Master's Thesis

Toward a sustainable energy future: Peak load shaving in commercial properties to reduce cost of energy

Department of Statistical Sciences
University of Cape Town

Tiffany Deanne Woodley

February, 2022

Minor Dissertation submitted in partial fulfilment of the requirements for the degree of Master of Science in Advanced Analytics
Supervised by Dr Juwa Nyirenda

Abstract

Transitioning from fossil fuel-based energy systems to renewable sources is a is a global environmental imperative. South Africa has a coal-based energy sector, and consumers could be incentivised to pursue renewable energy alternatives if these solutions were financially advantageous. In South Africa, commercial properties are billed per kWh and can incur an additional demand charge that often accounts for a substantial portion of the energy bill, depending on the load factor.

This thesis investigates peak load shaving as a solution for commercial properties to reduce their cost of electricity while supporting the transition to a greener energy future. Of the methods proposed for peak load shaving, reinforcement learning holds the greatest promise. However, its application in practice has been limited due to the "curse of dimensionality". To make reinforcement learning a feasible option for peak load shaving, this thesis introduces a novel approach that employs clustering the energy demand profile shapes and training separate learning agents to target specific demand shapes, thereby reducing the complexity of the problem presented to the individual agents.

The reinforcement learning model was trained on historical data from a commercial shopping centre in Cape Town using a hypothetical battery. Two scenarios were considered; the first assumed the absence of solar in the energy system while the second assumed its presence. Once trained, the learning agents were tested on unfamiliar energy data from the same shopping centre, and they achieved practical peak load shaving results. In Scenario 1 when using only a battery, monthly demand was reduced by 91 kW on average. Introducing a solar system in Scenario 2 increases uncertainty in the problem. The results, only demonstrated on one cluster, show the battery most often achieved a 50 kW reduction per day. In both scenarios, a learning agent trained on particular clusters of demand profiles was able to reduce peak energy demand for all unfamiliar days. Furthermore, in Scenario 2, the agent's learning progression indicated that the agent was learning to increase the battery output during the predominant peak. This suggests that our method's efficacy would improve with increased training time. If implemented, this approach could provide a practical peak shaving solution for the commercial shopping centre in Cape Town, effectively lowering their energy demand charges. This thesis has shown that clustering techniques used in conjunction with reinforcement learning is a promising approach when considering the peak shaving problem.

All code used in this thesis can be found at https://github.com/tiff-woodley/Peak-shaving.

Contents

List of Figures

List of Tables

1 Introduction

This study proposes the application of reinforcement learning to carry out the task of peak load shaving in order to improve the cost of electricity in commercial properties by reducing their demand charge. The purpose of this chapter is to introduce the study by contextualising the research and providing an overview of the research process. It includes the background to the research problem, the research objectives and the dissertation outline.

1.1 Background to Study

Globally, energy demand is at a record high [Mongia et al., 2021]. Modern lifestyles, technologies, and business practices are energy intensive, and fossil fuels are presently used to fulfil the majority of energy requirements around the world. While much research has gone into renewable energy technologies in recent years, a study by Zurfi et al [2017], showed that fossil fuels consistently accounted for 80 % of the global energy mix between 2009 and 2019. Due to their finite nature, fossil fuels are an infeasible long-term energy resource as shortages will eventually occur. Moreover, the burning of fossil fuels to produce energy is associated with environmental pollution and the release of greenhouse gasses (GHGs) that contribute to climate change. A sustainable energy future depends upon the incorporation of renewable energy resources into the global energy mix.

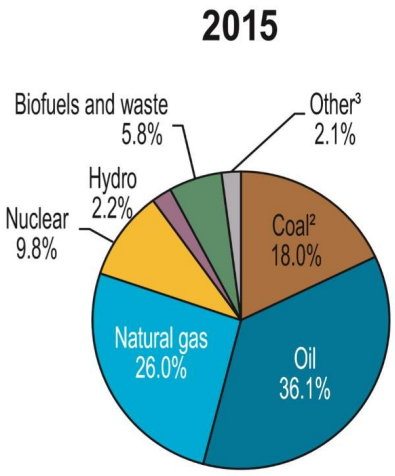


Figure 1: Pie chart showing the proportional contribution of different energy resources toward meeting global energy demands in 2015 [Crenshaw, 2017].

Following global trends, South Africa presently has a fossil fuel-based economy, with coal accounting for 90 % of the country's energy production [Bekun et al., 2019]. Coal is an inexpensive but environmentally detrimental,

non-renewable energy resource. The burning of coal releases GHGs that exacerbate climate concerns, and coal mining is associated with large-scale land degradation and biodiversity loss. In their 2019 report, the United Nations described climate change as "the defining issue of our time" [Conca, 2019]. Coal is responsible for almost half of global carbon emissions, and there is a strong argument within literature for phasing out the burning and mining of coal [Smith et al., 2021]. At the 2021 United Nations Climate Change Conference (COP26), many countries - including South Africa - pledged to phase out fossil fuels in pursuit of greener energy sectors [Smith et al., 2021].

Given the environmental impact of a fossil-fuel based energy sector, it is clear that developing renewable energy options is imperative globally. South Africa's warm, sunny climate is well suited to solar energy solutions. As it is weather-dependent, solar energy can presently only reduce the use of coal, not replace it. As such, continued research into energy storage systems has the potential to revolutionise the viability of the solar industry. Energy storage systems protect consumers from the inconvenience of sudden power outages and can reduce electricity bills through the application of systems such as peak load shaving (to be discussed in Subsection 1.1.2) [Zurfi et al., 2017]. Storage systems like these which help align the financial objectives of consumers with the objectives of creating a sustainable energy future, are a necessity. If we can incentivise consumers to integrate more renewables into their energy solutions by making them financially advantageous, we can start moving away from the reliability of fossil fuels.

1.1.1 Electricity in South Africa

In South Africa, electricity is billed based on the number of kWh used, although a time of use tariff may cause costs to vary at different times of the day [Kanzumba and Kusakana, 2019]. Consumers could incur an additional demand charge which is determined based on their maximum demand reached averaged over a predetermined period of the month (in most cases this period is half an hour), even if this limit is only reached once [Hohne et al., 2020]. Demand charges are intended to encourage users to balance their load; in other words, ensuring their energy demands remain stable throughout the day. A balanced load mitigates the need for power suppliers to build additional (and expensive) grid assets.

Larger commercial consumers usually have substantial energy loads and are thus responsible for greater demand spikes on the grid. Consequently these commercial consumers are often on a tariff which includes a demand charge. The demand charge for commercial consumers can account for a large portion of their energy bill depending on the load factor. For example, a study showed that demand charges account for between 30 to 70 percent of commercial consumers' total energy bills in the United States of America [McLaren et al., 2017]. It is therefore highly desirable to such companies to balance their energy loads

and reduce their demand charges.

1.1.2 Peak Load Shaving Overview

Peak load shaving (which from this point on will be referred to as peak shaving) is one way consumers incurring a demand charge can reduce their utility costs [Karmiris and Tenger, 2013]. As illustrated in Figure 2, peak shaving eliminates short-term energy demand spikes that set a higher demand peak by reducing the maximum amount of energy drawn from an electricity utility company. This is done by using an alternative energy source during periods of high demand to mitigate the proportion of that demand being drawn from the grid. By lowering the maximum demand on the grid, peak shaving smooths out the energy load and results in a better load factor. This in turn reduces the overall demand charges incurred by the consumer and limits the adverse impacts of energy demand spikes on the generating plants (which are predominantly fossil-fuel based) and the grid.

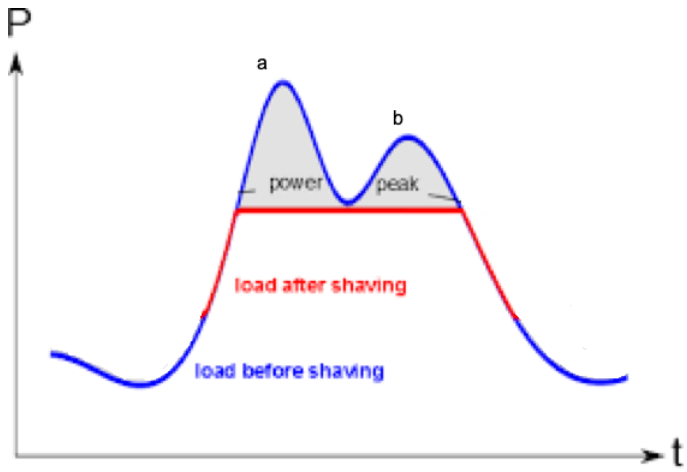


Figure 2: Peak Shaving Illustration: The blue line indicates the energy demanded from the grid over the course of the day, a) shows the morning peak energy demand and b) shows the afternoon peak energy demand. The grey area indicates where an energy storage system would output its energy in order to reduce these energy peaks resulting in a flatter energy demand as is denoted by the red line [Karmiris and Tenger, 2013].

These are some of the traditional systems used for implementing peak shaving:

- **Manual Intervention** - an individual can manually power-down certain equipment during peak periods, thereby reducing the maximum amount of power required from the grid. This approach is not viable in instances where the majority of equipment is instrumental to business operations.

Furthermore, the risk of missing peak periods increases when relying wholly on manual intervention.

- **Diesel Generators** - generators are used to provide additional power during peak periods thereby reducing the need for more power from the grid [Clifford, 1979]. Generators are expensive to maintain and run on diesel which is environmentally unfriendly.

- **Stand Alone Solar** - solar is an inexpensive renewable energy source that can be used to generally reduce electricity consumption and demand charges [Ihirwe et al., 2021]. However, cloud cover or shading can temporarily reduce solar generation and disrupt the effectiveness of peak shaving. The fact that solar is intimately dependent on the sun means that there can be small energy demand spikes in a customer's energy demand on the grid. Energy demand spikes refer to a short period is which the average demand is higher than the rest of the day. As seen in Figure 2, energy demand spikes most commonly occur in the morning and evening when the sun is rising or setting. Solar systems are consequently not generating maximum solar energy during these times. Batteries, generators, or the grid can be used to fill in these gaps that solar leaves.

A modern approach to mitigate the inefficiencies associated with these peak shaving systems is to use a micro-grid that combines a solar energy system with a supporting energy storage system [Oyewo et al., 2019]. Batteries have become an extremely popular energy storage system and are therefore often the energy storage system of choice. The solar energy produced reduces the energy demand on the grid, leaving behind smaller energy demand spikes. The energy storage system, generally a battery, can target these demand spikes that are left behind. As the demand spikes are smaller and narrower than the initial demand, a smaller battery system can be installed to reduce the capital outlay needed for the system. As such, this combined system lends itself to optimising the efficiency of peak shaving.

1.2 Research Objectives

Data for this paper was collected from a commercial shopping centre in Cape Town, whose identity will remain undisclosed for confidentiality reasons. This shopping centre has a solar energy system as part of their energy solution and is on a tariff that includes demand charge, making it a perfect candidate for peak shaving using a battery system. The main objective of this thesis is to minimise the monthly cost of electricity for this commercial consumer by reducing their demand charge through peak shaving, using a combined micro-grid consisting of the solar and battery system. The energy placed on the grid by the consumer averaged over a half hour interval t during the day can be modelled as follows in Equation 1:

$$E_t^{grid\ demand} = E_t^{total\ demand} - E_t^{solar\ generation} - E_t^{battery\ output} + E_t^{battery\ charging} \quad \text{for t in 1,...,48.} \tag{1}$$

Where the demand placed on the grid $E_t^{grid\ demand}$ equates to the total demand $E_t^{total\ demand}$ when there is no solar or battery system operating. The solar system $E_t^{solar\ generation}$ generates energy which the consumer uses in turn reducing the energy demand placed on the grid. The energy demand placed on the grid after the solar energy has been utilised can further be manipulated by outputting the battery system $E_t^{battery\ output}$ or charging it $E_t^{battery\ charging}$.

Deriving the most financial benefit from a battery system in a combined micro-grid requires optimal control of the battery system. Using the battery system optimally for the purpose of peak shaving involves deciding when to discharge and charge a battery system in such a way that demand costs are reduced, thus minimising electricity costs. This can be achieved by manipulating $E_t^{battery\ output}$ and $E_t^{battery\ charging}$ in order to reduce the maximum energy drawn from the grid over the course of a month, as outlined in Equation 2:

$$\text{Minimise}(\text{Max}(E_t^{grid\ demand}) \text{ for all t in each day of a month}) \tag{2}$$

To do this, a model will need to make the decision for how much power to output from/ input to the battery system for the next half-hour interval. Optimally controlling the battery system in a combined micro-grid is quite challenging owing to the wide range of variability and level of uncertainty associated with the consumer's energy demand, as well as due to meteorological factors affecting solar generation.

1.2.1 Shortfalls of existing methods

There has been much research into optimal control of energy storage systems for the purpose of peak shaving. Most commonly, researchers use a predefined control strategy algorithm, such as a set shave level. The algorithm is then optimised to historical data through methods like simulation or optimisation [Uddin et al., 2018]. Since these predefined strategies look solely at historical data they rely on the assumption that future energy demand requirements will follow the same distribution as historical data. Forecasting energy demand and solar generation is another research "hot-spot". Forecasts for the coming day can then be used to calculate battery output analytically [Uddin et al., 2018]. Forecasts for weather and demand need to be made at some set point in time and acted upon to make a decision for the strategy for an entire day. After forecasts are made, new information may come to light which would affect these forecasts however and so it is difficult to implement these. This leads to these methods assuming stationarity in the energy-usage pattern and not allowing for

new trends to emerge. This results in the need for added control algorithms such as prediction error functions or optimisation methods such as model predictive control to be layered into the solution. Emergent research in reinforcement learning offers a method to only focuses on the best action right now and so avoids this issue of prediction errors.

While theoretically promising, researchers have run into the "curse of dimensionality" when applying reinforcement learning to peak shaving problems. This thesis investigates how clustering techniques could be used to reduce the complexity of the problem in order to feasibly apply reinforcement learning to peak shaving. Clustering techniques are used to derive insight into the patterns of both energy demand and solar generation profiles. These insights are then used to create simplified environments and state spaces, reducing the complexity of the problem presented to reinforcement learning agents.

1.2.2 Main contributions of this thesis

The research problem will be solved in two phases. First, an optimal control signal will be used to operate a hypothetical battery installed at the commercial shopping centre serving as the trial location, but as if a solar system was not installed at the site (i.e. $E_t^{solar\ generation}$ will output zero at every half hour time interval). Once the solution to this simplified objective has been well defined, it will then be extended to be able to handle the added complexity of a solar system.

In summary, the main contributions of this thesis are as follows:

1. A clustering approach which can be used to gain better insight into the energy demand of a commercial entity.
2. A clustering approach which can be used to gain better insight into the expected profile shapes of solar generation and how these shapes will affect where the grid demand peaks will occur.
3. A reinforcement learning solution to generate an optimal battery control signal for a commercial shopping centre where there is no solar system installed.
4. A reinforcement learning solution to generate an optimal battery control signal for a commercial shopping centre where there is a solar system installed.

Section 2.3 provides a comprehensive overview of how these research objectives will be achieved by outlining this thesis' methodology.

1.3 Motivation for research

Electricity is a large operational cost for commercial consumers globally. In South Africa, grid-based electricity rates are estimated to increase substantially

over the next few years [Ting and Byrne, 2020]. As such, alternative energy projects that could offset electricity costs are of great interest to consumers with a high energy demand. Furthermore, Eskom - South Africa's state-owned power utility - is in crisis. Many of Eskom's coal-fired power stations are operating suboptimally due to failing infrastructure [Ting and Byrne, 2020]. Planned power outages, known as "loadshedding", have plagued the country since 2008 in an attempt to decrease demand on the national grid
[Lumka et al., 2021].

High electricity costs, poor utility provision, and environmental concerns associated with fossil fuel-based energy have increased the interest in renewable energy sources, such as solar, wind, nuclear and hydro-electric. The cost of solar energy per kWh is cheaper than Eskom's tariffs [Merven et al., 2021], and while batteries are currently expensive energy storage solutions, there is extensive research going into battery chemistry and up-scaling manufacturing efforts. As such, the cost of batteries is predicted to decrease dramatically over the next few years, improving the financial case for integrated battery projects over grid-based power in many instances [Mauler et al., 2021].

As batteries become more accessible, the installation of integrated battery systems with renewable energy solutions will likewise become more cost effective. For instance peak shaving with a battery system additionally offers consumers affected by high electricity demand charges the opportunity to reduce their utility costs. In order for peak shaving to be financially viable however, the battery has to be used optimally, offsetting demand on the grid enough to justify the capital outlay on the battery system. Reinforcement learning promises to improve existing peak shaving technologies through its dynamic nature, enabling the battery's potential to be used more efficiently.

If a wide range of commercial consumers utilise peak shaving, it will reduce the demand on the grid which will in turn minimise strain on poorly serviced utility infrastructure and decrease the need for costly capital investments into new power stations to meet demand in turn reducing the use of fossil fuels. Moreover, utilising peak shaving and renewable energy options, like solar, is pivotal to reducing our country's reliance on coal, and ensuring South Africa aligns with international imperatives towards a more sustainable, clean energy future [Oyewo et al., 2019].

The implementation of solar systems makes financial sense for commercial consumers in the long term. As such, it is likely that a greater number of electricity consumers will opt to integrate solar into their energy procurement strategies over time. To ensure that the most benefit is derived from these solar systems, it is imperative to ensure that peak shaving solutions are available that optimally cut consumer costs. It is thus that this paper explores the benefits reinforcement learning could bring to peak shaving technologies in combined solar micro-grids.

1.4 Structure of Research

The remainder of this thesis is concerned with evaluating previous methods used to solve the peak shaving problem, exploring and interpreting the data collected, discussing how the proposed models were built, and testing and evaluating these models against a baseline model. The subsequent chapters are structured as follows:

- Chapter 2 presents an overview of the literature on peak shaving and this thesis' proposed methodology.
- Chapter 3 gives an overview of the methods utilised in this thesis.
- Chapter 4 explores a commercial shopping centre's data and uses clustering techniques to derive insight into its energy demand trends.
- Chapter 5 defines and formulates the reinforcement learning agents for the purpose of peak shaving, with no solar system installed. This chapter also discusses the model's results based on the data collected from the commercial shopping centre.
- Chapter 6 uses clustering to derive insight into grid demand trends and how they are affected by solar generation.
- Chapter 7 discusses how adaptions can be made to the reinforcement learning agents used in Chapter 5, enabling them to work with a solar system. The chapter goes on to analyse the results of the adapted agents on the commercial shopping centre's energy profiles.
- Chapter 8 draws conclusions based on this thesis' findings and discusses potential directions for future research.

2 Literature Review

As noted in the previous chapter, optimally utilising battery capacity is fundamental to effective peak shaving when using a combined micro-grid system. In this section we review methods that have been used to control energy storage systems, including rule-based, forecasting, optimisation, and reinforcement learning methods. We also review methods for clustering demand profiles.

2.1 Control Methods

Traditional methods used to control energy storage systems can be divided into three categories: rule-based, forecasting, and optimisation-based methods. These methods generally need to be set ahead of time and are inflexible once being set, making them unable to account for the stochastic nature of energy demand and solar forecasts. While one may be unable to find instances of its practical application to peak shaving problems in literature, research suggests that reinforcement learning could be another viable method to control batteries in combined micro-grids. This approach has the potential to respond effectively to changes in the energy demand profiles throughout the day.

2.1.1 Heuristics or Rule-based methods

Heuristics or rule-based approaches are the most popular methods for controlling battery systems and have found application in real grid and combined micro-grid systems. Rule-based methods use historical data of past energy demand profiles to set parameters that govern battery output based on the assumption that future demand profiles will look similar.

A recent review paper by [Uddin et al., 2018] suggests that predefined shave levels are the most common rule-based control method applied to peak shaving. In this instance, a system is programmed so that the battery outputs a set amount when energy demand rises above a stipulated shave level that is set based on a consumer's historical energy demand profiles. While rule-based methods are popular due to their simplicity and ease of implementation [Uddin et al., 2018], these methods apply their policy to an expected profile shape calculated using a historical average. In practice, a profile can vary significantly from this average, leading to suboptimal battery usage. As such, the results of rule-based systems often need to be improved through optimisation techniques (see Subsection 2.1.3). Researchers have additionally attempted to increase the complexity of rule-based functions so as to improve their efficacy in peak shaving. Figure 3 is a flow diagram describing an example of a complex rule-based method for managing a battery system in order to decrease demand on the grid [Chua et al., 2016].

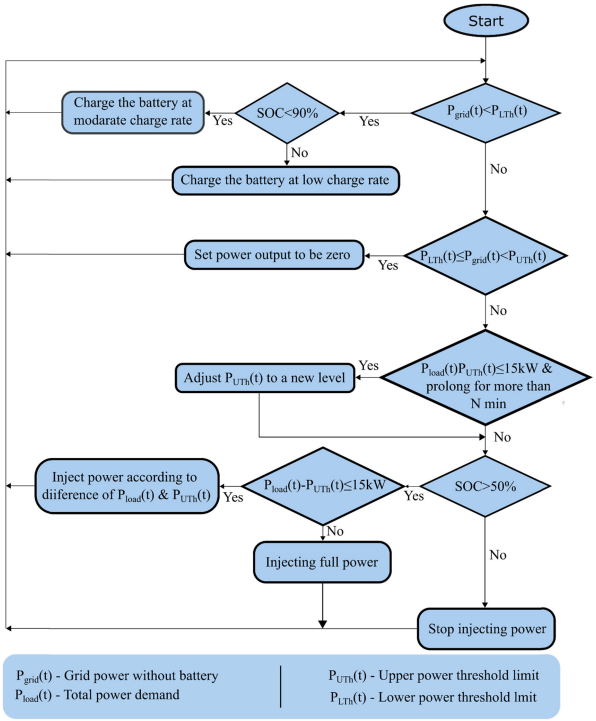


Figure 3: A more complex pre-defined control strategy suggested by Chau et al. [2016]. This strategy like simple shave levels still use threshold values to trigger when the battery should output, but also take into account the battery's state of charge (SOC) when determining how much the battery should output. The strategy also changes the threshold values if a peak energy spike lasts for longer than a set amount of time.

2.1.2 Forecasting

While rule-based control methods use historical energy demand profiles to make assumptions about future demand, forecasting methods plug historical energy profiles into a model in conjunction with other variables to predict future energy demands. Forecasting methods differ depending on how far into the future one is predicting. When peak shaving, one typically forecasts one day ahead using short-term load forecasting methods and then plans battery outputs accordingly. A systematic review of electrical load forecasting techniques found Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and time series models (such as ARIMA and ARMA models) to be among the most popular methods for forecasting short-term energy loads [Kuster et al., 2017]. These models are generally driven by variables pertaining to a building's characteristics and its occupancy, various environmental data, and historical electricity load data.

Energy demand predictions also differ depending on the time interval being predicted. The smaller the time interval the more complex the prediction problem becomes, particularly in terms of computational requirements. For the purpose of peak shaving, prediction intervals are commonly employed on a scale of half an hour or less, thereby complexifying the problem. Figure 4 provides a summary of the frequency that different forecasting techniques have been applied in research studies that are concerned with forecasting electricity loads for periods ranging from very short to long-term. ANNs, SVMs, and time series methods again proved most popular for forecasting the smaller time intervals that are of interest in peak shaving. Another less popular approach is the bottom up approach where the probability of different electrical devices being on is used to create an aggregated profile. While forecasting can assist in better planning battery output based on predicted demand peaks, forecasts can prove inaccurate due to the inherently random nature of energy demand and solar generation. As such, the risk of suboptimal battery usage remains. Moreover, forecasting models may need further optimisation in instances where peak shaving using a micro-grid is solved as part of a larger, more complex problem (discussed in Subsection 2.1.3).

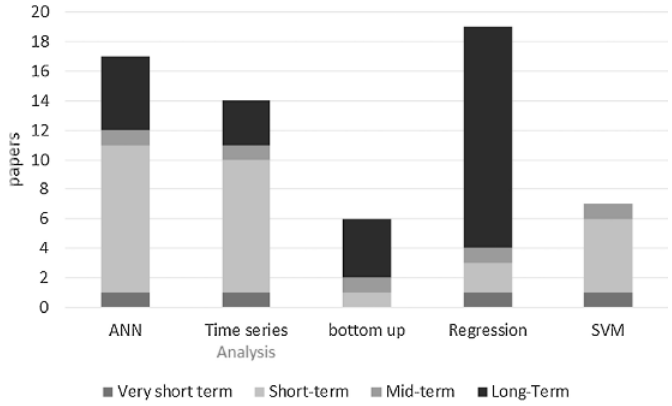


Figure 4: Distribution of papers about load forecasting separated by method and coloured by how far into the future is being predicted.

2.1.3 Optimisation based methods

Optimisation-based techniques cannot be used to solve peak shaving problems as a stand-alone method, but are rather used in conjunction with other control methods to improve the accuracy of a solution. For example, once a rule-based control method has been defined, the system can be optimised using techniques such as dynamic programming or MATLAB simulations [Oudalov et al., 2007, Rahimi et al., 2013, Son and Song, 2014]. As previously mentioned, suboptimal battery usage occurs when applying a static rule to a range of energy profile

shapes for the purpose of peak shaving. Optimisation techniques increase the efficacy of a rule-based solution, thus improving the efficiency of battery usage. Similarly, solutions based on forecasting methods may require further optimisation if peak shaving is being solved as part of a more complex problem. A micro-grid control system may have multiple objectives, such as reducing energy costs per kWh by taking advantage of time of use tariffs, or optimally charging the battery using excess solar to avoid either losing solar or exceeding feed-in limits to the national grid. In such cases, battery outputs will need to be optimised using optimisation techniques such as Model Predictive Control [Vedullapalli et al., 2019].

Model Predictive Control (MPC) has shown promise for applications where an Heating Ventilation and Cooling (HVAC) system is controlled in conjunction with a battery. This combined system is used to target two objectives namely, reducing the demand charge through peak shaving and optimising the kWh cost by manipulating the time of use tariff [Vedullapalli et al., 2019]. The use of MPC lends well to this application as it re-evaluates the optimisation problem at regular intervals allowing for corrections to be made on updated forecasts. Further, MPC has demonstrated its ability to take forecast errors into account on problems where solar feed-in to the national grid is an objective alongside additional objectives, other than peak shaving [Dongol et al., 2018]. These MPC-based energy management controllers usually show very good performance when attempting to achieve several objectives on forecasted profiles. Studies show that optimisation techniques for rule-based controls can successfully lower energy demand peaks, but not in a way that derives the greatest financial benefit from a battery system [Oudalov et al., 2007]. The source of this difficulty again lies in attempting to optimise a universal approach that is being applied to variable data.

2.1.4 Reinforcement learning

Reinforcement learning uses intelligent agents that make predictions and update their approach as their environment changes in a way that maximises a reward signal. The actions of reinforcement learning agents can be used to control battery outputs, directly affecting an energy demand curve. As such, reinforcement learning is well suited to peak shaving as it can solve the problem holistically. The problem of peak shaving is complex and consists of several interacting variables that are difficult to model. Reinforcement learning is usually applied as a model-free algorithm, increasing its appeal as a control method for peak shaving. Vázquez-Canteli and Nagy [2019], has provided a comprehensive summary of recent research done on energy demand management using reinforcement learning. The most widely used learning algorithms are Q-learning and SARSA. There are also extensions to the Q-learning algorithm and the more recent actor-critic methods [Lee and Powell, 2012, Du and Fei, 2008].

Research has also been conducted into using eligibility traces for faster con-

vergence. Different regression techniques have been applied across research to approximate the Q-table, such as randomised trees, support vector machines, artificial neural networks, echo-state networks and kernal based methods [Ruelens et al., 2014, Claessens et al., 2013, Yang et al., 2015, Shi et al., 2017, Chiş et al., 2015]. An e-greedy approach is the most common strategy underpinning action selection, followed by soft-max exploration. Vázquez-Canteli and Nagy [2019], also highlights that the "curse of dimensionality" as an overarching challenge across all research reviewed, arising from an extremely large and sparse state-action space. Data quantity is a limiting factor in reinforcement learning, such that the agent's learning potential and progress is reduced by a sparse state space. Non-linear feature extraction techniques, such as autoencoders [Ruelens et al., 2016], have also been investigated as a way to reduce the state space of a reinforcement learning agent.

Peak shaving falls under the umbrella of energy demand management as it modifies an energy consumer's demand. Mahapatra et al., [2017] was highlighted in the review paper as performing peak shaving using smart appliances, they used a neural-network based Q-learning approach and chose actions using a softmax function. Another paper performing peak shaving controlled a water heater and so user satisfaction was part of their optimisation which utilised Q-learning and epsilon-greedy exploration [Al-Jabery et al., 2014]. Some energy storage systems pull from multiple sources and as such multi-agent reinforcement learning solutions have been explored. Claessens et al. [2018], uses multiple agents to control an array of thermal loads in order to perform both arbitrage (optimising kwh cost by taking advantage of the TOU tariff) and peak shaving [Claessens et al., 2018]. The scheduling of charging electric vehicles as to not spike the demand on the grid has also been researched using multi-agent reinforcement learning by Dauer et al. [2013], who uses Q-learning with softmax exploration and Vaya et al. [2014], who uses Q-learning with epsilon-greedy exploration [Dauer et al., 2013, Vayá et al., 2014].

While reinforcement learning has been applied to energy demand management problems with limited success within existing literature, Vázquez-Canteli and Nagy [2019]'s paper suggests that further research be conducted in the following areas to more fully explore reinforcement learning's potential contribution to energy demand management problems:

- Pre-processing data in order to reduce the dimensionality of the state space;
- Utilising policy iteration with the purpose of finding a transition function for the system. This will help overcome data quantity limitations by turning the problem into a planning problem;
- Using data augmentation algorithms to increase the quantity of data available to learning agents.

2.2 Energy Demand Profile Clustering

As previously mentioned, popular rule-based control methods result in suboptimal battery usage when a single predefined strategy is applied to a range of variable profile shapes. In order to optimise a battery's potential this research can be improved upon by setting multiple set strategies using clustering techniques. Clustering historical energy demand profiles into groups, such that groups reflect similar profile shapes, enables strategies to be tailored to each data cluster. This notion of clustering energy demand profiles been intensively researched with much success. Wang et al. [2015], found that the most frequent clustering techniques used in energy demand clustering are: K-means, Fuzzy K-means, hierarchical clustering, and self-organising maps. Some research shows that the application of Principal Component Analysis (PCA) - a dimensionality reduction technique - as a pre-process to clustering can lead to better results. This is due to the fact that PCA compresses data by representing relevant information from a large dataset in fewer variables that capture the most variance in the data (the principal components).

2.3 Conceptual Framework

The concept of peak shaving has been extensively researched. Rule-based control methods that utilise a single predefined shave level are most commonly applied to the peak shaving problem. While this method has successfully reduced energy demand peaks, it has been unable to optimally utilise the battery's potential in a combined micro-grid as a static rule cannot take variable energy profile shapes into account. This inability to respond to variable data is a limitation that extends to most peak shaving methods documented in literature, and can result in inefficient demand reduction. The idea of using clustering to set multiple strategies that better represent the shape of historical energy profiles is a novel way to improve upon this research.

While much research has gone into clustering techniques, these techniques have not been practically applied to the peak shaving problem. Despite promising results in terms of creating suitable clusters of historical data, a challenge exists in deciding which cluster is most likely to represent a new day when it arrives. To address this limitation, an approach would need to be devised that can identify underlying patterns within clusters in order to meaningfully predict within which cluster a new day falls. Forecasting methods also lead to practical challenges as they fail to take the dynamism of a new day into account, increasing the risk of a battery running out of capacity and missing the peak. While additional layers, such as prediction error functions, can be added to reduce this risk, optimal utilisation of the battery's potential is still not achieved.

Reinforcement learning is being explored as a solution to the peak shaving problem in order to address the shortcomings associated with the aforementioned approaches. Reinforcement learning, which adjusts battery outputs according

to fluctuating energy demand patterns within a day, has the potential to improve the efficacy of battery use, in turn increasing the likelihood of the battery discharging during the peak and thus saving costs. If reinforcement learning is to achieve effective results within the complex environment of a combined micro-grid however, further research is required into dimensionality reduction techniques for the state space of reinforcement learning agents, as well as research into data augmentation and generation. The following section (Section 2.4) highlights how this thesis aims to enhance the feasible application of reinforcement learning to peak shaving by reducing the complexity of the problem.

2.4 Proposed Methodology

Peak shaving methods described in extant literature are either overly simplistic, resulting in sub-optimal battery usage, or overly complex, resulting in the "curse of dimensionality". This thesis explores a novel approach that reduces the complexity of the peak shaving problem so that more complex methods, such as reinforcement learning, can be practically employed. Clustering energy demand profiles and tailoring predefined control strategies to each cluster is a promising approach in terms of reducing the complexity of the problem. In order to be practically applicable to the peak shaving problem however, an approach is needed that can allocate a new, unseen day to an appropriate cluster. This requires analysis of factors within clusters in order to identify similarities that can inform this allocation of new days to clusters. Once achieved, reinforcement learning agents can be created for each individual cluster, drastically reducing the complexity of an agent's environment as energy demand profiles within a cluster reflect similar profile shapes. Reinforcement learning agents actively learn energy demand trends by exploring their environment and analysing the reward function (described in Section 4.3). As such, an energy demand forecasting profile is unnecessary, further reducing the complexity of the state space.

2.4.1 Methodology for energy system without a solar energy system

In order to employ this novel approach to the peak shaving problem, we first attempt to solve the problem in a simplistic energy system that excludes a solar component (i.e. $E_t^{solar\ generation} = 0$ for all t). The following steps are employed to create an optimal control signal for a simplistic energy system that excludes solar (under the assumption that the battery system is charged to full capacity in the evening during periods of low demand):

1. An array of clustering methods are used to group energy demand profiles. The method that most accurately clusters similar profile shapes will be selected.

2. Clusters are analysed to identify key factors / identifiers that underpin the similarity among profile shapes. To elaborate: is it the day of the week, or the season of the year that contributes to a particular profile

shape? These identifiers enable a new, unseen day to be allocated to an appropriate cluster.

3. Reinforcement learning agents are trained to find the best actions (i.e. the battery's output for the next half hour $E_{t+1}^{battery\ output}$) on total demand profiles within each cluster. As profile shapes within each cluster are similar, the complexity of the environment within which the agent is learning is dramatically reduced.

4. Trained learning agents are tested on a year's worth of unfamiliar data. For each new day, an agent will be selected based on the key identifiers that link that day to a particular cluster.

2.4.2 Methodology for energy system including a solar energy system

To extend the application of this approach to include a solar system in a combined micro-grid, the total energy demand placed on the grid must first be considered. When a solar system is not in use, the total energy demand placed on the grid ($E_t^{grid\ demand}$) is equivalent to the total energy demand of the consumer ($E_t^{total\ demand}$). In instances where a solar system is employed however, the energy demand placed on the grid is equivalent to the total energy demand of the consumer less the amount of solar energy generated by the consumer ($E_t^{solar\ generation}$).

As exemplified in Equation 1, incorporating a solar system into an energy system affects a consumer's overall energy demand on the grid. To account for the added complexity a solar system adds to the process of clustering grid demand profiles, the approach detailed in steps 1-4 above must be adapted. The following steps are employed to create an optimal control signal for an energy system that includes solar:

1. An array of clustering methods are used to group energy demand profiles $E^{total\ demand}$. The method that most accurately clusters similar profile shapes will be selected.

2. $E^{total\ demand}$ clusters are analysed to identify key factors / identifiers that underpin the similarity among profile shapes. These identifiers enable a new, unseen day to be allocated to an appropriate cluster.

3. Grid demand energy profiles $E_t^{grid\ demand}$ are clustered such that profile shapes that are most similar are grouped together.

4. Reinforcement learning agents are trained to generate an optimal battery control signal $E^{battery\ output}$ for each grid demand cluster.

5. Trained learning agents are tested on a year's worth of grid demand unfamiliar data. For each new day in the testing process, an agent will be selected as follows:

6.1 Each new day that arises is allocated to an energy cluster found in Step 1. The average profile shape from that cluster is used to represent the consumer's expected energy demand for that day.

6.2 There are a number of companies that specialise in forecasting the solar generation for the day ahead. As such, we assume that the solar forecast for the day ahead is known. For the purpose of this thesis a day's actual solar data is used to represent the solar forecast (based on the assumption that solar forecasts are one-hundred percent accurate).

6.3 The upcoming day's demand on the grid is forecasted by subtracting the solar energy forecast (see Step 6.2) from the consumer's energy demand forecast (see Step 6.1).

6.4 The upcoming day's forecasted grid demand is allocated to a grid demand cluster based on which cluster best represents the forecast's profile shape.

6.5 The appropriate agent is selected based on the cluster that is chosen.

3 Methods Overview

This chapter provides a high-level overview of the methods to be used in solving the peak shaving problem outlined in this thesis. Sections 3.1 and 3.2 outline principal component analysis and self-organising maps which are used for pre-processing and visualisation of data. Section 3.3 overviews the clustering techniques used for deriving insight into the energy demand data. Section 3.4 introduces reinforcement learning, and section 3.5 finishes the chapter off with an explanation of neural networks which this thesis uses as a function approximator for the reinforcement learning agents.

3.1 Principal Component Analysis

Principal component analysis is commonly used for noise reduction, data compression, pattern recognition and data visualisation. It takes correlated variables and transforms them into a set of fewer uncorrelated variables using eigen-decomposition or single-value decomposition where vectors capturing the most variation in the data are found. Expressing the data in fewer variables (principal components) is useful as it allows fewer variables to be given to methods, thereby reducing the complexity whilst still being able to give the methods the relevant information. For a more in-depth overview into principal component analysis refer to Shlens, [2014].

3.2 Self-Organising Maps

Self-Organising maps are an unsupervised learning algorithm, with their main use being for visualisation of data as they preserve the topology of the input data. Self-organising maps can take data points which are in a high dimensional input space and map them to a lower dimension by grouping the points onto a two/three dimensional lattice of neurons. The mapping leads to neurons with similar weightings being physically closer together on the map. Since similar neurons lie close to one another on the map, it makes self-organising maps a useful tool for visualising clusters.

In order to map the input variables to the neurons, neurons are first initialised with random weightings. An input variable x is then randomly selected and presented to the network. The similarity between the input variable and each of the weight vectors is calculated and the neuron with the highest similarity is deemed the winner. The winner's weights w^{winner} are then adjusted to be closer to the input variables

$$w^{winner} = w^{winner} + \beta(x - w^{winner}) \tag{3}$$

where β is the learning rate. Furthermore, neurons in close proximity to the winning neuron are also adjusted, the closer the neuron is to the winning neuron the greater the adjustment. This process is repeated for a predetermined number of iterations [Kohonen et al., 1996].

3.3 Clustering

Clustering is a form of unsupervised learning, taking unlabelled data and segmenting it into groups (clusters). Clustering methods aim to find clusters which have high inter-cluster similarity and low intra-cluster similarity. This similarity can be calculated using different distance measures. The distance measure chosen would depend on the type of data being used. Clustering methods are broken up into different categories based on the way they segment the data, these categories include distance-based, density-based, partitioning-based and model based methods [Rokach and Maimon, 2005].

3.3.1 DBscan

Density Based Spatial Clustering of Applications with Noise (DBscan) is a density based clustering method [Çelik et al., 2011]. Since this method is based on the density of the data, it is able to cluster groups of different shapes whilst being robust to outliers and noise. Although useful, this method needs user input in order to find meaningful groupings [Khan et al., 2014]. The user input required is to determine the ϵ (maximum distance between points for them to be connected) and $minpPts$ (the number of points in range ϵ required for a point to be required a core point). The algorithm is very sensitive to these variables, and if data has clusters with varying densities it can make it very hard to determine clusters within the data.

Once the parameters have been decided upon, the algorithm operates as follows:

- For each observation, the points which lie within an ϵ radius are found. If a point has $minPts$ or higher other points within it is radius it is classified as a core point.
- Using a neighbour graph connect all of the connected components between the core points.
- For non-core points assign to nearest cluster if they are within distance ϵ of a core point within the cluster, otherwise classify them as noise.

3.3.2 K-means

K-means is a partitioning-based clustering method. The k-means algorithm iteratively assigns clusters by randomly choosing a predetermined number of centroids and assigning all the data points to their closest centroid, the centroids are then recalculated using the mean value of the assigned data points. This process continues until the cluster groupings stabilise [Likas et al., 2003].

3.3.3 K-medoids

K-medoids is similar to K-means and shares the same algorithm, the only difference being instead of using centroids, actual data points from the input data called medoids are used. Medoids are the most-central observation in the cluster and using them makes the algorithm more robust to outliers [Zhang and Couloigner, 2005].

3.3.4 Hierarchical Clustering

There are two types of hierarchical clustering: divisive and agglomerative. Agglomerative hierarchical clustering puts each data point into its own cluster and consecutively merges clusters until all the data points have been merged into one. The consecutive merging creates a dendrogram which can be used to decide on the number of clusters [Nielsen, 2016]. The process of merging differs based on what linkage technique is being used, linkage techniques include:

- **Average linkage** which calculates how well a data point will fit into a cluster by averaging the distance between itself and all points within the cluster.
- **Complete linkage** calculates the distance between the data point and the furthest data point in the cluster.
- **Single linkage** calculates the distance between the data point and the closest data point in the cluster.

Divisive hierarchical clustering works in the opposite direction, starting with all data points in one cluster and splitting them step by step until they are all in their own cluster.

3.3.5 Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are a model-based form of clustering. Model-based clustering makes the assumption that the data has several separate subpopulations within the data with their own distributions that mix together to create the overall distribution. Instead of using distance measures clusters are represented by a paramatised distribution, in the case of a GMM the assumption is the underlying subpopulations come from a Gaussian distribution. GMMs find clusters by optimising the fit of the data to the gaussian models of each cluster by tweaking the distribution parameters using the EM algorithm [Maugis et al., 2009].

3.4 Linear Programming

Linear programming is an optimisation technique which is concerned with maximising or minimising a convex function over a convex set. In Linear programming both the objective function and constraints are linear functions of the decision variables. A standard linear program can be expressed as follows:

$$\begin{aligned} \text{Minimise } & c^t x \\ \text{subject to } & Ax \geq b \\ & x \geq 0 \end{aligned}$$

Here $x \in R^n$ is the vector of decision variables, and $c \in R^n$; $A_1, \ldots, A_m \in R^m$ and $b_1, \ldots, b_m \in R$ are constant parameters. A linear program can be solved to optimality quite efficiently using the simplex algorithm. While an efficient solution method is available for solving linear programming problems, the difficulty lies in being able to formulate an optimisation problem into a linear programming model. Formulating a linear programming model involves identifying the decision variables, objective function, and constraints of the problem, and writing the objective function and constraints as a linear function of the decision variables. Decision variables are the variables whose values are to be determined by the simplex algorithm. The values obtained by the simplest algorithm constitute the solution to the problem. Constraints are restricting the decision variables. Often, multiple transformations must be performed to ensure that the problem is linear and convex. Linear programming is widely used and appears in many problem areas, such as telecommunications networks, production planning and power systems [Matousek and Gärtner, 2007].

3.5 Reinforcement Learning

Reinforcement learning is a comprehensive framework used for the purpose of solving problems which can be modelled as Markov Decision Processes (MDPs) [Sutton and Barto, 2018]. Its main objective is to maximise a reward signal through the state transitions of the MDP. In the context of reinforcement learning the MDP is seen as the environment the reinforcement learning agent interacts with, and is defined using tuples $< S, A, P, R, \lambda >$ where:

- S is the set of states
- A is the set of actions
- $P_{ss'}$ is a state transition probability matrix
- R is the reward function
- and λ is the reward discount factor

In an MDP all states have the Markov property, meaning the probability of transitioning to a new state is solely based on the current state as defined in Equation 4.

$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, \ldots, S_t] \tag{4}$$

In order to transition between states, an action a in $\mathbf{A}$ need to be taken based on a probability distribution over the action set, called the policy $\pi(s)$. In order to solve the reinforcement learning problem, an optimal policy $\pi_*(s)$ needs to be found. $\pi_*(s)$ is the policy in which an agent always takes the action which will maximise the accumulated future rewards and is equivalent to acting greedily with respect to $V_*(s)$ - a value table where each state's value is the accumulated discounted reward an agent can get by taking optimal actions from that state onward. Both $\pi_*(s)$ and $V_*(s)$ can be solved using an array of different dynamic programming methods.

In most practical real-world problems, the MDP is generally unknown or too large to be fully defined i.e. the state transition probabilities $P_{ss'}$ are not known. This means the only information we can attain from the MDP is by sampling episodes from the environment. In turn this leads to the need for methods which can extend the dynamic programming methods to work with unknown model dynamics. These methods are known as model-free methods.

Model-free prediction methods can be used to iteratively update the value table of a given policy $V_\pi(s)$ by averaging future returns from each state under the given policy $\pi(s)$ as seen in Equation 5. These methods differ in how they calculate the returns G_t.

$$V(S_{t+1}) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G_t - V(S_t)) \tag{5}$$

Monte Carlo simulation calculates the return of a state by sampling multiple episodes to completion starting from the given state and averaging the discounted rewards from the samples.

$$G_t = R_{t+1} + \lambda R_{t+2} + ... + \lambda^{T-1} V(S_T) \tag{6}$$

Temporal Difference Learning (TD(n)), is a method where the return is calculated looking only $n+1$ steps ahead to calculate the return. For example TD(0) looks at only 1 step ahead: even though this return is calculated using values which may not have been updated, it has been proven to converge to the correct values [Sutton and Barto, 2018].

$$G_t = R_{t+1} + \lambda V(S_{t+1}) \tag{7}$$

These ways of calculating the returns have their limitations. Monte Carlo creates state values with high variance since it looks far into the future, but despite this high variance these values are unbiased as they are not predicting any future rewards. TD(0) on the other hand is biased due to state values being updated with predicted values for future states rather than true returns but in turn has lower variance. These two methods create a trade-off between bias and variance.

The n-step in TD(n) can be used as a way to try balance the bias variance trade-off, where increasing the number n increases both the bias and variance.

TD(∞) looks to the end of all episodes and is equivalent to Monte Carlo learning.

$$G_t^{(n)} = R_{t+1} + \lambda R_{t+2} + ... + \lambda^{n-1} R_{t+n} + \lambda^n V(S_{t+n}) \tag{8}$$

There lies another method TD(λ) which averages returns of different lengths in order to try strike a balance between the bias and variance.

$$G_t^{\lambda} = (1-\lambda)\Sigma_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)} \tag{9}$$

Although these methods allow us to predict the value function for a given policy, we still need to focus on the problem of control which involves optimising the policy. In order to optimise the policy we need to take actions which maximise the reward.

$$\pi^{'}(S) = \text{argmax}_{a \in A} R_s^a + P_{ss'}^a V(S^{'}) \tag{10}$$

Since the $P_{ss'}^a$ is unknown, Q-values $Q(s,a)$ can be used instead, $Q(s,a)$ is the value of taking a given action from a given state.

$$\pi^{'}(S) = \text{argmax}_{a \in A} Q(S,a) \tag{11}$$

In some cases there is direct access to the environment and data can be generated by directly interacting with the environment. In these cases on-policy approaches such as SARSA are useful, where Q-values are updated by sampling episodes using a given policy which is iteratively improved by acting greedily with respect the updated Q-values.

$$Q(S,a)_{t+1} \leftarrow Q(S,a)_t + \alpha(R + \lambda Q(S^{'}, a^{'})_t - Q(S,a)_t) \tag{12}$$

In other cases we do not have direct access to the environment but rather data of some processes that have been sampled under different policies. This opens up a need for off-policy approaches such as Q-learning. Q-learning uses episodes run under different policies: instead of using actions sampled from the current policy the Q-values are updated using the value of taking the action which maximises the current Q-values.

$$Q(S,a)_{t+1} \leftarrow Q(S,a)_t + \alpha(R + \lambda \text{max}_{a'} Q(S^{'}, a^{'})_t - Q(S,a)_t) \tag{13}$$

Although it is desirable to act greedily in terms of the value table, if the value table has not converged to the true value table, acting greedily can stunt the exploration of the whole state space and in turn not allow better solutions to be found. There are different approaches for solving this dilemma, the most common one being the epsilon-greedy approach. The epsilon-greedy approach is similar to a greedy approach the exception being that instead of always acting greedily it takes a random action with the probability of epsilon. This random exploration guarantees that the true value table can be found as iterations tend towards infinity.

In many practical reinforcement learning problems, the environment's state space is extremely large/sparse and calculating/storing a value for every Q-value becomes unfeasible. This opens up room for function approximators, which take in the state and predict the values of actions allowing for a much smaller representation of the value table and for interpolation to be made in between states, allowing for these methods to handle a very large/continuous state space [Li, 2017].

3.6 Neural Networks

An artificial neural network (ANN) is a feed-forward network in which data is fed in through an input layer and moved forward through hidden layers to an output layer which gives the function mapping of the input. An ANN can be represented by composing different functions together [Aurélien, 2019]. These functions are connected in a chain, for example:

$$f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x}))) \tag{14}$$

Where each function represents a hidden layer within the network, the longer the chain the greater the depth of the network hence the name "deep learning" that is often associated with neural networks. This chain of functions gives the network a large amount of flexibility allowing them to approximate many different complex non-linear functions in order to achieve statistical generalisation [Goodfellow et al., 2016].

The simplest form of an ANN is the perceptron as can be visualised in Figure 5. The perceptron consists of a weighted average of the inputs passed through an activation function. In this case the activation function is a sigmoid function which can be modelled using Equation 15:

$$\varphi(\cdot) = \frac{1}{1 + e^{-x}} \tag{15}$$

The activation function transforms the linear weighted sum into a non-linear function. There are many different types of activation functions that can be used such as step, relu and tanh. To see more activation functions and how they differ refer to [Nwankpa et al., 2018].

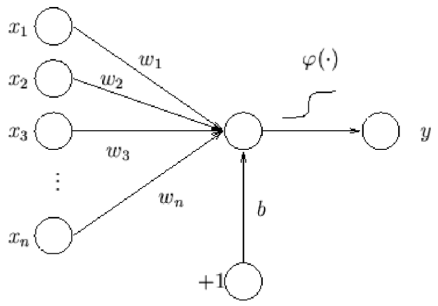


Figure 5: A diagram of a perceptron [Hemmatinezhad et al., 2022]. The output y is a weighted sum of the inputs passed through an activation function.

Since the perceptron is not differentiable it cannot be trained with a closed form solution like linear regression can, instead training instances have to be fed through the network one at time and the weights adjusted to drive the function the network is approximating $f(x)$ closer to the function captured in the training data $f'(x)$. This can be done using the perceptron learning rule [Aurélien, 2019].

$$w_i^{next\ step} := w_i + \eta(y - \hat{y})x_i \tag{16}$$

Where w_i is the ith input's weight, η is the learning rate, $\hat{y}$ is the models predicted output, y is the target output from the training data, and x_i is the ith input of the training input. This weight update pushes the weights closer towards being able to map the target output from the given input.

Despite being able to capture non-linear functions the perceptron has its limitations and is incapable of capturing some trivial problems [Aurélien, 2019] such as the XOR function. This lead to the development of the multi-layer perceptron more commonly knows as the ANN.

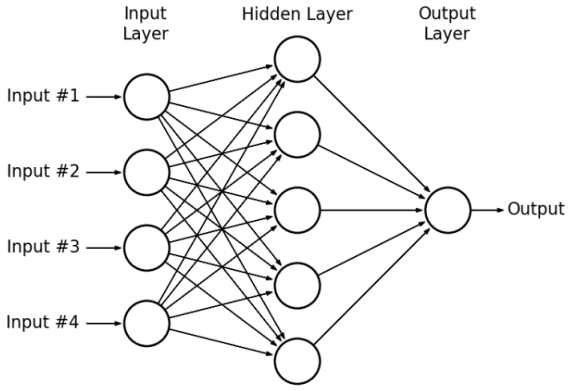


Figure 6: A diagram of an artificial neural network. The first layer is the input layer, the second layer is a hidden layer where each neuron's input is a weighted sum of the input nodes. The final layer represents the output neuron [Hassan et al., 2015].

An ANN is created by connecting one or more layers of perceptrons working in parallel, these layers are known as the hidden layers. This chaining effect as discussed above allows for more complex functions to be approximated.

In order to train this larger network backpropagation is used. For each training instance the input is fed through the network, the error between the output and target calculated and the error propagated back through the network in order to move the weights in a direction which will push the predicted values closer to the targets. Backpropagation is similar to the perceptron learning rule, but since we now have chained functions partial derivatives are used to determine how much each weight contributed to the error and update them accordingly. To get a more thorough explanation of back propagation refer to LeCun et al, [1988].

Once the network has been trained generating predictions is a simple task, a forward pass of the new inputs can be done to attain the predictions in the output layer.

4 Data Exploration and Energy Demand Cluster Analysis

Deriving insight into energy demand trends at the commercial shopping centre is crucial when trying to reduce the complexity of the peak shaving problem. This chapter focuses on the techniques used to find cluster groupings within the data. It uses these groupings to gain key understanding into why grouped energy demand trends are similar. It further investigates the effects of temperature on the energy demand as a substantial part of a commercial load is comprised of energy used by Heating, Ventilation, and Air Conditioning (HVAC) systems.

4.1 Commercial Shopping Centre Metered Energy System Data

The data-set contains information collected from energy meters and the solar system's monitoring system installed on the rooftop of a commercial retail shopping centre. The data was compiled by averaging the measured data over 30 minute intervals over the period of 2018 and 2019. A list of measured variables can be found in Table 1.

The energy demand data measures the total energy consumption of the commercial shopping centre. This is used in this chapter to gain insight into the energy demand trends. The solar generation data accounts for the energy outputted by the installed solar system. The measured temperature is also a useful variable as energy demand can be affected by the temperature, due to HVAC systems making up a large percentage of the load [Vázquez-Canteli and Nagy, 2019].

Variable	**Unit**	**Time interval**
Energy Demand	kW	30 minutes
Solar Generation	kW	30 minutes
Temperature	$°C$	30 minutes

Table 1: A table of the commercial shopping centre's metered energy system variables

The format of the given variables can be seen in Table 2.

Date	**0:00**	**0:30**	...	**23:00**	**23:30**
01/01/2018					
02/01/2018					
...					
30/12/2019					
31/12/2019					

Table 2: Data format of the given variables.

4.2 Energy Demand Profile Normalisation

As weather fluctuates over the seasons of the year, the energy demand can be affected. This can be seen in Figure 7, where seasonal trends in the form of changing magnitudes can be visualised over the course of the year. Energy demand rises from approximately September reaching a peak between January and March thereafter falling to lower demands over the middle of the year. There were unusual reductions in the energy demand between October and November and again between November and December. The reductions between November and December are likely due to the reduced traffic at shopping centres over Christmas and Boxing day.

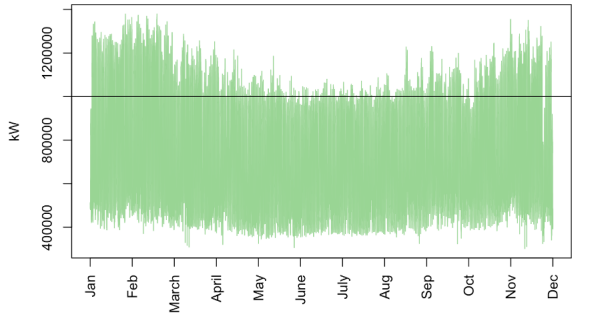


Figure 7: The energy demand of the commercial shopping centre plotted across a year in green, the black line shows 1 MW as reference to help better visualise the seasonal energy demand fluctuations over the year.

In the context of peak shaving when trying to reduce the monthly maximum demand, if the day in a billing cycle (a month) with the maximum demand peak is missed, no savings can be attained that month despite best efforts in reducing the energy demand peaks on other days. Although the battery can be used to only target days were suspected maximum magnitude of the energy demand is high, this runs the risk of missing the day with the actual highest peak. In order to reduce the risk of missing the maximum energy demand spike in a month when solving this problem we aim to target daily energy demand peaks. Targeting daily energy demand spikes reduces the chance of underestimating the demand requirements of a day ahead.

When targeting the daily energy demand spikes the overall magnitude of the spike is inconsequential due to limited energy capacity in the battery. The battery's energy capacity has to be used in such a way that it does not deplete before an energy demand spike is over as this will result in limited or no peak demand reduction. This means the battery system can only reduce the peak by an amount limited by its energy capacity and does not depend on the total magnitude of the energy demand peak. What is of importance to know is when

a peak is occurring and its overall shape. For this reason the energy demand profiles are normalised to reduce the variation introduced by magnitude. This will help when we cluster the energy demand profiles as the clustering techniques can focus on the shape of the profiles without magnitude having an effect on the found clusters.

The data was normalised using a min-max scaler defined in Equation 17,

$$ED_t^{norm} = \frac{ED_t - ED_{min}}{ED_{max} - ED_{min}} \tag{17}$$

Where, the normalised energy demand profile at time t, namely ED_t^{norm} is calculated by scaling the original energy demand value ED_t using the minimum and maximum energy demand reached over the course of the day ED_{min}, ED_{max}.

4.3 Daily and Monthly Trends

When we analyse the average energy demand profiles for each day of the week in Figure 8, we can see how the energy demand shape is affected by the trading hours. In general, weekdays have similar demand shapes, whilst Saturdays and Sundays differ with energy demand dropping earlier than on weekdays. This is expected as the shopping centre closes earlier on weekends.

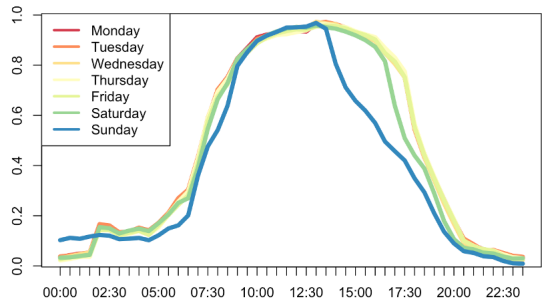


Figure 8: The averaged energy demand profile for each day of the week. The colours which represent each day are tabulated in the legend in the top left corner.

When we examine the average energy demand profiles for each month in Figure 9, we can see that during winter months in the middle of the year (namely June, July, August) there is an increased demand in the earlier hours of the morning. Although interesting, these fluctuations in the early morning do not hold much importance in the context of the demand charge reduction as they occur during times of low demand usage.

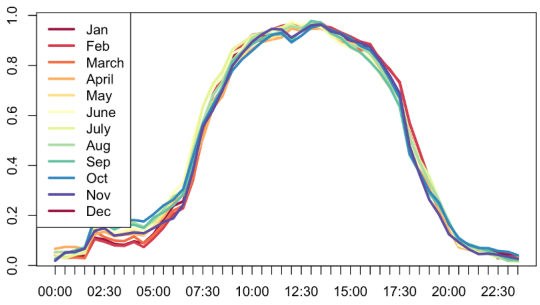


Figure 9: The averaged energy demand profile for each month of the year. The colours which represent each month are tabulated in the legend in the top left corner.

4.4 Principal Component Analysis

The metered energy data is averaged over 30 minute time intervals as described in Section 4.1. This results in each day's energy demand profile comprising of 48 variables. In other words, each day's energy demand profile can be thought of as a point in a 48 dimensional space. Principal Component Analysis (PCA) is used to reduce the 2019 energy demand profile's dimensions from high to low whilst still retaining the important information.

The information explained or variation captured by each of the principal components is proportional to the height of each bar in the scree plot found in Figure 10. The first five components explain approximately 83% of the data and the rest of the individual components capture very small amounts of variation. Going forward the first five principal components will be used to represent the 48 variable energy demand profile sacrificing some amount of the signal to gain a more wieldy data set.

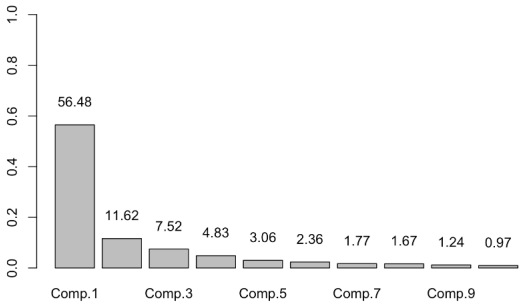


Figure 10: Scree Plot: Each bar represents the variation (information) explained by the given component. The most variation is captured by the first component and the captured variation decreases as the components increase.

Figure 11 shows how the days disperse across the first two principal components. Two distinct clusters emerge, weekdays and Saturdays (1,2,3,4,5,6) form a predominant cluster. Saturdays (6) accumulate to the right of the cluster and weekdays to the left. The second cluster is made up with a majority of Sundays (7). Investigating the loading values of the first principal component shows us that the time periods between 3 pm and 7 pm contribute the most information or variation in the direction of the principal component. The first principal component is therefore indicative of when the load starts to drop, with earlier dropping energy demand profiles being further right (Sundays) and later dropping energy demand profiles being further left (Weekdays).

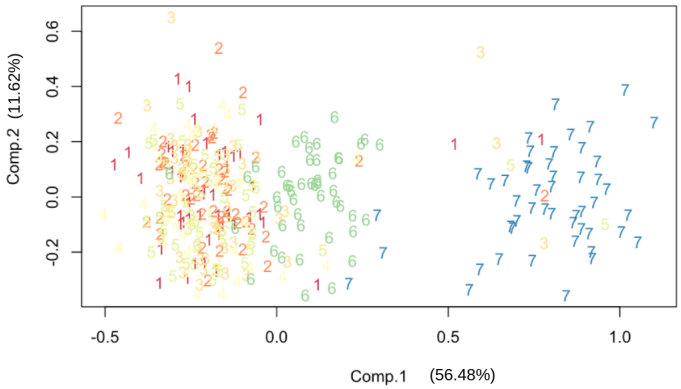


Figure 11: Energy demand profiles plotted across the first two principal components with points coloured by day of week and represented by a number. The numbers 1 to 7 represent the weekdays in order from Monday to Sunday.

Figure 12 shows the energy demand profiles across the first two principal compo-

nents coloured by month, this shows us some of the seasonality across the second principal component. The loading values of the second principal component are higher between 4:30 a.m. and 9 a.m., the times associated with the increased energy demand in the winter mornings as shown in Section 4.3. This implies that the second principal component is a measure of some of the seasonality.

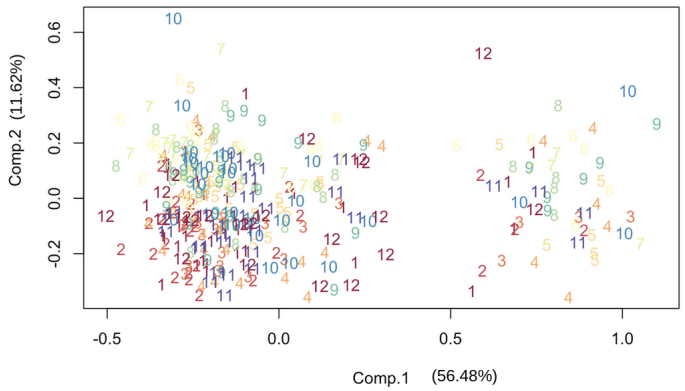


Figure 12: Energy demand profiles plotted across the first two principal components with points coloured by month and represented by a number. The numbers 1 to 12 represent the months in order from January to December.

4.5 Outliers

In order for useful clusters within the energy demand profiles to be found the variation within found clusters needs to be reduced. Since outlying profiles increase the variation it is important to investigate why these outliers occur. To find these outlying energy demand profiles, the Desnsity Based Spatial Clustering of Applications with Noise (DBSCAN) is utilised to identify energy demand profiles that fall into low density areas. The ability of DBscan to identify outliers depends on the settings of two parameters: $minPts$ and ϵ as described in Subsection 3.3.1. In our experiments values of 10 and 0.2 for $minPts$ and ϵ were used respectively. The outlying energy demand profiles identified can be seen in Figure 13. When visualising how the profiles disperse across the first two principal components (to be explained in Section 4.5) we see that the outlying profiles fall much higher on the second principal component.

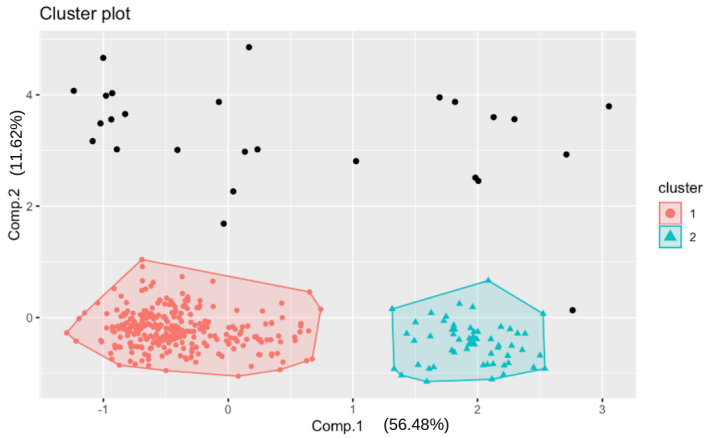


Figure 13: DBSCAN output: profiles plotted across the first two principal components. Two distinct clusters can be seen in red and blue with the black dots indicating the energy demand profiles which were classified as outliers or noise.

The outlying demand profiles are explored further in Figure 14 to determine what characterises them from the other profiles and whether this characterisation could be have been used to preempt them. The commonality between the found outliers were periods of zero energy consumption: this can be caused by load shedding or faulty metering equipment. Since the drops in energy consumption last for two hour periods consistent with load-shedding intervals, they are more likely to have been caused by these planned power outages.

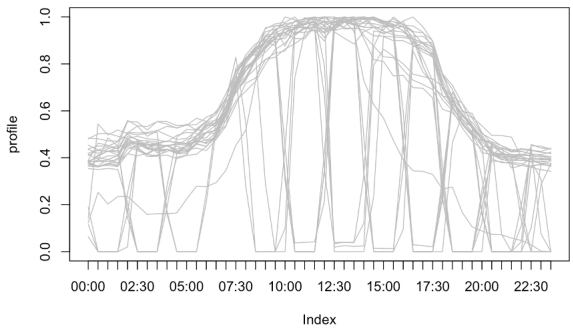


Figure 14: Plots of the found outlying profiles, identified by the sudden drops in energy demand.

In order to limit increased variation between the clusters we are attempting to find, the outliers are identified and removed so that we can better find the

true patterns in the energy demand profiles. Caution will have to be exercised regarding anomalies when implementing the system as the peak shaving system is modelled on the assumption that the energy demand profiles follow trends. For example, if there is equipment testing that needs to be done which will increase the energy demand and deviate from general demand trends, the testing should be planned for a period of low demand as to not increase the maximum. Further to this, if trading hours are changed, the algorithm would need to be adjusted. Since the found outlying energy demand profiles have periods of zero energy demand, they should not cause interruptions to the battery control signal's objective to reduce the peak demand. If the energy demand drop happens during what would have been a peak, the battery will have more capacity left giving it greater potential to reduce the maximum demand.

4.6 Clustering Methods

The clustering methods explored were K-means, K-mediods, gaussian mixture models and hierarchical clustering. These methods are explained in Section 3.3. The metric used to calculate the distance between the profiles represented by five principal components was euclidean distance, where the distance between two points p and q of dimension n can be calculated using equation 18:

$$d(p,q) = \sqrt{\sum_{i=0}^{n}(p_i - q_i)^2} \tag{18}$$

K-means, K-mediods and gaussian mixture models require a user's input of the number of clusters to be found before they are able to be run. R's **NBclust** library utilises 30 different indices used for analysing the number of clusters apparent in the data to propose the best number of clusters determined by combining the results of the indices [Charrad et al., 2014]. Using **NBclust** the number of clusters apparent in the data was investigated, the library indicated towards the data containing five clusters. Therefore each clustering method was used to try find five clusters within the data.

Our intent is to find clusters containing energy demand profiles with similar shapes. In order to quantify this similarity of shape the original data which represents the shape needs to be used and not the five principal components used to cluster the data. Correlation is a good way to determine the similarity of shape as it quantifies the strength of a relationship between two variables. A Pearson's correlation coefficient r can be calculated using Equation 19, where x and y are variables which can represent two energy demand profiles. The final calculated correlation coefficient gives a value representing how much of the variance in the two variables is captured by their combined relationship (co-variance) indicated by a value between -1 and 1.

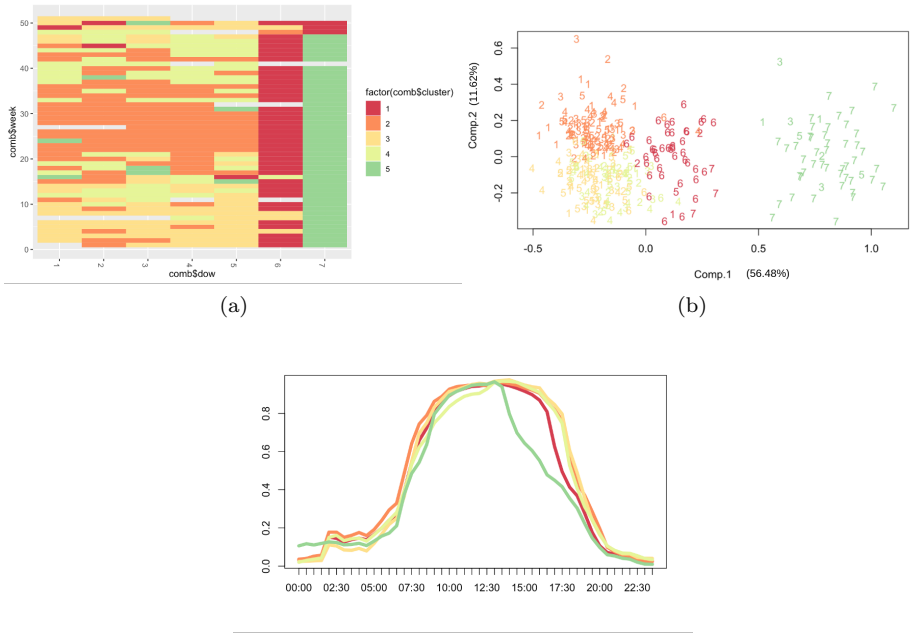
$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})\sum(y_i - \bar{y})}} \tag{19}$$

Each cluster should contain energy demand profiles of similar shapes. Within a cluster if we calculate the correlation between each pair of profiles and average them we will have a measure that gives a good idea of how similar the profiles in the cluster are. By averaging this measure across all found clusters we can get an overall idea of how well a clustering technique is doing across all clusters. This averaged measure across the clusters will be referred to as average cluster correlation and will be used as a comparison to decide between the clustering techniques. The average cluster correlation for each of the clustering techniques are tabulated in Table 3:

Clustering Method	Average Cluster Correlation
K-means	0.9874
K-mediods	0.9873
Hierarchical - Single	0.9834
Hierarchical - Complete	0.9754
Hierarchical - Average	0.9849
Hierarchical - Centroid	0.9836
Gaussian Mixture Model	0.9844

Table 3: A tabulation of the average cluster correlation calculated for each of the different clustering techniques explored.

The clustering technique with the highest average cluster correlation is the K-means algorithm. The found clusters using this method can be visualised on the PCA plot in Figure 15 along with the clusters over the calendar year and the average profiles of the found clusters. Looking at the overall pattern of the clusters we see that Saturdays and Sundays seem to be broken up into their own clusters in red and green respectively. Weekdays are separated into three clusters along the second principal component coloured in orange, yellow and dark yellow. If we look at the calendar the three clusters consisting mainly of weekdays can be seen to change seasonally over the year with the winter months in orange and the light yellow cluster being the transition into the darker yellow warmer months. These three clusters split across the second principal component are indicative of the seasonal changes seen in the profiles in the early mornings (this was explored in Section 4.3). Given that these are times of lower energy demand during the day and these seasonal differences between the demand profiles will not affect the peak energy demand the three clusters consisting mainly of weekdays can be combined in the context of our objectives.



(a) (b)

(c)

Figure 15: K-means resultant clusters: Subfigure a illustrates the clusters over a calendar year, subfigure b shows the energy demand profiles across the first two principal components coloured by cluster number and subfigure c shows the average profile of each cluster.

4.7 Eliciting Common Factors

Finding clusters helps to identify patterns inherent in the historical data, but when dealing with new data, profiles cannot directly be grouped in with an already defined cluster. In order to be able to group a new day to an already defined cluster, the clusters found are used to elicit similar factors between the found clusters.

Cluster 1, predominantly made up of Saturdays had a few exceptions of different days of the week within the cluster. These exceptions are listed in Table 4 and are all associated with either the festive season in December or the Easter holidays where the shops might have introduced different trading hours. The 26/04/2019 is the Friday before Freedom Day which falls within the Easter holidays. The rest of the days were either Sundays or Weekdays during the December festive season. This means that the Sunday's trading hours were extended to later during the festive season, and some of the weekdays close a bit earlier causing the energy demand to ramp down slightly earlier. The profiles found within this cluster can be seen in Figure 16.

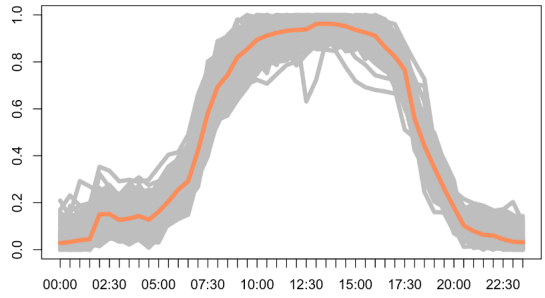


Figure 16: A plot of energy demand profiles found in cluster 1 in grey, with the orange line representing the cluster average energy demand profile.

Date	Type of Day
26/04/2019	Day before Freedom Day
15/12/2019	Day before Day of Reconciliation
16/12/2019	Day of Reconciliation
22/12/2019	Festive Day
24/12/2019	Festive Day
29/12/2019	Festive Day

Table 4: A table of the days within cluster 1 that were not a Saturday, they all relate to the Easter and festive holidays.

Cluster 2 was created by merging the three clusters which span over the second principal component (orange, light orange and yellow) and is predominantly made up of weekdays, the energy demand profiles within this cluster can be seen in Figure 17. The red line indicates the average trend of the profiles within the cluster. The energy demand profiles ramp down close to 6 p.m consistent with weekday trading hours. There were three days grouped into this cluster that were not weekdays, these exceptions were all Saturdays and fell within the summer months. Saturday's generally drop demand slightly earlier than weekdays (as shown in Section 4.2) and likely ramped down later because traffic of shoppers was pushed later with people out enjoying the summer during the day. Since weekdays and Saturdays do not fall too far apart on the PCA plot as seen in Figure 11 these three exceptions were ignored.

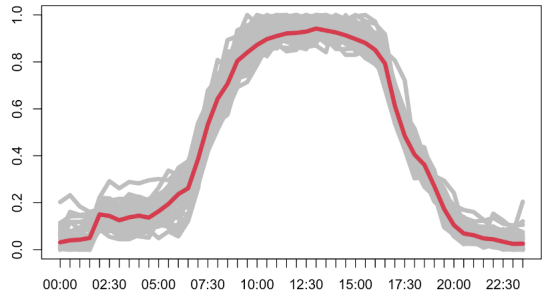


Figure 17: A plot of energy demand profiles found in cluster 2 in grey, with the red line representing the cluster average energy demand profile.

Date	Type of Day
26/01/2019	Saturday in Summer
23/02/2019	Saturday in Summer
21/12/2019	Saturday in Summer

Table 5: A table of the days within cluster 2 that were not weekdays, all these days occurred on a Saturday during the summer months.

Cluster 3 consists of majority Sundays but had some weekdays grouped into the cluster, on closer inspection these weekdays were all public holidays and are listed in Table 6. Public holidays have similar trading hours to Sundays and so it makes sense that these would be grouped together.

The public holidays that were not grouped in this cluster were on a Saturday, a Monday and a Tuesday. Public holidays which fall on a Saturday have the same trading hours as a general Saturday and so will fall into cluster 1. The other two public holidays which fell on the Monday and Tuesday fell in the December Festive season where trading hours had been changed.

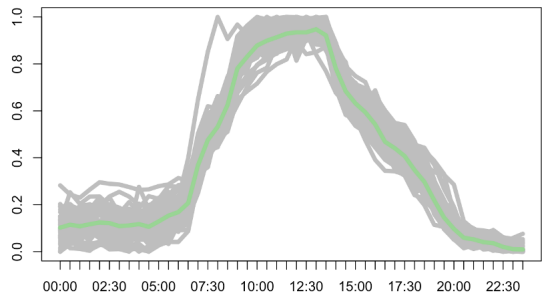


Figure 18: A plot of energy demand profiles found in cluster 3 in grey, with the green line representing the cluster average energy demand profile.

Date	Type of Day
19/04/2019	Good Friday
22/04/2019	Family Day
01/05/2019	International Workers' Day
08/05/2019	National and Provincial Government Elections
17/06/2019	Youth Day
09/08/2019	National Women's Day
24/09/2019	Heritage Day
25/12/2019	Christmas Day

Table 6: A table of the days within cluster 3 that were not a Sunday, all these anomalies occured on public holidays.

Based on the similar factors elicited from the found clusters, a new day can be grouped into one of the defined clusters by following these rules:

- Saturdays are grouped with cluster 1.
- Weekdays are grouped with cluster 2.
- Sundays are grouped with cluster 3.
- Public holidays which fall on weekdays are grouped with cluster 3.
- Days that fall within the festive season and Easter holidays which have different trading hours are grouped with the cluster which closest resembles the updated trading hours.

4.8 Temperature Effects

Although the energy demand profiles within the found clusters are similar they still have some unexplained variation. Temperature is analysed to see if it can help explain some of this variation. Looking at the relationship between average temperature and the daily maximum energy demand we see there is definite positive correlation between them. The days with a higher average temperature have a higher maximum demand. This makes intuitive sense as days with higher temperatures would lead to more power being drawn from the HVAC systems. If we reached cooler temperatures we would also require more heating.

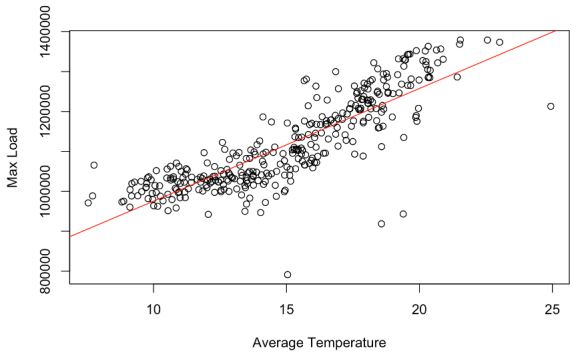
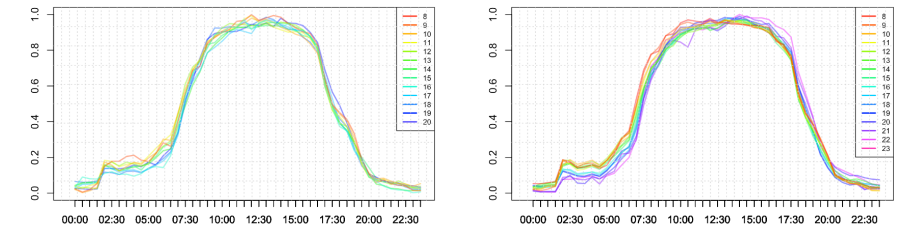
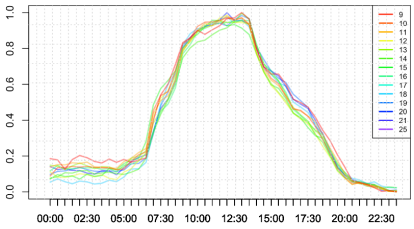


Figure 19: Scatter plot of the maximum energy demand per day vs average temperature of each day with a line of best fit in red.

Since we have more interest in the shape of the profiles than the maximum half hour energy used, the relationship between temperature and energy demand shape is further investigated. In order to visualise if the shape of the profile is affected by the average temperature, we look at Figure 20. From this we can see that the average day temperature does affect the shape of the energy demand profile with the lower temperatures causing the energy demand profile to ramp up and down earlier. Therefore the temperature creates a shift (time delay) in the profiles. This phenomenon is more prevalent in cluster 1 comprised mainly of weekdays. There is also a difference in the energy consumption in the early morning, with colder days drawing more power. Since we are interested in reducing the maximum peak power this difference in the shape of the profile in the early morning is not of direct interest.



(a) (b)



(c)

Figure 20: Plots of the found clusters coloured by average temperature: Subfigure (a) shows cluster 1, Subfigure (b) cluster 2 and Subfigure (c) cluster 3. The plots show there is a shift in the energy demand profiles associated with temperature, this shift is the most predominant in cluster 2.

4.9 Conclusion

Meaningful cluster groupings of the energy demand profiles were found. On closer inspection of these found clusters, we were able to draw some useful conclusions about why the clustered energy demand profiles were similar. This is extremely useful in the context of this thesis' objective as it can be used to reduce the environment of our reinforcement learning agents going forward. This reduced complexity can be achieved by creating separate agents for each cluster, where each agent's environment will be less complex as the intra-cluster variation has been removed from it. The reinforcement learning agents will be able to learn the trend of the cluster's energy demand profiles through experiencing the environment and the use of a reward signal (to be defined in Section 5.4), this negates the need to explicitly give an agent a predicted day ahead energy demand in turn drastically reducing the state space. This would not be possible without the found similarities of factors between clusters as new days need to be assigned to a cluster so that the correct agent can be used.

The temperature creating a time delay in the profiles, accounts for some of

the variation seen between profiles. When using reinforcement learning this variation caused by temperature can be accounted for by adding average temperature into the state space of the reinforcement learning agents. By adding this variable into the state space we give the agent more information about the cause in variation between the energy demand profiles within the clusters.

5 Peak Shaving on Load Profiles using Reinforcement Learning Agents

This chapter outlines the reinforcement learning agents trained for the purpose of demand charge reduction on a commercial shopping centre. This chapter puts complete emphasis on the energy demand usage of the property by acting as if their was no solar system installed. The reinforcement learning agents make use of insights derived from the clusters found in the previous chapter. Once the reinforcement learning agents have been fully explained the chapter continues on to the training and testing methodology used and the results they derived.

Chapter 7 will extend the reinforcement learning agents from this chapter to incorporate a solar power system on the commercial shopping centre's roof.

5.1 Reinforcement Learning Agent Overview

The "curse of dimensionality" has been encountered when attempting similar problems as the way in which a daily energy requirements can transition varies. This variation depends on factors such as day of the week, season etc. A single agent can be trained to learn these variations between demand profiles by adding such factors into the agent's state. However, learning this variation will lead to the need for longer training times and larger data sets resulting in stunted practical capabilities.

In the previous chapter we were able to attain meaningful cluster groupings of the energy demand profiles, and extract key identifiers between them. Using the found clusters three separate environments are created, with each environment having reduced variation in how it transitions between states. This reduced variation is due to each environment only comprising of energy demand profiles which behave in similar patterns. Separate agents are trained on each of the less complex environments, creating agents better suited to reducing the energy demand peak on a specific cluster's energy demand profile pattern. The goal of each agent will be to find an optimal approach for the given cluster's average energy demand profile and alter it slightly to take into consideration the risk associated with the small differences between the energy demand profiles within the cluster.

In order to further reduce the complexity of the problem the state space was considered. If one were to directly give the agents the average energy demand trend the agent's state would have to include 48 continuous variables which would result in a extremely large and sparse state space. Instead of adding in these 48 variables, a reward function (explained in Section 5.3) which helps the agent learn the average energy demand trends through experiencing the environment is used. This ability to learn the energy demand trends through experiencing the environment negates the need for a complex agent state.

In order to solve this reduced problem Q-learning was used as it learns the optimal policy directly (the pseudo-code for Q-learning can be seen in Algorithm 1). The reward signal (which is detailed in Section 5.3) gives the agent instantaneous feedback for actions taken, allowing the reward to propagate back through the action-value function faster. This faster feedback helps reduce the bias that using TD(0) (traditionally used with Q-learning) brings. The code was adapted from a stock trading tutorial [The-Lazy-Programmer, 2018].

Algorithm 1 Q-learning Algorithm (Adapted from [The-Lazy-Programmer, 2018])

Initialize Q(s,a) randomly where s ∈ S and a ∈ A
for each episode **do**
Initialise S;
  for each step **do**
    Choose A according to current policy derived from Q (e.g ϵ-greedy);
    Take action A, and observe R, S';
    $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \lambda max_a Q(S', a) - Q(S, A)]$;
    $S \leftarrow S'$
  end for
end for

The exploration method (or policy) used when training the agents was the ϵ-greedy approach, with a decaying ϵ to ensure convergence of the q-value function. The pseudo-code for the decaying ϵ-greedy approach is outlined below.

Algorithm 2 Decaying epsilon greedy (Adapted from [The-Lazy-Programmer, 2018])

Random ← random number generated between 0 and 1
if Random > ϵ **then**
  Take greedy action
else
  Take random action
end if
Decay ϵ

Despite the reduction in the problem's complexity the resulting Q-value table would still be very large. In order to capture the value table in a more compressed way and to allow for faster learning through extrapolating between states, function approximation is used. The function approximator utilised was an artificial neural network, the specifics of which are outlined in Section 6.4.

5.1.1 State space

Given the reward function (detailed in Section 5.4) the average energy demand requirements of a cluster do not need to explicitly be given to the agent. This results in a compressed agent state consisting of the following variables:

- *Time*
- *Capacity left in the battery*
- *Previous maximum peak power reached*
- *Average day temperature predicted*

The *time* variable is a discrete value between 1 and 48 based on a day's half-hour time values. This variable gives the agent information about where during the day the agent is currently positioned. The time of day is crucial in the agent's state as energy demand patterns are dependent on time.

The *capacity left in the battery* (measured in kWh) is a limiting factor in how much the agent can reduce the energy demand peak going forward. And is therefore important for the agent to know in order to reduce the peaks.

The *maximum kW* value that has already been reached during the day is an important variable for the agent, this is due to the fact that once a maximum demand has been reached the agent cannot do better despite its efforts going forward. Thus, its actions going forward can be altered, for example, by being more cautious because a greater saving would no longer be possible.

The *average temperature* was added to the agent's state. As seen in Chapter 5 it accounts for some of the variation of the energy demand profiles within the clusters as it creates a small time shift. Since we use the daily average, it remains static during episodes. In this chapter the agent is trained and tested with and without average temperature in order to compare how effective this variable is in the agent's state.

This agent state detailed above does not directly equate to the environment state as we do not have enough information to account for all the dynamics of reality, and so in the context of this problem the MDP is only partially observable. Despite the environment state not being fully observable the chosen state variables give a good overview of the agent's position within the environment.

Since a neural network is used as the function approximator all the state variables were normalised so that issues with unbounded parameters would not be encountered.

5.2 Actions

Since our objective is to create an optimal control signal for the battery, our actions relate to outputs of the battery that are possible over a half hour interval (our time frame). In order to create appropriate actions we must take into consideration how a battery system operates.

5.2.1 Battery

The sizing of the battery system stipulates how much power and energy the battery system can output. The battery system is comprised of two parts: the inverter and the battery storage. The inverter limits the amount of power the battery can output at any given time and is measured in kW. The battery storage is the amount of energy potential the battery holds and is measured in kWh. For example, if the inverter size was 10 kW, the inverter would not be able to output more than 10 kW of power at any given time, thereby limiting the maximum demand reduction to 10 kW. The battery storage would be the limiting factor in the time period the inverter could output a given amount of power for: for example if the battery storage had the rated energy capacity of 20 kWh, it would be able to hold a constant power output of 10 kW for 2 hours or 5 kW for 4 hours etc. These output indications outlined above assume that the battery has a depth of discharge (D.o.D) capability of 100 %. In reality the D.o.D is battery specific and varies depending on the technology thereby reducing the rated energy capacity and not being able to utilise the full rated storage potential.

The battery modelled in this scenario has the following sizing:

- Inverter - 200 kW
- Battery Storage - 800 kWh

The commercial shopping centre operating without the installed solar storage system has a single long energy demand peak. This long flatter demand peak means the battery would need to output for a longer duration and so a larger storage size was chosen where the battery could output at maximum output (the inverter size) for 4 hours. The battery and inverter size can be adjusted based on financial optimisation in future simulations.

For this application it was assumed that the battery would be recharged in the evening when there was low energy demand, charging during the day when the demand is higher can be risky as it could lead to increasing the demand charge which is detrimental to our objective.

Since this thesis focuses on optimal control rather than the financial modelling of the system, it is assumed that the battery system has no efficiency losses. In practice this is not the case and the energy lost due to inefficiencies would need to be taken into account in the financial modelling of the system.

5.2.2 Agent Actions

The actions the agent would be able to take would be the power outputs of the inverter which are limited to the inverter size. Q-learning maximises over all possible actions to calculate its targets and using a continuous action space can lead to an unwieldy problem. For the purpose of reduced complexity a set of discrete power outputs are used as the potential actions of the agent. Increasing the choice of actions in the action set will lead to an agent having more freedom in how it can output power but also a more complex problem to solve. For this exercise agents were trained using the simplified action set:

- $[0, 50, 100, 150, 200]\ kW$

Although the agents will have the choice among all actions in the action set, if the battery capacity is depleted, the agent will be forced to take the action of outputting 0 kW.

5.3 Reward

The reward is broken up into two parts outlined in this section. The first part focusing on an instantaneous reward which helps the agent learn the cluster average energy demand trends and negates the need for a forecast demand profile to be incorporated into the state. The second part gives a reward to the agent at the end of an episode which will help the agent adapt its approach to take into consideration the deviations between energy demand profiles in a cluster.

5.3.1 Reward Part 1

Figure 21 shows the maximum demand reduction that can be attained on a day if the day ahead demand requirements are known. The blue line shows the demand profile before peak shaving is done. The orange line shows the optimum battery output given the capacity within the battery and the green line shows the demand profile after the optimum battery output has been applied.

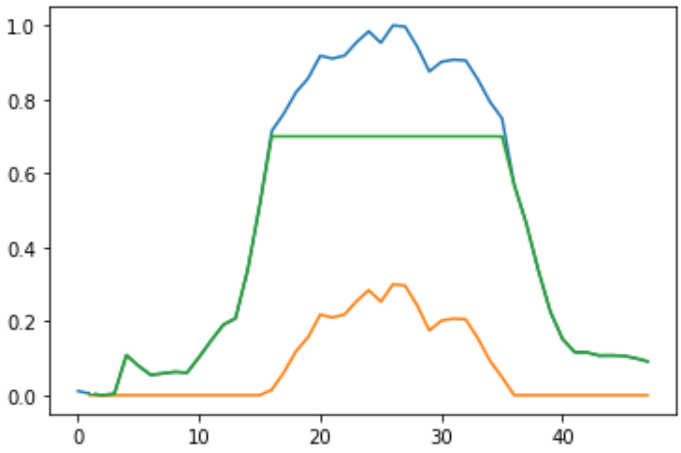


Figure 21: A plot showing the maximum demand reduction that can be achieved with a limited capacity within a battery: The blue line indicates the energy demand before the intervention of a battery, the orange line indicates the optimal battery output and the green line indicates the energy demand after the battery intervened.

This optimum battery output given a day ahead demand requirements can be calculated using the pseudo-code in Algorithm 3. The algorithm starts by setting the target demand reduction to the maximum demand reached in the day ahead demand requirements. It then proceeds to incrementally reduce the target demand reduction and calculate the day's battery output required to achieve this reduction. At each increment the battery's output (orange line) is integrated to find the area under it is curve (which is equivalent to the capacity used, measured in kWh). If the battery's capacity has been overshot (making the proposed battery output impossible) the demand reduction that can be achieved is reduced to the previous increment.

Algorithm 3 Function to calculate potential maximum demand reduction

```
Require: Function takes in current time step
  level = max of average cluster profile
  curve = episode[current time step:48]
  while True do
      output = curve - repeat(level, 48 - current time step)
      output[output<0] = 0
      kWh used = sum (output/2)                ▷ Integrate the area of output
      if kWh used < battery capacity then
          level = level - 10 kW
      else
          Break
      end if
  end while
  if max(curve - output) > max already reached then
      Return 1 - max(curve - output)
  else
      Return 1 - max already reached
  end if
```

This algorithm for working out optimal battery outputs is extremely useful but requires the day ahead demand requirements to be known ahead of time. At the start of each day the energy demand requirements are unknown and the best projection we have of them would be the average of the demand profiles from the cluster the day has been assigned to. Thus, this average cluster profile would be the best approximation of the demand requirements and can be used to calculate the approximate demand reduction potential that can be derived from the battery for a day ahead.

Furthering this, Algorithm 3 allows a time step to be given as an input. This means that potential demand reduction can be calculated from any point during the day. Using the assigned cluster's average demand profile, at each step the potential demand reduction can be calculated. If at any step the potential reduction has been reduced, a negative reward can be given to the agent indicating the action taken in the step was not optimal. The negative reward is proportional to the amount that the potential energy demand reduction was reduced (Equation 20). This reward system will allow the agent to propagate negative rewards for non-optimal actions quickly and will help the agent learn the optimum output for the cluster's average energy demand profile.

$$reward_1 = \phi(Potential_{before\ action\ is\ taken} - Potential_{after\ action\ is\ taken}) \quad (20)$$

Since the demand potential is worked out with an unconstrained battery output rather than a discretised one, this reward function will create a more complex

reward gradient for the agent to learn. Receiving small negative rewards for optimal actions, thereby making the process of minimising these negative rewards more difficult. Although this method creates a more complex reward gradient the computational benefit of not having to compute a linear constrained optimisation at each time step in the training process (to calculate the optimal output constrained by the discretised battery outputs) outweighs this con. If training on a machine with more computational power, computing the optimal binary output of the battery at each time step would improve how the agents could learn using this reward.

5.3.2 Reward Part 2

The second part of the reward comes at the end of an episode when the maximum demand reduction is calculated. A positive reward is given proportional to the total energy demand reduction. This reward is calculated on the actual profile (which is known at the end of the day), rather than the average profile allowing for the risk associated with variation in the days to be accounted for in the reward. Even if the agent varied from the best output strategy for the average profile but still was able to generate savings, this reward would counterbalance the negative rewards received from reward 1.

This reward takes longer to propagate back through the q-value function as the cumulative reward is discounted using $\gamma = 0.99$. The reward is discounted using Equation 21, were the the greater the time step n the greater it is discounted.

$$r = r_1 + r_2 \times \gamma^1 ... + r_n \times \gamma^{n-1} \tag{21}$$

5.3.3 Total Reward

The combined reward function is

$$Reward = -Reward_1 \times \alpha + Reward_2 \times \beta \tag{22}$$

Where α and β can be changed to alter the importance that is put onto each part of the reward. Increasing α puts more emphasis on the agent learning an approach for the average demand profile whilst increasing β puts more emphasis on taking into account the variation in the individual profiles. For the trained agents the parameters chosen were fractions to keep the reward size manageable and were chosen to be equal to put equal importance on each reward. The chosen parameters were:

- $\alpha = \frac{1}{1000}$
- $\beta = \frac{1}{1000}$

5.4 Environment

A class in *python* was created for the environment which held all the needed information about the environment and contained functions which followed a similar format used in AI gym [Brockman et al., 2016]. Since the real environment cannot be directly accessed, sampled episodes from the historic profile data is given to the environment class. The environment included the following functions:

- *reset*: initialises the environment at the start of each day
- *step*: performs a state transition
- *dispatch*: takes an action
- *curve max dispatch*: calculates the potential demand reduction that can be achieved given a profiles and battery capacity
- *get obs*: returns current state

The *reset* function initialises the environment by setting the battery capacity to its maximum capacity, time to zero, day ahead battery outputs to zero and the potential maximum reward to the maximum potential of the day ahead average cluster profile. It then returns the initial state using the *get obs* function (a function used to return the current state of the agent).

The *step* function (Algorithm 4) takes in an action, proceeds with taking the action using the *dispatch* function (Algorithm 5). This *dispatch* function updates the capacity of the battery and the battery output based on the given action. To be noted the code was written using an object orientated approach, and so when an algorithm shows a variable being updated, it is updated in the object. The *step* function then calculates the reward with the help of the *curve max dispatch* function (shown in more detail in the reward Section 6.2) and returns the next state, reward received and a done indicator (True if the episode is over).

Algorithm 4 Step Function (Adapted from [The-Lazy-Programmer, 2018])

```
assert action is in the action space
dispatch(action)
current demand = demand[time step] - battery output[time step]
if current demand ≥ max value then:
    max value = current demand
end if
done = time step == 47
reward = 0
if done then:
    max before = max(demand)
    max after = max value
    reward = reward + (max before - max after)/1000
end if
current potential = curve max dispatch()
reward = reward - (previous potential - current potential )/1000
previous potential = current potential
time step += 1
return get obs(), reward, done, info
```

Algorithm 5 Dispatch Function

```
if There is enough capacity to take chosen action then
    battery output = action value
    capacity = capacity - action value/2
else
    battery output = capacity*2
    capacity = 0
end if
battery outputs[current time step] = battery output
```

5.5 Function Approximator

The function approximator used was an artificial neural network, it is non-linear nature makes it suitable for capturing the non-linear time aspect of the changing energy demand. In order to use and train the neural network to capture the function of the action value table there were a few considerations, namely:

- If sequential inputs were used to train the network they would be highly correlated which would stunt the training of the network. To resolve this issue, experience replay was used where an experience buffer of inputs is populated and then randomly sampled. The algorithm used to implement experience replay can be found in Algorithm 5.

- Since the targets ($R+\lambda max_a Q(s,a)$) used contain predictions, true gradient descent cannot be performed, but rather an approximation of gradient descent. This can lead to instability in a neural network, therefore a copy period was added where a network uses the same network parameters over the copy period's number of iterations before it is updated with new parameters.
- Since we are creating a network to capture the value of all actions that can be taken from a given state, the network would have multiple outputs (the number of actions).

The structure of the neural networks is comprised of an input layer of 3 or 4 nodes (depending on if average temperature is added to the state space) which represent the state variables, 2 hidden layers with 10 neurons and an output layer of 5 nodes where each node represents an action the agent can take. The networks weights are randomly initialised before training begins.

In order to train the initialised network, 5 000 episodes are sampled through to train the model using Algorithm 6. For each episode Algorithm 4 is called where for each time step in an episode an action is taken based on the ϵ-greedy algorithm. The experience buffer is then updated using the *add experience* function (Algorithm 7) and the *train model* function (Algorithm 8) is called.

Algorithm 6 Function to play episode (Adapted from [The-Lazy-Programmer, 2018])

while not Done **do**
  Sample action using ϵ-greedy algorithm given current observation
  previous observation $\leftarrow$ observation
  observation,reward,done $\leftarrow$ return from step function given chosen action
  Add experience (previous observation, action, reward, observation, done)
  Call train model function
end while

The *add experience* function propagates the experience buffer until the maximum experiences has been reached (10000), once the maximum threshold has been reached experiences from the beginning of the buffer are removed before new ones are appended.

Algorithm 7 Function to add experience (Adapted from [The-Lazy-Programmer, 2018])

if length(experience buffer) >= maximum experiences **then**
  Pop first element from the experience buffer
else
  Append element to the experience buffer
end if

The *train model* function does nothing if the length of the experience buffer is less than min experiences (100), otherwise a batch of size 32 is sampled from it randomly and used to perform an iteration of training on the network.

Algorithm 8 Function to train model

if length(experiences) $<$ min experiences **then**
Do not do anything
else
Randomly sample (batch size amount) of observations from experience buffer
Use trained network to predict next q values
Calculate the targets using the observations and next q values
Train the network using the states as the inputs and targets as the outputs
end if

Each batch is used to iteratively train the model using the Adam optimiser. The Adam optimiser is suited towards handling sparse gradients in environments which are quite noisy [Kingma and Ba, 2017]. The targets of the network outputs were the Q-values stipulated in Equation 20.

$$Target = R + \lambda max_a Q(s, a). \tag{23}$$

The loss function used was the squared error between the current network outputs and the targets.

$$\sum_{1}^{No\ actions} (output - target)^2 \tag{24}$$

5.6 Training platform

In order to train the agents 2019 data is used, a set of three agents are trained for two different agent states (one including average temperature and the other not) on each of the three found clusters in Chapter 4. Training agents with and without temperature gives us the ability to assess if the addition of average temperature into the state space helps the agents to learn. Reinforcement learning typically takes a long time to train, with the reduction of the state space and use of a reward signal which allows for instantaneous feedback the agents will learn a lot faster. For computational reasons, the training process was allowed to run for 5 000 iterations for each agent, but for improved results the agent could be trained over more iterations. The value of ϵ used when training the agent was calculated based on the iteration number n using Equation 22, resulting in a decaying ϵ as the iterations increased.

$$\epsilon = 1/\sqrt{n+1} \tag{25}$$

5.7 Testing platform

The main objective of peak shaving is to reduce the maximum energy demand reached during a month over half hour intervals in order to optimise electricity costs. With batteries being expensive we need to ensure that we are deriving the most potential from them. The best way to understand how well the agents are operating is to test it over a year's data. In order to decide which agent should be used on an unseen day the key identifiers found in Chapter 5 are used to find the most suitable cluster and the agent is chosen accordingly. The trained agents are tested on both 2018 and 2019 data. Testing on both seen and unseen data gives us an indication on how well the agents generalise to new data.

In order to quantify how well the agents perform in terms of energy demand reduction, the maximum energy demand of each day without intervention of the agent is found and used as the first bench mark. Any results with a maximum energy demand below this bench mark will show that the agents are generating demand reductions. Furthermore, the optimal demand reduction (the reduction that can be achieved if the day ahead energy demand requirements are known ahead of time) is calculated in order to see the maximum reduction that could be achieved given the battery size and used as another benchmark. Since the agents are limited by a discretised output, a third benchmark is used to see how the agents compare against an optimal output given this binary output constraint. To calculate the optimal output given the binary output constraint, the problem is set up as a constrained optimisation problem and linear programming is used to calculate the optimal battery output. The linear programming problem is set up as follows:

A binary matrix namely Y size 48×5 is created to choose between the 5 actions at each half hour time period. Each of the 48 row represents a binary vector with a 1 in the column of the action to be chosen and 0's in the rest.

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

A matrix A of size 48×5 is created where each of the 48 rows reflects the 5 possible discrete battery output values that could be taken at any half hour time period.

$$A = \begin{bmatrix} 0 & 50 & 100 & 150 & 200 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 50 & 100 & 150 & 200 \end{bmatrix}$$

A matrix *Battery Output* of size $48x1$ can be calculated by matrix multiplying Y by A^T. Each row of this matrix will reflect the chosen value that the battery output at that time period.

$$Battery\ Output = Y \cdot A^T$$

The capacity used of the battery during the day can be calculated by summing the *Battery Output* matrix and dividing it by 2. The reason it needs to be divided by 2 is that the battery outputs are only for 30 minute intervals and as explained in Subsection 5.2.1 if a battery were to output 10 kW for 30 minutes it would result in the battery using 5 kWh.

$$Capacity\ Used = \frac{\sum Battery\ Output}{2}$$

The maximum reduction of the demand can be calculated by finding the difference between the demand and the battery output and taking the maximum.

$$Max\ Reduction = Max[Demand - Battery\ Output]$$

In order to reduce the maximum demand we aim to reduce $Max\ Reduction$.

Minimize: $Max\ Reduction$

In order to solve the problem we have to set constraints

Subject to:
The capacity used needs to be less than or equal to the capacity in the battery.

$$Capacity\ Used \leq 800$$

Each step has to be less than the current max reduction as to not take actions which increase the current maximum.

$$[Demand - Battery\ Output]_t \leq Max\ Reduction$$

The values in the Y matrix have to all be binary so that they can select between the actions. In order to ensure that only 1 action is selected at each time step the sum of each row of Y needs to equal to 1.

$$\sum Y_t = 1$$

The comparison of the agent results to the first benchmark (no intervention) will allow us to see the demand reduction being achieved by the agents. The optimal binary constrained reduction will allow us to see how well the agents use the full potential of the storage system, given the binary output constraint. The unconstrained optimal reduction will contrast the binary constrained reduction with how much could be reduced if the full potential of the battery could be derived by not having any limitations placed on its output, allowing for a more tailored output to be created. Taking into consideration that without precise day ahead forecasts the optimal usage cannot be achieved, a measure of retention

is used to quantify how optimally the battery system is operating, the equation to calculate retention is as follows:

$$retention = \frac{savings_{agent}}{savings_{optimal}} \tag{26}$$

This shows the percentage of the savings the agents were able to attain compared to the optimal savings. These three benchmarks will give a good indication of how well the system operates.

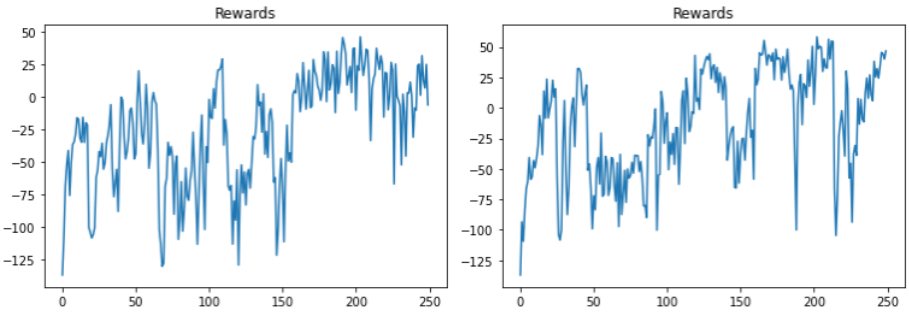
5.8 Results

In order to ascertain if temperature would be a valuable addition to the state space, as discussed above reinforcement learning agents were created with and without temperature in the state space. The results refer to both sets of agents but for ease of reading some of the figures can be found in the Appendix.

In the first three subsections daily results for each cluster are presented, these daily results are then pulled together to determine how the array of agents perform as a combined system over a year.

5.8.1 Cluster 1

The first agents (one with temperature in its state space and one without) were trained using only energy demand profiles found in cluster 1 (comprised predominately of Saturdays). Figure 22 shows the undiscounted accumulated rewards (total rewards received over a day / episode) averaged over 50 episode iterations for both the agents. Both agents can be seen to be steadily increasing with periods of dips indicating that the agents were exploring in the attempt to find an optimal solution. At the end of the 5000 training episodes the results can be seen to still be improving, indicating that with lengthened training time the agent's optimally could be increased.

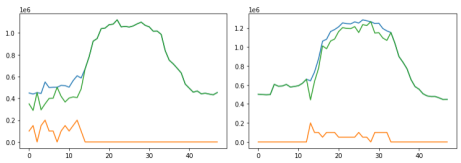


(a) (b)

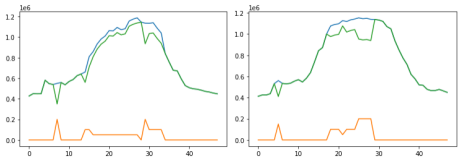
Figure 22: Cluster 1 reinforcement learning agent's undiscounted accumulated rewards averaged over 50 episode iterations: Subfigure (a) represents the agent trained without temperature and Subfigure (b) represents the agent trained with temperature.

Figure 23 visually demonstrates the learning progression of the agent trained with temperature in its state space. By episode 20 the agent outputs randomly and runs out of battery capacity quite quickly resulting in no demand reduction for the day. By episode 100 we see the agent moving its output toward the middle of the day as it learns it is more advantageous, but still resulting in minimal savings. By the final episode 500 the agent has an improved steady

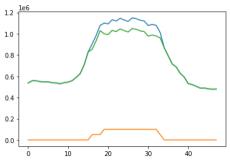
output during the middle of the day. Similar learning progress can be seen for the agent excluding temperature from its state, this progress can be visualised in Figure 44 in the Appendix



(a) (b)



(c) (d)

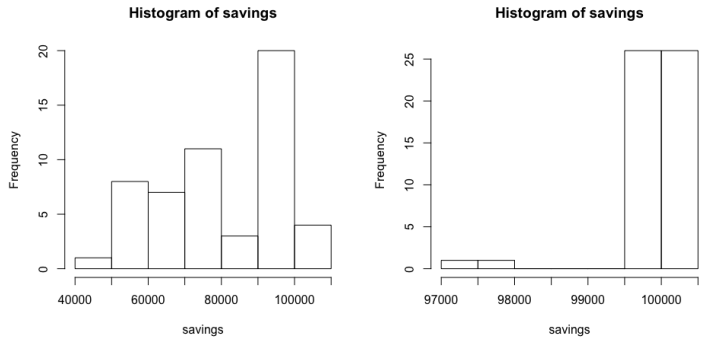


(e)

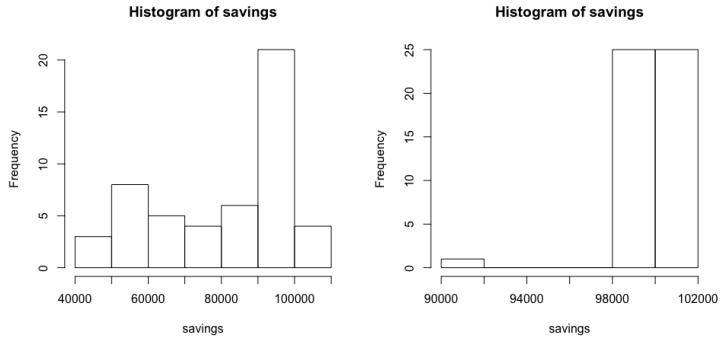
Figure 23: Learning progression of the agent trained with temperature in its state space: Subfigure (a) shows episode 20, Subfigure (b) shows episode 100, Subfigure (c) shows episode 500, sub-figure d shows episode 1000 and sub-figure e shows episode 5000

Figure 24 shows the demand reduction performance of the agents on the daily 2018 and 2019 profiles. Both agents performed well, generating a demand reduction on all the days within cluster 1 for both 2018 and 2019 data. The good performance on both year's data is a really good indication that the agents are able to generalise well to unseen days and that our key identifiers used to group new profiles into cluster 1 are appropriate. The battery output strategy used by the agents was quite static across all the tested days, this is a good sign as it means the agents are finding an overall best strategy for cluster 1. Since this overall strategy was able to generate demand reduction on all profiles the agent's approach was able to account for inherent risk caused by variation between energy demand profiles within cluster 1. Although both agents performed well, the agent with temperature in its state space produced better results more consistently. This tells us that giving the agent temperature as a state variable helps explain some of the inter cluster variation. This is illustrated in the histograms shown in Figure 26 with the more consistent agent attaining 100 kW

reduction on all but two of the days in 2018 and all but one of the days in 2019. The days where the agent didn't derive the 100 kW it was still able to generate a reduction close to 90 kW. The agent with temperature in its state space outperformed the agent without with the respective average demand reductions on 2018 being 100 kW and 80 kW. Since the agents with temperature in their state space performed more optimally, the rest of this subsection will refer to this agent unless specified otherwise.



(a) (b)

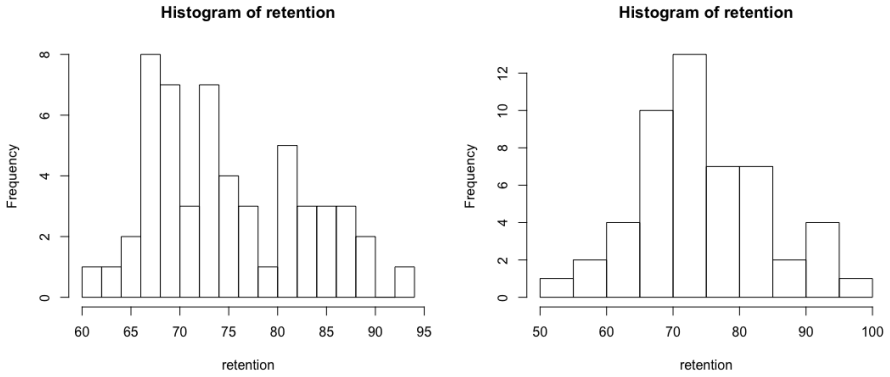


(c) (d)

Figure 24: Histograms of the daily savings derived from the agents on cluster 1: Subfigure (a) shows the distribution of savings attained on the 2018 data using the agent trained without temperature, Subfigure (b) shows the distribution of savings attained on the 2018 data using the agent trained with temperature, Subfigure (c) shows distribution of savings attained on the 2019 data using the agent trained without temperature and Subfigure (d) shows the distribution of savings attained on the 2019 data using the agent trained with temperature.

The fact that the agents were able to derive reductions on all the days is a promising sign about the systems practical application. But as stated before

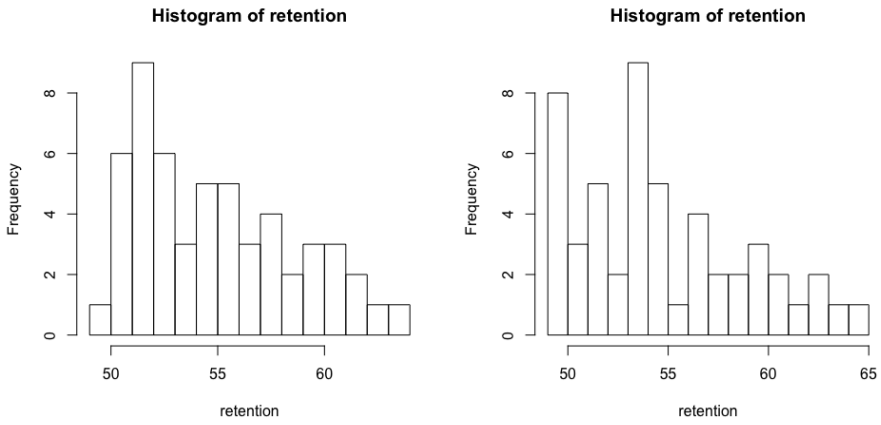
our main objective in this thesis is to maximise the savings we can derive from the battery's potential and so Figure 25 shows the distribution of the retention (detailed in Section 5.7) of the battery's potential acquired given the constraint of the binary output. On average the agent was able to attain 75.11% on 2018 data and 74.05% on 2019 data. Keeping in mind that without perfectly accurate forecasts of the energy demand it would be impossible to attain completely optimal results, this agent is able to generate a high percentage of the the optimal result. This means this approach takes advantage of a large portion of the battery's potential. The agent was able to generate a demand reduction of 95.43% on one of the days in 2018. On the lower end of the scale, worst case the agent generated 50.01% of the optimal savings.



(a) (b)

Figure 25: Daily retention values including the binary output constraint: Subfigure (a) shows the results of the 2018 data and Subfigure (b) shows the results of the 2019 data.

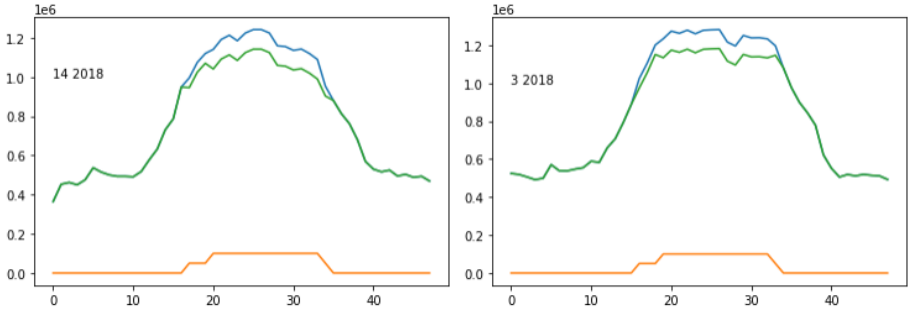
Our agents had the limitation of a binary output and so the previous benchmark would be a fair comparison for them given this constraint. Figure 26 shows the retention values calculated using an optimal solution with no constraints. The agent attained 50% or above on this benchmark on all days within this cluster but on average did not do as well with 61.85% on the 2018 data and 54.74% on the 2019 data. In spite of the limitation the agents were still able to do fairly well, and if extended to use a continuous output have the potential to derive even more benefit from the battery.



(a) (b)

Figure 26: Daily retention values with no output constraints: Subfigure (a) shows the results of the 2018 data and Subfigure (b) shows the results of the 2019 data.

It may seem counter-intuitive that the agent is deriving the same reduction on most days but having varied retention values. The reasoning behind this can be visualised in Figure 27 by comparing the days which had the best and worst retention. Whilst both days ended up with a 100 kW reduction, they had different retention values. The profile with the lower unconstrained retention of 52.63% is not as wide as the profile with greater unconstrained retention of 71.42% meaning it had more potential to save with the given capacity and therefore has a lower retention percentage. The best and worst days of the other agents trained on this cluster can be found if Figures 49, 50 and 51 in the Appendix.

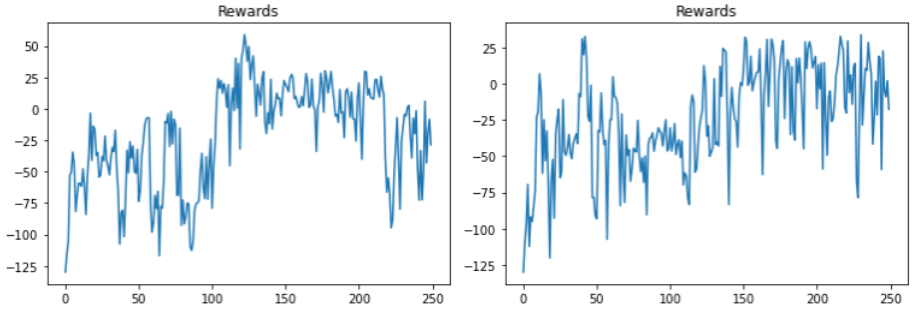


(a) (b)

Figure 27: Demand reduction overview of the best and worst retention days on the 2018 data: Subfigure (a) shows the worst unconstrained retention of 52.63% and Subfigure (b) shows the best unconstrained retention of 71.42%

5.8.2 Cluster 2

Cluster 2 is comprised mostly of Weekdays distinguished by their later trading hours. As seen with cluster 1 both agents have periods of exploration and were still on the up trend at the end of their training. The visual progression of learning for both the agents is similar to the agents for cluster 1 and can be found in Figures 45 and 46 in the Appendix.



(a) (b)

Figure 28: Cluster 2 reinforcement learning agent's undiscounted accumulated rewards averaged over 50 episode iterations: Subfigure (a) represents the agent trained without temperature and Subfigure (b) represents the agent trained with temperature.

Similar to the battery outputs seen in the previous subsection (Subsection 5.8.1), the agents utilise a static approach to the energy demand profiles indicating an overall solution adapted to the inherent variation has been found. Figure 29

shows the daily demand reduction achieved by the agents on days within the cluster. The agent with temperature in its state space outperformed the agent without with the respective average demand reductions on 2018 being 87 kW and 76 kW. The agent trained with temperature in its state space was able to achieve a demand reduction on all of the days in both 2018 and 2019 with a majority of the demand reduction sitting at 100 kW. The lowest demand reduction seen in 2018 was 47 kW and the highest 100 kW. The average retention achieved on 2018 given the binary constraint was 67.76% and without the constraint 55.39%. Figures 58 and 59 in the Appendix elaborate on the retention results achieved.

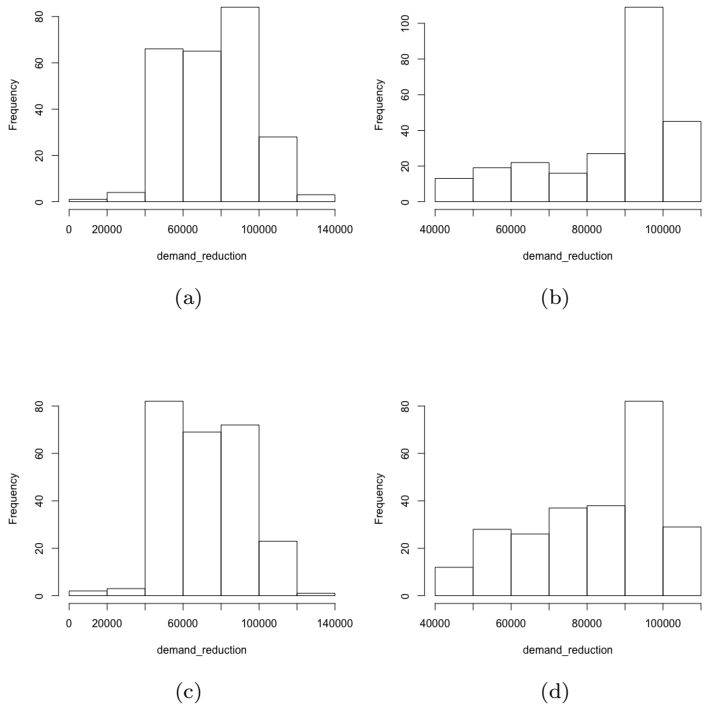
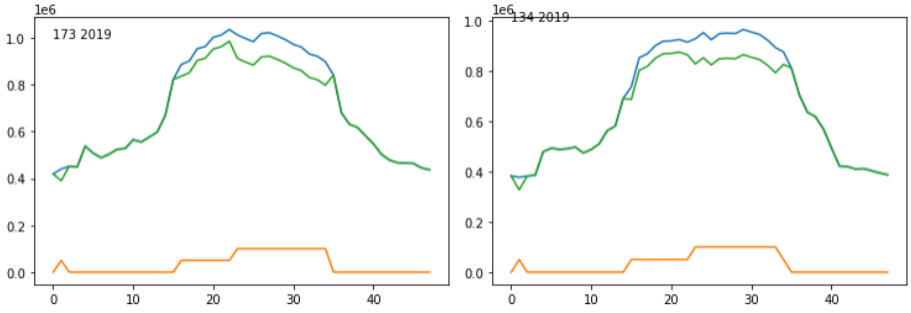


Figure 29: Histograms of the daily demand reductions derived from the agents on cluster 2: Subfigure (a) shows the distribution of the reductions attained on the 2018 data using the agent trained without temperature, Subfigure (b) shows the distribution of the reductions attained on the 2018 data using the agent trained with temperature, Subfigure (c) shows distribution of the reductions attained on the 2019 data using the agent trained without temperature and Subfigure (d) shows the distribution of the reductions attained on the 2019 data using the agent trained with temperature.

Figure 30 shows the worst and best days based on the unconstrained retention.

The agent learnt that in general the second half of the day was slightly higher and increased its output in the afternoon. This increased output worked well for the profile on the right as it followed this trend but did not perform as well on the profile on the left as there was a small spike during the first half of the day. The best and worst days of the other agents trained on this cluster can be found if Figures 52, 53 and 54 in the Appendix.

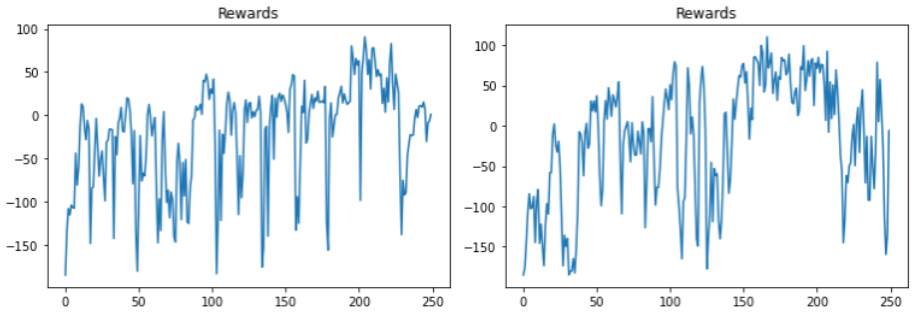


(a) (b)

Figure 30: Demand reduction overview of the best and worst retention days on the 2019 data: Subfigure (a) shows the worst unconstrained retention of 50.00% and Subfigure (b) shows the best unconstrained retention of 64.94%

5.8.3 Cluster 3

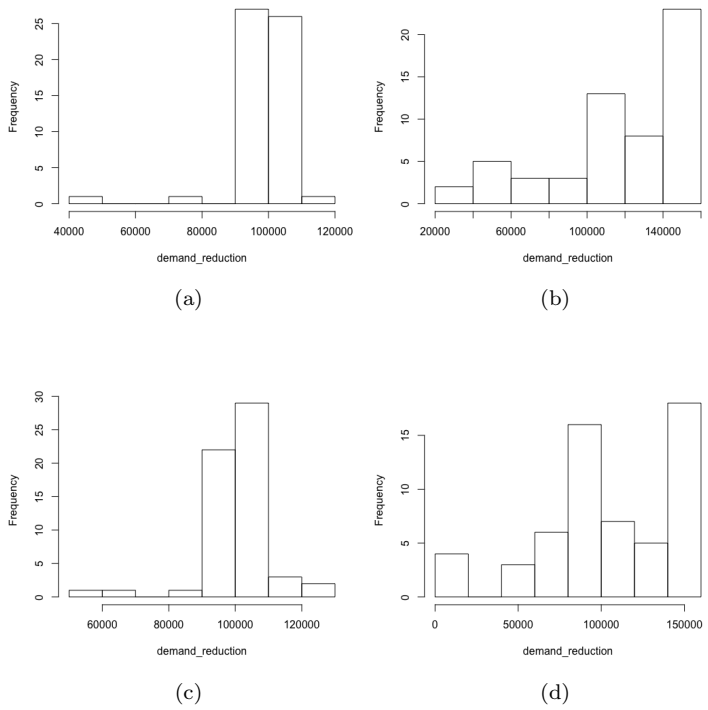
The third agents were trained on cluster 3 profiles which comprise of Sundays and public holidays. The average reward from the agents over each 50 episode iteration can be seen in Figure 31. At 5000 episodes the results are still improving for both agents and with added training time could lead to a near optimal agents. The visual progress of the agents can be seen in Figures 47 and 48 in the Appendix.



(a) (b)

Figure 31: Cluster 3 reinforcement learning agent's undiscounted accumulated rewards averaged over 50 episode iterations: Subfigure (a) represents the agent trained without temperature and Subfigure (b) represents the agent trained with temperature.

Figure 32 shows the daily demand reduction achieved by the agents on days within cluster 3. The agent with temperature in its state space outperformed the agent without with the average demand reductions on 2018 being 118 kW and 100 kW respectively. The agent trained with temperature in its state space was able to achieve a demand reduction on all of the days in 2018 and all but one day in 2019 with a majority of the demand reduction sitting at 100 kW. The lowest demand reduction in 2018 was 22 kW and the highest was 150 kW. The average retention achieved with the binary constraint was 63.52% and without the constraint was 59.34%. Figures 60 and 61 in the Appendix elaborate on the retention results achieved.

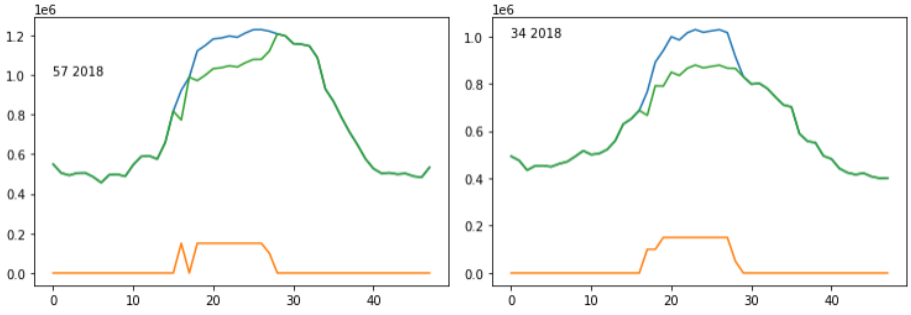


(a) (b)

(c) (d)

Figure 32: Histograms of the daily demand reductions derived from the agents in cluster 3: Subfigure (a) shows the distribution of the reductions attained on the 2018 data using the agent trained without temperature, Subfigure (b) shows the distribution of the reductions attained on the 2018 data using the agent trained with temperature, Subfigure (c) shows distribution of the reductions attained on the 2019 data using the agent trained without temperature and Subfigure (d) shows the distribution of the reductions attained on the 2019 data using the agent trained with temperature.

Figure 33 shows the energy demand profiles which resulted in the best and worst daily binary constrained retention. Sundays and public holidays generally have early drop off due to the trading hours. The day that was missed in 2019 did not follow this trend as the demand dropped off much later. The date of this was the 25^{th} of December and as it was Christmas day the trading hours were probably altered. This needs to be accounted for when clustering as this profile was incorrectly clustered based on its shape. The best and worst days of the other agents trained on this cluster can be found in Figures 55, 56 and 57 in the Appendix.



(a) (b)

Figure 33: Demand reduction overview of the best and worst retention days on the 2018 data: Subfigure (a) shows the worst unconstrained retention of 14.36% and Subfigure (b) shows the best unconstrained retention of 78.94%

5.8.4 Monthly results

All the results tested on the separated clusters were brought together in order to get a full picture into how the agent would derive actual savings over the year on a month by month basis. The results from the agent trained with temperature on the 2019 training data are tabulated in Table 7 and the results derived on the 2018 test data are tabulated in Table 8. The monthly average binary constrained retention values were extremely similar averaging at 64.77% over the 2019 months and 63.04% over the 2018 months. Since the system of agents behaves well on the unfamiliar data this is great indication that the system would generalise in a practical setting. In 2018 the month with the greatest binary constrained retention was January with 72.89% and the lowest was May with 53.51%. In 2019 the month with the greatest binary constrained retention was June with 74.43% and the lowest was August with 50.50%.

The average monthly demand reduction on the unfamiliar 2018 from the agent trained without temperature was 78.3 kW these are significantly lower than the above mentioned trained agents who achieved 91 kW. From these results it is evident that temperature is a useful variable to have in your state. The monthly results for this agent are tabulated in Table 8 and 9 in the Appendix.

Since the system of agents were able to derive results on all but 1 day in 2018 and 2019 it appears that our forward looking clustering approach is working well on unseen data.

Month	Optimal Binary Savings (kW)	Optimal Continuous Savings (kW)	Agent Attained Savings (kW)	Binary Battery Retention (%)	Continuous Battery Retention (%)
1	134	178	97	72.89	54.23
2	120	168	86	71.24	50.99
3	149	199	100	66.91	50.31
4	139	191	100	71.73	52.46
5	200	200	142	71.20	71.20
6	186	200	127	53.51	63.55
7	119	164	71	59.51	43.32
8	132	182	81	61.49	44.56
9	138	186	96	69.04	51.35
10	147	191	100	68.15	52.36
11	153	200	100	65.38	50.00
12	171	200	86	50.38	43.00
Avg	149	188	99	64.77	52.49

Table 7: Table of results derived from the system of agents trained with temperature in their state space on the 2019 data.

Month	Optimal Binary Savings (kW)	Optimal Continuous Savings (kW)	Agent Attained Savings (kW)	Binary Battery Retention (%)	Continuous Battery Retention (%)
1	120	150	64	53.90	43.00
2	130	157	67	51.22	42.57
3	134	160	99	74.16	62.11
4	157	180	100	63.68	55.56
5	140	170	100	71.57	58.82
6	134	163	100	74.43	61.22
7	118	147	87	73.69	59.21
8	123	148	62	50.58	42.08
9	200	200	150	59.06	59.05
10	135	159	59	43.68	37.15
11	150	180	100	66.67	55.55
12	135	160	100	73.90	62.50
Avg	149	164	91	63.04	53.23

Table 8: Table of results derived from the system of agents trained with temperature in their state space on the 2018 data.

6 Cluster Analysis: Grid Demand after Solar

The addition of a solar system alters the energy demand the shopping centre has on the grid by lowering it when solar energy is produced.

$$E_t^{grid\ demand} = E_t^{total\ demand} - E_t^{solar\ generation} - E_t^{battery\ output} \tag{27}$$

The solar profiles themselves can be quite volatile, and therefore can be tricky to forecast forward due to meteorological factors such as cloud cover and wind. Companies specialise in solar power forecasting by making use of satellite data to track weather movements. For the purpose of this thesis we assume the day ahead solar generation as a known, however when implementing the system the solar forecasts from a forecasting company can be substituted in instead.

6.1 Self-Organising Maps

Considering that we are interested in reducing energy demand spikes on the grid, we are focused on how the solar generation impacts the grid demand and changes where the energy demand spikes occur. We therefore look at $E_{grid\ demand}$ the grid impact after the solar energy has been consumed. A self-organising map (SOM) is applied to the 2019 grid demand profiles and the resultant map can be visualised in Figure 34. Similar profiles sit closer together on the map and this helps us to visualise how the grid demand profiles can vary.

When peak shaving it is important to know when a demand peak is occurring, if it is at the beginning of the day, at the end of the day, or if there are multiple peaks. Where the peaks occur will depend on the shape of the solar profile, solar generally follows a fairly curved shape ramping up in the morning and down in the evening, this is affected by factors such as cloud cover where the curved shape can have dramatic drops in generation. For instance, if there is cloud cover in the afternoon the grid demand energy peak is likely to occur in the afternoon. The SOM shows us that profiles with larger spikes in the afternoon are placed in the bottom right of the map, profiles with greater spikes in the evening are shown in the top middle, profiles with equal peaks are closer to the middle and profiles in the top left have longer peak period similar to the previous chapter indicating the solar system having a constant low output throughout the day.

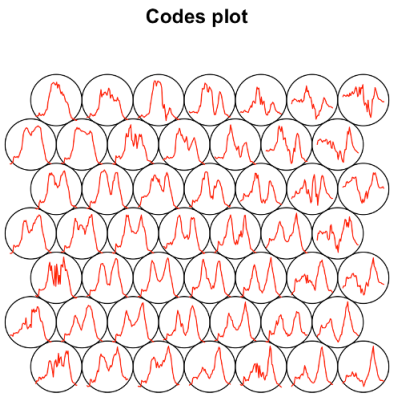


Figure 34: Self-organising map codes plot of the 2019 grid demand profiles from the commercial shopping centre.

6.2 Clustering

Since we are interested in where the predominant peaks lie, the neurons of the SOM were clustered manually so that the clusters could be based solely on where the energy peaks occur and ignore the affect of the small fluctuations within the profiles. The chosen clusters can be seen in Figure 35.

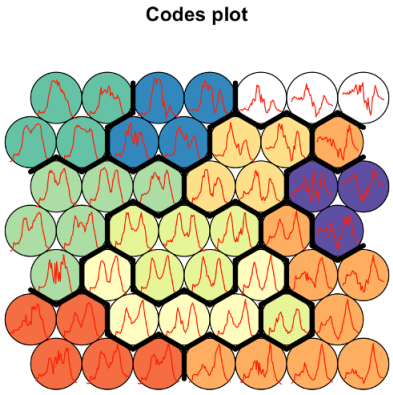


Figure 35: Self-organising map codes plot of the 2019 grid demand profiles from the commercial shopping centre coloured by cluster.

The problem encountered in Chapter 4 when trying to assign unseen days to already defined clusters holds for these found grid demand clusters. Grouping day ahead grid demand profiles is not as straight forward as with energy demand

profiles, as it is harder to elicit factors behind why a solar profile shape will occur. As previously mentioned companies specialise in day ahead solar generation prediction, and so in order to cluster a day ahead grid demand profile to a defined cluster the following methodology was used:

1. The insights derived from clustering the energy demand $E_{total\ demand}$ (outlined in Chapter 4) are used to generate a forecast of the day ahead total energy demand. This is done by using the chosen cluster's average energy demand as the prediction.

2. The predicted grid demand $ED_{grid\ demand}$ can then be calculated by subtracting the day ahead solar generation E_{solar} from the predicted energy demand $E_{total\ demand}$.

3. The day can then be assigned to a grid demand cluster based on which cluster average has the highest correlation (introduced in Equation 17) with the predicted grid demand profile. Figure 36 shows how the cluster with the highest correlation is chosen.

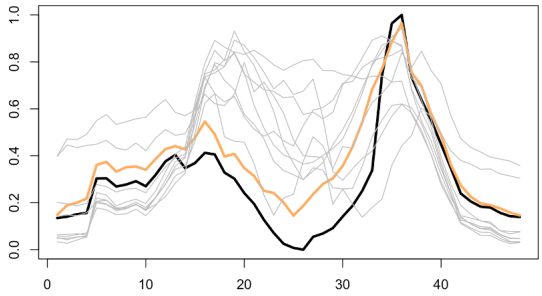
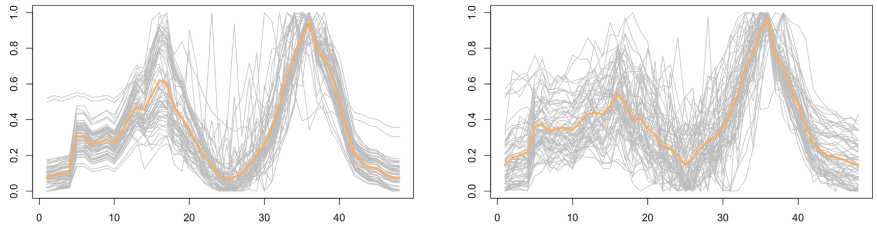


Figure 36: A plot showing the average grid demand profile of each of the clusters in grey, with a predicted grid demand profile from 2018 shown in black. The average cluster grid demand profile that produced the highest correlation value of 0.925 was the medium orange cluster, it is cluster average is represented by the thick medium orange line.

All profiles associated with the medium orange cluster can be seen in Figure 37. Looking at the grouped profiles we see that there is more variation within the cluster compared to the clusters found in Chapter 4, but despite this increased variation the grid demand profiles still tend to follow a similar shape. The most variation within the profiles happens during the morning peak. The clustered 2019 data shows the grid demand profiles associated with the manually chosen clusters. The 2018 data shows the profiles grouped into the medium orange cluster based on the methodology outlined above. The grouped 2018 grid demand profiles show similar trends to the 2019 cluster giving us a good indication

that the methodology outlined above is working. Interestingly the 2018 grouped data has less variation than the 2019 data.



(a) (b)

Figure 37: The graphs show all the grid demand profiles found within the medium orange cluster: Subfigure (a) shows the 2018 data which was clustered based on the outlined methodology, Subfigure (b) shows the 2019 data manually clustered based on the SOM neurons.

6.3 Conclusion

This chapter highlights a novel way to cluster day ahead grid demand profiles in turn allowing us to extend our previous reinforcement learning approach to energy systems incorporating a solar system. If a reinforcement learning agent is created for each cluster it can learn how to target specific grid demand peaks related to the cluster in turn drastically reducing the complexity of the overall problem. This will allow for reinforcement learning agents to be a feasible solution for peak shaving in combined micro-grids despite the "curse of dimensionality" that has been run into in the past.

7 Reinforcement Learning System for Peak Shaving including a Solar System

This chapter extends the reinforcement learning agents introduced in Chapter 5 to incorporate the existing solar system installed at the commercial shopping centre. The previous chapter explored clustering grid demand profiles, this chapter utilises the found clusters to create a system of reinforcement learning agents in a similar manner to Chapter 5. The splitting of data should help account for some of the inherent complexity introduced by the solar system. A few small changes are made to the agents (defined in Chapter 5) to better suit this new application, namely the parameters used in the reward signal and the addition of pre-training the network.

Chapter 6 resulted in 10 clusters being found, in order to create a fully operational peak shaving system 10 agents would need to be trained to encompass all profiles shapes the system could encounter. This chapter shows how an agent would run on one of the found clusters, in this case the medium orange cluster found in the previous chapter, as proof of concept of how the whole system could operate if it were extended by training agents on the other 9 agents.

7.1 Reward

In order to adjust to the new more volatile environment, the reward parameters were altered slightly. As explained in Section 5.3 there are two parts to the reward function, the first part reward 1 encourages the agent to learn a strategy to best suit the average energy demand profile in the cluster. The second part reward 2 is received at the end of an episode and is proportional to the total demand reduction for the day and helps to account for the variation between profiles within the cluster. When training the agent for this new application where a solar system is included greater emphasis was put onto reward 2 in order to encourage the agent to take the variation between profiles in the cluster into account since the addition of a solar system has increased it. The updated reward parameters used are as follows:

- $\alpha = \frac{1}{10000}$
- $\beta = \frac{1}{1000}$

7.2 Training the Reinforcement Learning Agent

The increased variation in this application results in the agent requiring a much longer training time to learn this new environment. To try reduce the training time to account for limited computational resources the network was pre-trained on the average grid demand profile for 1250 episodes with a constant epsilon of 0.1. Using these found parameters as the initial parameters during the full training process gives the network a better starting point in order to learn the

more complex environment faster. To further give the agent a better chance of learning the training iterations on all the 2019 grid demand profiles in the cluster was increased to 10000.

7.3 Testing the Reinforcement Learning Agent

Given only one agent is trained, only a portion of the 2018 and 2019 data that falls within the cluster is used to test the system. The 2019 data had clusters that were already formed and grid demand profiles which fell into the medium orange cluster that were used to train the agent were also used to test the agent. In order to decide which days in 2018 the agent would be tested on the following process outlined in Section 6.2 was used on each day:

1. Factors elicited from the energy demand clusters were used to predict the day ahead energy demand using the cluster average.
2. The solar generation was subtracted from the energy demand prediction to get the grid demand prediction.
3. The day was assigned to a grid demand cluster based on the highest correlation found between the predicted grid demand and cluster averages.

7.4 Results

The medium orange cluster's average grid demand profile is indicated in blue in Figure 37. The profile has two energy demand peaks, one in the morning and a larger one in the afternoon. The larger peak is only slightly bigger and so both peaks would need to be targeted in order to derive full benefit out of the battery's capacity. As explained in Section 7.2 the reinforcement learning agent is first trained on the cluster's average energy demand profile to attain better starting parameters for the neural network. The battery output at the end of the pre-training can be seen in Figure 38 indicated by the orange line. The agent targets both the morning and afternoon demand peaks of the average profile.

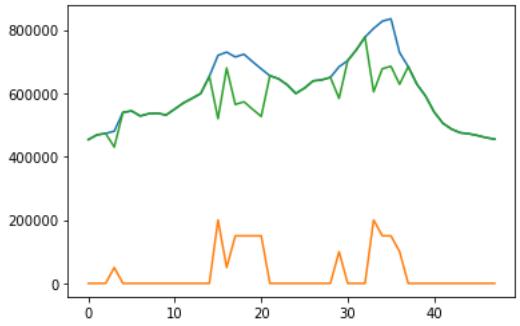


Figure 38: Reinforcement learning agent's battery output after pre-training on the cluster's average profile.

Figure 39 shows the undiscounted accumulated rewards averaged over 50 episode iterations for the agent whilst training on all the 2019 grid demand profiles found in the cluster. Over the first 50 iterations we see the reward steadily increasing, the agent then goes through dips with periods of lower increase, this shows the agent exploring the state space and needing more time to increase the average reward. Another consideration would be that the agent is receiving a more complex reward gradient due to the potential being calculated with a continuous battery output rather than a discretised one (as explained in Section 5.3) making the process of maximising the rewards less straight forward, also increasing the training time required.

Figure 39: Reinforcement learning agent's undiscounted accumulated rewards averaged over 50 episode iterations.

Figure 40 shows the learning progression of the agent, some of the variation between the grid demand profiles within the cluster can be seen when looking at the different episodes. The variation tends to come through in the magnitude of the initial peak. After the first 50 episodes the agent outputs randomly but in the times where the two demand peaks occur showing the initial starting values helped the agent to start in a better position. As the episodes progress the agent starts to explore the environment and begins to output near the beginning, running out of capacity before the second peak occurs. Over the following episodes it starts to correct itself and shifts towards the second peak

targeting the whole day. As can be seen by the variation in the profiles the initial peak does not occur at the same times and so the agent starts to target the middle of the day as well to account for the risk associated with this. By the end of the 10000 episode training the agent still targets the whole of the middle of the day but is starting to learn that saving capacity earlier on during the day and increasing the output near the bigger second peak is advantageous. With increased training time the agent can become more optimal.

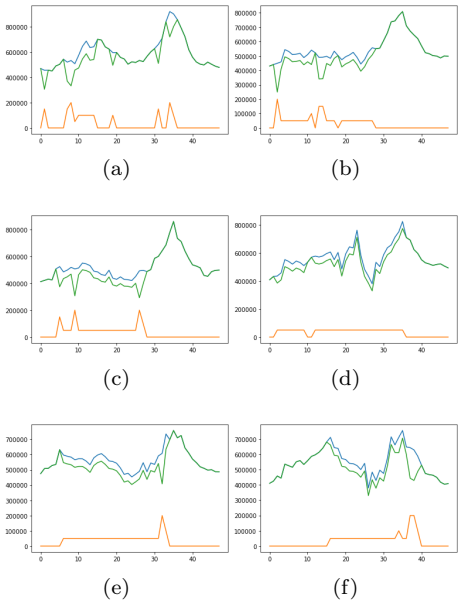
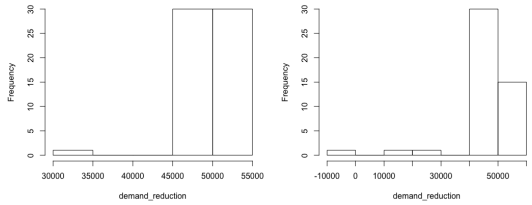


Figure 40: Learning progression of the agent: Subfigure (a) shows episode 50, Subfigure (b) shows episode 200, Subfigure (c) shows episode 500, Subfigure (d) shows episode 1000, Subfigure (e) shows episode 5000 and Subfigure (f) shows episode 10000.

The agent was able to attain savings on every day in 2018 and all but 1 in 2019. The agent was predominantly getting 50 kW savings each day. The agent was conservative and output a set amount throughout the times of the day where it thought the demand could peak (as can be seen in Figure 43) not increasing the output near the second peak as we saw it starting to learn to do it at the end of the training process. The average retention attained given the the binary output constraint was 25.58% and 24.92% without the constraint on the 2018 data and 26.30% and 24.75% on the 2019 data. These retention binary constrained and unconstrained values are closer together than the retention values seen in Chapter 5, this is because with the smaller peaks the optimal use of the battery often results in a reduction of the full inverter size with both the binary constrained and unconstrained outputs. Figures 62 and 63 in the

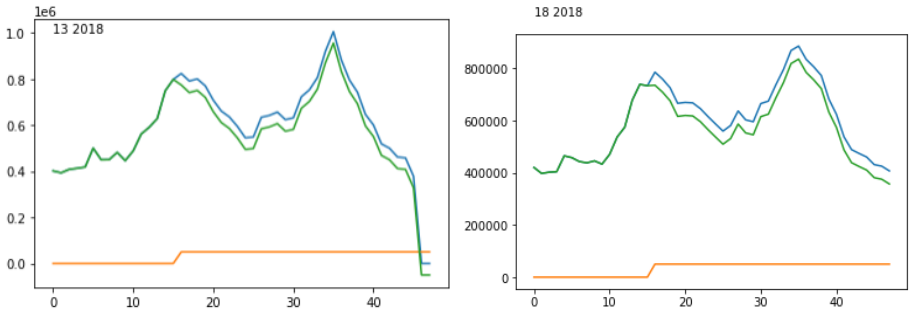
Appendix elaborate on the retention results achieved.



(a) (b)

Figure 41: Histograms of the daily savings derived from the agent: Subfigure (a) shows the distribution of savings attained on the 2018 data using the agent and Subfigure (b) shows the distribution of savings attained on the 2019 data.

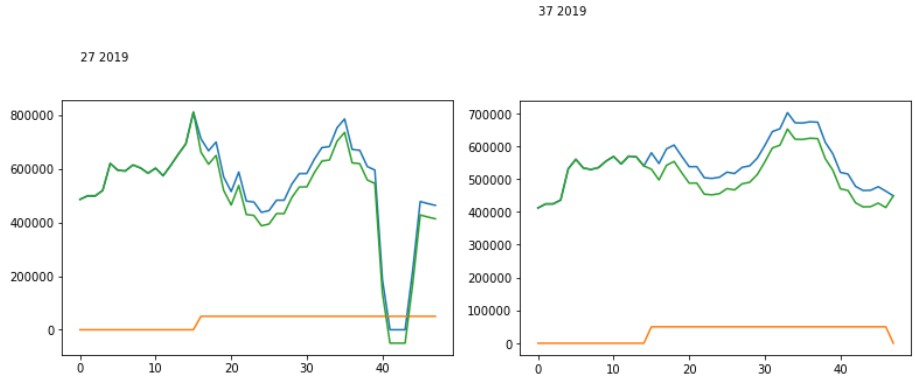
Figure 42 shows the agent at its best and worst in 2018 in terms of unconstrained retention, on the worst day the agent only attained 30 kW savings as it missed part of the first peak as the peak ramped up faster than the other profiles within the cluster. As compared to the best day which ramped up slightly later and received the full 50 kW reduction.



(a) (b)

Figure 42: Demand reduction overview of the best and worst retention days on the 2018 data: Subfigure (a) shows the worst unconstrained retention of 15.96% and Subfigure (b) shows the best unconstrained retention of 27.78%.

Figure 43 shows the agent at its best and worst in 2019 in terms of retention. On the worst day the agent achieved no reduction, this is because the initial peak was larger than the second and peaked early. Not achieving a reduction only occurred once across the 49 profiles in the cluster in 2019 and so the clustering methodology used to group unseen days appears to be working well overall.



(a) (b)

Figure 43: Demand reduction overview of the best and worst retention days on the 2019 data: Subfigure (a) shows the worst unconstrained retention of 0% and Subfigure (b) shows the best unconstrained retention of 31.25%.

8 Conclusion

8.1 Overview

South Africa's coal-based energy sector is in crisis; faulty infrastructure and poor service delivery is unable to meet the growing energy demand placed on the national grid. Moreover, the environmental imperative of transitioning from fossil fuel-based energy systems to renewable sources is being increasingly advocated globally. Peak shaving using a combined micro-grid has the potential to decrease the maximum energy demand a consumer places on the national grid, while capitalising on renewable solar energy resources. However, peak shaving is only financially viable if the battery system in a combined micro-grid is optimally utilised. Prior research suggested that reinforcement learning could be applied to the peak shaving problem as a means to optimise battery control. However, researchers encountered the "curse of dimensionality" and failed to achieve practical results. This thesis investigated a novel approach to the peak shaving problem, whereby forward based clustering was used to reduce the complexity of the environment presented to reinforcement learning agents, thus mitigating the "curse of dimensionality".

In this study, trained learning agents were able to achieve practical results based on unfamiliar historical energy data obtained from a commercial shopping centre in Cape Town. These results could be implemented to achieve a working peak shaving solution for the shopping centre if a battery system is installed. In instances when a solar system was excluded from the energy system, the reinforcement learning agent was able to reduce the maximum monthly energy

demand placed on the national grid by 91 kW. An average binary constrained retention of 63.04% across all months was achieved. If practically implemented, this reduced peak energy demand would translate to a reduced demand charge for the customer. Promising results were also achieved in instances when a solar component was included in the energy system. A learning agent trained to a particular cluster typically derived a 50 kW reduction in energy demand on all unfamiliar days. While an average binary retention of only 25.58% was achieved, the learning progression shows that the agent is learning to increase the output by the second peak, suggesting that retention can be improved with increased training time.

While learning agents were only trained and tested for one of the clusters found in instances when a solar component was included in the energy system, this initial exploration illustrates how the approach could work if it was extended to include an entire system. It is encouraging that a forward clustering approach enables the learning agent to derive practical results in an extremely volatile environment. This method successfully reduces the "curse of dimensionality", improving the feasibility of reinforcement learning for application to the peak shaving problem. Nevertheless, the effectiveness of learning agents could be further improved through additional research. Possible improvements for the reinforcement learning approach to peak shaving are outlined in the following subsection.

8.2 Future Recommendations

This thesis illustrates that forward clustering used in conjunction with reinforcement learning is a promising approach when considering the peak shaving problem. There is much scope to improve the efficiency of the approach however. A summary of the major limitations associated with the approach, as well as recommendations to potentially overcome these drawbacks, is outlined below:

- For the purpose of this investigation, discrete battery outputs were used to reduce the complexity of the problem. Discretized outputs limit learning agents, such that it is worthwhile exploring alternative reinforcement learning methods (such as actor critic) that support a continuous action space. A continuous action space would allow learning agents to generate more versatile outputs, increasing the efficiency of battery use. Moreover, a continuous action space would simplify the reward gradient, enabling agents to be trained faster.
- Training time (which was restricted due to computational capacity) further limited the learning agents. Increasing training time would improve results.
- When training the reinforcement learning agents, parameters (such as discount rate, epsilon, and the variables used to balance the reward) were

chosen manually. To improve agent parameters, optimisation techniques could alternatively be employed.

- As mentioned, there are a host of companies that specialise in solar generation forecasts for the day ahead. For the purpose of this study (which utilised historical solar data for the actual day in place of a forecast), the solar generation forecasts were assumed to be 100% accurate. Methodology that utilises true solar generation forecasts would provide a more realistic reflection of how the energy system would operate in practice.

- This study focused on a single commercial property in Cape Town. Future research could extend this investigation by applying our methodology to multiple sites and evaluating the efficacy of our approach across sites. Moreover, the study site had accumulated only two years' worth of data and utilises just one battery size in their energy system. It would be beneficial for future researchers to acquire a more extensive historical dataset, that exceeds two years, to train and test the learning agents. As the peak shaving problem can also be framed as a financial optimisation problem, we recommend investigating the optimal battery size for a site by comparing the financial gains achieved when using different battery sizes in an energy system.

- Since this study only focuses on a single site and battery size not covered in other research, it is hard to draw comparison to other methods. A comparative study bench marking different peak shaving methods on the same data set and battery size, will give a good indication of how different methods perform relative to each other.

- Finally, this investigation aimed to reduce energy demand peaks on each day of the month, while demand charges are computed based on energy demand across the month as a whole. We chose this approach for its lower risk compared to targeting specific days in a month. Our research could be extended if the magnitude of an upcoming day's energy demand could be predicted and used to determine whether or not the learning agent should output at all on that specific day.

References

[Al-Jabery et al., 2014] Al-Jabery, K., Wunsch, D. C., Xiong, J., and Shi, Y. (2014). A novel grid load management technique using electric water heaters and q-learning. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 776–781. IEEE.

[Aurélien, 2019] Aurélien, G. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

[Bekun et al., 2019] Bekun, F. V., Emir, F., and Sarkodie, S. A. (2019). Another look at the relationship between energy consumption, carbon dioxide emissions, and economic growth in south africa. *Science of The Total Environment*, 655:759–765.

[Brockman et al., 2016] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym.

[Charrad et al., 2014] Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. (2014). Nbclust: An r package for determining the relevant number of clusters in a data set. *Journal of Statistical Software*, 61(6):1–36.

[Chiş et al., 2015] Chiş, A., Lunden, J., and Koivunen, V. (2015). Optimization of plug-in electric vehicle charging with forecasted price. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2086–2089. IEEE.

[Chua et al., 2016] Chua, K., Lim, Y., and Morris, S. (2016). Energy storage system for peak shaving. *International Journal of Energy Sector Management*, 10(1):3 – 184.

[Claessens et al., 2013] Claessens, B. J., Vandael, S., Ruelens, F., De Craemer, K., and Beusen, B. (2013). Peak shaving of a heterogeneous cluster of residential flexibility carriers using reinforcement learning. In *IEEE PES ISGT Europe 2013*, pages 1–5. IEEE.

[Claessens et al., 2018] Claessens, B. J., Vanhoudt, D., Desmedt, J., and Ruelens, F. (2018). Model-free control of thermostatically controlled loads connected to a district heating network. *Energy and Buildings*, 159:1–10.

[Clifford, 1979] Clifford, H. (1979). Peak shaving via emergency generator. In *INTELEC - 1979 International Telecommunications Energy Conference*, pages 316–318.

[Conca, 2019] Conca, K. (2019). Is there a role for the un security council on climate change? *Environment: Science and Policy for Sustainable Development*, 61(1):4–15.

[Crenshaw, 2017] Crenshaw, M. (2017). *Analysis of vortex tube applications in hydrogen liquefaction*. PhD thesis.

[Dauer et al., 2013] Dauer, D., Flath, C. M., Ströhle, P., and Weinhardt, C. (2013). Market-based ev charging coordination. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 2, pages 102–107.

[Dongol et al., 2018] Dongol, D., Feldmann, T., Schmidt, M., and Bollin, E. (2018). A model predictive control based peak shaving application of battery for a household with photovoltaic system in a rural distribution grid. *Sustainable Energy, Grids and Networks*, 16:1–13.

[Du and Fei, 2008] Du, D. and Fei, M. (2008). A two-layer networked learning control system using actor–critic neural network. *Applied mathematics and computation*, 205(1):26–36.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[Hassan et al., 2015] Hassan, H., Negm, A., Zahran, M., and Saavedra, O. (2015). Assessment of artificial neural network for bathymetry estimation using high resolution satellite imagery in shallow lakes: Case study el burullus lake. *International Water Technology Journal*, 5.

[Hemmatinezhad et al., 2022] Hemmatinezhad, M., Gholizadeh, M., Ramezaniyan, M., Shafiee, S., and Ghazi Zahedi, A. (2022). Predicting the success of nations in asian games using neural network.

[Hohne et al., 2020] Hohne, P., Kusakana, K., and Numbi, B. (2020). Model validation and economic dispatch of a dual axis pv tracking system connected to energy storage with grid connection: A case of a healthcare institution in south africa. *Journal of Energy Storage*, 32:101986.

[Ihirwe et al., 2021] Ihirwe, J. P., Li, Z., Sun, K., Bimenyimana, S., Wang, C., Asemota, G. N. O., Nduwamungu, A., and Mesa, C. K. (2021). Solar pv minigrid technology: Peak shaving analysis in the east african community countries. *International Journal of Photoenergy*, 2021:5580264.

[Kanzumba and Kusakana, 2019] Kanzumba and Kusakana (2019). Optimal electricity cost minimization of a grid-interactive pumped hydro storage using ground water in a dynamic electricity pricing environment. *Energy Reports*, 5:159–169.

[Karmiris and Tenger, 2013] Karmiris and Tenger, T. (2013). Peak shaving control method for energy storage.

[Khan et al., 2014] Khan, K., Rehman, S. U., Aziz, K., Fong, S., and Sarasvady, S. (2014). Dbscan: Past, present and future. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, pages 232–238.

[Kingma and Ba, 2017] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

[Kohonen et al., 1996] Kohonen, Oja, Simula, Visa, and Kangas (1996). Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358–1384.

[Kuster et al., 2017] Kuster, C., Rezgui, Y., and Mourshed, M. (2017). Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society*, 35:257–270.

[Lee and Powell, 2012] Lee, D. and Powell, W. B. (2012). An intelligent battery controller using bias-corrected q-learning. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[Li, 2017] Li, Y. (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

[Likas et al., 2003] Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461.

[Lumka et al., 2021] Lumka, M., Thobile, H. I., Pearl, N. N., Zukile, Q., and Bruwer (2021). Load shedding and its influence on south african small, medium and micro enterprise profitability, liquidity, efficiency and solvency.

[Matousek and Gärtner, 2007] Matousek, J. and Gärtner, B. (2007). *Understanding and using linear programming*. Springer Science & Business Media.

[Maugis et al., 2009] Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009). Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3):701–709.

[Mauler et al., 2021] Mauler, L., Duffner, F., Zeier, W. G., and Leker, J. (2021). Battery cost forecasting: a review of methods and results with an outlook to 2050. *Energy & Environmental Science*.

[McLaren et al., 2017] McLaren, J. A., Gagnon, P. J., and Mullendore, S. (2017). Identifying potential markets for behind-the-meter battery energy storage: A survey of us demand charges. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

[Merven et al., 2021] Merven, B., Burton, J., and Lehmann-Grube, P. (2021). Assessment of new coal generation capacity targets in south africa's 2019 integrated resource plan for electricity.

[Mongia et al., 2021] Mongia, G., Kaur, I., Marwah, R. G., and Malyan, A. (2021). A comprehensive review of the different methods and materials for the construction of mfcs and their effect on the performance of mfc.

[Nielsen, 2016] Nielsen, F. (2016). Hierarchical clustering. In *Introduction to HPC with MPI for Data Science*, pages 195–211. Springer.

[Nwankpa et al., 2018] Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

[Oudalov et al., 2007] Oudalov, Cherkaoui, R., and Beguin, A. (2007). Sizing and optimal operation of battery energystorage system for peak shaving application. In *Proceedings of the IEEE powertech conference*, pages 621–25. IEEE.

[Oyewo et al., 2019] Oyewo, A. S., Aghahosseini, A., Ram, M., Lohrmann, A., and Breyer, C. (2019). Pathway towards achieving 100 % renewable electricity by 2050 for south africa. *Solar Energy*, 191:549–565.

[Rahimi et al., 2013] Rahimi, Zarghami, Vaziri, and Vadhva (2013). A simple and effective approach for peak load shaving using battery storage systems. In *Proceedings of the North American Power Symposium, IEEE*, pages 1–5. IEEE.

[Rokach and Maimon, 2005] Rokach, L. and Maimon, O. (2005). Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer.

[Ruelens et al., 2016] Ruelens, F., Claessens, B. J., Quaiyum, S., De Schutter, B., Babuška, R., and Belmans, R. (2016). Reinforcement learning applied to an electric water heater: From theory to practice. *IEEE Transactions on Smart Grid*, 9(4):3792–3800.

[Ruelens et al., 2014] Ruelens, F., Claessens, B. J., Vandael, S., Iacovella, S., Vingerhoets, P., and Belmans, R. (2014). Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning. In *2014 Power Systems Computation Conference*, pages 1–7. IEEE.

[Shi et al., 2017] Shi, G., Liu, D., and Wei, Q. (2017). Echo state network-based q-learning method for optimal battery control of offices combined with renewable energy. *IET Control Theory & Applications*, 11(7):915–922.

[Smith et al., 2021] Smith, P., Beaumont, L., Bernacchi, C., Byrne, M., Cheung, W., Conant, R., Cotrufo, F., Feng, X., Janssens, I., Jones, H., Kirschbaum, M., Kobayashi, K., LaRoche, J., Luo, Y., McKechnie, A., Penuelas, J., Piao, S., Robinson, S., Sage, R., Sugget, D., Thackeray, S., Way, D., and Long, S. (2021). "essential outcomes for cop26". *"Global Change Biology"*.

[Son and Song, 2014] Son, S. and Song, H. (2014). Real-time peak shaving algorithm using fuzzy wind power generation curves for large-scale battery energy storage systems. *International Journal of Fuzzy Logic and Intelligent Systems*, 14(4):305–312.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[The-Lazy-Programmer, 2018] The-Lazy-Programmer (2018). Artificial intelligence: Reinforcement learning in python.

[Ting and Byrne, 2020] Ting, M. B. and Byrne, R. (2020). Eskom and the rise of renewables: Regime-resistance, crisis and the strategy of incumbency in south africa's electricity system. *Energy Research Social Science*, 60:101333.

[Uddin et al., 2018] Uddin, M., Romlie, M. F., Abdullah, M. F., Abd Halim, S., Abu Bakar, A. H., and Chia Kwang, T. (2018). A review on peak load shaving strategies. *Renewable and Sustainable Energy Reviews*, 82:3323–3332.

[Vayá et al., 2014] Vayá, M. G., Roselló, L. B., and Andersson, G. (2014). Optimal bidding of plug-in electric vehicles in a market-based control setup. In *2014 Power Systems Computation Conference*, pages 1–8. IEEE.

[Vedullapalli et al., 2019] Vedullapalli, D. T., Hadidi, R., and Schroeder, B. (2019). Optimal demand response in a building by battery and hvac scheduling using model predictive control. In *2019 IEEE/IAS 55th Industrial and Commercial Power Systems Technical Conference (I CPS)*, pages 1–6.

[Vázquez-Canteli and Nagy, 2019] Vázquez-Canteli, J. and Nagy, Z. (2019). Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied Energy*, (235):1072–1089.

[Yang et al., 2015] Yang, L., Nagy, Z., Goffin, P., and Schlueter, A. (2015). Reinforcement learning for optimal control of low exergy buildings. *Applied Energy*, 156:577–586.

[Zhang and Couloigner, 2005] Zhang, Q. and Couloigner, I. (2005). A new and efficient k-medoid algorithm for spatial clustering. In *International conference on computational science and its applications*, pages 181–189. Springer.

[Zurfi et al., 2017] Zurfi, A., Albayati, G., and Zhang, J. (2017). Economic feasibility of residential behind-the-meter battery energy storage under energy time-of-use and demand charge rates. In *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 842–849.

[Çelik et al., 2011] Çelik, M., Dadaşer-Çelik, F., and Dokuz, A. (2011). Anomaly detection in temperature data using dbscan algorithm. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pages 91–95.

9 Appendix

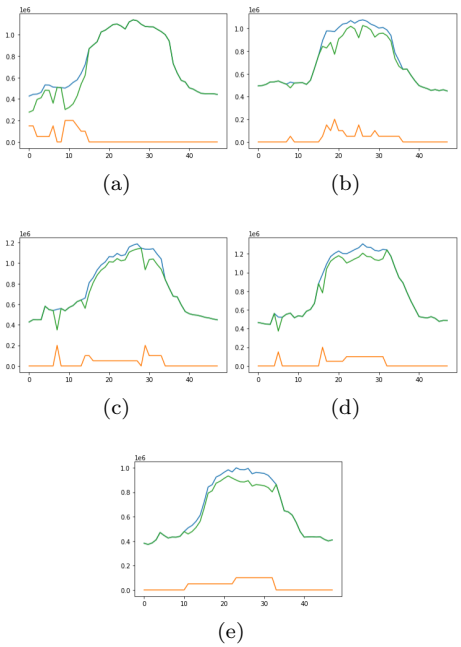


Figure 44: Learning progression of the agent trained without temperature in its state space on 2019 cluster 1 data: Subfigure (a) shows episode 20, Subfigure (b) shows episode 100, Subfigure (c) shows episode 500, Subfigure (d) shows episode 1000 and Subfigure (e) shows episode 5000

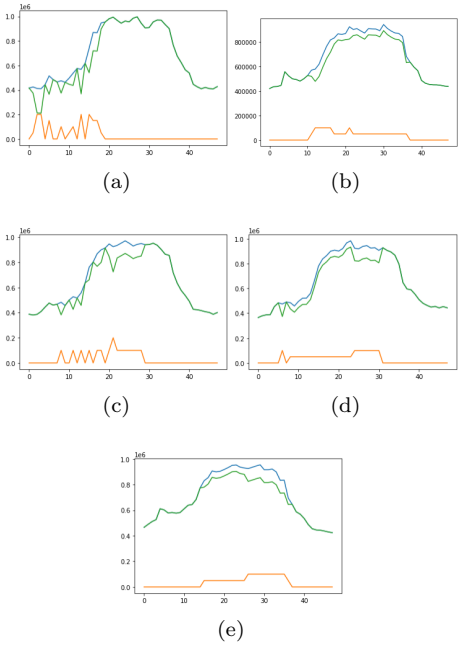
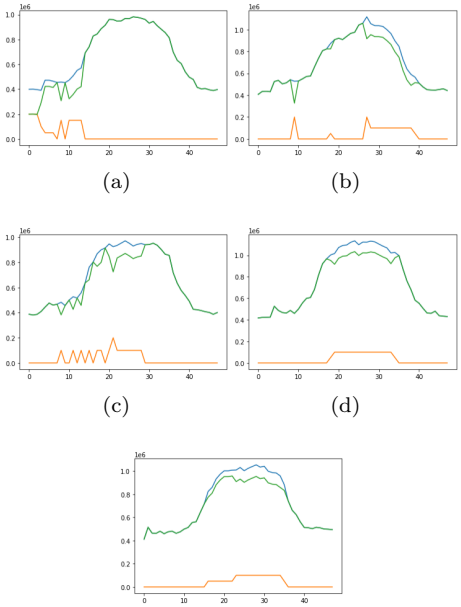
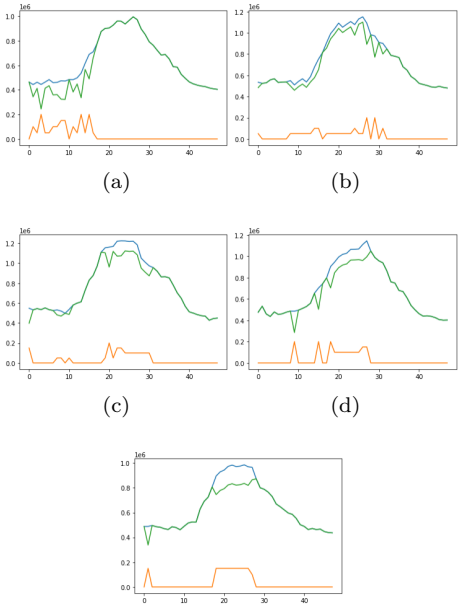


Figure 45: Learning progression of the agent trained without temperature in its state space on 2019 cluster 2 data: Subfigure (a) shows episode 20, Subfigure (b) shows episode 100, Subfigure (c) shows episode 500, Subfigure (d) shows episode 1000 and Subfigure (e) shows episode 5000



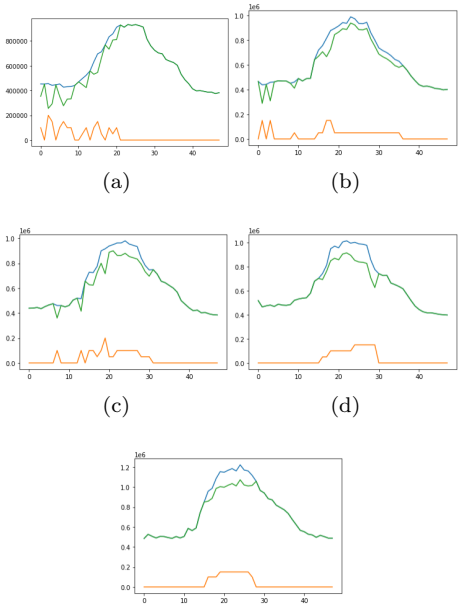
(a) (b) (c) (d) (e)

Figure 46: Learning progression of the agent trained with temperature in its state space on 2019 cluster 2 data: Subfigure (a) shows episode 20, Subfigure (b) shows episode 100, Subfigure (c) shows episode 500, Subfigure (d) shows episode 1000 and Subfigure (e) shows episode 5000



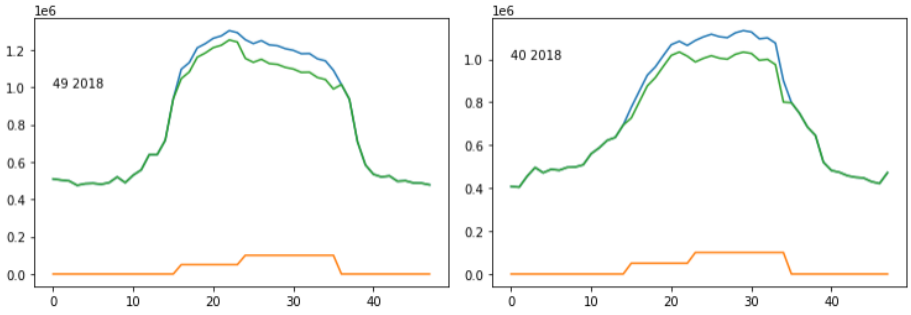
(a) (b) (c) (d) (e)

Figure 47: Learning progression of the agent trained without temperature in its state space on 2019 cluster 3 data: Subfigure (a) shows episode 20, Subfigure (b) shows episode 100, Subfigure (c) shows episode 500, Subfigure (d) shows episode 1000 and Subfigure (e) shows episode 5000



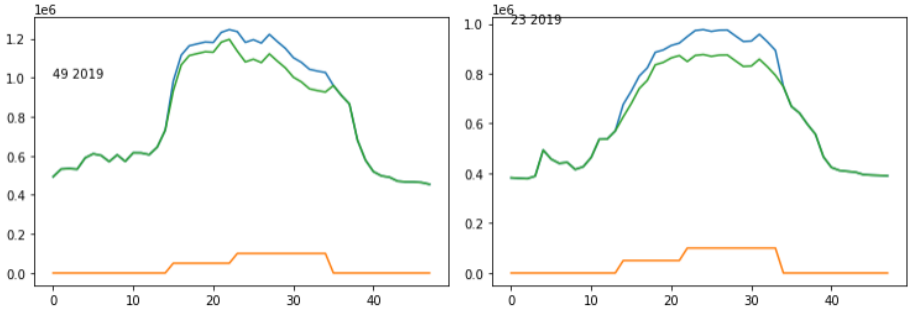
(a) (b) (c) (d) (e)

Figure 48: Learning progression of the agent trained with temperature in its state space on 2019 cluster 3 data: Subfigure (a) shows episode 20, Subfigure (b) shows episode 100, Subfigure (c) shows episode 500, Subfigure (d) shows episode 1000 and Subfigure (e) shows episode 5000



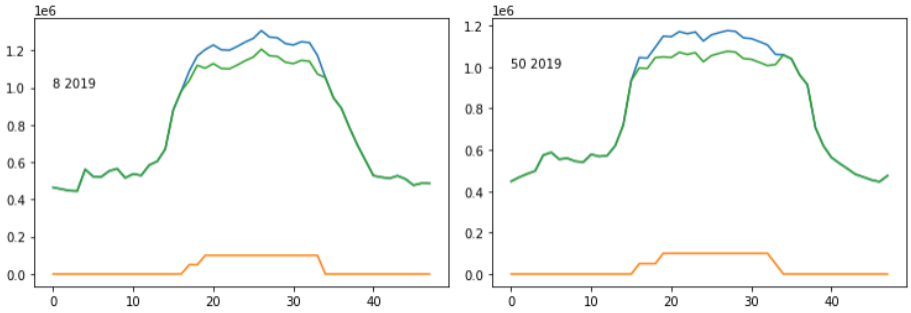
(a) (b)

Figure 49: Demand reduction overview of the agent trained without temperature in its state space on the best and worst retention days on the cluster 1 2018 data: Subfigure (a) shows the worst unconstrained retention of 27.78% and Subfigure (b) shows the best unconstrained retention of 66.45%



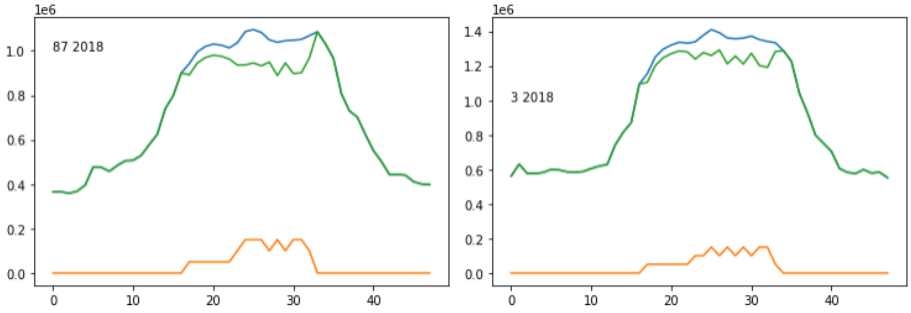
(a) (b)

Figure 50: Demand reduction overview of the agent trained without temperature in its state space on the best and worst retention days on the cluster 1 2019 data: Subfigure (a) shows the worst unconstrained retention of 25.51% and Subfigure (b) shows the best unconstrained retention of 61.35%



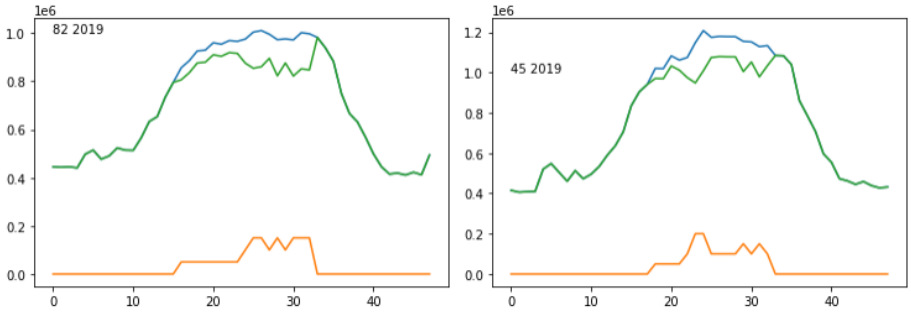
(a) (b)

Figure 51: Demand reduction overview of the agent trained with temperature in its state space on the best and worst retention days on the cluster 1 2018 data: Subfigure (a) shows the worst unconstrained retention of 50.00% and Subfigure (b) shows the best unconstrained retention of 64.93%



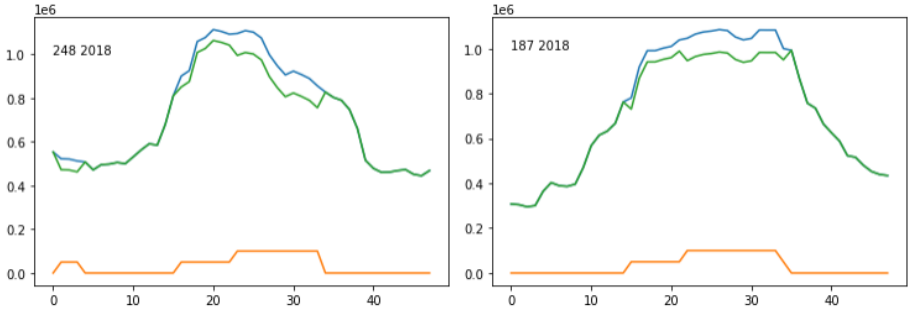
(a) (b)

Figure 52: Demand reduction overview of the agent trained without temperature in its state space on the best and worst retention days on the cluster 2 2018 data: Subfigure (a) shows the worst unconstrained retention of 6.83% and Subfigure (b) shows the best unconstrained retention of 70.95%



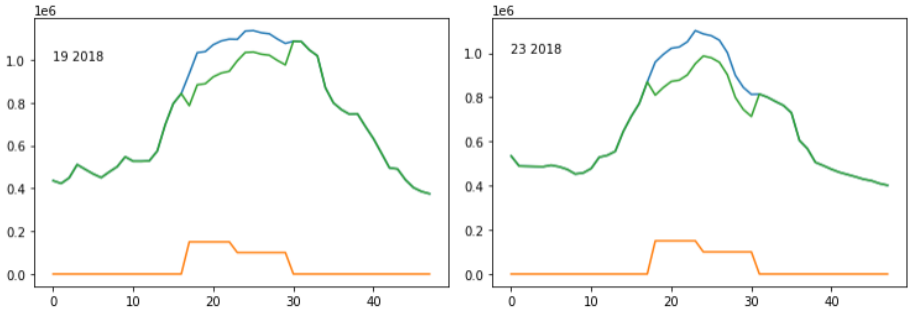
(a) (b)

Figure 53: Demand reduction overview of the agent trained without temperature in its state space on the best and worst retention days on the cluster 2 2019 data: Subfigure (a) shows the worst unconstrained retention of 6.42% and Subfigure (b) shows the best unconstrained retention of 61.90%



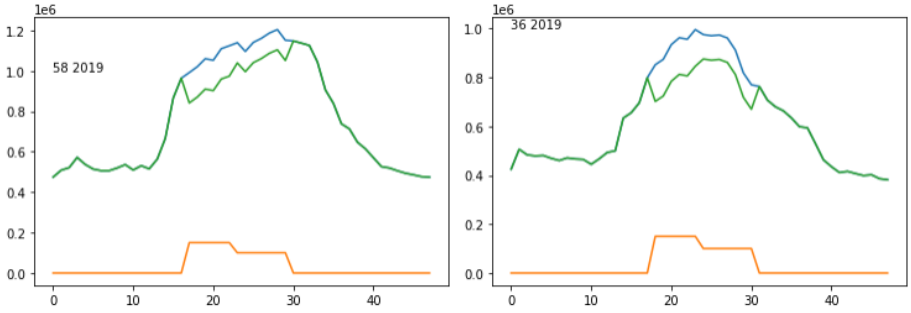
(a) (b)

Figure 54: Demand reduction overview of the agent trained with temperature in its state space on the best and worst retention days on the cluster 2 2018 data: Subfigure (a) shows the worst unconstrained retention of 26.31% and Subfigure (b) shows the best unconstrained retention of 71.43%



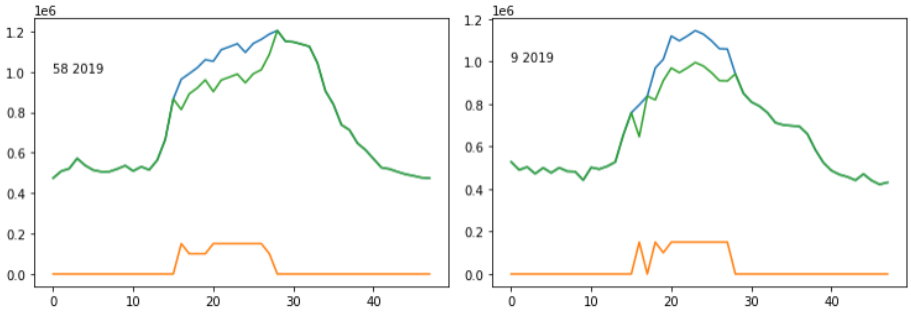
(a) (b)

Figure 55: Demand reduction overview of the agent trained without temperature in its state space on the best and worst retention days on the cluster 3 2018 data: Subfigure (a) shows the worst unconstrained retention of 31.05% and Subfigure (b) shows the best unconstrained retention of 75.00%



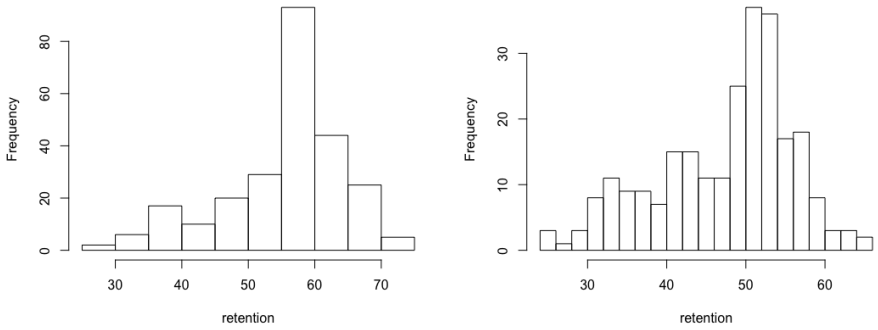
(a) (b)

Figure 56: Demand reduction overview of the agent trained without temperature in its state space on the best and worst retention days on the cluster 3 2019 data: Subfigure (a) shows the worst unconstrained retention of 28.15% and Subfigure (b) shows the best unconstrained retention of 62.13%



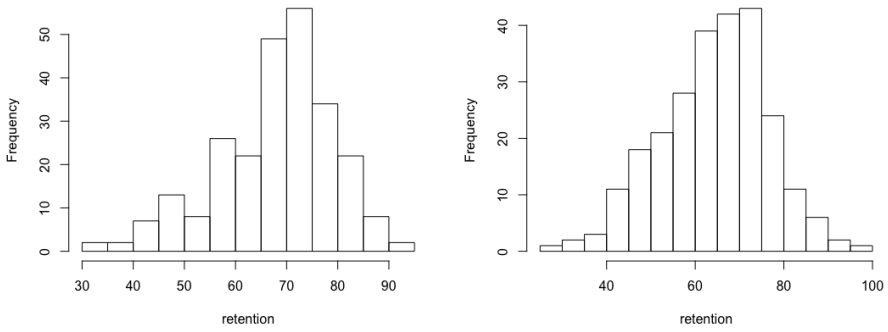
(a) (b)

Figure 57: Demand reduction overview of the agent trained with temperature in its state space on the best and worst retention days on the cluster 3 2019 data: Subfigure (a) shows the worst unconstrained retention of 0.00% and Subfigure (b) shows the best unconstrained retention of 75.00%



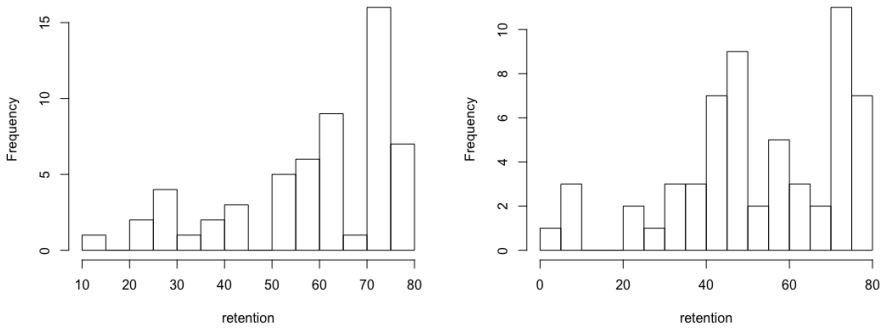
(a) (b)

Figure 58: Daily retention values without a constraint on cluster 2: Subfigure (a) shows the results of the 2018 data and Subfigure (b) shows the results of the 2019 data.



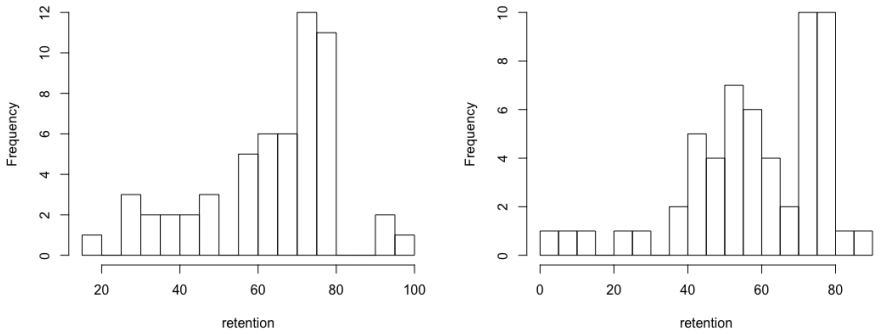
(a) (b)

Figure 59: Daily retention values including the binary output constraint on cluster 2: Subfigure (a) shows the results of the 2018 data and Subfigure (b) shows the results of the 2019 data.



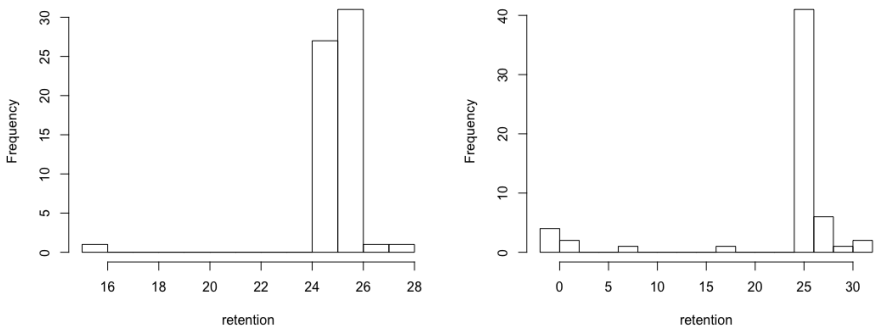
(a) (b)

Figure 60: Daily retention values without a constraint on cluster 3: Subfigure (a) shows the results of the 2018 data and Subfigure (b) shows the results of the 2019 data.



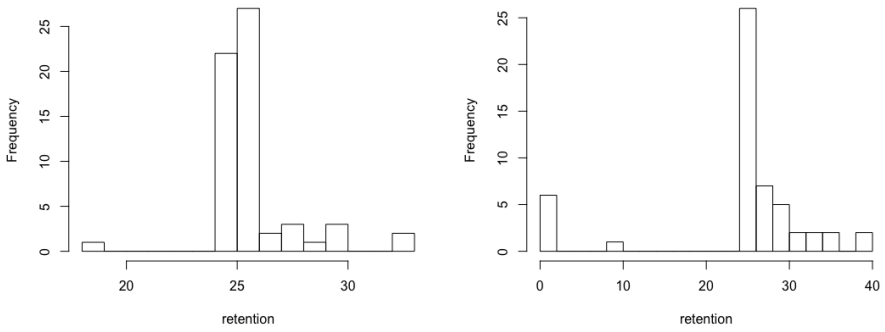
(a) (b)

Figure 61: Daily retention values including the binary output constraint on cluster 3: Subfigure (a) shows the results of the 2018 data and Subfigure (b) shows the results of the 2019 data.



(a) (b)

Figure 62: Daily retention values without a constraint on grid demand data: Subfigure (a) shows the results of the 2018 data and Subfigure (b) shows the results of the 2019 data.



(a) (b)

Figure 63: Daily retention values including the binary output constraint on grid demand data: Subfigure (a) shows the results of the 2018 data and Subfigure (b) shows the results of the 2019 data.

Month	Optimal Continuous Savings (kW)	Optimal Binary Savings (kW)	Agent Attained Savings (kW)	Binary Battery Retention (%)	Continuous Battery Retention (%)
1	150	120	73.8	61.7	49.2
2	157	130	58.4	44.8	37.3
3	160	134	82.5	61.5	51.6
4	160	157	100	63.7	55.6
5	180	140	96.5	69.1	56.8
6	170	134	75.0	55.8	45.9
7	163	118	86.8	73.7	59.2
8	147	123	49.6	40.3	33.6
9	200	200	100	50.0	50.0
10	159	135	59.1	43.7	37.1
11	180	150	100	66.7	55.6
12	160	135	58.2	43.0	36.4
Avg	164	139.7	78.3	56.2	47.3

Table 9: Table of results derived from the system of agents trained without temperature in their state space on the 2018 data.

Month	Optimal Continuous Savings (kW)	Optimal Binary Savings (kW)	Agent Attained Savings (kW)	Binary Battery Retention (%)	Continuous Battery Retention (%)
1	178	132	96.5	72.9	54.2
2	168	120	24.0	20.0	14.3
3	199	149	100.0	66.9	50.3
4	191	139	79.7	57.2	41.8
5	200	200	100.0	50.0	50.0
6	200	187	100.0	53.5	50.0
7	164	119	67.7	56.7	41.3
8	182	132	52.7	40.0	29.0
9	186	138	65.0	47.0	35.0
10	191	147	79.5	54.2	41.6
11	200	153	100.0	65.4	50.0
12	200	171	77.2	45.2	38.6
Avg	188	149	78.5	52.7	41.7

Table 10: Table of results derived from the system of agents trained without temperature in their state space on the 2019 data.