
HERA 7kW Vertical	1	7000	26240	37485.71	link
FLTNXY POWER 800W 48V	1	800	569527979	71187.50	link
10000W Vertical Axis	1	10000	27979	27979.00	link

Table A.5: Prices of diesel generators.

Name of unit	**Num units**	**Capacity (W)**	**Price (R)**	**Price Per Kw**	**Link**
45 kVA 3-Phase Diesel Generator	1	45000	43125	958.33	link
Diesel Generator 6.5 kVa	1	6500	22800	3507.69	link
30KVA 3 Phase	1	30000	125999	4199.97	link
8Kva Diesel generator	1	80000	35000	437.505494.95	link
20kVA 3-Phase FAW	1	20000	109899	5494.95	link

Table A.7: Prices of lithium-ion battery units.

Name of unit	**Num units**	**Capacity (W)**	**Price (R)**	**Price Per Kw**	**Link**
Freedom Won Lite Commercial	1	100000	623986	623986.00.33	link
Goscor Powerwall Battery		10240	53590	523339.84	link
Hubble Lithium HV System		51200	409160	799140.63	link
Outback Power AGM Battery		113000	688276	609093.81	link
Blue Carbon High Voltage		100000	679259	679259.00	link

Appendix B

Additional environment information

Table B.1: Version Notes Summary

Version	Additions
v0.1	AC load profile PV Generation profile Time of use indicators (1:Off-peak, 2:Standard, 3:Peak)
v0.2	Changed observation space to hold power balances rather than separate load and generation values (generation - load = power balance). This was done to simplify the observation space.
v0.2.1	Started using Weights and Biases to log results and models. Reduced the number of forecasted steps in the observation space from 24 to 3. This was done to further simplify the observation space.
v0.3	Load shedding, (load shedding forecasts were included in the agent's observation space so it was aware of oncoming blackouts) Reward structure was modified to include unmet demand with the addition of load shedding the Vec_env wrapper was used to speed up training.
v1.0	Hybrid grid! - AC and DC load Wind turbine profile for AC generation PV array profile for DC generation modified observation space redefined standby and purchase functions to include the new loads and generation forms.
v1.1	Rectifier and inverter power tracking improved logging with Weights and Biases.
v1.2	Diesel Generator reward based on real prices included a demand charge in the reward.

Table B.3: Version Notes Summary Continued

Version	**Additions**
v2.0	Introduced a continuous action space to overcome demand charge issues.
v2.1	Added intermediate penalty for every purchase that exceeds the previous max demand. Included the current max demand in the agent's observation space.
v2.2	Scaled the observation space. All the values in kW were normalised with the same scaling function to range -1, 1 Time of use rates scaled to range 0,1 Action space defined as -1,1 for the agents sampling distributions. Is scaled up to with its own scaling function in the environment.
v2.3	Flattened the observation space to be a 1D vector as it can be processed more efficiently. Tested agents with increased learning rates and decreased vf_coef to encourage more exploratory learning.
v2.4	Tested shorter episodes of one month to see if the agent would learn to predict and avoid the demand charge. Fixed some major bugs with scaling
v2.5	Transition back to normal episode lengths (70% of the year) for training. Removed the demand charge.
v2.6	Cleaned code Working continuous action space (no demand charge)
v2.7	Added structure to use actual forecasts rather than perfect vision Tried a rolling demand charge penalty Implemented a purchase limit

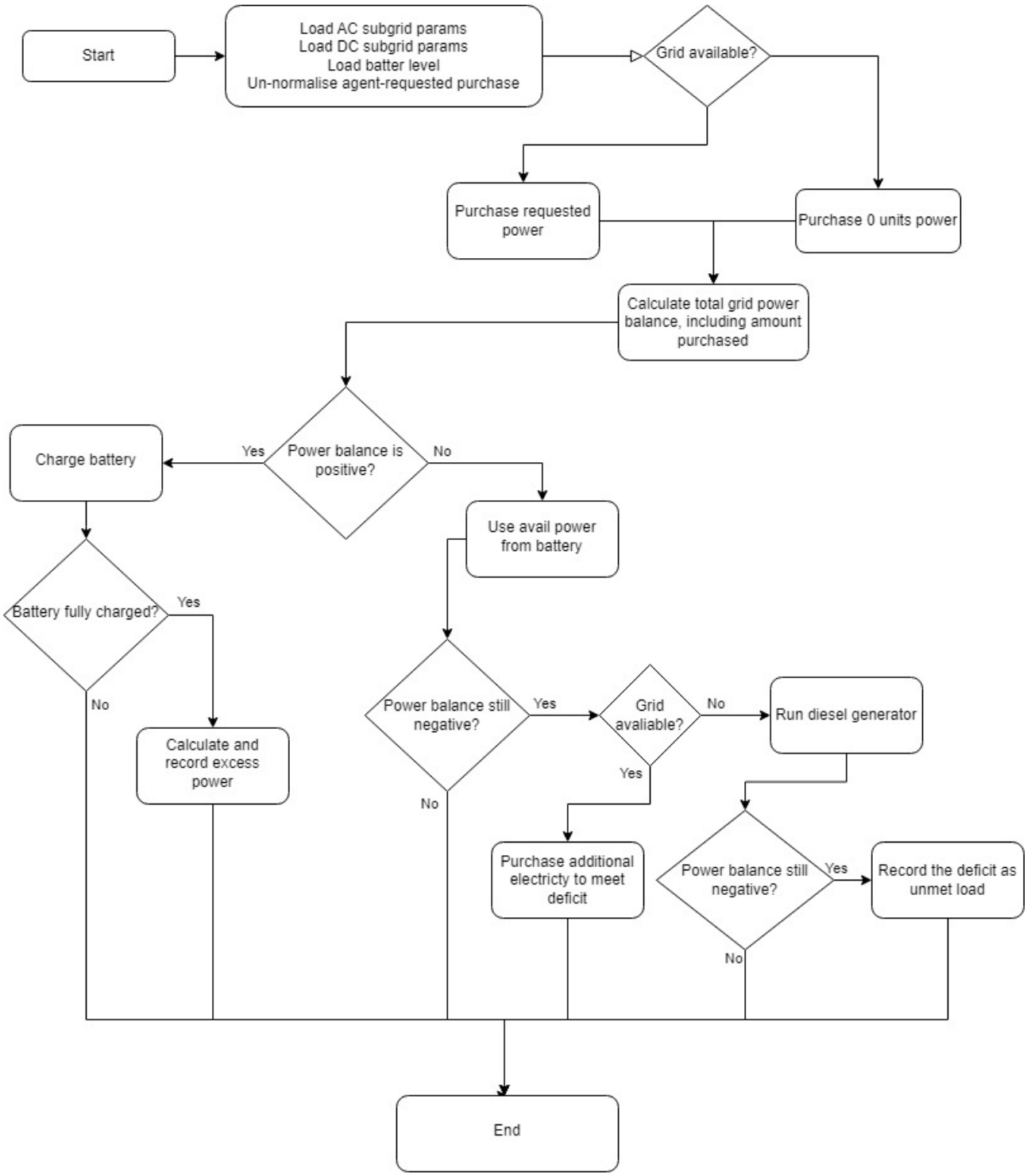


Figure B.1: Rule-based system for management of power in the microgrid

Appendix C

Ethics Forms

UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA · UNIVERSITEIT VAN KAAPSTAD

PRE-SCREENING QUESTIONNAIRE OUTCOME LETTER

STU-EBE-2023-PSQ000596
2023/08/09

Dear Julian Banks,

Your Ethics pre-screening questionnaire (PSQ) has been evaluated by your departmental ethics representative. Based on the information supplied in your PSQ, it has been determined that you do not need to make a full ethics application for the research project in question.

You may proceed with your research project titled:

INTELLIGENT ENERGY MANAGEMENT SCHEME OF A HYBRID MICROGRID USING
MACHINE LEARNING TECHNIQUES

Please note that should aspect(s) of your current project change, you should submit a new PSQ in order to determine whether the changed aspects increase the ethical risks of your project. It may be the case that project changes could require a full ethics application and review process.

Regards,

Faculty Research Ethics Committee

Figure C.1: Ethical clearance