Utrecht University

Faculty of Sciences

Consumer Energy Demand Distribution Forecasting using Clustering Strategies and Applications in Robust Peak Shifting with Batteries

Energyworx Supervisor:

Rinse Veltman
Energyworx

First Supervisor:

Dr. ir. J.M. van den Akker
Utrecht University

Second Supervisor:

Dr. S. Dirksen
Utrecht University

Master Thesis

Pieter Knops

Computing Science and
Mathematical Sciences.

June 20, 2023

Abstract

In recent years, the electric grid is becoming increasingly overloaded. At the same time, dependence on electric energy is growing, and production is becoming less plannable and predictable due to a growing share in renewable energies.
To alleviate the resulting peak stresses on the electric grid, we perform peak shifting using batteries at the neighborhood scale. The (dis)charging schedules are based on a self-made hourly consumer demand forecasting approach.
The LSTM architecture is proposed for forecasting, and compared to an LR approach. Instead of a point-forecast, we implement a distribution forecast, allowing for accurate modelling of the stochasticity of consumer demand. Clustering is used for efficient training, and improved by our ReClustering technique, in which consumers are reclassified based on forecasting performance.
For planning batteries, we use 5 or 10 Tesla Powerwall batteries for peak shifting the hourly demand of the 500 consumers at hand. We propose and prove optimality of a fully polynomial time approximation scheme for the offline case, and evaluate different strategies in the online approach, in which a rolling horizon approach is used to solve for a time horizon of 24 hours into the future. A variable battery reserve is included in our proposed approach.
Our LSTM forecasting technique outperforms the shift (a persistence model) and linear regression approaches. Clustering significantly improves results. ReClustering also contributes to improved results.
Our battery planning algorithm outperforms the baseline approaches that do not use our forecasting approach, but only when the information in the percentiles of the distribution forecast are used for deciding on battery reserve, showing one of the advantages of distribution forecasting over point forecasting.

Keywords: Clustering, Demand Forecasting, Distribution Forecasting, Peak Shifting, Battery Planning

Acknowledgements

Firstly, I want to thank my supervisors M.Sc. Rinse Veltman from Energyworx who introduced me to the industry related to Energy Data, Dr. Sjoerd Dirksen who helped me to focus on the mathematical quality and Dr. ir. Marjan van den Akker helping me focus on the main objectives. Without the help of my supervisors, doing this thesis would have been much harder. Also, their feedback helped me in spotting mistakes and write in a more structured manner. I am thankful for the time they invested in this thesis.
Furthermore, I want to thank Ellert van der Velden from Energyworx for his time providing me with consumer demand data.
I want to thank Dr. Kirien Whan from the KNMI. Apart from providing me with weather forecasting data, she took the time to show me what thinks I could do to use this data most effectively.
Also, I want to thank Pieter Janssen from Enexis. He told me about the challenges of a Distribution Grid Operator. This helped me to formulate the peak shifting problem so that it also is useful in practice.
Lastly, I would like to thank my family and friends, especially my parents, and Jennifer de Smidt. It was great to be able to tell about the latest struggles and progressions, again and again. This also helped me to get a clear high-level view of my own thesis.

Contents

Chapter 1

Introduction and Background

1.1 Introduction

The Electricity Grid (EG) in the Netherlands[1, 2], as well as in various other countries is becoming overloaded, and in some locations, there are long waiting times for new connections to the grid[3]. At the same time, the dependence on electrical power is rising. For instance, the Dutch government wants households to be decoupled from gas and stimulates the use of electrical vehicles.
In addition, energy production is becoming less predictable as solar and wind power, that are heavily dependent on weather, have an increasing share in the electric energy production. Another challenge of using renewables, is that their peak production is not when there is also a high energy demand [4], causing further imbalances on the energy grid. This is an increasing problem.
For instance, In the Netherlands, on Saturday, April 23rd in 2022, the production of electric energy by renewables has been higher than demand for multiple hours[5]. However, during that month, only about 15% [6] of the electric energy production has been from renewables. This is not unique for the Netherlands or that time: When the share of green energy becomes more than 30%, this causes problems on network stability [7] as the resulting imbalance between production and consumption causes frequency and voltage instability [8].

To gain more insight in energy consumption, the European Parliament decided on conducting a cost-benefits analysis on smart meter deployment, resulting in a target penetration level of smart metering systems above 80% by 2020 in 16 European counties (including the Netherlands) [9]. Most countries have started this roll-out around 2013[10]. As such, energy consumption data has gotten far more precise with data per consumer and per hour (or sometimes per 15-minute) intervals, enabling more precise research on energy consumption characteristics.

While researching solutions for this problem of imbalances, it is crucial to have an energy demand forecast [11]. In this project, we will create an electrical energy demand forecast mechanism for many individual consumers in the Rotterdam area in the Netherlands. We focus on short-term forecasting up to 24 hours into the future.
We do not only give a forecasted value, but we also give a distribution forecast. This is used to reliably plan the (dis)charging of a battery pack for load shifting, placed inside a neighborhood. With load-shifting, peak stresses on the electric grid are reduced, and more renewables can be used. For this, forecasting has to be reliable. If forecasting mistakes are made, a battery might be discharged too fast and be empty too early. The forecasted energy demand and confidence of that demand are used to plan (dis)charging of the battery reliably.

This document is segmented in three parts.
In this chapter, we firstly introduce the Energy Markets, its parties and the (for this thesis) important aspects of generation in Section 1.2. In Section 1.3, we go over related research, split into the subjects forecasting and battery planning. In Section 1.4, we describe what our goals are, and what we are adding in this research field. In Section 1.5, we describe the data that we use for our models, and do some exploratory analysis of demand data in Section 1.6.

The second part, Chapter 2, is about clustering consumers and forecasting energy demand, in which first the methods are explained in Section 2.1 and then results are shown in Section 2.2. Our findings are discussed in Section 2.3.
In Chapter 3, we use our forecasting approach for load shifting. The model is described in Section 3.1. Offline algorithms and optimality proofs are given in Section 3.2 and the translation to the online case is written in Section 3.3. In Section 3.4, the experimental setup is described, of which results are shown in Section 3.5. We conclude with our findings about load shifting in Section 3.6.

1.2 Background: Energy Markets, Parties and Generation

The energy market consists of multiple types of actors, each with their own roles. In this section, we explain the energy market, who its players are, what goals and obligations they have, and how they relate to each other. Also, we explain one reason why high demand peaks cause high electricity prices.

1.2.1 Parties

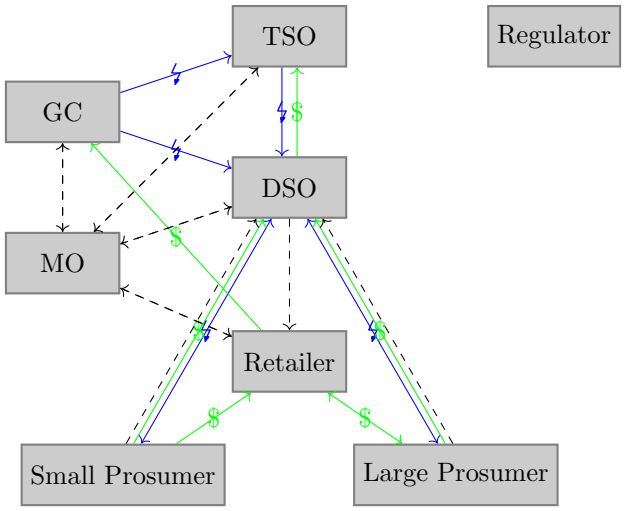


Figure 1.1: A selection of the most important parties in the energy market and how they relate to each other. Visualized are energy streams (blue), money streams (green) and information streams (black, dashed).

In the energy market, multiple players are present, directly or indirectly, all with different roles. The *European Network of Transmission System Operators for Electricity* (ENTSO-E) published the Harmonized Electricity Market Role Model, describing over 40 roles [12]. We give a summary of the most important parties, and what roles they more or less have. Note however, that this division of roles between parties can be slightly different between countries. The most important parties are as follows (also see Figure 1.1).

Regulators — In a liberalized energy market, regulators are responsible for creating and enforce rules on the market. For example, there are rules about which company can control what components, how energy availability is guaranteed and what changes in electricity billing can be made and how early that has to be communicated to the consumers. The Autoriteit Consument & Markt (*Authority Consumer and Market*, ACM) is the regulator of the Dutch electricity market.

MO — The *Market Operator* (MO) maintains the market systems where energy is offered en bought and directs the money to its proper destination. The APX Group is responsible for this in the Netherlands.

GC — *Generator Companies* (GCs). These companies are the energy producers and often have a diverse portfolio of generation plans, such as Coal Plants, Gas Plants, Solar Farms, Hydropower Plants and Wind-turbine Plants. They sell generation capacity on the long-term energy market to retail companies and when the expected demand of the close future becomes known, these companies calculate the cheapest possible way to generate the requested electricity, an optimization problem called the Unit Commitment problem. The resulting production planning often consists of a mix of renewables plus a cheap baseload, and more expensive but responsive generation plants to be able to produce short peak loads. Well-known generator companies in the Netherlands are: Essent, Engie, RWE generation NL and Eneco.

TSO — *Transmission System Operators* (TSO) control and maintain the high voltage energy network. Often, these companies have a national monopoly and are paid by the DSOs. TSOs transport energy from the production sites of GCs to elsewhere in the country, or even to other countries. TSOs are responsible for balancing the market, and closely monitor and electricity stability and quality[1]. The TSO of the Netherlands is TenneT, which also serves large parts of Germany.

DSO — *Distribution System Operators* (DSO), sometimes called *Distribution Network Operators*, or DNOs for short, are responsible for the distribution of energy from the national high voltage power lines (and some directly connected GCs, which is then called embedded generation) down to the lower voltages and the energy users. Often, these companies have regional monopolies. These companies are normally also responsible for communicating the data from smart meters to the retail companies. Every party that is connected to the distribution grid pays a connection fee to the DSO[2].
Examples of Dutch DSOs are: Liander, Enexis and Stedin. Stedin is the DSO for the Rotterdam area.

Retailer — The retail companies bridge the gap between generator companies and the energy users. It is the retailers' responsibility to buy future energy, such that, when users want to use electricity, there is enough production that can be used. If the energy use of clients of the retailer is higher than the amount of energy the retailer has bought at GCs, they have to buy additional energy, which can be very expensive. Additionally, if the amount of reserved energy from GCs is too low, the regulator can fine retail companies. The retailer is a so-called *Balance responsible party*. Hence, forecasting consumer energy load is a core business for these companies.
Well-known Dutch retailer companies are: Greenchoice, Eneco and Innova Energy.

Small Prosumers — Consumers, here called Small Prosumers, are one kind of end users, and are under contract of one energy Retailer. A notably big part of consumers now have *PhotoVoltaic* (PV) panels and are producing electricity. For some homes, the production during clear, sunny weather is more than their energy usage, making them energy producers. These Small Prosumers compare retailer offers and choose a retailer that can offer them low, fixed or flexible prices.

Large Prosumers — Bigger energy prosumers get commercial contracts from energy retailers. Sometimes, multiple retailers make a prosumer hand-tailored offers with some arrangements. For example, greenhouses can help retailers balance their total energy usage by turning their lights on when the total energy demand lies lower than what the retailer had forecasted. Therefore, Some large prosumers also act

[1] Voltage and frequency are usually used to describe energy stability and quality.
[2] In the Netherlands, this is different. Dutch retail companies collect the fees that consumers have to pay to the DSO.

as balancing companies. Examples of large prosumers are hospitals, universities, industry, greenhouses and factories.

BC — Balancing Companies form an emerging type of company (and hence, are not included in Figure 1.1). Their goal is to make profit of the temporal price differences on the energy market. This has as result that the energy demand and prices become more stable and predictable. Some examples of this are households that use home-batteries for energy trading and startups like Habitat Energy, Harmony Energy and Lens.

1.2.2 Markets

The energy market consists of multiple submarkets, each having their own timescale and serving different goals.

Capacity Market — In the Capacity Market, retail companies can buy generation capacity for some years into the future. This market guarantees that there is enough supply and helps to prevent an energy shortage, and thus blackouts. When a company sells capacity, it must have this capacity available for the buyer during the contracted time. Companies are not allowed to oversell capacity.

Day-Ahead Market — In the Day-Ahead Market, the energy for tomorrow is bought from the electricity producers. All retail companies communicate their per hour[3] demand forecast and, together with the supply from the producers, an agreement on price will be made, resulting in different prices for different hours of the day ahead.

Intraday Market — The intraday has an even lower timescale granularity, often 15-minutes. Some markets have multiple timescales simultaneously (for example, 15-minute, 30-minute and 60-minute intervals). Energy can be bought or sold up until 5 minutes[4] before the start of the interval. Because more information is available on the energy demand in the near-future, energy demand forecasts are adjusted and the differences in expected total demand are traded on the Intraday Market.

Balancing Market — If a retail company has a real time shortage or surplus of energy, it can buy or sell that energy on the Balancing Market, usually under high financial losses, or gains. The goal of this market, is that any energy surplus is bought, and any energy deficit is compensated, such that the energy network remains stable.

Frequency Markets — There are also frequency control markets, which have a very low timescale from seconds to minutes. The frequency control markets keep the frequency of the electricity grid in the right bounds. For the European electricity grid, the frequency should be 50Hz. If the frequency is too low, this represents an energy shortage. Grid operators can buy additional electricity on the different frequency control markets for maintaining a stable 50Hz network. In the same way, a too high frequency represents an energy surplus. Generation should in that case be lowered directly.

1.2.3 Energy Generation

As mentioned, GCs have a portfolio of different modes of generation. Conventional generation can be divided in three parts: baseload generation, intermediate generation and peaking generation. Baseload Generation (for example Coal and Nuclear Plants) has as benefit that it is the cheapest way of producing electricity. However, it is not very flexible. Start up and shutdown costs are

[3] Timescales may differ between countries.
[4] These lead-times vary between countries and are usually becoming shorter as technology progresses.

high, and so are the starting and stopping times (called ramp up and cooldown times).
Peaking Power however, is very flexible. If additional energy must be generated in a short time or in a short notice, this generating mode is used. A mayor drawback is that this generation mode is very costly. Gas power plans are a good example of this generation mode.
In between Baseload Generation and Peaking Power, we have intermediate power. It is not the most flexible, but also not the most expensive generation mode.
Furthermore, there is Renewable Generation. But this production is often far from plannable, so the above generation modes together must perfectly match the difference between renewable generation and energy usage in the cheapest way. If the difference between renewable generation and energy usage varies greatly over time, more peaking power is used. In contrast, if the difference is more stable, a bigger part of the energy production can come from the cheaper baseload production. Hence, additionally to supply and demand, also energy usage variability and predictability plays an important role in energy prices. The more stable the difference between renewable generation and energy usage, the bigger part of the conventional energy can be generated using baseload generation. This reduces the need for peaking power and hence, lowers costs.

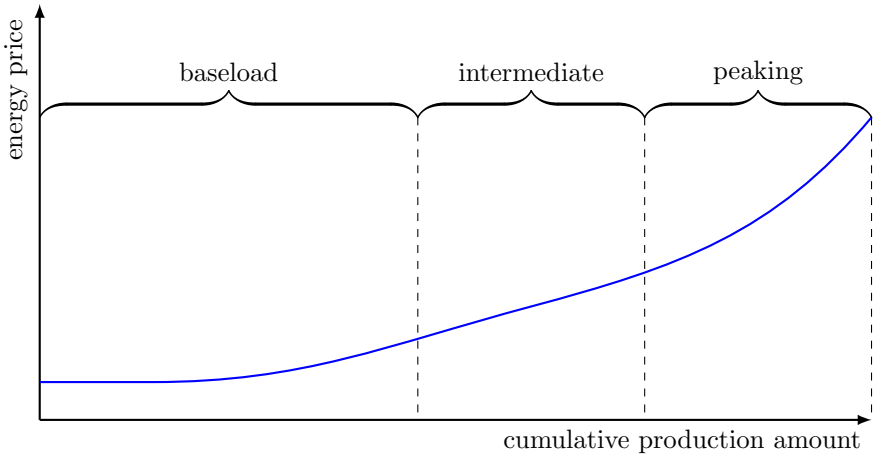


Figure 1.2: Higher production amounts corresponds to higher generation cost. In addition, peaking capacity has to be flexible to be able to quickly generate more, or less energy.

1.3 Related Literature

1.3.1 Clustering & Forecasting

Quilumba et al. [13] categorizes the forecasting horizons of energy usage in 4 parts: Very-short term forecasting up to a day, short-term forecasting up to two weeks, medium-term forecasting up to three years and long-term forecasting longer than three years. We focus on short-term forecasting, which is a well-researched topic. Much research done in this field is based on machine learning techniques.

For instance, Nizami et al. [14] used a neural network to forecast monthly energy usage with a horizon of one year for the total of Saudi-Arabia. The neural network was a very small feed-forward neural network, with two layers of four neurons. The input parameters they used were temperature, humidity, solar radiation and population size, showing that weather is an important predictor of energy usage. They found that this approach performed much better than the model of their previous work[15], in which they used a regression technique based on the same inputs. Furthermore, they argued that the neural network approach is easier to create, since no information about the relationships between the input variables and energy demand has to be modelled.

While the first papers modelled top-level energy consumption, the paper by Quilumba et al. [13],

written in 2014, performed forecasting on the consumer-level. Powered by the introduction of smart meters (resulting in data about power consumption profiles of lower levels) they showed that top-level forecasts can be made more accurate when based on consumer-level data. Additionally, they have been one of the first to show that clustering consumers, based on usage similarity, increases forecasting performance. They decided on the number of clusters based on resulting predictive performance, which was the main objective.
Around the same time, Lavin et al. [16] also used a clustering approach, based on the electric energy usage, but now on 1000 commercial businesses. They found out that basing this clustering on euclidean distance worked adequately. The amount of clusters was decided by visual inspection, and they mention that determining it more automatically is important for the future.
A more sophisticated way of clustering was introduced by Bañales et al. [17], using a dataset of 5000 households. Clustering was not performed based on the full timeseries of smart meter data, but was based on *representative days* that were mined from each consumer by clustering all its daily load profiles, resulting in two stages of clustering. Also, using only a consumer's representative days instead of the full timeseries for clustering functioned as a feature reduction method. They concluded that this technique also improved top-level forecast accuracy.

The use of clustering to improve forecasting is not unique to energy demand forecasting. Lokmic et al. [18] have made a forecasting mechanism to predict the date on which a financial check will be presented, to analyze future cashflow. They used clustering on checks to get multiple homogeneous groups before they used neural networks to create forecasts.
Closer to our field, Feng et al. [19] have used clustering on forecasting short term solar power yield, but not on clustering similar objects, since they only investigated one power plant. They used clustering based on similar timeframes, which where then labelled accordingly, for training a forecaster. This different type of clustering similar times of one timeseries is also used in the financial world by Oh et al. [20] for predicting interest rates. They use a change-point detection algorithm to divide the timeseries into intervals. These groups of intervals are then used for training different forecasting models.

The short-term forecasting method itself has also seen improvements over time. Mocanu et al. [21] proposed a factored conditional restricted Boltzmann machine, a deep learning method, for single-meter load forecasting and showed that it outperforms shallow (recurrent) neural networks and support vector machines. They argued that deep learning enabled learning higher levels of abstraction. This was also showed by Muzaffar et al. [22], who used a long-short term memory (LSTM) neural network, which outperformed AutoRegression Moving Average (ARMA) based models. Also, Kong et al. [23] successfully used the LSTM architecture, argued that LSTM networks are able to learn subtle consumption patterns of households and showed that it outperforms conventional neural networks and k-nearest neighbor approaches. Furthermore, they show that forecasting on consumer-level and then aggregating for the top-level is more accurate than to directly forecast the top-level energy consumption, just like Quilumba et al. [13] and Bañales et al. [17] have concluded. Outside load forecasting, LSTM networks were also successfully used in forecasting traffic [24], stock market indices [25], air pollution levels [26] and flooding [27]

1.3.2 Uncertainty and Peak Shifting

In the area of peak shifting by batteries, we start with two simple battery dispatching strategies called OFFON and RealTime (RT), introduced by Nottrott et al. [4]. The OFFON strategy simply charges at full speed and to full capacity at a predefined off-peak charging time, and discharges at full speed at peak-time, regardless of the energy demand. In RT, the battery is charged in the same way, but the discharging speed follows the demand, but as the battery capacity is limited and too low, the battery will be empty too quickly. This only removes the first part of the peak, and thus, has only minor reduction in maximum peak value. They mentioned that the OFFON strategy is robust and suboptimal, as demand does not change the discharging schedule and thus, the battery will never run empty sooner than planned. In contrast, RT is far from robust, but if the demand peaks are small, RT is optimal. Nottrott et al. [4] propose the OPT strategy, which is a rolling-horizon Linear Programming optimization strategy using demand forecasting. They

show that using the forecasts, and rescheduling upon forecast refinement, makes it possible for the battery discharging speed to closely follow the shape of the demand during peak usage, and hence giving the greatest reduction in peak demand value.

Rahimi et al. [28] also proposed to use a moving horizon approach. Their approach however, does not use Linear Programming and claims to be simpler. They use a (forecasted) rolling average of the energy demand, and plan the battery in such a way that the resulting demand follows the rolling average. Since the rolling average is not flat, the resulting demand will also not be flat. Hence, this approach is not optimal for reducing the maximum peak value.
Furthermore, forecast errors still make battery dispatching strategies suboptimal. Hanna et al. [29], also using rolling windows in their approach, showed that more accurate forecasting leads to more optimal plannings. Also, they concluded that solar panels alone could not alleviate the high demand peaks, as solar panel production is instable.

Using a battery for peak shifting can result in a financial profit. Usually, energy prices during peak hours are much higher than during valley hours. The paper by Li et al. [30] uses this aspect in their objective function. They want to maximize the profit from charging and discharging their batteries, together with increasing voltage stability. They use peak and valley thresholds as hard constraints. For calculating these, they use a load and PV forecast. Also, they decide on the trade-off between profit and voltage stability. They test this on 4 batteries in the IEEE 33-node system, a digital and artificial microgrid that is also used in other papers, ranging back to 1989. They use a time-granularity of 15-minutes and solve per day. A robustness survey on the performance when testing the system on actual data and optimizing on forecasted data is not done.

In the paper by Brouwer et al. [31], a much closer examination of forecasting performance is done, to make more robust battery dispatching schedules. Again, a rolling-horizon approach is used to solve the peak-shifting problem for a week on a time granularity of 15 minutes. To create robust schedules, they calculated the average error and error correlation (between two timesteps) of a forecasting method to create hundreds of testing scenarios to put into a Linear Programming formulation. They saw that accurate forecast is important up until 6 hours into the future. After that, big changes between forecasted values and actual values are not significantly affecting performance.

Hanif et al. [32] have developed a framework that can be used for different timescales of operation. In this way it is possible to exploit energy price arbitrage, and at the same time, the frequency can be stabilized. Exploiting energy usage and peak shifting result in somewhat similar battery schedules. In peak shifting, the demand during the peak is lowered by moving it into times that have a lower demand by using a battery. While using energy arbitrage, a battery is charged when energy is cheap, and discharged when energy is more expensive. As demand peaks tend to coincide with higher energy prices, these problems have similar outcomes.
Furthermore, Hanif et al. [32] propose robust optimization strategies to deal with errors in forecasting the energy market prices as well as demands. They introduce an uncertainty budget, as robust optimization can be too conservative. Distributions of the forecasted variables are not calculated or used.
Finally, and most close to our technical optimization approach, Johnson et al. [33] give a simple algorithm, based on algebra, for minimizing the maximum demand and prove optimality in the offline case.

1.4 Research Objectives

In this thesis, we create a forecasting method for up to 24 hours ahead, per hour. In contrast to the related papers, we do not give a point-forecast, but give probability distributions for the energy demand, as commonly done for weather forecasting. This is relatively new for energy consumption forecasting.
The forecasting approach will use a clustering mechanism, like Quilumba et al. [13], Lavin et al. [16] and Bañales et al. [17]. For each cluster, the forecasting model is trained separately. What

we add here, is a clustering procedure that we call ReClustering, in which clusters are iteratively improved based on forecasting performance. To our best knowledge, this is a new approach. The initial clustering will be based on the euclidean distance between some aggregated features. With this, we want to answer the question "How can we accurately create a load forecasting algorithm, and how can clustering improve results?". Different numbers of clusters will be tested, since the best clustering might not result in the best forecasting.
We do not invest time in researching how we can use consumer location. We do not test what spatial interpolation of weather stations works best. With this in mind, we have selected the Rotterdam area. The resulting smaller differences in weather between consumers is assumed to not be of interest. This is something that we leave for future research.

Furthermore, a robust optimization method for peak shaving on neighborhood level using batteries will be introduced. Here, having the forecasted load distributions (instead of a point-forecast) is crucial, since only then we can make claims about reliability of the optimization, and more importantly, know when the forecasting method is reliable and when to act more conservative. Having this distribution forecast, and using it in robust peak shifting, is something we introduce to the field and is a main strength of this research.
Peak shaving is done on the sum of energy consumption of a collection of consumers. We want to reduce the morning and evening peaks as much as possible by discharging the battery at those times. Batteries are charged when the net demand is lowest. This is achieved by minimizing the maximum demand and maximizing the minimum demand, something that is also called *bandwidth reduction*.
Since the forecast (E.g., the forecast of 9:00 tomorrow) will be increasingly better as time progresses (and 9:00 comes closer), the optimization scheme will use a rolling horizon approach, where we recalculate the forecasted values to modify the battery (dis)charging plans. Here, the central research question is "How can we use an energy demand distribution forecast in a reliable optimization algorithm for short-term load shifting?".
The way this will be set up, allows us to test different battery and subgrid sizes. By this, we want to learn at what scale (where in the hierarchy) of the distribution grid, peak shifting on consumer demand is most effective, and how predicted distributions can aid battery planning.

1.5 Data Overview

1.5.1 Energy Usage Data

We have acquired energy usage data from 1772 consumers in the Rotterdam area. This data is from November 2020 until November 2022, and has a time granularity of 15 minutes. The timezone is Universal Coordinated Time (UTC) and it is kept in that timezone. Otherwise, a choice must be made about if we will be using daylight saving time (DST) or not, and switching from and to DST has as burden that once per year, we skip an hour, and once per year, there is a duplicated hour. Dealing with the resulting artifacts is not worth having localized notation.
In the usage data, we suffer from missing values[5], stretching for multiple days. If a consumer has more than 7 consecutive days of missing data, we remove this consumer. We fill in other missing data by interpolation: we take the average demand of this consumer of 7 days ago and 7 days into the future. If the data for 7 days into the future or 7 days into the past is unknown, we take the value that is present as new value. If they are both missing, we exclude the consumer from the dataset.
The resulting dataset consists of 804 consumers in which 1.4% of the total amount of data has been interpolated. Of course, there are more precise methods of interpolation, such as using a nearest neighbor approach or using representative days, but as this isn't the focus of this thesis, we have chosen for this simple interpolation method. See Figure 1.3 for an example of filling in missing days.
In September 2021, an unusual spike in energy production has been spotted exactly at 00:00 local time, on different days for different consumers and not for all consumers. Assuming that this is

[5] This is a known issue.

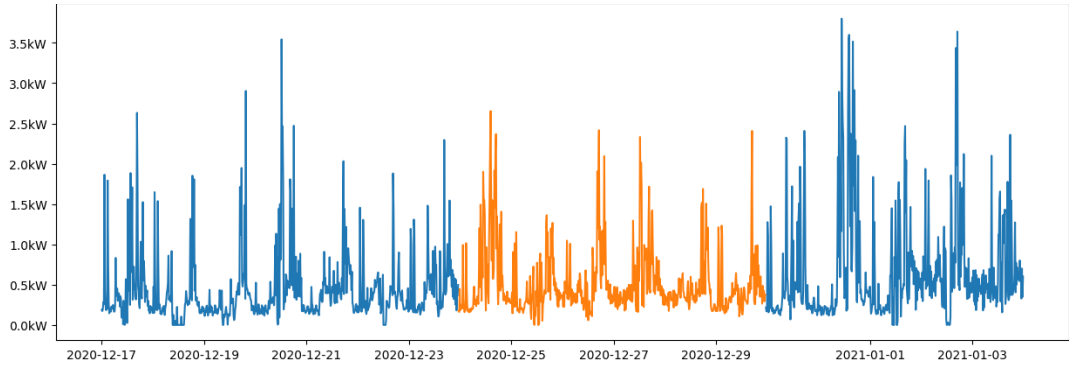


Figure 1.3: Example of interpolation for one consumer. The original data is in blue, and has 6 missing days from December 24th up until December 30th. In this timespan, the interpolated values are in orange.

due to an event like a software-update, we decided to find the biggest spikes and remove them. If the energy consumption at midnight is more than 3 times bigger than the sum of the energy consumption 15 minutes before and the energy consumption 15 minutes after midnight, we replace the value at midnight by the average of the energy consumption 15 minutes before and after midnight. This was done for 78 consumers.
Furthermore, consumption timeseries from 136 smart meters have been excluded because of the unexplainable behavior these meters (see Figure 1.4), or because PV production was mixed into the usage data. By these preprocessing steps, we end up with a dataset with 668 consumption timeseries.

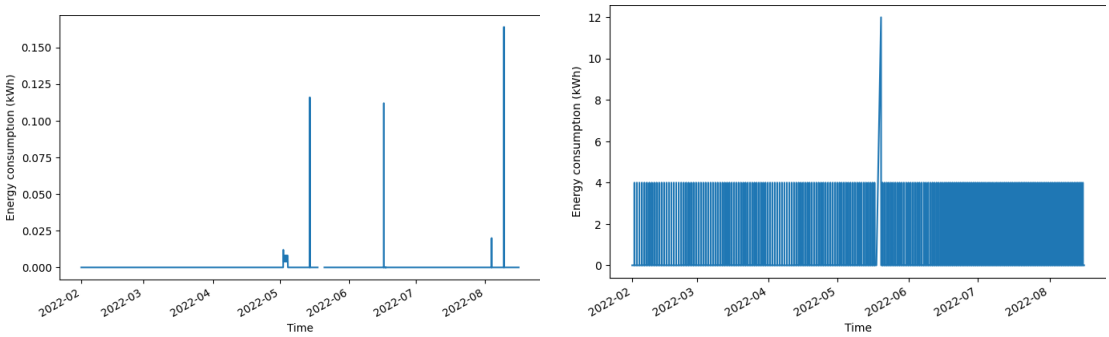


Figure 1.4: Two examples of the data of two smart meters that have been excluded from the dataset.

Since we have slightly less than two years of usage data, we will only use data from 2021 as training data. In this way, we prevent introducing an unwanted bias due to over-representation of one or more seasons of the year. Data from January 2022 up to and including May 2022 is used as validation data, and June 2022, up to November 12 is used for testing (Also see Figure 1.9).
Furthermore, we will use 500 consumers for our training dataset, and use the remaining 168 consumers for additional testing and replacing if needed.

1.5.2 Weather Data

From the Royal Dutch Meteorological Institute, or *Koninklijk Nederlands Meteorologisch Instituut* (KNMI), we have obtained hourly historical forecast data for the centerpoint of the Rotterdam area, also starting from November 2020. The data originates from the Harmonie model, which is run every 6 hours and forecasts 48 hours into the future. The Harmonie model is ran multiple times on slightly different inputs. Then, these multiple predictions are used as an ensemble, calculating the mean and standard deviations for the final prediction.

We have access to 10 runs per day, all initiated on 00:00 UTC. From these 10 runs, we also calculate the mean and standard deviation, resulting in 14 columns, for means and standard deviations of pressure, temperature, relative humidity, total cloud cover, visibility, wind speed and wind direction.

Also in this weather data, there are missing days. To fill in the gaps, a linear interpolation method is used, based on time of the day. For example, while calculating values for 8:00 AM on September 25th, we use the average of the values at 8:00 AM on September 24th and at 8:00AM on September 26th. If multiple consecutive days are missing we use the enclosing days that are present. For instance, if the 7th and 8th of November are missing, we use the 6th and 9th of November for interpolation.

In addition to the historical forecasting data, we also have gathered historical weather data, observed at the weather station of Rotterdam operated by KNMI. This data is freely available at B

1.6 Exploratory Analysis

In this section, an exploratory analysis of the energy demand is done to find some characteristics that are useful to know for developing the forecasting methods.

Firstly, the relative consumption per consumer is calculated by dividing the timeseries by their average values. Then, we take the average over all consumers. The resulting plots are visible in Figure 1.5 for energy consumption.
What draws the attention in the consumption plot, is the effect of DST visible by the "shift" of 1 hour from April to October. Also, it is clear that energy consumption is lowest during the night, increases when people wake up, a bit lower during the day and high around dinner time (16:00 during the summer, 17:00 during the winter). Furthermore, during the winter, more energy is consumed.

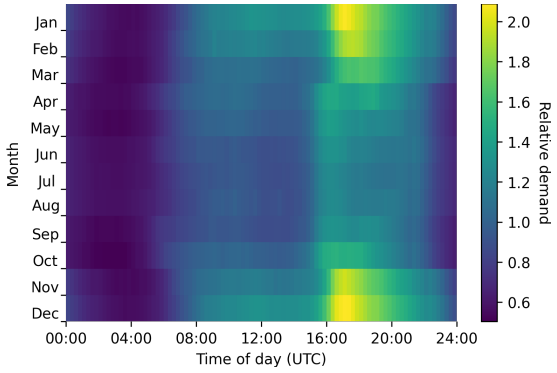


Figure 1.5: Average relative consumption over all consumers over different times and months ($n = 806$).

Taking days of the week separately (see Figure 1.6), we see again that dinner-time forms a time of high demand. Also, during the weekend, more energy is used, except during the early-morning. Two demand bumps can be seen around 21:00 and 22:00 UTC. Further investigation showed that these bumps are both from 23:00 local time, exactly when the off-peak time, with lower energy costs, starts. This also shows the effect of energy prices on consumers.
In Figure 1.7, the same plot is made, but now for only one individual household. Some behavioral observations can be made. It is visible that on Friday evenings and in the late afternoon on Saturday, the energy consumption is relatively high. One of the reasons for this could be that this

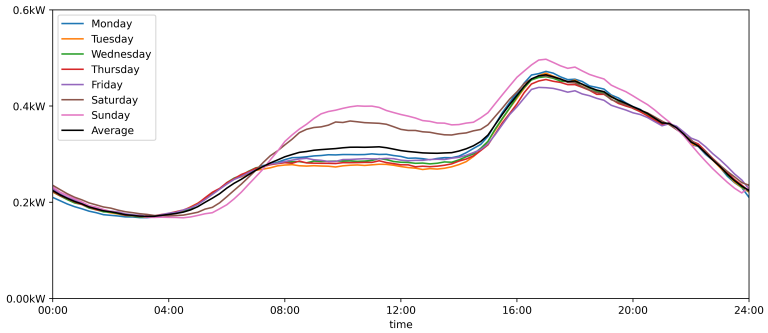


Figure 1.6: Average relative consumption over all consumers ($n = 805$).

consumer is using an energy consuming PC at that times. Of course, this might not be the case, but this is a showcase of the important role of consumer behavior in energy consumption.
Furthermore, household characteristics are important. In Figure 1.8, we have the total energy demand per month for two households. We see that the energy consumption in the colder winter months of Household II is relatively higher than that of Household I, which might be because Household I has a much better isolation than Household II. Here again, the reason for this difference between households is not known, however, these observations argue for treating different households differently in the forecasting mechanism. This makes a strong point for the use of clustering.

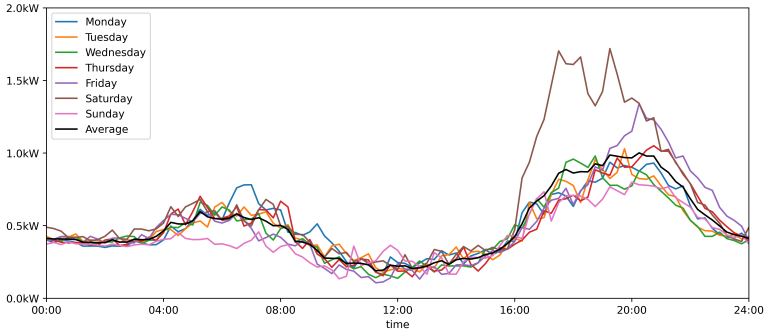


Figure 1.7: Average consumption for one consumer over different weekdays of 2021, including the average of the whole week.

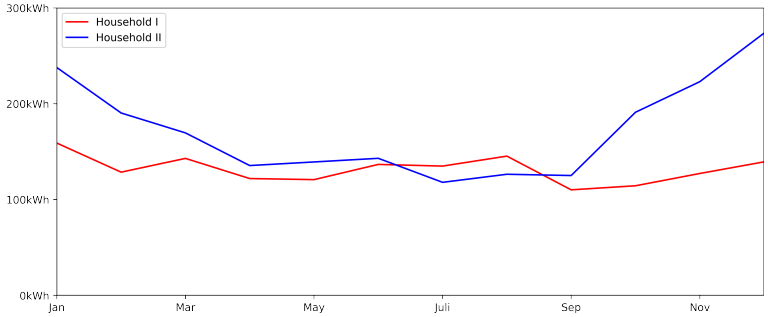


Figure 1.8: Total energy consumption per month over 2021 for two different households.

We fitted a Linear Regression model (from the `sklearn` python library) to the energy consumption data and the historical weather station data. This is done for every consumer in the training dataset, and for every 2-hour interval (so for data from 00:00 to 02:00, 02:00 to 04:00, etc.) separately. The average of the absolute coefficients from the model are given in Table 1.1. From this

table, we can interpret which weather features look most promising to use as input features for the forecasting model, and which do not. For instance, the predictive value of Wind Gust Speed is very low if also the Wind Speed itself is used. Of course, this Linear Regression can only use linear effects of the weather variables, but we can be quite sure that Air Pressure, Wind Gust Speed, Rain, Snow, Factor Sunshine and Clouds are of lesser importance. Temperature, Dewpoint Temperature, Solar Radiation and Relative Humidity seem to be important predictive features.

Weather Variable	Consumption
Wind Speed	0.178
Wind Direction (Sine)	0.040
Wind Direction (Cosine)	0.037
Temperature	0.881
Dewpoint Temperature	0.856
Fraction Sunshine	0.044
Solar Radiation	0.078
Clouds	0.032
Precipitation	0.100
Relative Humidity	0.495
Air Pressure	0.059
Wind Gust Speed	0.187
Horizontal Visibility	0.082
Rain	0.033
Snow	0.030
Thunder	0.045

Table 1.1: Mean Absolute Linear Regression Coefficients for the model where weather variables are the input, and consumption is the output. Before fitting, all the values are normalized linearly to the range between 0 and 1.

Finally, a post-hoc analysis of the differences between different training sets was done on energy usage and the two of the most important input features present in the Weather Forecast data: Temperature and Relative Humidity (see Figure 1.9). Not all weather characteristics in the validation and testing datasets have been previously seen in the training dataset, which might reduce forecasting performance. Temperatures in 2022 were much higher than in 2021, especially during the testing set. Furthermore, relative humidity has been much lower during the validation and testing set than during the training set. Also, energy consumption in 2022 is much lower than in 2021, possibly caused by the higher temperature and lower humidity.

Figure 1.9: Total daily usage, minimum and maximum daily temperature and average daily relative humidity for the training data (red), validation data (green) and testing data (blue).

Chapter 2

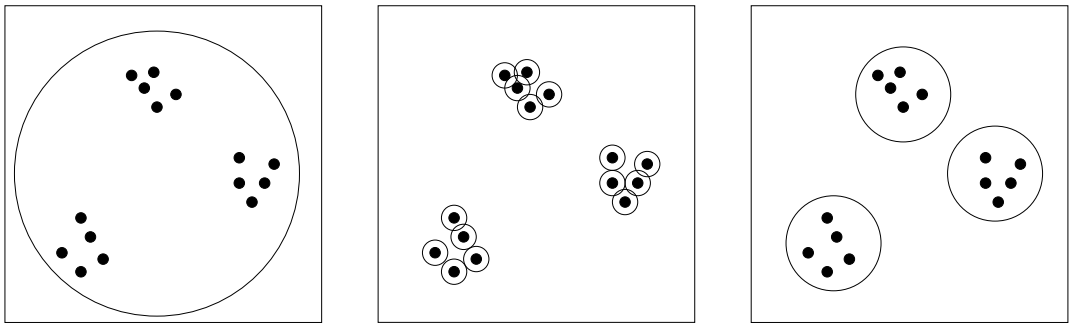
Demand Forecasting

2.1 Methods

Our forecasting mechanism consists of multiple steps. Firstly, consumers are clustered into groups that have similar energy consumption. Then, for each consumer cluster, we train a forecaster based on a Long-Short Term Memory (LSTM) Neural Network. These are compared to a Linear Regression approach, and a simple baseline forecaster. Lastly, we improve performance by ReClustering, in which we cluster based on forecasting performance, instead of similarity. This section explains the technical details of these techniques.

2.1.1 Consumer Clustering

Then forecasting on multiple entities, there are two extremes. One is to train for every entity a different model, which will be computationally expensive. The other extreme is to create one single big model that is trained on all entities. However, underlying processes that result in a future value might be too dificult to be all grasped in one model.
In the middle lies using clusters. For every cluster, a separate model is trained for forecasting. Bigger clusters might suffer from the fact that they must be general and precise at the same time. On the other hand, using smaller clusters, we might not have enough data to train the model effectively. When clustering consumers based on energy demand, data is sometimes *weather normalized*. In weather-normalization, it is tried to filter out the effect of weather. We do not do this, since the differences between weather of different consumers, all located in the area of Rotterdam, are minor. If in future research a dataset is used where the consumers are located farther away from each other, weather will be more diverse and is thought to have a bigger influence on the resulting clusters.



(a) General trained forecaster. (b) Individually trained forecaster. (c) Per-cluster trained forecaster.

Figure 2.1: Different forecasting training levels.

We use the K-Means algorithm on Euclidean distance for the clustering. Before this is done, three methods of dimension reduction are used. For each household separately, we calculate the following feature types.

- Representative days, to capture daily effects.
- Average weekly profile, to capture weekly behavior.
- Total monthly consumption, to capture seasonal behavior.

Before calculating all these datapoints, household timeseries are normalized such that their average demand is exactly 1. Now, we no longer refer to household demand, but we refer to relative household demand.

Representative day clustering

To calculate representative days, the demand timeseries of a household is separated into timeseries spanning for only one day. Then, these timeseries are again normalized such that every day has the same total consumption (by dividing the timeseries by the mean consumption during that day), after which K-Means clustering is executed on the timeseries. The resulting clusters of days are converted to representative days by calculating the average profile of all days in the cluster.

Average weekly profile

Secondly, the average weekly profiles are calculated. To do this, we separate the timeseries of a household into multiple timeseries, one for each week. The granularity is reduced to 3 hours after which the timeseries are normalized so that the sum of consumption is the same in every week. Then, the average weekly profile is calculated by taking the average over the values over all weeks to capture weekly effects.

Total monthly consumption

Household energy demand will also be driven by seasonal effects. Some households will have a bigger relative demand in the winter than other households. This can be caused by differences in ways houses are isolated, for example. See Figure 1.8 for an example on this effect.
To capture these effects, the household timeseries are separated into months. Then, the sum over all months separately is calculated for the last feature type.

Combining feature types

It might be possible that one feature type is more important than other feature types. To test this, each feature type is given a weight. One for representative days, one for weekly profiles and one for total monthly consumptions. The higher the weight, the more important the features are for the K-Means clustering.

We will test 66 combinations for these three weights. These combinations are formed by dividing 10 parts of 10% importance between the three different feature types.

We use the "Average In-Cluster Distance" as performance metric. It is calculated as

$$\text{Average In-Cluster Distance}(C) = \frac{1}{\sum_i |C_i|(|C_i|-1)} \sum_i \sum_{a,b \in C_i, a \neq b} \|a-b\|.$$

Here, C_i are the clusters of clustering C, and $\|a-b\|$ is the Euclidean distance between the timeseries demands of consumer a and b.

2.1.2 Forecasting Models

Multistep Forecasting

The forecasting method is then performed per cluster. For the forecasting, we use the original timeseries of all customers in a cluster. We will forecast multiple 60-minute timesteps into the future using a *Multi-Step Forecasting approach*. Taieb et al. [34] describe 3 main ways to do this[1]:

- *Direct Multi-step Forecasting*: In a Direct Multi-step Forecasting strategy, a separate model is trained for each horizon. Therefore, it is sometimes called the Independent strategy. In our case, this means that we would have to train and tune 24 different independent models, which does not seem feasible. It has as an advantage that the forecast of the future is based only on information that is known in the present, and does not need to use intermediate forecasts.
  This approach can be summarized as
  $$f_{t+\Delta t} = M_{\Delta t}(o_t, o_{t-1}, o_{t-2}, \dots),$$
  where $f_{t,t+\Delta t}$ is the forecast done at timestep t, for timestep $t+\Delta t$, $M_{\Delta t}$ is one of the trained models and o_t are vectors of the observed input variables at timestep t.

- *Recursive Multi-step Forecasting*: This is different in Recursive Multi-step Forecasting strategy. Here, only one model is created. So to be able to create a forecast of tomorrow (timestep 96), the preceding values have to be known. This is solved by recursively forecasting values, and use these values for the forecast of the next value. If any additional timeseries are used only as input (we for instance will have temperature as an input timeseries), these can be used in this approach.
  The Recursive Multi-step Forecasting approach is summarized as
  $$f_{t+\Delta t} = M(f_{t+\Delta t-1}, \dots, f_{t+1}, o_t, o_{t-1}, \dots),$$
  where we have the same notation as in the Direct Multi-step Forecasting approach.

- *Multiple Output Forecasting*: In the Multiple Output Forecasting strategy, the trained model has the different forecasts directly in one output vector. Now, the forecasted values, up to T timesteps into the future, are calculated as
  $$(f_{t+1}, f_{t+2}, \dots, f_{t+T}) = M(o_t, o_{t-1}, o_{t-2}, \dots).$$

As the Multiple Output Forecasting is the most straightforward to implement, and computationally the least expensive, this type is chosen. It does not have the computational complexity of Direct Multi-step Forecasting, and it is not needed to work on a way to use pas outputs as inputs.

Forecasting Models

We will use the following three different forecasting models.

Shift: As forecast, we simply use a past value. In other words, this model assumes that energy consumption is periodic. For a forecast for multiple hour into the future, we can take the value of 24 hours, 48 hours and 7 days before as predicted demand. For example, for the forecast of today at 14:00, the observed energy demand of yesterday at 14:00 will be used. It will be tested what time shift works best.

LR: A Linear Regression (LR) Forecasting algorithm. It assumes that the values from the predictive timeseries we use, are linearly related to the energy demand. This makes this model intuitive and simple, but it might not be the best performing method.

LSTM: As our most important and promising approach, we will use a Long-Short Term Memory (LSTM) Neural Network. This kind of neural network is able to capture trends, which is important for demand forecasting. LSTMs are assumed to be bad at learning long-term trends. To solve this, a modification to LSTMs could be made. However, due to time constraints, this will be a point for future work.

[1] The other two approaches they describe, are combinations of the former

Long-Short Term Memory (LSTM) Model

Our focus is on the *Long-Short Term Memory* approach. LSTM finds its origins in the Recurrent Neural Network (RNN) structure. This is a Neural Network in which previous intermediate values (outputs of individual neurons) are stored and used again as input to the neuron itself (see Figure 2.2 for a single-step schematic). In this way, previous computations on values from the past are used in the next computations. Each time input variables are fed into an RNN, the RNN gives the forecasted value as output and remembers some intermediate values for the time the RNN is used for the next forecast.
Remember that basic feed forward neurons calculate their output $y_t \in \mathbb{R}$ as

$$y_t = g(\mathbf{w}\mathbf{x} + b),$$

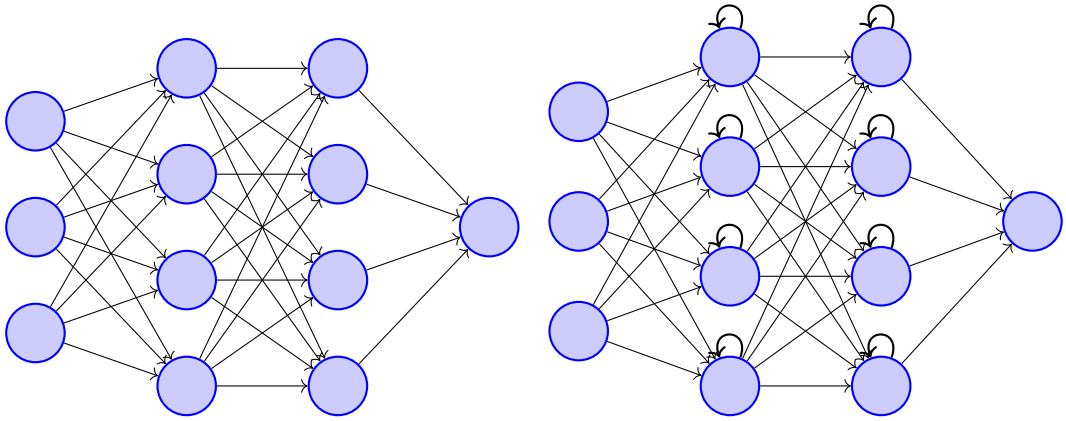
where, $\mathbf{x} \in \mathbb{R}^I$ is the vector of I inputs and $w \in \mathbb{R}^I$ is the vector of corresponding weights, with identical size, for the neuron and $b \in \mathbb{R}$ is the bias parameter. The activation function is denoted with g, for which often the sigmoid function is chosen.

For an RNN cell, we have the following two equations.

$$\begin{aligned} \boldsymbol{h}_t &= \text{sigmoid}(\boldsymbol{b} + W\boldsymbol{h}_{t-1} + V\boldsymbol{x}_t), \\ \boldsymbol{y}_t &= \tanh(U\boldsymbol{h}_t + \boldsymbol{c}). \end{aligned}$$

Here, $\boldsymbol{x}^t \in \mathbb{R}^I$ is the input vector of I input features at timestep t and $\boldsymbol{y}_t \in \mathbb{R}^O$ the output vector of size O. Additionally, this neuron has a *hidden state* vector of size H, denoted by $\boldsymbol{h}_t \in \mathbb{R}^H$. This state is used in the next timestep as input. The matrices $U \in \mathbb{R}^{H\times O}$, $V \in \mathbb{R}^{H\times I}$, and $W \in \mathbb{R}^{H\times H}$ are weight matrices, and $\boldsymbol{b} \in \mathbb{R}^H$ and $\boldsymbol{c} \in \mathbb{R}^O$ are bias vectors. The activation functions sigmoid and tanh are calculated component-wise. These not need to be sigmoid and tanh, also other activation functions can be used.

Over long timeseries, RNN approaches have been found to often suffer from the vanishing and exploding gradient problems, making learning temporal dependencies hard. The LSTM architecture alleviates this problem[35, 36], allowing to learn long-term effects.



(a) Feed Forward Neural Network. (b) Recurrent Neural Network.

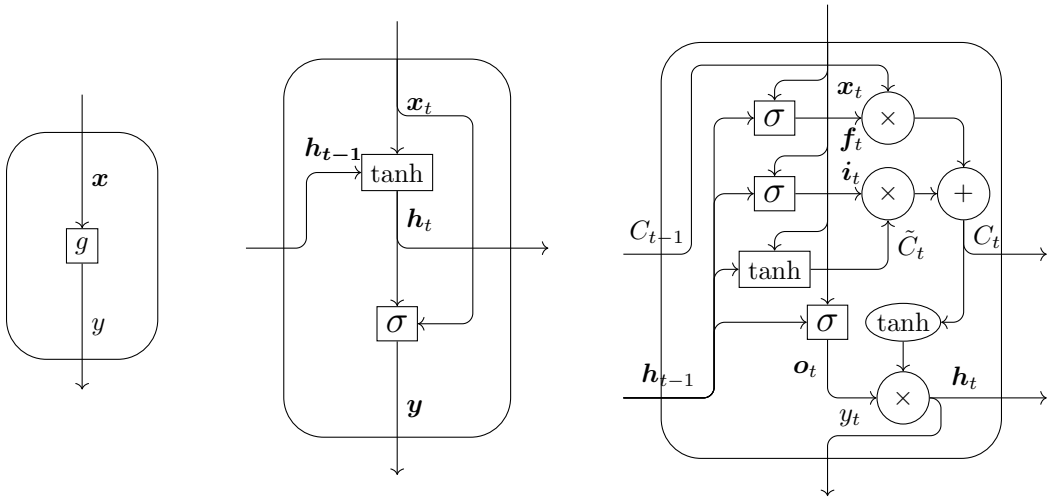
Figure 2.2: Schematic overview of a Feedforward Neural Network and a Recurrent neural network.

The LSTM has two state components. A *hidden state*, denoted by $\boldsymbol{h}_t \in \mathbb{R}^H$ that represents short-term memory, and the *internal state* (sometimes called the *cell state*) of size S, denoted by $C_t \in \mathbb{S}$, representing long-term memory.

The computations inside a LSTM cell are as follows.

$$
\begin{aligned}
\boldsymbol{i}_t &= \text{sigmoid}(W_i \boldsymbol{h}_{t-1} + V_i \boldsymbol{x}_t + \boldsymbol{b}_i), \\
\boldsymbol{o}_t &= \text{sigmoid}(W_o \boldsymbol{h}_{t-1} + V_o \boldsymbol{x}_t + \boldsymbol{b}_o), \\
\boldsymbol{f}_t &= \text{sigmoid}(W_f \boldsymbol{h}_{t-1} + V_f \boldsymbol{x}_t + \boldsymbol{b}_f), \\
\tilde{C}_t &= \tanh(W_c \boldsymbol{h}_{t-1} + V_c \boldsymbol{x}_t + \boldsymbol{b}_c), \\
C_t &= \boldsymbol{i}_t \odot \tilde{C}_t + \boldsymbol{f}_t \odot C_{t-1}, \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(C_t), \\
\boldsymbol{y}_t &= \boldsymbol{h}_t.
\end{aligned}
$$

Here, $\odot$ denotes the element-wise multiplication operator, and the activation functions sigmoid and tanh are computed element-wise on their input. Furthermore, $\boldsymbol{i}_t$, $\boldsymbol{o}_t$ and $\boldsymbol{f}_t \in \mathbb{R}^H$ denote input, output and forget gates respectively. These are all calculated based on the input $\boldsymbol{x}_t$, the hidden state from the last timestep $\boldsymbol{h}_{t-1}$, gate-specific weights W_i, W_o, $W_f \in \mathbb{R}^{H \times H}$, and V_i, V_o, $V_f \in \mathbb{R}^{H \times I}$ and biases $\boldsymbol{b}_i$, $\boldsymbol{b}_o$, $\boldsymbol{b}_f \in \mathbb{R}^H$. Then, the potential new cell state $\tilde{C}_t \in \mathbb{R}^H$ is calculated, also based on the input, hidden state and weight vectors $W_c \in \mathbb{R}^{H \times H}$ and $V_c \in \mathbb{R}^{H \times I}$ and a bias vector $\boldsymbol{b}_c \in \mathbb{R}^H$. This potential new cell state is put through the input gate and added to the old cell state, which is firstly put to the forget gate. Now, we have the new cell state $C_t \in \mathbb{R}^H$. This new cell state is put through the output gate to calculate the output and new hidden state. See Figure 2.3 for a schematic visualization of the "normal" Feed-Forward NN cell, an RNN cell and a LSTM cell.



(a) FFNN Cell. (b) RNN Cell. (c) LSTM Cell.

Figure 2.3: Schematic visualization of the "Normal" Feed-Forward NN, Recurrent NN and LSTM cells.

2.1.3 Feature selection

Before creating a LSTM model, a correlation study will be done to decide what weather features to use. At daytime, for example, temperature and energy usage are both generally higher. This is a correlation that is thought to have no causality, and thus no predictive power. To remove these effects, we will calculate Time-Normalized correlations.

In addition to the weather features, the following features are added to this single-consumer model.

Hour of Day	Between 0 and 23.
Day of Week	Between 0 and 6, where the week starts on Monday.

Month	Month number starting at 0, between 0 and 11.
Public holiday	National holidays in the Netherlands. The Public holiday variable equals 1 on national holidays, and 0 otherwise.
Height of Sun	This feature represents the vertical angle of the sun in Rotterdam, throughout time.
Recent Consumption	In the recent consumption, we have the consumption of the consumer up to 24h into the past.

Each selected feature is translated to a timeseries as input for the LSTM, where we use the data from 24 hours into the past and 24 hours into the future. For the recent consumption, no data from the future is used.

Output and Loss Function

As we are not only interested in the expected energy demand, but also want to know how certain we are, we give as output a distribution forecast. To decide what kind of distribution is used, a first small model was created to check what kind of prediction errors are made (See Figure A.1 for an example). There seems to be a highly stochastic aspect of when these peaks occur, and what size these peaks have. This problem has also come up in related literature about short-term forecasting on household level. Haben et al. [37] solves this by changing the loss function such that a missed peak has a lower penalty when, close to that peak, the forecast did predict a spike in energy consumption. A different way of solving this is proposed by Herraiz-Cañete et al. [38]. They use information about the electric appliances that a household has, and based on that, builds coincidence behavior of these appliances. This information, however, has a high privacy sensitivity, and is not available in our case. Instead, we translate the model to a stochastic model that gives

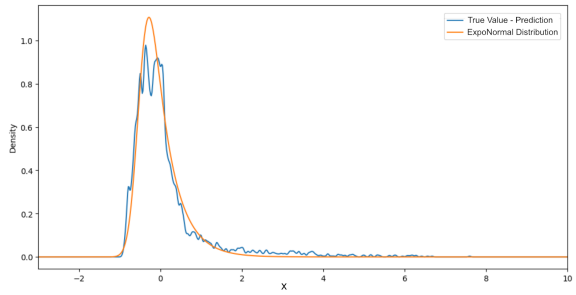


Figure 2.4: Prediction error and a loosely fitted EMG Distribution.

24 distributions of energy consumption as output, one for each hour in the forecasting horizon. For this, the *Exponentially Modified Gaussian* (EMG) distribution is used, as it has visually an acceptably good fit to the prediction error (See Figure 2.4). We say that $\text{EMG}_{\mu,\sigma,\lambda}$ has an EMG Distribution with parameters $\mu \in \mathbb{R}$, $\sigma \in \mathbb{R}_{\geq 0}$ and $\lambda \in \mathbb{R}_{>0}$, if

$$\text{EMG}_{\mu,\sigma,\lambda} = N_{\mu,\sigma} + E_\lambda, \text{ where } N_{\mu,\sigma} \sim \mathcal{N}(\mu, \sigma^2), \text{ and } E_\lambda \sim \text{Exp}(\lambda),$$

where $N_{\mu,\sigma}$ and E_λ are independent.

This makes it intuitively a good fit; The Exponential Distribution is able to describe the peak values, and the Gaussian Distribution can describe the non-peak values. Lastly, the log-likelihood loss Score can be easily approximated using easily, which is a requirement for learning efficiently.

The distribution forecast can be translated to a point forecast by taking the expected value of the EMG Distribution, given by

$$\mathbb{E}(\text{EMG}_{\mu,\sigma,\lambda}) = \mathbb{E}(N_{\mu,\sigma} + E_\lambda) = \mathbb{E}(N_{\mu,\sigma}) + \mathbb{E}(E_\lambda) = \mu + \frac{1}{\lambda}.$$

If we want to filter out the effect of peaks, the median of the distribution is preferred above the expected value of the distribution. We have not found a closed form expression of the median of the EMG distribution, also not in literature. Hence, the median is numerically approximated using the `scipy.stats` package, which under the hood uses Brent's method to find the value for which the cumulative density function equals 0.5. Brent's method combines tree root finding algorithms: Root bracketing, interval bisection, and inverse quadratic interpolation [39].
The resulting median will be used to calculate the SMAPE score, and the expected value will be used to calculate the NRMSE score.

As the probability density function of the EMG distribution with parameters $\mu \in \mathbb{R}$, $\sigma \in \mathbb{R}_{\geq 0}$ and $\lambda \in \mathbb{R}_{>0}$ is the convolution of the probability density functions of the Gaussian distribution with parameters μ and σ, and the Exponential distribution with parameter λ. For an observed energy usage $\mathbf{y}$, its probability density function is given by [40]

$$f(\mathbf{y}; \mu, \sigma, \lambda) = \frac{\lambda}{2} e^{\frac{\lambda}{2}(2\mu + \lambda\sigma - 2\mathbf{y})} \operatorname{erfc}\left(\frac{\mu + \lambda\sigma - \mathbf{y}}{\sqrt{2}\sigma}\right).$$

The erfc function is defined as

$$\operatorname{erfc}(z) = \frac{2}{\pi} \int_z^\infty e^{-t^2} \mathrm{d}t$$

and takes values in $[0, 2]$.

From the probability density function, the log-likelihood loss score for one predicted timestep, is calculated as

$$\ln(f(\mathbf{y}; \mu, \sigma, \lambda)) = \ln\left(\frac{\lambda}{2} e^{\frac{\lambda}{2}(2\mu + \lambda\sigma - 2\mathbf{y})} \operatorname{erfc}\left(\frac{\mu + \lambda\sigma - \mathbf{y}}{\sqrt{2}\sigma}\right)\right) \tag{2.1}$$

$$= \ln\left(\frac{\lambda}{2}\right) + \frac{\lambda}{2}(2\mu + \lambda\sigma - 2\mathbf{y}) + \ln\left(\operatorname{erfc}\left(\frac{\mu + \lambda\sigma - \mathbf{y}}{\sqrt{2}\sigma}\right)\right), \tag{2.2}$$

where we use the observed energy usage $\mathbf{y}$ and μ, σ, and λ, which come as direct output from the model. As we are taking a logarithm from the output of the erfc function, which can produce arbitrarily small values, we add a small ϵ of 10^{-20} in practice, to ensure that the logarithm can still be calculated, as the logarithm of 0 does not exist.
In this approach of using distributions, the number of outputs from the model triples from 24 (one for every hour), to 72 (three parameters for every hour).

2.1.4 Performance measures

In the field of energy demand forecasting, distribution forecasting is not standard. In all related literature (See Section 1.3), point forecasting is used. As performance measures, the Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are used most frequently. As a result, the log-likelihood loss function is not comparable with the loss functions that are used in this field. Therefore, we add variants of the MAPE and RMSE functions. Firstly, the Symmetric Mean Absolute Percentage Error (SMAPE) will be given. As the actual demands is equal to zero multiple times, the MAPE will be undefined. Secondly, we give the Normalized RMSE (NRMSE) performance. They are as follows.

$$\operatorname{MAPE}(\mathbf{y}, \mathbf{f}) = \sum_{i=0}^{n-1} \left| \frac{\mathbf{f}_i - \mathbf{y}_i}{\mathbf{y}_i} \right|,$$

$$\operatorname{SMAPE}(\mathbf{y}, \mathbf{f}) = 2 \sum_{i=0}^{n-1} \left| \frac{\mathbf{f}_i - \mathbf{y}_i}{\mathbf{f}_i + \mathbf{y}_i} \right|,$$

$$\operatorname{NRMSE}(\mathbf{y}, \mathbf{f}) = \frac{1}{\overline{\mathbf{y}}} \sqrt{\frac{\sum_{i=0}^{n-1} (\mathbf{f}_i - \mathbf{y}_i)^2}{n}}.$$

Here, $\mathbf{y}$ is a vector of n observed values with mean $\overline{\mathbf{y}}$ and $\mathbf{f}$ is the vector of the corresponding n predicted values from our forecast method.

2.1.5 Baseline models

A new baseline model is required to make fair comparisons based on log-likelihood loss, since the log-likelihood loss can't be computed for the Shift approach. We use Linear Regression (LR) as second baseline model and use the same kinds of input and outputs as for the LSTM for the model, such that it can be tested in the same way. To lower complexity, the LR approach is not given historic values and also has the output for only one hour. In this way, the LR method calculates a μ, σ and λ for the distribution for a given time t and consumer c as

$$\mu_{t,c} = \text{softplus}\left(\sum_i a_i^{\mu} o_{(i,t,c)} + b^{\mu}\right),$$

$$\sigma_{t,c} = \text{softplus}\left(\sum_i a_i^{\sigma} o_{(i,t,c)} + b^{\sigma}\right), \text{ and}$$

$$\lambda_{t,c} = \text{softplus}\left(\sum_i a_i^{\lambda} o_{(i,t,c)} + b^{\lambda}\right),$$

where a_i^{μ}, a_i^{σ}, a_i^{λ}, b^{μ}, b^{σ}, and b^{λ} are the LR parameters to be fitted. The variables $o_{(i,t,c)}$ are input datapoints for feature i, at time t and for consumer c. Many of these input datapoints are the same over the different consumer, since all consumers have the same weather and calendar data. The softplus function is used to ensure that the log-likelihood exists for all LR outputs, as calculating the log-likelihood loss is not possible for a $\sigma_{t,c}$ equal to zero. Furthermore, negative $\sigma_{t,c}$ and $\lambda_{t,c}$ do not make sense.

2.1.6 Multiple Consumers

Our goal is not to create a forecast for a single consumer. We want to have a forecast method to predict the future energy consumption for all households in the dataset. We could try to use the above method for all consumers separately, but as there are hundreds of households in our dataset, this will not be computationally feasible on the systems that we use. In practice, this problem becomes even worse, as retail companies do not have hundreds of contracted households, but have millions.

Without using clusters, the whole collection of households is used as training data for training one forecasting neural network. Technically, this means that we repeatedly and randomly pick consumers and timeslices in our training batches during training.

Since the collection of training data is very big (around 4.4 million examples) iterate over the whole dataset takes a long time. Therefore, we redefine an epoch as follows. We call the time in which the neural network sees $\frac{1}{n}$th (not necessary unique) of the number of examples in the training dataset, in which n is the number of households in the data, an epoch. If we use n epochs for training, the neural network has seen as many examples as there are in the dataset. One epoch of data has the same size as the data we have for one consumer.
Also, the LR approach does not use the complete collection of training data. We have the same definition as an epoch for this approach.

The LSTM network is now used for many households, all with a different consumption given the exact same inputs (apart from the consumption history input). This might be difficult to learn for a neural network, so we add features that makes households different, on features that are thought to have predictive power. For this, we calculate for every household the correlation between all selected weather features, and energy consumption.

A number of experiments will be done by hand to determine some hyperparameters, like batch size, number of LSTM layers and lookback time. We will test batch sizes of 16, 32, 64 and 128, one to three LSTM layers and a lookback time of one day, two days and a week. Final values will be chosen based on training log-likelihood loss, and for the batch size we choose the value which is faster for the system and results in stable convergence.

Then, Bayesian hyperparameter tuning used to more systematically determine the best values for the size for both LSTM layers independently, the dropout value between the LSTM layers, the learning speed and the number of epochs. In Table 2.1 are the bounds that are used during hyperparametertuning. The layer size bounds and number of epochs are chosen such that the computational complexity stays within reason, such that it takes no more than about two hours for the used system. The dropout and learning rate upper- and lowerbound are chosen such that, based on some tests that were done by hand, the optimal values are expected to be within bounds. For the hyperparameter tuning we will use the Gaussian process `Optimizer` from the `skopt` library. The Gaussian process has, in literature, proven to be useful for hyperparametertuning[41, 42, 43] and comes also as a popular "out of the box" option in the Machine Learning package `Keras`.

Variable	min	max
Layer size	30	80
Epochs	50	200
Dropout	0.0	0.2
Learning rate	0.0001	0.006

Table 2.1: Upper- and lowerbounds for different variables that are used during hyperparametertuning.

Separating all consumers into groups that are alike is thought to increase forecasting performance (see Section 2.1.1). Therefore, this approach is also used for forecasting on clusters. The hyperparameters that are selected by hand are kept the same, but the automated hyperparametertuning is done for each cluster separately.

2.1.7 ReClustering

The method of clustering that we have defined, with its performance measure, might not fall in line with what it means for a cluster to be easy to train for a neural network. Finding the best way to define an "easiness to train" metric might not even be possible a priori. To test this, we have come up with the *ReCluster* Mechanism.
This mechanism has the goal to iteratively improve the forecasting performance by assigning households to clusters for which the corresponding trained LSTM performs best on the training dataset. The ReCluster Algorithm is given in Algorithm 1.

Algorithm 1 ReCluster method

Input: TrainOn(cluster): A function that outputs a trained LSTM given the cluster of consumers that are used to train on.
Input: TrainLoss(LSTM, consumer): A function that outputs the training Log-Likelihood loss of the given LSTM prediction network on the given consumer.
Input: consumers: a sequence of consumers.
Input: $\mathcal{C}_{i+1}$: a sequence of m clusters of consumers.
Input: s: the number of ReClustering steps.
Output: $\mathcal{C}_s$: Improved clustering.
Output: $\mathcal{M}_s$: Improved forecasting models.

procedure ReCluster
  $i \leftarrow 0$
  repeat
    $\mathcal{M}_i \leftarrow \langle \text{TrainOn}(\mathcal{C}_{(i,0)}), ..., \text{TrainOn}(\mathcal{C}_{(i,m-1)}) \rangle$
    for all $j \in \{0, ..., m-1\}$ **do**
      $\mathcal{C}_{(i+1,j)} \leftarrow \{c \in \text{consumers} : j = \text{argmin}_{k \in \{0,...,m-1\}} \text{TrainLoss}(\mathcal{M}_{(i,k)}, c)\}$
    end for
    $\mathcal{C}_{i+1} \leftarrow \langle \mathcal{C}_{(i+1,0)}, ..., \mathcal{C}_{(i+1,m-1)} \rangle$
    $i \leftarrow i + 1$
  until $i = s$.
end procedure

Preferably, the performance calculations for the ReClustering are done on the validation dataset. However, since the validation dataset has only a length of 5 months, the validation dataset has a seasonal bias. To avoid overfitting, the forecasting performance is therefore calculated on the training dataset.

The number of times the ReClustering algorithm can be executed, heavily relies on the computation time that is available and how many combinations of clusters and consumers are present. When executing the ReClustering algorithm for 5 times on 10 clusters, 50 times a neural network is trained, and 50 times the performance of a neural network is tested on all consumers.
Therefore, we limit ourselves to 3, 5 and 10 clusters and execute the ReClustering algorithm for 3 iterations. Hyperparameter tuning is only done before ReClustering, to keep it feasible for ReClustering to be tested on a home PC.

2.2 Results

2.2.1 Clustering

For each of the three feature types explained in Section 2.1.1, we compare clustering performance between clustering based on each feature type separately, and random clusters in Figures A.4, A.5 and A.6. It becomes visible that the average consumptions are different between clusters. These differences do not show up for the randomly formed clusters, so clustering on the separate feature types works as expected.

While doing these tests, it turned out that for more than a couple representative days, the KMeans clustering is heavily determined by peaks in consumption (see Figure A.3), showing that using too many representative days might not work properly. For fewer representative days, however, clear patterns emerge. For example, in Figure 2.5, Household A has in general days with a stable, but higher consumption around 7:00, 15:00 and 17:00 UTC (representative day 2), and a day in which consumption the bulk of consumption is later on the day (representative day 1), around 9:00 and 20:00 UTC. On the other hand, for Household B, early consumption is generally around 8:00 UTC, with some days a really high consumption around 16:00 UTC. One could conclude that Household B is using an electric or an induction stove. Furthermore, the problem of "the curse of dimension-

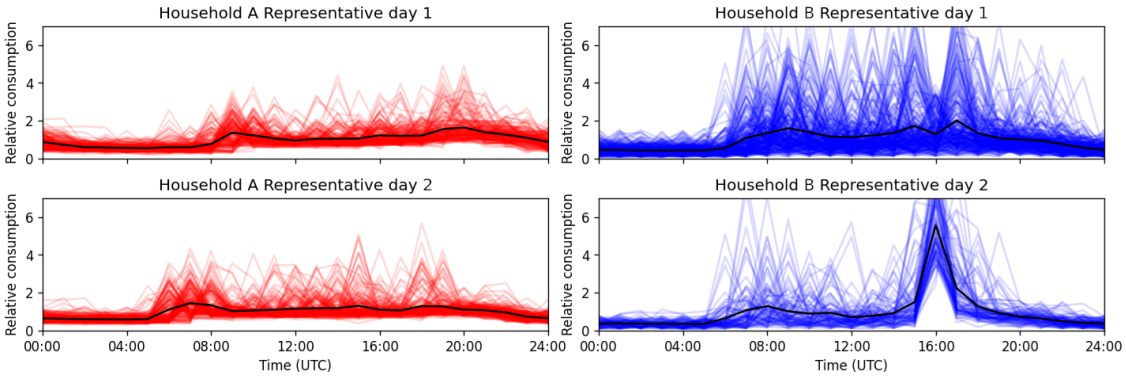


Figure 2.5: Visualization of the result of representative days clustering for two household and two representative days per household. Days in a cluster are plotted in red and blue, and the resulting representative days are plotted with a black line.

ality" arose for more representative days, so it was needed to reduce the final number of features on which we perform clustering. Taking 3 representative days, the above techniques result in 140 features for each household. 72 from representative days, 56 from weekly profile, and 12 from total monthly consumption.

Finding the weights

Each feature type is given a weight. To find the optimal weights, 65 combinations of weights are tested on clustering performance (Results are visible in Figure 2.6.
The weights (3,5,2) for Seasonal Effect, Weekly Effect and Representative Days respectively, were best on average, but not statistically significantly different from the results obtained from the weights (3,6,1) or (3,7,0), for example. The chosen weights for the clustering method are (3,5,2).
In Figure 2.7, the effect of the number of clusters on In-Cluster distance of the final clustering mechanism is plotted.

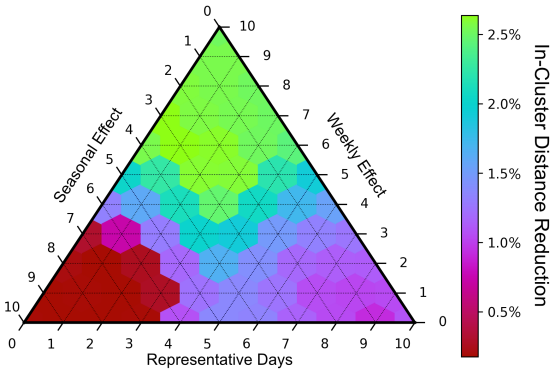


Figure 2.6: Average Reduction in In-Cluster distances for some combinations of feature weights. Each hexagon represents the average over four separately formed clusterings per feature weight combination.

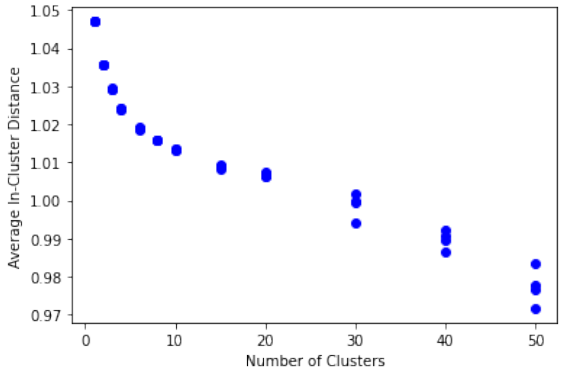


Figure 2.7: Average In-Cluster distances for the final clustering mechanism for some number of clusters. Results are an average of four clustering restarts.

2.2.2 Baseline Forecaster: Shift Approach

The performance of the baseline Shift forecasting method is calculated on the training plus validation set (see Table 2.2). Using the observed energy usage of 7 days into the past gives the best results. With the shift of 7 days, this mechanism has a performance of 44.0% on SMAPE and of 0.657 on NRMSE on the testing set.

Time Shift	SMAPE	NRMSE
1h	38.7%	0.487
2h	47.6%	0.667
3h	53.4%	0.758
4h	58.3%	0.825
6h	65.1%	0.916
8h	68.4%	0.969
10h	68.3%	0.992
12h	68.0%	0.999
1 day	44.9%	0.679
2 days	46.8%	0.706
7 days	44.8%	0.643
14 days	46.2%	0.658

Table 2.2: SMAPE and NRMSE scores for the Shift forecasting method for different shifts in time, on the training plus validation set.

2.2.3 Feature Selection and hyperparameter tuning

See Table 2.3 for the results of the correlation study of consumption and weather features. Based on these results, we see that we can exclude the visibility feature, as it has by far the lowest Time-Normalized correlation. For the other six features, we add the mean of the weather forecasts, and the standard deviations of the forecasts as an input feature to the model.

2.2.4 Results without clustering

Hyperparametertuning

Looking to best log-likelihood on the validation dataset, a batch size of 32, two LSTM layers and a lookback time of 24 hours looked most promising. For the automatically tuned hyperparameters, we found that the best parameters are 41 LSTM units for layer 1, 80 LSTM units for layer 2, a training time of 200 epochs (corresponding to a training duration of about 2 hours on an AMD Ryzen 7 2700X at 4Ghz), a dropout rate of about 0.054 and a learning rate close to 0.0024.

Variable	Correlation	Time-Normalized Correlation
Pressure	0.021	0.123
Relative Humidity	0.083	0.125
Temperature	0.068	0.140
Total Cloud Cover	0.029	0.109
Visibility	0.090	0.017
Wind Speed	0.073	0.109
Wind Direction	0.036	0.118

Table 2.3: Average correlation between weather features and energy consumption per consumer. For the Time-Normalized Correlation, data is split per month and per hour of day.

The number of LSTM units of layer 2 and the number of epochs are against the upper bounds for those parameters, showing that the underlying processes that lead to a consumption pattern for all consumers individually, are complicated and need a stronger neural network than what we allowed.

Results

Looking at the results (see Figure 2.8) of this forecasting approach without clustering, we see that the LSTM approach outperforms the Linear Regression and Shift approach on all performance metrics. On SMAPE, the LSTM beats the Shift on the Testing set with a SMAPE of 42.4% against 45.1%. The LR approach performs significantly worse, with a SMAPE of 59.9%. Looking at NRMSE scores, which penalizes missing spikes much harder than the SMAPE score does, the LSTM still performs best, but now the LR approach outperforms the Shift approach. Possibly because with the distribution forecast, the LR and LSTM approaches can model peaks better than the Shift approach.
On the main score, the log-likelihood loss, the LR approach is also significantly outperformed by the LSTM approach. The Shift approach is not present in these scores, since the Shift approach does not have the needed distribution to calculate the log-likelihood loss as output.

2.2.5 Results with clustering

In this subsection, we use the clustering mechanism described in Section 2.1.1 on a clustering of 3, 5 and 10 clusters, and compare it to forecasting performance with no clustering, which is equivalent to a clustering with only one cluster.

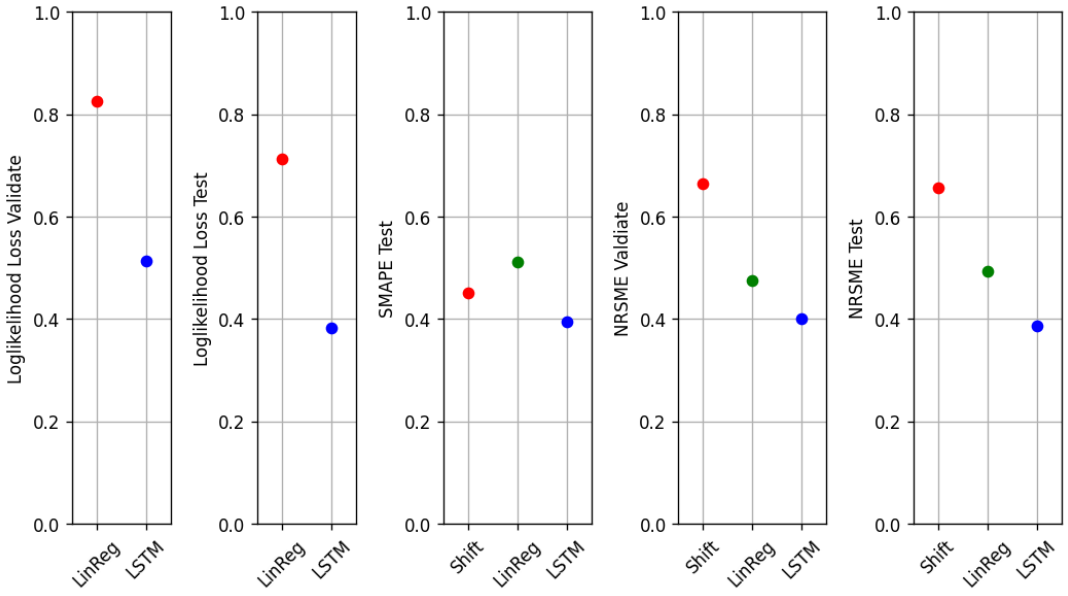


Figure 2.8: Performance of the Shift, LR and LSTM forecasting mechanisms on five performance metrics: log-likelihood loss on the validation and testing set, SMAPE on the testing set and Normalized RMSE (NRMSE) on the validation and testing set. For all performance measures, lower is better. Note that the Shift mechanism is missing for the log-likelihood loss performance metric, since it is not possible to compute this metric for the Shift Mechanism.

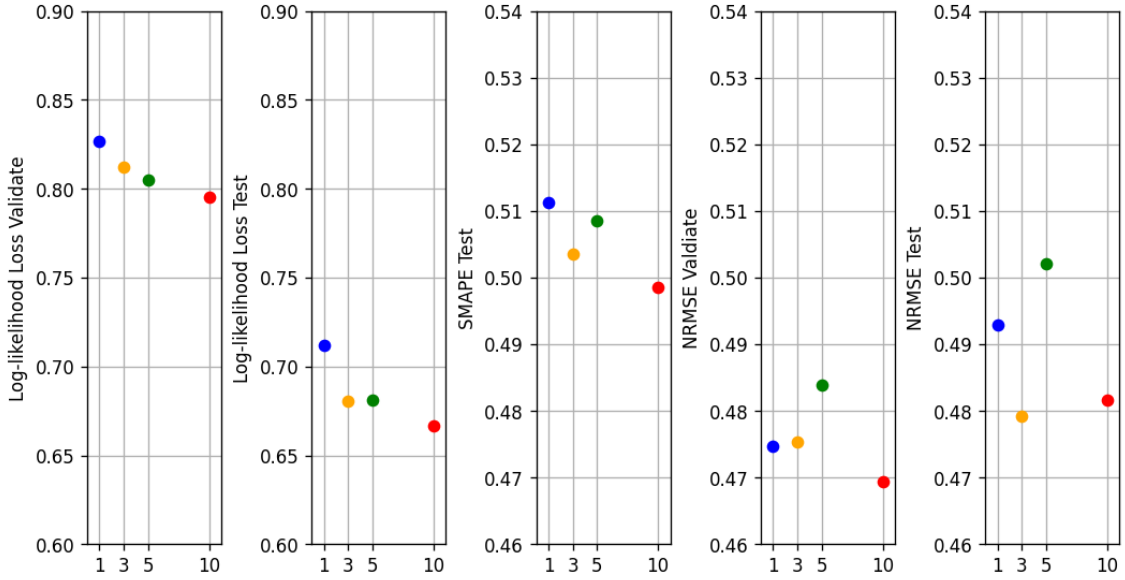


Figure 2.9: Performance of the LR approach using different numbers of clusters on five performance metrics: log-likelihood loss on the validation and testing set, SMAPE on the testing set and NRSME on the validation and testing set. For all performance measures, lower is better.

From the results of clustering combined with the Linear Regression approach (see Figure 2.9), we see generally that with more clusters, log-likelihood losses are lower on the Train and validation sets. SMAPE scores are not significantly different and for NRMSE, no trend is distinguishable.

On the other hand, for clustering in combination with the proposed LSTM approach (see Figure 2.10), we see that with more clusters, the log-likelihood on the validation set decreases, although with diminishing returns. On the testing set, there is even a higher loss with 10 clusters than with 5 clusters, which seem the optimal number of clusters (amongst 1, 3, 5 and 10 clusters). The same is true in regard to SMAPE score and NRMSE on the testing set. Again, the NRMSE on the validation set is consistently lower for more clusters, but also with diminishing returns and with almost no difference between 5 and 10 clusters.

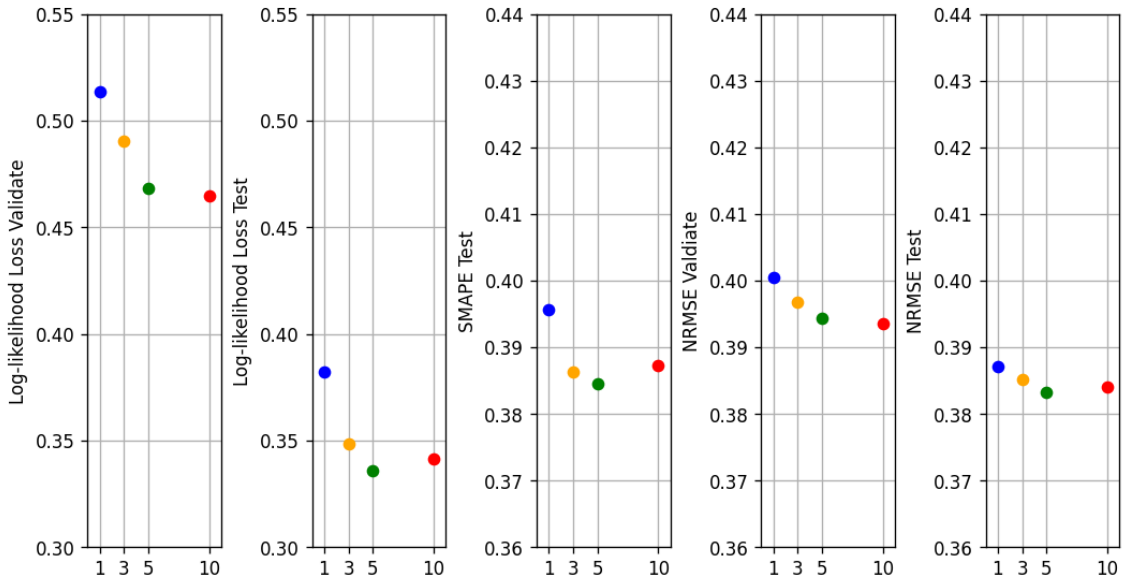


Figure 2.10: Performance of the LSTM approach using different numbers of clusters on five performance metrics: log-likelihood loss on the validation and testing set, SMAPE on the testing set and NRSME on the validation and testing set. For all performance measures, lower is better.

2.2.6 ReClustering

Results are visualized in Figure 2.11. It becomes clear that the ReClustering can significantly improve performance, but not in an extremely stable manner. If we take the LSTM models and clusters of the reclustering step which has the lowest validation score, the resulting log-likelihood losses are 4.8% lower for 3 clusters, 5.2% lower for 5 clusters and 2.7% lower for 10 clusters relative to when no ReClustering has ben used. Furthermore, note that the resulting log-likelihood Loss of executing ReClustering on three clusters is better than using five clusters and no ReClustering. With respect to SMAPE, only minor improvements were found, if at all significant (Figure A.10). For NRMSE, only a decrease in performance was seen (see Figure A.11). Looking at the clustering scores of the newly formed clusters (see Figure 2.12, left), we see that clusters become worse if we follow our definition of a good clustering (in Section 2.1.1), even though the forecasting performance (in Loglikelihood Loss) is increasing. From Figure 2.12, right, we conclude that fewer consumers tend to be reclustered in later ReClustering iterations. In Figures A.7, A.8 and A.9 in Chapter A, mutation matrices are plotted.

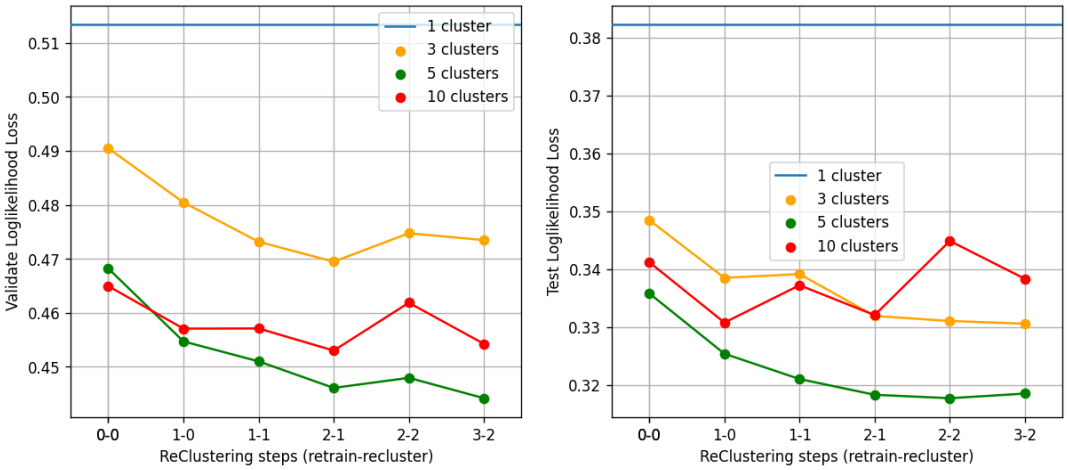


Figure 2.11: Performance of the ReClustering approach on 1 (blue), 3 (yellow), 5 (green) and 10 (red) clusters, based on the Loglikelihood Loss on the Validation dataset (left) and the Testing dataset (right). The x-axis notation has two numbers. The first represents the number of recluster steps, and the second represents the number of retraining steps.

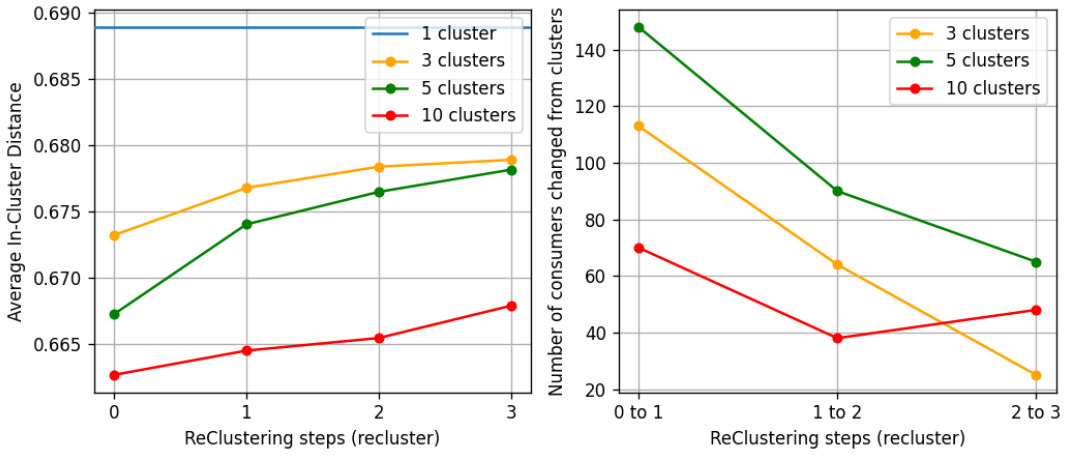


Figure 2.12: Left: Cluster score based on the clustering performance we have defined in Section 2.1.1. Right: Number of consumers that is changed from cluster.

2.3 Forecasting Discussion

After doing extensive data validation and cleanup, we have created a well-performing short-term energy demand forecasting LSTM approach for households that is able to model the peaky behavior of consumer energy usage, and that beats the base Shift and LR approaches. We created a two-step method of using the KMeans clustering to create good clusters that increase forecasting performance of our proposed LSTM approach, compared to no clustering, although with deminishing returns. Furthermore, we have introduced the ReClustering mechanism, which modifies the clusters based on original forecasting performance. It has been observed that ReClustering improves log-likelihood losses significantly. We can conclude that the ReClustering mechanism has a tendency to form more optimal clusters than our KMeans clustering approach. Finally, with ReClustering, fewer clusters can be used to have the same, or even a higher, forecasting accuracy. Using ReClustering, training costs could be reduced significantly without sacrificing performance.

A main challenge during this research was the scarcity and low quality of data. Compared to related literature, the number of consumers in our dataset is small. For example, the time frame of consumption data less than two years, limiting us to only one year of data for training. Since energy consumption depends heavily on weather, and one year of weather cannot reflect all types of weather for a region, results are expected to improve significantly if using more years of data. Also, ReClustering is thought to be less effective as we had less than one year in our validation dataset.
Furthermore, by missing days in consumption and weather forecast data, interpolation had to be done. This is also expected to have had a negative effect on performance.
Additionally, faulty data was present in the original data. It has been a tedious job removing this kind of data, and most certainly, not all faulty data has been removed, as we decided to only remove data when we were confident about it being faulty. The dataset is too large to check everything by hand. High quality data that is validated and precisely interpolated if needed, is crucial for effective forecasting.
When using energy data for learning, an automated and reliable workflow of data validation and cleanup is needed to establish a dataset with high quality.

In future research, distributions other than the ExpoNormal distribution should be evaluated, such as the Weibull distribution, or the Skewed Gaussian distribution. Also, the different approaches should be tested on different and bigger datasets and on computing clusters, allowing for bigger LSTM networks and longer training.

The ReClustering approach gives better clusters that are not formed based on a human-driven definition of good clusters. If we can find out the differences between consumers in different clusters formed by the ReClustering approach, we come closer to discovering what makes a cluster easy to train on. ReClustering is now done based on Loglikelihood Loss, but it should also be tested on SMAPE and NRMSE.
By ReClustering, clusters change. This will definitely also have as effect, that optimal hyperparameters change. If hyperparameter tuning had been done before, during, and after ReClustering, results could be better, and at least, improvements can become stable. For this, ReClustering should be sped up by using a model that is easier to train. Using our LSTM approach during ReClustering is computationally a big task. ReClustering based on a shallow direct neural network might also improve the clusters and can be done much faster than the current approach. When this is indeed much faster, Hyperparameter tuning can be incorporated in every retrain step, possibly causing performance to improve every step. After executing this reworked ReClustering approach, the LSTM networks can be trained on the clusters produced by this faster ReClustering approach.
Faster ReClustering also would allow a dynamic version. Splitting bigger clusters that have a relatively low forecasting performance in two, or merging clusters that are smaller than a certain threshold with another cluster most nearby.

Chapter 3

Load Shifting

3.1 Model

At this point, we believe that we have an accurate individual consumer load forecasting technique. In this chapter, we use the best performing forecasting mechanism for reliable Peak Shifting for lowering energy demand during peak hours, for which we use a battery system. The goal is to minimize the maximum hourly usage per day, by discharging the battery at the times with the highest load. Combining this with charging when demand is lowest, this comes down to minimizing the difference between the highest and lowest load for each day. This is also called *bandwidth minimization*.

3.1.1 Grid Model

We have a synthetic grid model of a neighborhood (See Figure 3.1). In this model, we have a network with the following components.

- Households consuming energy.
- A big battery.
- The external grid.

The internal connections are visualized, but are not present in our boilerplate model. In other words, inside our energy grid, all connections have unlimited capacity and no transmission losses. Furthermore, we do not model grid substations that, for instance, change the voltage level because of the small, granular scale that we are using.

Figure 3.1: Schematic overview of the synthetic electric grid with a variable number of households.

3.1.2 Total Demand Distribution

For this grid, we want to know the forecasted total energy demand that is used by the consumers (without the effect of the battery). Earlier in this thesis, we have created individual consumer load forecasts. We assume these to be, conditionally on the forecasting inputs, independent of each other. The forecasts were given by three parameters per hour t and per consumer c, namely $\mu_{t,c}$, $\sigma_{t,c}$, and $\lambda_{t,c}$. The assumed independence allows us to calculate the total energy usage of a collection of consumers. The demand distribution of one consumer is given by

$$\begin{aligned}&\text{Demand}_t(c) \sim \text{ExpoNormal}(\mu_{t,c}, \sigma_{t,c}, \lambda_{t,c}), \text{ so}\\&\text{Demand}_t(c) \sim N_{\mu_{t,c},\sigma^2_{t,c}} + E_{\lambda_{t,c}} \text{ where } N_{\mu_{t,c},\sigma_{t,c}} \sim \mathcal{N}(\mu_{t,c}, \sigma_{t,c}) \text{ and } E_{\lambda_{t,c}} \sim \text{Exp}(\lambda_{t,c}).\end{aligned}$$

We can write the total distribution, for a group of consumers C, as

$$\begin{aligned}\text{Demand}_t(C) &= \sum_{c\in C} \text{Demand}_t(c)\\&= \sum_{c\in C} N_{\mu_{t,c},\sigma_{t,c}} + \sum_{c\in C} E_{\lambda_{t,c}}.\end{aligned}$$

Since we have assumed independence on forecasting inputs, we can use

$$\sum_{c\in C} N_{\mu_{t,c},\sigma_{t,c}} \sim N_{\sum_{c\in C}\mu_{t,c},\sqrt{\sum_{c\in C}\sigma^2_{t,c}}},$$

and write the sum of exponential distribution as a hypoexponential distribution (also called the generalized Erlang distribution), denoted as Hypo.
Then, we finally have

$$\text{Demand}_t(C) \sim N\left(\sum_{c\in C}\mu_{t,c}, \sum_{c\in C}\sigma^2_{t,c}\right) + \text{Hypo}(\{\lambda_{t,c} : c \in C\}).$$

In practice, we are interested in the percentiles of the forecasted distributions. These are approximated by repeatedly sampling from the normal distribution for all consumers and from the individual exponential distributions of each individual consumer, after which the sum is taken to end up with one samples. In practice, for the 10% percentile, 1000 samples are taken and the 100st smallest value is taken as result, and for the 90% percentile, the 900st smallest value is taken. This is done for each timestep separately. Every 10% percentile is computed.
These approximated percentiles are about 0.1% off from the values from the more precise approximation with 10000 samples. However, taking 10000 samples for 500 consumers, for each hour in a five months period, and each forecasting horizon up until 24 hours into the future is a lot of computational work, we decided to use 1000 samples.

The resulting top-level forecast is used for peak-shifting by planning battery charging and discharging. We charge the battery at times that have a low forecasted net load and discharge the battery at times that have a high net energy load. The forecaster helps us with determining when we should have fully loaded the battery, and how low we can make peak demands. Using the battery, the consumer demand is changed in shape. The resulting shape, which is requested from the external grid, is called the *request* shape, and has as goal to lower the stress on the electric grid, compared to the demand shape.

3.2 Offline Algorithms

During this thesis, three different approaches have been tried. A *Linear Programming* (LP) formulation, a *Dynamic Programming* (DP) approach and lastly, algebraic equations.
The LP and DP approaches were computationally expensive to calculate. In the LP formulation, bookkeeping was a big part and the objective function itself rather simple.

For the DP approach, the battery level had to be discretized, which resulted in a running time of $\mathcal{O}(Tl^2)$, in which T is the amount of timesteps, and l is the number of possible battery levels. Already for a small number of battery levels, running times were unacceptably large.
Lastly, we have the algebraic equation approach. By dividing the problem in parts and solving them by doing algebraic analysis, computational work has been made much faster.

The following variables are used in this and following sections:

Energy demand d_t:	For every time t in the planning horizon, d_t denotes the total energy demand at timestep t for the group of consumers for which peak shifting is done.
Energy request r_t:	For every time t in the planning horizon, r_t denotes the amount of requested energy at timestep t for the group of consumers, together with the battery.
Charge c_t^+:	For every time t in the planning horizon, c_t^+ denotes the amount of energy that is used to charge the battery at timestep t.
Discharge c_t^-:	For every time t in the planning horizon, c_t^- denotes the amount of energy that is discharged from the battery at timestep t.
Maximum charge speed $c_{\max}^+$:	Maximum battery charge speed. The amount of energy used for charging in one timestep can never exceed this value.
Maximum discharge speed $c_{\max}^-$:	Maximum battery discharge speed. The amount of energy discharged in one timestep can never exceed this value.
Battery capacity $b_{\max}$:	The maximum amount of energy that can be stored in the battery.
Battery charge b_t:	The amount of energy stored in the battery at the beginning of timestep t, for which b_0 is the amount of charge in the battery at the start of time.
Battery efficiency e:	The round-trip efficiency of the battery.
Upperbound u:	The requested energy in one timestep cannot exceed this value.
Lowerbound l:	The requested energy in one timestep can never be lower than this value.

We assume a finite planning horizon, so we have all above variables for $0 \leq t < T$ In practice, charging losses occur during charging, storing, and discharging. Since the battery cycle time is relatively short, storing losses (also called *self-discharge* losses) are neglegible[44]. Furthermore, discharging losses are calculated during charging, such that only one calculation has to be done. This slightly reduces precision, but simplifies algorithmic work. Given charge and discharge efficiencies e^+ and e^-, the overall efficiency is given by $e = e^+e^-$.

We have split the offline case into four parts, increasing in extensiveness. See Table 3.1 for the differences between these cases. All cases are solved by first finding the optimal upper- and lowerbound, and then calculating with what net loads the bounds can be achieved. The last case is suitable to be used for testing.

3.2.1 Case I: Infinite battery capacity, battery empty at start

If we have infinite battery capacity, the amount of requested energy can be made arbitrarily high (you can always charge more), so we can set our lowerbound equal to the optimal upperbound. Hence, only the upperbound is interesting to calculate in this case.
For the upperbound, notice that there are no constraints apart from the fact that all energy

Case	Start charge	Battery capacity	Efficiency	Speed	Run time	Optimality
I	0	∞	1	∞	$\mathcal{O}(n^2)$	Proved
II	$b_0 \geq 0$	∞	1	∞	$\mathcal{O}(n^2)$	Proved
III	$b_0 \geq 0$	$b_{\max}$	1	∞	$\mathcal{O}(n^2)$	Proved
IV	$b_0 \geq 0$	$b_{\max}$	$e \in (0, 1]$	$c^+_{\max}, c^-_{\max} > 0$	$\mathcal{O}(n \log_2(p))$	FPTAS

Table 3.1: Different offline battery shifting models, increasing in extensiveness.

discharged into the network has to be charged into the battery before. Hence, for any time t, we must have

$$b_t = \sum_{\tau=0}^{t} (u - d_\tau) \geq 0. \tag{3.1}$$

If $d_\tau < u$, the battery can charge $u - d_\tau$ at time τ. If $d_\tau > u$, the battery must discharge $d_\tau - u$ at time τ. The amount of charged energy minus the amount of discharged energy cannot be negative.

We calculate the best upperbound (i.e., the lowest upperbound) that satisfies this constraint, and take that as our solution, called u^{ALG}. We have

$$u^{ALG} = \max_{0 \leq t < T} \frac{1}{t+1} \sum_{0 \leq \tau \leq t} d_\tau.$$

The optimal request schedule is achieved by requesting u^{ALG} during all timesteps. The optimality of this upperbound follows from the proof of optimality of Case II, where we then use $b_0 = 0$.

Consider Figure 3.2 as an example. We must have that, for the upperbound, the sum of areas in which charging can take place (areas A_{14} and C_{14}) is bigger than the sum of areas in which discharging must take place (areas B_{14} and D_{14}), since we start with an empty battery. The height of the best upperbound that fits this requirement is equal to the mean of all demands up to time $t = 14$. However, this requirement does not only count for $t = 14$. This must be true for all $0 \leq t \leq T$. Hence, we calculate the average for all times t, and take the maximum of those.
In Figure 3.3, it is visible that the best upperbound is not equal to the overall average demand. We must have that the area of A and C is larger than the area of B and D. At the end of the timeseries in the example, there can be some additional charging, but this cannot be used earlier in time for lowering the upperbound. See Figure S1[1] for an animation of this.

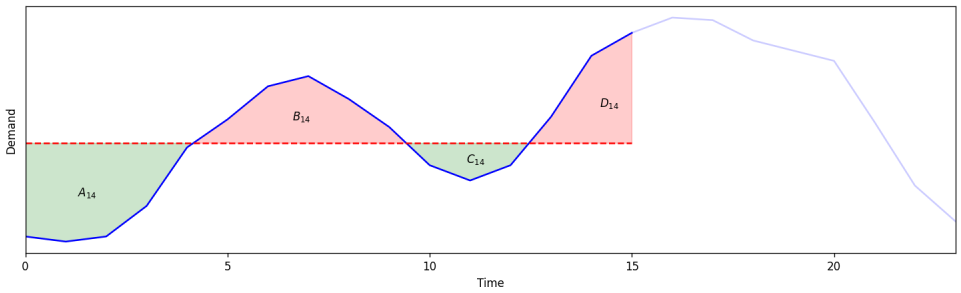


Figure 3.2: Example of one demand series and the optimal upperbound up to $t = 14$, visualized by the red dashed line. Four areas below and above this upperbound are named (from left to right) A_{14}, B_{14}, C_{14} and D_{14}.

[1] The supplemental materials are available on pdknops.nl/supplemental_material.html

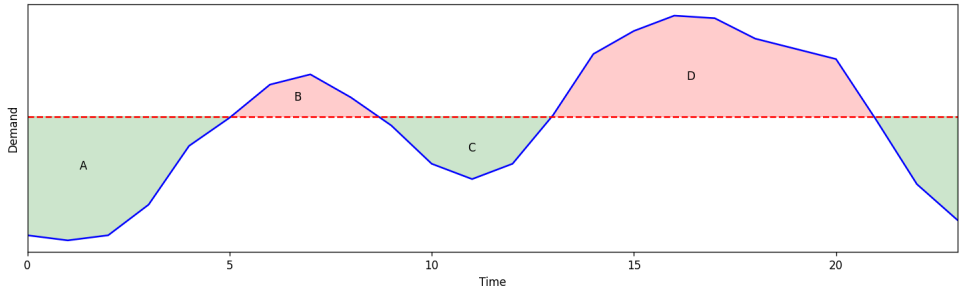


Figure 3.3: Example of one demand series and the optimal upperbound visualized by the red dashed line. Four areas below and above this upperbound are named (from left to right) A, B, C and D.

3.2.2 Case II: Infinite battery capacity, arbitrary battery load at start

The above case can be extended by allowing for arbitrary battery loads at the start, denoted with b_0. The constraint in Equation (3.1) from Case I is extended to be suitable for this case. For any time t, we must have a positive battery charge, and we get a maximum battery charge by making the request equal to the upperbound. Hence, we have

$$b_t = b_0 + \sum_{\tau=0}^{t} (d_\tau - u) \geq 0. \tag{3.2}$$

In other words: Up until any timestep t, the maximum amount of energy charged plus the starting charge, minus the minimum amount of energy charged cannot be negative.
The best upperbound that satisfies this constraint is

$$u^{ALG} = \max_{0 \leq t < T} \frac{1}{t+1} \left(\sum_{0 \leq \tau \leq t} d_\tau - b_0 \right).$$

We calculate this upperbound and take it as solution, resulting in an optimal upperbound.

Proof. Suppose this is not the optimal upperbound. Then, there is a feasible $u^* < u^{ALG}$, and so, there is a time t in $\{0, 1, \ldots, T-1\}$ such that

$$u^* < \frac{1}{t+1} \left(\sum_{0 \leq \tau \leq t} d_\tau - b_0 \right).$$

At the end of timestep t, the battery has a maximum charge of

$$\begin{aligned}
b_0 + \sum_{0 \leq \tau \leq t} (u^* - d_\tau) &= b_0 + (t+1)u^* - \sum_{0 \leq \tau \leq t} d_\tau \\
&< b_0 + (t+1)\frac{1}{t+1} \left(\sum_{0 \leq \tau \leq t} d_\tau - b_0 \right) - \sum_{0 \leq \tau \leq t} d_\tau \\
&= b_0 + \sum_{0 < \tau \leq t} d_\tau - b_0 - \sum_{0 \leq \tau \leq t} d_\tau \\
&= 0,
\end{aligned}$$

which means that the battery is discharged to a strictly negative state, which is not possible and a contradiction with our assumption that u^* is feasible. We conclude that u^{ALG} must be the optimal upperbound. □

3.2.3 Case III: Battery capacity

The third case adds the maximum battery capacity. This introduces the first constraint for the lowerbound; The lowerbound must be such that the battery does not have to be overcharged. We therefore now have a separate upper and lowerbound.
Furthermore, not more than $b_{\max}$ of energy can be stored for a big peak to be reduced. This puts an additional constraint on the upperbound, namely that for any time t_1 and t_2, with $t_2 > t_1$, the amount of battery discharged minus the amount of energy charged must be smaller than the maximum battery capacity $b_{\max}$. In other words,

$$\forall t_1, \forall t_2 > t_1 \colon \sum_{\tau=t_1}^{t_2} (d_\tau - u) \leq b_{\max}.$$

Together with Equation (3.2), and working out u, we see that an upperbound that satisfies this requirement is given by

$$u^{ALG} = \max \left\{ \max_{0 \leq t < T} \frac{1}{t+1} \left(\sum_{0 \leq \tau \leq t} d_\tau - b_0 \right), \max_{0 \leq t_1 < T-1} \max_{t_1 \leq t_2 < T} \frac{1}{t_2 - t_1 + 1} \left(\sum_{t_1 \leq \tau \leq t_2} d_\tau - b_{\max} \right) \right\}. \tag{3.3}$$

During planning, we calculate this and take it as upperbound. Calculation is possible in $\mathcal{O}(n)$ space and $\mathcal{O}(n^2)$ time by using memoisation.
This is also the optimal upperbound.

Theorem 1. *The calculated upperbound u^{ALG} from Equation (3.3) is the optimal upperbound.*

Proof. Suppose that u^{ALG} is not optimal. i.e., there is a feasible $u^* < u^{ALG}$. Then there are two possibilities. Either (A)

$$u^* < \max_{0 \leq t_1 < T-1} \max_{t_1 \leq t_2 < T} \frac{1}{t_2 - t_1 + 1} \left(\sum_{t_1 \leq \tau \leq t_2} d_\tau - b_{\max} \right)$$

or (B)

$$u^* < \max_{0 \leq t < T} \frac{1}{t+1} \left(\sum_{0 \leq \tau \leq t} d_\tau - b_0 \right).$$

Suppose (A), then there are t_1 and t_2 between 0 and T with $t_1 \leq t_2$ such that

$$u^* < \frac{1}{t_2 - t_1 + 1} \left(\sum_{t_1 \leq \tau \leq t_2} d_\tau - b_{\max} \right).$$

In that case, the minimum net discharge (when requesting no less than the upperbound) in timespan $[t_1, t_2]$ is equal to

$$\begin{aligned}
\sum_{t_1 \leq \tau \leq t_2} (d_\tau - u^*) &= \sum_{t_1 \leq \tau \leq t_2} d_\tau - (t_2 - t_1 + 1) u^* \\
&> \sum_{t_1 \leq \tau \leq t_2} d_\tau - (t_2 - t_1 + 1) \frac{1}{t_2 - t_1 + 1} \left(\sum_{t_1 \leq \tau \leq t_2} d_\tau - b_{\max} \right) \\
&= \sum_{t_1 \leq \tau \leq t_2} d_\tau - \sum_{t_1 \leq \tau \leq t_2} d_\tau + b_{\max} \\
&= b_{\max},
\end{aligned}$$

which means that from time t_1 up to time t_2, the battery has a net discharge of more than its capacity, which is not feasible and a contradiction with our assumption.

Suppose (B), then there is a t between 0 and T such that

$$u^* < \frac{1}{t+1}\left(\sum_{0\leq\tau\leq t} d_\tau - b_0\right).$$

Then, from the start of time to the end of timestep t, the battery is discharged from the start of time minimally (when requesting the upperbound) an amount of

$$\begin{aligned}\sum_{0\leq\tau\leq t}(d_\tau - u^*) &= \sum_{0\leq\tau\leq t} d_\tau - (t+1)u^* \\ &> \sum_{0\leq\tau\leq t} d_\tau - (t+1)\frac{1}{t+1}\left(\sum_{0\leq\tau\leq t} d_\tau - b_0\right) \\ &= \sum_{0\leq\tau\leq t} d_\tau - \sum_{0\leq\tau\leq t} d_\tau + b_0 \\ &= b_0,\end{aligned}$$

which means that from the start to time t, the battery has a minimum net discharge of more than b_0, which is not feasible and a contradiction with our assumption.

We conclude that in both possible cases, the optimal $u^* < u^{ALG}$ is not feasible, which proves that u^{ALG} is optimal. □

For the lowerbound, it must be assured that the battery is not overcharged. This is done using two equations. The first is needed to make sure that, for all timesteps t in the horizon, it is not required to charge more than $b_{\max} - b_0$ from time 0 to time t, so

$$\forall t : \sum_{\tau=0}^{t}(l - d_\tau) \leq b_{\max} - b_0.$$

Furthermore, between two timesteps t_1 and t_2, with $t_2 > t_1$, not more than $b_{\max}$ can be charged, so the lowerbound must be such that the minimum amount charged is still less than $b_{\max}$. In other words,

$$\forall t_1,\, \forall t_2 > t_1 \text{:}\ \sum_{\tau=t_1}^{t_2}(l - d_\tau) \leq b_{\max}.$$

Combining these two equations and working out the lowerbound l, we get as best possible lowerbound

$$l^{ALG} = \min\left\{\min_{0\leq t_1<T-1}\min_{t_1\leq t_2<T}\frac{1}{t_2 - t_1 + 1}\left(b_{\max} + \sum_{t_1\leq\tau\leq t_2} d_\tau\right), \tag{3.4}$$

$$\min_{0\leq t<T-1}\frac{1}{t+1}\left(b_{\max} - b_0 + \sum_{0\leq\tau\leq t} d_\tau\right)\right\}. \tag{3.5}$$

This lowerbound can also be computed in $\mathcal{O}(n)$ space and $\mathcal{O}(n^2)$ time using memoization. Furthermore, this lowerbound is optimal.

Theorem 2. *The calculated lowerbound l^{ALG} from Equation (3.3) is the optimal lowerbound.*

Proof. Suppose that l^{ALG} is not optimal, i.e., there is a feasible $l^* > l^{ALG}$. Then there are two possibilities. Either (A)

$$l^* > \min_{0 \le t_1 < T-1} \min_{t_1 \le t_2 < T} \frac{1}{t_2 - t_1 + 1}(b_{\max} + \sum_{t_1 \le \tau \le t_2} d_\tau),$$

or (B)

$$l^* > \min_{0 \le t < T-1} \frac{1}{t+1}(b_{\max} - b_0 + \sum_{0 \le \tau \le t} d_\tau).$$

Suppose (A), then there are t_1 and t_2 between 0 and T with $t_1 \le t_2$ such that

$$l^* > \frac{1}{t_2 - t_1 + 1}(b_{\max} + \sum_{t_1 \le \tau \le t_2} d_\tau).$$

In that case, the minimum net charge in between t_1 and t_2 is equal to

$$\begin{aligned}
\sum_{t_1 \le \tau \le t_2} (l^* - d_\tau) &= (t_2 - t_1 + 1)l^* - \sum_{t_1 \le \tau \le t_2} d_\tau \\
&> (t_2 - t_1 + 1)\frac{1}{t_2 - t_1 + 1}(b_{\max} + \sum_{t_1 \le \tau \le t_2} d_\tau) - \sum_{t_1 \le \tau \le t_2} d_\tau \\
&= b_{\max} + \sum_{t_1 \le \tau \le t_2} d_\tau - \sum_{t_1 \le \tau \le t_2} d_\tau \\
&= b_{\max},
\end{aligned}$$

which means that the battery is charged from time t_1 up to time t_2 more than its capacity, which is not feasible and a contradiction with our assumption.

Suppose (B), then there is a t between 0 and T such that

$$l^* > \frac{1}{t+1}(b_{\max} - b_0 + \sum_{0 \le \tau \le t} d_\tau).$$

Then, from the start of time, the battery is charged, additionally to the start charge of b_0, minimally an amount of

$$\begin{aligned}
\sum_{0 \le \tau \le t} (l^* - d_\tau) &= (t+1)l^* - \sum_{0 \le \tau \le t} d_\tau \\
&> (t+1)\frac{1}{t+1}(b_{\max} - b_0 + \sum_{0 \le \tau \le t} d_\tau) - \sum_{0 \le \tau \le t} d_\tau \\
&= b_{\max} - b_0 + \sum_{0 \le \tau \le t} d_\tau - \sum_{0 \le \tau \le t} d_\tau \\
&= b_{\max} - b_0,
\end{aligned}$$

which means that from the start to time t, the battery is charged more than $b_{\max} - b_0$. Since the battery started with load b_0, the battery has a load of more than $b_{\max}$ at timestep t, which is not feasible and a contradiction with our assumption.

We conclude that in both possible cases, the optimal $l^* > l$ is not feasible, which proves that l^{ALG} is optimal. □

It might be that for some problem instance, the optimal upperbound is lower than the optimal lowerbound. In that case, the battery has been sufficient to stabilize the demand and the coming calculations are not needed, as we can take any value between l^* and u^* as stable request. We assume that we do not have a battery that is big enough for this, and thus, that for any problem instance, we have $u^* \ge l^*$.
It remains to be proven that the optimal upperbound and the optimal lowerbound can be calculated separately. Firstly, we will construct the request scheme, and show that it is feasible. Then, we are able to show that the upperbound and lowerbound are respected.

From bounds to request scheme.

Now, we have an optimal upper- and lowerbound for the requests. What should be chosen as request scheme is not directly known. For being able to satisfy the upperbound, there is some minimum battery level at each timestep t, and for the lowerbound, there is a maximum battery level. After each timestep t, the battery should be between some values $(\underline{b}_t, \overline{b}_t)$, which we call the boundaries of the allowed battery levels, or battery targets, at time t.
These can be calculated by going backwards from time T to the start. The only battery restriction at time T, is that the battery cannot be charged over capacity, and cannot have a negative charge. Hence, $\underline{b}_T = 0$ and $\overline{b}_T = b_{\max}$. Then, at time t, $0 \leq t < T$, we need to charge the battery such that in the next timestep, the battery targets also can be reached.

For the upperbound on the battery boundaries, we have that the target for time t is equal to that of time $t+1$, plus what we have to discharge at time t, minus what we can charge at time t. Formally,

$$\underline{b}_t = \max\left\{0, \underline{b}_{t+1} + \max\{0, d_{t+1} - u\} - \max\{0, u - d_{t+1}\}\right\} = \max\left\{0, \underline{b}_{t+1} + d_{t+1} - u\right\}. \quad (3.6)$$

Likewise, for the lowerbound of the battery target, we have that the target for time t is equal to that of time $t+1$, minus what we have to charge at time t, plus what we can discharge at time t. Formally,

$$\overline{b}_t = \min\left\{b_{\max}, \overline{b}_{t+1} - \max\{0, l - d_{t+1}\} + \max\{0, d_{t+1} - l\}\right\} = \min\left\{b_{\max}, \overline{b}_{t+1} + d_{t+1} - l\right\}. \quad (3.7)$$

We formulate a theorem about these battery bounds.

Theorem 3. *We have $\overline{b}_t \geq \underline{b}_t$ for all $0 \leq t \leq T$.*

Before we prove this theorem, let us introduce and prove two lemmas.

Lemma 4. *We have $\underline{b}_t \leq b_{max}$ for all $0 \leq t \leq T$ and each feasible upperbound u.*

Proof. Assume that this is not the case, so that we have a time t between 0 and T for which we have $\underline{b}_t > b_{\max}$. Then, pick the first time s after t for which we have $\underline{b}_s = 0$. Such a time certainly exists, since T is an example of such a time s.
We can write

$$\underline{b}_t = \underline{b}_s + \sum_{\tau=t}^{s-1} (d_{\tau+1} - u) = \sum_{\tau=t}^{s-1} (d_{\tau+1} - u) > b_{\max},$$

which means that the upperbound is such that minimum net discharge between time t and s is greater than $b_{\max}$. This means that the upperbound u has not been feasible, which is a contradiction with our premise of a feasible upperbound. □

Lemma 5. *We have $\overline{b}_t \geq 0$ for all $0 \leq t \leq T$ and for all feasible lowerbounds l.*

The proof of this lemma goes analogously with the proof of Lemma 4 and is omitted.

Now, we can give the proof of Theorem 3

Proof. Remember that $\underline{b}_T = 0$ and $\overline{b}_T = b_{\max}$, so we have $\underline{b}_T \leq \overline{b}_T$. Now the proof will be a backwards induction proof.
We assume that $\underline{b}_{t+1} \leq \overline{b}_{t+1}$ Looking at Equations (3.6) and (3.7), to prove that $\underline{b}_t \leq \overline{b}_t$, we must show that (A) $b_{\max} \geq 0$, (B) $b_{\max} \geq \underline{b}_{t+1} + d_{t+1} - u$, (C) $\overline{b}_{t+1} + d_{t+1} - l \geq 0$ and (D) $\overline{b}_{t+1} + d_{t+1} - l \geq \underline{b}_{t+1} + d_{t+1} - u$.

(A) It is trivial to see that $b_{\max} \geq 0$.

(B) Since we have that $\underline{b}_{t+1} + d_{t+1} - u \leq \underline{b}_t$ (see Equation (3.6)), and $\underline{b}_t \leq b_{\max}$ (see Theorem 4), we also have

$$\underline{b}_{t+1} + d_{t+1} - u \leq b_{\max}.$$

(C) Since we have that $\overline{b}_{t+1} + d_{t+1} - l \geq \overline{b}_t$ (see Equation (3.7)), and $\overline{b}_t \geq 0$ (see Theorem 5), we have

$$\overline{b}_{t+1} + d_{t+1} - l \geq 0$$

.
(D) We have assumed that $\underline{b}_{t+1} \leq \overline{b}_{t+1}$ (induction hypothesis). Together with $l \leq u$, we also know that

$$\overline{b}_{t+1} + d_{t+1} - l \geq \underline{b}_{t+1} + d_{t+1} - u$$

.
Since all elements in the maximum function for $\underline{b}_t$ are smaller or equal to all the elements in the minimum function for $\overline{b}_t$, we conclude that we have also $\underline{b}_t \leq \overline{b}_t$. Using induction, we have shown this for all $0 \leq t \leq T$. □

After calculating the allowed battery levels, we can calculate the minimum delta $\underline{\delta}_t$ and maximum delta $\overline{\delta}_t$ for the battery charge, for all timesteps, based on the lowerbound, the upperbound and the allowed battery levels. These $\underline{\delta}_t$ and $\overline{\delta}_t$ describe how the battery must be charged or discharged, to respect all battery related constraints and respect the upper and lowerbound.
For $\underline{\delta}_t$, the most negative change that the battery can have on its charge, we have that it must satisfy the lowerbound and the minimum allowed battery levels. So we have

$$\underline{\delta}_t = \max\{l - d_t, \underline{b}_{t+1} - b_t\}. \tag{3.8}$$

Now, $\overline{\delta}_t$, the most positive change that the battery can have on its charge, must satisfy the upperbound and the maximum allowed battery level. For $\overline{\delta}_t$, we have

$$\overline{\delta}_t = \min\{u - d_t, \overline{b}_{t+1} - b_t\}. \tag{3.9}$$

All these calculations can be done in $\mathcal{O}(n)$ time. Notice that both $\underline{\delta}_t$ and $\overline{\delta}_t$ can be negative. However, we always have $\underline{\delta}_t \leq \overline{\delta}_t$. This is formulated in a theorem and proven.

Theorem 6. *We have $\underline{\delta}_t \leq \overline{\delta}_t$ for all $0 \leq t \leq T$.*

Proof. We must show

$$\underline{\delta}_t \leq \overline{\delta}_t,$$

or equivalently,

$$\max\{l - d_t, \underline{b}_{t+1} - b_t\} \leq \min\{u - d_t, \overline{b}_{t+1} - b_t\}.$$

Since $u \geq l$, and $\underline{b}_{t+1} \leq \overline{b}_{t+1}$, we have $l - d_t \leq u - d_t$ and $\underline{b}_{t+1} - b_t \leq \overline{b}_{t+1} - b_t$.
Furthermore, we have to show that

$$\underline{b}_{t+1} - b_t \leq u - d_t \text{ and } l - d_t \leq \overline{b}_{t+1} - b_t.$$

We will only do the first one, since the second one goes analogously.
We have that $\underline{b}_{t+1} - b_t \leq \overline{b}_{t+1} - b_t$ (see Theorem 3), and since the maximum battery target is feasible for the lowerbound, we have $\overline{b}_{t+1} - b_t \leq l - d_t$. Lastly, we have $l - d_t \leq u - d_t$. Putting these together, we acquire

$$\underline{b}_{t+1} - b_t \leq u - d_t.$$

Since all elements inside the minimum function in Equation (3.9) are greater or equal to all elements inside the maximum function in Equation (3.8), we conclude that we must have $\underline{\delta}_t \leq \overline{\delta}_t$. This is applicable to all $0 \leq t < T$, so we have proven that $\underline{\delta}_t \leq \overline{\delta}_t$ for all $0 \leq t \leq T$. □

Because of Theorem 6, we see that for any time t there exists an r_t that is feasible with respect to the battery constraints, and respects the upper and lowerbound. This is formulated in the following theorem.

Theorem 7. *For any time t, take any δ_t such that $\underline{\delta}_t \leq \delta_t \leq \overline{\delta}_t$. Take as resulting request $r_t = d_t + \delta_t$. This request r_t satisfies the battery target levels and the upper- and lowerbound.*

Proof. For every time t with $0 \leq t < T$, let us be given such δ_t, $0 \leq t < T$ with $\underline{\delta}_t \leq \delta_t \leq \overline{\delta}_t$. Take as resulting request $r_t = d_t + \delta_t$.
Then we have

$$\begin{aligned} r_t &= d_t + \delta_t \\ &\leq d_t + \overline{\delta}_t \\ &\leq d_t + u - d_t \\ &= u, \end{aligned}$$

so r_t respects the upperbound.
Likewise, we see that

$$\begin{aligned} r_t &= d_t + \delta_t \\ &\geq d_t + \underline{\delta}_t \\ &\geq d_t + l - d_t \\ &= l, \end{aligned}$$

which shows that r_t respects the lowerbound.
Furthermore, we have

$$\begin{aligned} b_{t+1} &= b_t + \delta_t \\ &\leq \overline{b}_t + \overline{\delta}_t \\ &\leq \overline{b}_t + \overline{b}_{t+1} - b_t \\ &= (\overline{b}_t - b_t) + \overline{b}_{t+1} \\ &\leq \overline{b}_{t+1}, \end{aligned}$$

so with such a r_t, we remain below the maximum battery levels.
Analogously, we also have

$$\begin{aligned} b_{t+1} &= b_t + \delta_t \\ &\geq \underline{b}_t + \underline{\delta}_t \\ &\geq \underline{b}_t + \underline{b}_{t+1} - b_t \\ &= (\underline{b}_t - b_t) + \underline{b}_{t+1} \\ &\geq \underline{b}_{t+1}, \end{aligned}$$

so with r_t, we also are above the minimum battery levels. □

Any request scheme where we choose r_t in this fasion, is feasible and respects the upper and lowerbound. Since there are for any time t feasible r_t (since $\underline{\delta}_t < \overline{\delta}_t$), such a feasible request scheme always exists. This also proves that the lowerbound and upperbound could indeed be calculated separately.

In the experiments, we decided to take the average of $\underline{\delta}_t$ and $\overline{\delta}_t$. Any value between those two values results in a feasible request scheme, but by taking the average, we prevent having a bias to a more filled or to a more empty battery. This average is then translated back into the amount charged, or the amount discharged, at timestep t. See Figure 3.4 for an example of the result.

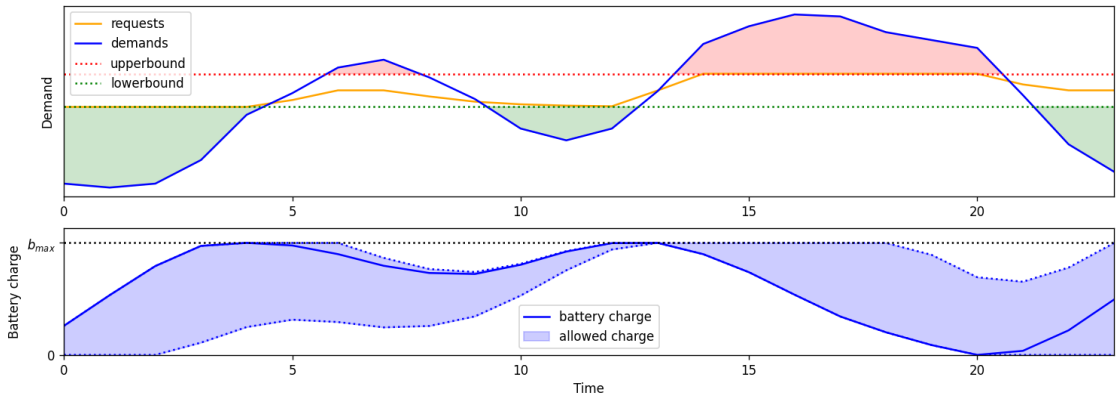


Figure 3.4: Example of a demand timeseries, with optimal lower and upperbound, the allowed battery levels and the resulting request scheme and battery charge.

3.2.4 Case IV: Maximum (dis)charging speed and efficiency.

In the most general case, we have a round-trip efficiency of the battery of below 100%, and maximum charge and discharge speeds. We conjecture that this proof is $\mathcal{NP}$-complete, but we have been unable to find a reduction to prove this statement. However, in other research to a similar problem (for instance the one by Johnson et al. [33] discussed in the Related Literature), also no polynomial solutions were found.

We have formulated LP formulations for the best feasible upper- and lowerbounds. For the upperbound, we have to find the lowest feasible upperbound such that every request is lower than that upperbound (Equation (3.11)), every request is equal to the demand plus what is charged (accounting for efficiency) minus what is discharged (Equation (3.12)) and that the resulting change in battery charge is accounted for (Equation (3.13)). The maximum charge and discharge speeds cannot be exceeded (Equations (3.14) and (3.15)). The battery cannot be overcharged (Equation (3.16)), and finally, charge, discharge and battery level cannot be negative (Equation (3.17)).

$$
\begin{aligned}
\min \quad & u && && && (3.10)\\
\text{s.t.} \quad & r_t && \leq u && \forall t \in \{0, ..., T-1\} && (3.11)\\
& d_t + \frac{1}{e} c_t^+ - c_t^- && = r_t && \forall t \in \{0, ..., T-1\} && (3.12)\\
& b_t + c_t^+ - c_t^- && = b_{t+1} && \forall t \in \{1, ..., T-1\} && (3.13)\\
& c_t^+ && \leq c_{\max}^+ && \forall t \in \{0, ..., T-1\} && (3.14)\\
& c_t^- && \leq c_{\max}^- && \forall t \in \{0, ..., T-1\} && (3.15)\\
& b_t && \leq b_{\max} && \forall t \in \{0, ..., T-1\} && (3.16)\\
& b_t, c_t^+, c_t^- && \geq 0 && \forall t \in \{0, ..., T-1\} && (3.17)
\end{aligned}
$$

Likewise, the LP formulation for the best feasible lowerbound is as follows.

$$
\begin{aligned}
\max \quad & l \\
\text{s.t.} \quad & r_t && \geq l && \forall t \in \{0, ..., T-1\} \\
& d_t + \frac{1}{e} c_t^+ - c_t^- && = r_t && \forall t \in \{0, ..., T-1\} \\
& b_t + c_t^+ - c_t^- && = b_{t+1} && \forall t \in \{1, ..., T-1\} \\
& c_t^+ && \leq c_{\max}^+ && \forall t \in \{0, ..., T-1\} \\
& c_t^- && \leq c_{\max}^- && \forall t \in \{0, ..., T-1\} \\
& b_t && \leq b_{\max} && \forall t \in \{0, ..., T-1\} \\
& b_t, c_t^+, c_t^- && \geq 0 && \forall t \in \{0, ..., T-1\}
\end{aligned}
$$

As previously stated, these LP formulations contain a lot of bookkeeping, and are computationally more difficult to solve than the equations that we have introduced for solving this problem.

However, given a candidate upperbound, we can check feasibility by an $\mathcal{O}(n)$ calculation. This can be done by simulating the battery for every timestep where we charge maximally (respecting the upperbound and battery constraints), and check if constraints are not violated for that upperbound. This can also be done analogously for the lowerbound.

We use a bisection algorithm to find the optimal upper- and lowerbound. For the upperbound, we take as highest possible upperbound the maximum demand $\max_t d_t$ and for the lowest possible upperbound the minimum demand $\min_t d_t$ as interval to search in, since the optimal upperbound will be between the lowest and largest demand.
Starting with this interval, denoted as $(\underline{u}, \overline{u})$, we calculate the middle point m of the interval and check if using m as the upperbound is feasible. If so, the interval can be reduced to $(\underline{u}, m)$. If m as upperbound is not feasible, then the interval is reduced to $(m, \overline{u})$. This is repeated recursively up until the size of the interval is smaller than some predetermined precision ϵ. This is more formally written in Algorithm 2, which forms a Fully Polynomial Time Approximation Scheme (FPTAS).

Algorithm 2 Bisection search to the best upperbound

```
1: procedure BisectionUpper(d, u, ū)
2:     if ū − u < ε then
3:         return u
4:     end if
5:     m ← (u + ū)/2
6:     if m feasible upperbound then
7:         BisectionUpper(d, u, m)
8:     else
9:         BisectionUpper(d, m, ū)
10:    end if
11: end procedure
```

Theorem 8. *Algorithm 2 forms a FPTAS for finding the best feasible upperbound.*

Proof. Let us be given a maximum approximation error $\epsilon > 0$, and a problem instance with demands d_t with $0 \leq t < T$, maximum battery capacity $b_{\max}$, efficiency e and maximum charge and discharge speeds $c_{\max}^+$ and $c_{\max}^-$. Furthermore, denote with $\underline{u}_s$ and $\overline{u}_s$ the search interval after s steps. Denote the total number of recursive steps used in Algorithm 2 with S. We have set $\underline{u}_0 = \min_i d_i$ and $\overline{u}_0 = \max_i d_i$. Furthermore, we know that S is the smallest number such that

$\overline{u}_S - \underline{u}_S < \epsilon$, and thus

$$\left(\frac{1}{2}\right)^S \left(\max_i d_i - \min_i d_i\right) < \epsilon$$
$$\frac{\max_i d_i - \min_i d_i}{\epsilon} < 2^S$$
$$\log_2\left(\frac{\max_i d_i - \min_i d_i}{\epsilon}\right) < S.$$

Since S is the smallest integer number that satisfies this equation, we know that

$$S < \log_2\left(\frac{\max_i d_i - \min_i d_i}{\epsilon}\right) + 1.$$

Furthermore we know by construction that the optimal upperbound u^* is in the interval $[\underline{u}_S, \overline{u}_S]$. Together with the fact that $\overline{u}_S - \underline{u}_S < \epsilon$ and that u^{ALG}, the solution found by Algorithm 2, is equal to $\underline{u}_S$, we have $u^{ALG} < u^* + \epsilon \leq (1+\epsilon)u^*$, assuming that $u^* > 1$[2].
Since we have less than

$$\log_2\left(\frac{\max_i d_i - \min_i d_i}{\epsilon}\right) + 1$$

steps of $\mathcal{O}(n)$, this algorithm has running time

$$\mathcal{O}\left(n \log_2\left(\frac{\max_i d_i - \min_i d_i}{\epsilon}\right)\right),$$

which is polynomial in the problem size and polynomial in the precision ϵ. Hence, Algorithm 2 forms a FPTAS for the best feasible upperbound. □

Finding the lowerbound goes analogously to finding the upperbound, and the algorithm for this (See Algorithm 3) also forms a FPTAS. We have omitted this proof, since it goes analogously to the proof of Theorem 8. As initial search interval, we use the same interval as for the upperbound.

Algorithm 3 Bisection search to the best lowerbound

```
1: procedure BisectionLower(d, l, l̄)
2:     if l̄ − l < ε then
3:         return l
4:     end if
5:     m ← (l + l̄)/2
6:     if m feasible lowerbound then
7:         BisectionLower(d, m, l̄)
8:     else
9:         BisectionLower(d, l, m)
10:    end if
11: end procedure
```

These upper and lowerbounds are also converted to a request scheme by firstly calculating the allowed battery levels. The equations for this are a bit more extensive than for case III, to account for the maximum (dis)charging speed, and the round-trip efficiency. Equation (3.6) is replaced by

$$\underline{b}_t = \max\left\{0, \underline{b}_{t+1} - \max\{0, \min\{e \cdot (u - d_{t+1}), c^+_{\max}\}\} + \max\{0, d_{t+1} - u\}\right\}, \tag{3.18}$$

and Equation (3.7) is replaced by

$$\overline{b}_t = \min\left\{b_{\max}, \overline{b}_{t+1} + \max\{0, \min\{d_{t+1} - l, c^-_{\max}\}\} - \max\{0, e \cdot (l - d_{t+1})\}\right\}. \tag{3.19}$$

[2] If we have $\min_i d_i > 1$, this is the case. If not, we can do $\log_2\left(\frac{1}{\min_i d_i}\right)$ extra iterations, which is linear in the problem size, and thus also results in a PFTAS. For simplicity, this is left out of the proof.

Furthermore, for the maximum and minimum deltas, we now have

$$\underline{\delta}_t = \max\left\{ l - d_t, -c^-_{\max}, \max\left\{ \frac{1}{e}\left(\underline{b}_{t+1} - b_t\right), 0 \right\} - \max\{b_t - \underline{b}_{t+1}, 0\} \right\} \tag{3.20}$$

instead of Equation (3.8), and instead of Equation (3.9), we have

$$\overline{\delta}_t = \min\left\{ u - d_t, \frac{1}{e}c^+_{\max}, \max\left\{ \frac{1}{e}\left(\overline{b}_{t+1} - b_t\right), 0 \right\} - \max\{b_t - \overline{b}_{t+1}, 0\} \right\}. \tag{3.21}$$

The extra terms in these equations account for the minimum and maximum charge speeds, and for the difference of effect of charging and discharging by the efficiency factor.
We do not give proofs of Theorems 3 and 6 for this extended case, since this would be a notational burden, and it is expected that the proofs go analogously with the proofs of Theorems 3 and 6. The request scheme is calculated in the same manner as in Case III.

3.3 Online Algorithms

For the online case, the most extensive case from the offline algorithms must be used, as we model battery round-trip efficiency, starting charge of the battery, maximum capacity, et cetera.

The difference between the offline and the online case is that future demands are not known in the online case. However, we do have a forecasted energy demand. A request planning can be made based on the forecast. This planning is dictated by the upper- and lowerbound, calculated using Algorithms 2 and 3, where the unknown demand is replaced by the forecasted demand. Then, Equations (3.18) and (3.19) are used to determine the allowed battery charges. Finally, the average of $\underline{\delta}_t$ and $\overline{\delta}_t$, calculated using Equations (3.20) and (3.21), is used to determine the request scheme.

However, observed values will be different from forecasted values. Therefore, it might be that the planned request scheme turns out to be infeasible. It is tried to satisfy the upper- and lower-bound, but battery constraints can never be violated. Hence, it makes sense to be a bit conservative with (dis)charging the battery, to leave some room for forecasting errors. In this way, we have 5 different online strategies, where two are baseline approaches, one an approach without a reserve, and the last two are approaches with a reserve.

Offline: The case where future demands are known in advance, this results in the optimal peak shift.

Shift-24h: In this approach, the schedule is based on the 24h Shift forecasting technique. This serves as a benchmark that should be improved upon.

Shift-7d: Here, the schedule is based on the 7d Shift forecasting technique. This approach also serves as benchmark.

LSTM-FM: Here, our proposed forecasting technique will be used, but the forecasted distribution is not leveraged. Instead, the mean of the forecasts will be used.

LSTM-BR: To prevent the battery from running out of energy, or the battery getting filled at capacity too fast, we reserve a percentage of the battery capacity. If we take a percentage of α as reserve, the planning cannot discharge the battery under α percentage, or charge the battery for more than $1 - \alpha$ percent. This variable α is a value to be fit.

LSTM-PT: Our LSTM forecasting approach offers a distribution forecast, which can exploited. Since we cannot assume independence of the consecutive distributions[3], only the

[3] We have assumed independence given forecasting input, but consecutive forecasts will have different inputs because input values change over time.

distribution of the first hour can be used. We use the mean forecast, as well as the 10% and 90% percentiles to decide on battery reserve. We use $\underline{\alpha}$ times the difference between the mean and 10% percentile as the capacity reserve, and $\overline{\alpha}$ times the difference between the 90% percentile and the mean as the battery load reserve. In this way, for the case of an overprediction (the demand lies in the higher percentiles of the forecasted distribution) we have a battery reserve to be able to serve the higher demand, and in the case of an underprediction (the demand lies in the lower percentiles of the forecasted distribution), we still have enough capacity to use the energy not used to charge the battery. See Figure 3.5 for a visualized example of this.
The variables $\underline{\alpha}$ and $\overline{\alpha}$ are values to be fit.

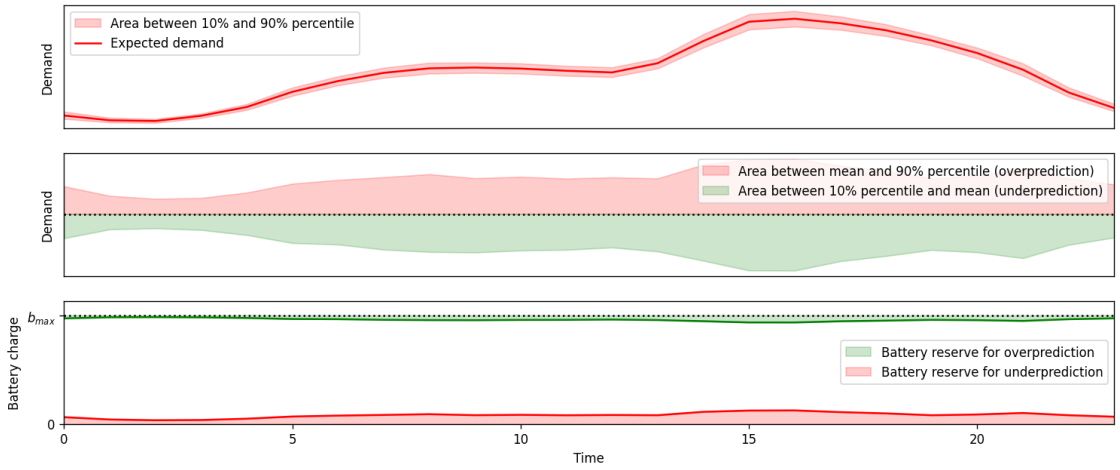


Figure 3.5: Example of the calculation of battery reserves with percentiles. Making the reserves dependent on the percentiles makes it dynamically linked to the uncertainty of the forecast. In the case of an overprediction, less energy is used, but we have reserved battery capacity to use that energy for charging. Likewise, in the case of an underprediction, the energy demand is higher than we expected, but we have reserved battery load to supply the grid for the underpredicted demand.

Furthermore, these online strategies are implemented in a Rolling-Horizon approach, where we plan 24 hours into the future, but reschedule every hour. From each planning, we try to meet the request of the first hour of that schedule. Then, the actual demand for that hour becomes known, and a more accurate forecast can be made. Based on this newly available information, the next schedule is made. This is done in a continuous fashion.

The reserves are implemented by changing the battery starting charge. For instance, if we take a 5% reserve in the LSTM-BP approach and the battery is currently loaded to 20% of its maximal charge, the lowerbound is calculated on a starting charge of 15% and the upperbound is calculated on a starting charge of 25%.

3.4 Experimental Setup

As input data, we use the mean and percentiles of the forecasted distributions, as described in Section 3.1. Tests are done on six populations, namely the whole of the 500 households, and 5 groups of 100 households, such that every household is used in the tests twice.
The data that we used during validation of the forecasting approaches (January 2022 to May 2022) is used to fit variables for the LSTM-BR and LSTM-PT approaches, and the testing data (June 2022 to October 10th 2022) is used to test the different approaches. For fitting variables, the objective is to have the lowest mean daily bandwidth, since the goal of using the batteries, is to

minimize the daily bandwidth.

In the tests, we will use the Tesla Powerwall 2, as it has been used also in some other peak shifting research papers. The following advertised specifications are important and used in the model (See Table 3.2).
For the case with all 500 consumers, tests will be done on 5 and 10 Tesla Powerwalls. For the five groups, tests will use 1 and 2 Tesla Powerwalls per group, also totalling in 5 and 10 Powerwalls. In this way, conclusions about the effect of the size of the subgrid and battery size can be done.

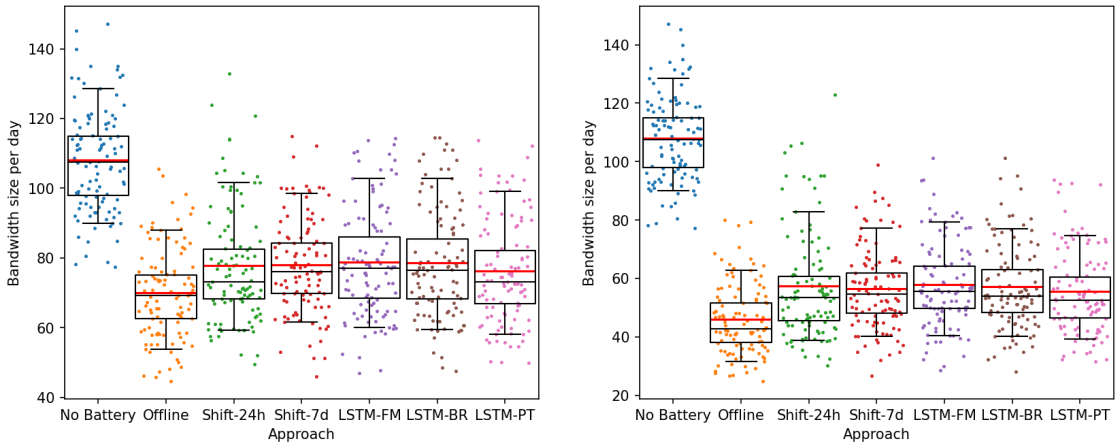
Usable capacity	13.5 kWh
Round-trip efficiency	90%
Max continuous (dis)charge	5 kW

Table 3.2: Specifications of the Tesla Powerwall 2[45].

As performance measure, the average daily bandwidth will be used. For this, the request shape will be separated into days, then we will subtract the minimum of each day from the maximum from each day, after which the average value is computed. This score should be minimized.
Furthermore, the maximum loads are calculated per day, and there will be made a separation between aggregating the five groups on mean demand and total demand. This also allows forming conclusions about on what scale load shifting can be done best, and how the distribution formed by summing consumer distribution forecasts performs.

3.5 Results

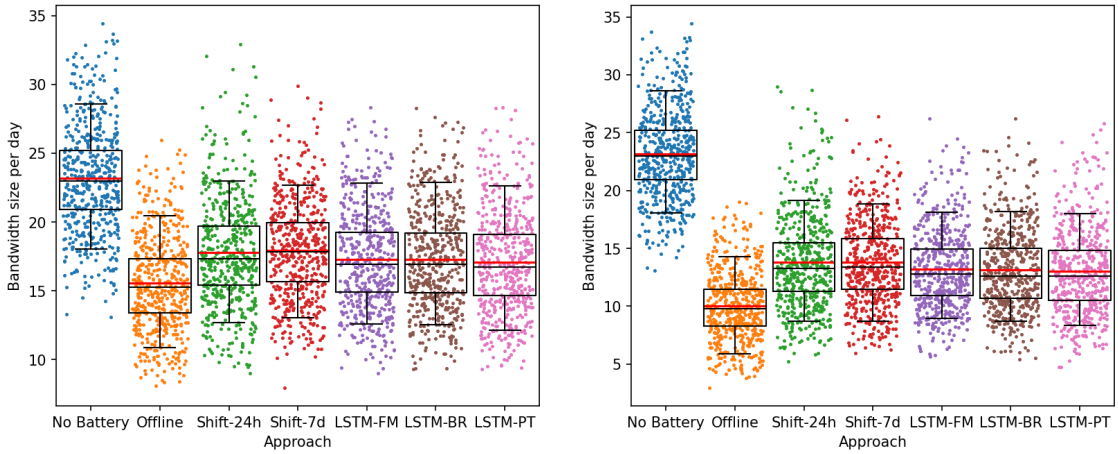
For the case with all 500 consumers the bandwidth results are shown in Figure 3.6, and the results for the maximum daily energy usage are shown in Figure A.12. It is visible that the shift approaches are better than the approaches based on forecasting where no percentiles are used (LSTM-FM and LSTM-BR). However, for the case with five groups of 100 consumers, all forecasting approaches perform significantly better than both shift approaches (See Figure 3.7 for the bandwidth results, and Figure A.13 for the maximum demand results).



(a) Five batteries.

(b) Ten batteries.

Figure 3.6: The test cases with all 500 consumers and five batteries (left) or ten batteries (right) for the case where no battery is used, the offline case and the five other approaches. Each dot represents the size of the bandwidth of a day in the data. The mean daily bandwidth per approach is plotted in red lines, and boxplots are visualized for the 10%, 25%, 50%, 75% and 90% percentiles.



(a) One battery.

(b) Two batteries.

Figure 3.7: The test cases with 5 groups of 100 consumers and one batteries (left) or two batteries (right) for the case where no battery is used, the offline case and the five other approaches. Each dot represents the size of the bandwidth of a day in the data for one of the five groups. The mean daily bandwidth per approach is plotted in red lines, and boxplots are visualized for the 10%, 25%, 50%, 75% and 90% percentiles.

In all online approaches, a big gap to the optimal Offline approach is present. The Offline approach has in all test cases 10% more bandwidth reduction than online approaches, compared to the case without a battery (See Table 3.3 for the bandwidth results, and Table A.1 for the maximum demand results).

	All 5	Groups 5 Mean	Groups 5 Total	All 10	Groups 10 Mean	Groups 10 Total
Offline	35.3%	32.8%	31.3%	57.5%	56.9%	55.6%
Shift-24h	27.9%	23.2%	26.3%	46.8%	40.5%	46.3%
Shift-7d	28.7%	22.6%	25.3%	47.7%	40.6%	46.8%
LSTM-FM	27.1%	25.4%	27.0%	46.4%	43.0%	46.5%
LSTM-BR	27.3%	25.2%	27.1%	47.0%	43.2%	46.9%
LSTM-PT	**29.3**%	**26.2**%	**28.4**%	**48.6**%	**43.9**%	**48.0**%

Table 3.3: Percentage of bandwidth reduction per approach and per test case. The best online approach per test-case is written in bold.

Furthermore, the approach using the percentiles of the distribution performs best in all test cases. For the groups, load shifting based on the LSTM predictions does work better than using the Shift approaches, but for the case where all 500 consumers are used at the same time, the LSTM-FM and LSTM-BR approaches do not beat the Shift approach.
Lastly, the LSTM-PT approach is working significantly better for the cases where all 500 consumers are used instead of the cases where we have five groups of 100 consumers. This difference is not found for the LSTM-FM and LSTM-BR approaches.

	All 5	Group 5 Mean	Group 5 Sum	All 10	Group 10 Mean	Group 10 Sum
1st Approach	LSTM-PT	LSTM-PT	LSTM-PT	LSTM-PT	LSTM-PT	LSTM-PT
2nd Approach	Shift-24h	LSTM-FM	LSTM-FM	Shift-7d	LSTM-BR	Shift-7d
p-value	0.0131	$1.723 \cdot 10^{-8}$	$1.304 \cdot 10^{-5}$	0.1001	$7.250 \cdot 10^{-4}$	0.0112

Table 3.4: Statistical p-values between the best approach (LSTM-PT for all cases) and the second-best approach. The p-values are calculated using a paired t-test on the daily bandwidth.

It becomes clear that LSTM-PT is also significantly the best approach, except in one case (see

Table 3.4 p-value of 5% taken as acceptance criterion). Note that the Group 5 Mean and Group 10 Mean have lowest p-values, while they have the most datapoints for the paired t-test (for every group, all tested days are used, resulting in $5 \cdot 104$ datapoints).

3.6 Discussion

We have defined the problem of peak shifting and solved it by bandwidth reduction. We have developed fast, exact methods for cases without maximum (dis)charging speeds and battery inefficiencies, with proven optimality. For the case with a maximum (dis)charging speed and battery efficiency, we have developed a Fully Polynomial Time Approximation Scheme, which can find solutions arbitrary close to the optimal solution.
Translating these algorithms from the Offline case to Online variants, we have developed multiple rolling horizon planning approaches. These approaches are successful in peak shifting and minimizing request bandwidth. Only using a couple of home batteries (only 5 to 10 home batteries for 500 households), bandwidths can be reduced with almost 50%, alleviating the stress on the electric grid.
Our proposed approach, LSTM-PT, beats the benchmark approaches. Without the use of percentiles, the benchmark would not have been beaten consistently, which underlines the importance of uncertainty modelling by using distribution forecasts.
Furthermore, the approach using forecasting percentiles becomes relatively better for larger groups of consumers, advocating the use of shared home batteries, as opposed to individual deployment of home batteries. As possible cause of this phenomenon can be that it might be easier to forecast for larger groups, since those tend to have more stable demand shapes. Additionally, spikes in consumption might even out over more consumers.
Creating a distribution forecasting approach and using that for robust planning is a strength of this study. This combination proved useful, and is something that, to the best of our knowledge, has not been seen in related literature before.

However, a few things should be taken into account. Most importantly, tests could only be done on around 100 days. One effect of this could be seen in the significantly lower p-values in the tests with more datapoints, which means that conclusions are limited by testing set size. Additionally, only one season (the summer) has been present in the testing data. To be able to do conclusions of the real world impact, all seasons should be tested.
Results might be less generalizable to other locations. Only the area of Rotterdam, a highly urbanized area, was contained in the data. Conclusions can be different in different, less urbanized, parts of the Netherlands.

In the future, the economical effect of using the batteries should be tested. Energy demand peaks are usually when energy is more expensive. Peak shifting might reduce the total electricity cost, on top of reducing the stress on the electrical grid.
More importantly, renewable forecasting should be included in the peak shifting algorithm. This can be done by incorporating renewables into the energy demand forecasting approach of the first part of this thesis. After that, it is also possible to do research to decoupling parts of the electrical grid in self-sustainable isolated microgrids.
Furthermore, the gap between the online approaches and the Offline case should be ought to be bridged. The gap reflects what improvement is possible if more is known in advance. This gap can be lowered by improving forecasting accuracy and inventing new online approaches.
This could also be combined. Instead of first predicting distribution parameters of the consumption, and then making a schedule based on those parameters, an approach could be designed in which the optimal battery reserves are predicted by a forecasting approach.

Bibliography

[1] Puck Voorhoeve. "Overbelast stroomnet raakt bedrijven en woningbouw." In: *NOS* (May 1, 2022). URL: https://nos.nl/nieuwsuur/artikel/2427174-overbelast-stroomnet-raakt-bedrijven-en-woningbouw (visited on 09/27/2022).

[2] Hanna L van Sambeek, Marisca Zweistra, Gerwin Hoogsteen, Ivo AM Varenhorst, and Stan Janssen. "GridShield—Optimizing the Use of Grid Capacity during Increased EV Adoption." In: *World Electric Vehicle Journal* 14.3 (2023), p. 68.

[3] Rei Syziu, JG Han Slootweg, WF van den Akker, and M Bongaerts. *Smart grid solution on Dutch MV & LV congested networks to increase demand side hosting capacity*. 2022.

[4] A Nottrott, Jan Kleissl, and Byron Washom. "Energy dispatch schedule optimization and cost benefit analysis for grid-connected, photovoltaic-battery storage systems." In: *Renewable Energy* 55 (2013), pp. 230–240.

[5] "Voor het eerst meer stroom opgewekt met zonnepanelen en windturbines dan verbruikt." In: *NOS* (Apr. 28, 2022). URL: https://nos.nl/nieuwsuur/artikel/2427174-overbelast-stroomnet-raakt-bedrijven-en-woningbouw (visited on 09/27/2022).

[6] "Elektriciteitsbalans; aanbod en verbruik." In: *Centraal Bureau voor de Statistiek* (Sept. 26, 2022). URL: https://www.cbs.nl/nl-nl/cijfers/detail/84575NED (visited on 10/05/2022).

[7] Matthias Huber, Desislava Dimkova, and Thomas Hamacher. "Integration of wind and solar power in Europe: Assessment of flexibility requirements." In: *Energy* 69 (2014), pp. 236–246.

[8] Steven Stoft. *Power system economics: designing markets for electricity*. Vol. 468. IEEE press Piscataway, 2002.

[9] European Union. "Directive 2009/72/EC of the European Parliament and of the Council of 13 July 2009 concerning common rules for the internal market in electricity and repealing Directive 2003/54/EC." In: *Official Journal of the European Union* 52 (2009), pp. 55–93.

[10] European Commission. "Cost–Benefit Analyses & State of Play of Smart Metering Deployment in the EU-27." In: *Official Journal of the European Union* (2014).

[11] Ivana Kiprijanovska, Simon Stankoski, Igor Ilievski, Slobodan Jovanovski, Matjaž Gams, and Hristijan Gjoreski. "Houseec: Day-ahead household electrical energy consumption forecasting using deep learning." In: *Energies* 13.10 (2020), p. 2672.

[12] "The harmonized electricity market role model, version: 2022-01." In: *European Network of Transmission System Operators for Electricity* (2022).

[13] Franklin L Quilumba, Wei-Jen Lee, Heng Huang, David Y Wang, and Robert L Szabados. "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities." In: *IEEE Transactions on Smart Grid* 6.2 (2014), pp. 911–918.

[14] SSAK Javeed Nizami and Ahmed Z Al-Garni. "Forecasting electric energy consumption using neural networks." In: *Energy policy* 23.12 (1995), pp. 1097–1104.

[15] Ahmed Z Al-Garni, Syed M Zubair, and Javeed S Nizami. "A regression model for electric-energy-consumption forecasting in Eastern Saudi Arabia." In: *Energy* 19.10 (1994), pp. 1043–1049.

[16] Alexander Lavin and Diego Klabjan. “Clustering time-series energy data from smart meters.” In: *Energy efficiency* 8.4 (2015), pp. 681–689.

[17] Santiago Bañales, Raquel Dormido, and Natividad Duro. “Smart meters time series clustering for demand response applications in the context of high penetration of renewable energy resources.” In: *Energies* 14.12 (2021), p. 3458.

[18] Larisa Lokmic and Kate A Smith. “Cash flow forecasting using supervised and unsupervised neural networks.” In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium.* Vol. 6. IEEE. 2000, pp. 343–347.

[19] Cong Feng, Mingjian Cui, Bri-Mathias Hodge, Siyuan Lu, Hendrik F Hamann, and Jie Zhang. “Unsupervised clustering-based short-term solar forecasting.” In: *IEEE Transactions on Sustainable Energy* 10.4 (2018), pp. 2174–2185.

[20] Kyong Joo Oh and Ingoo Han. “An intelligent clustering forecasting system based on change-point detection and artificial neural networks: Application to financial economics.” In: *Proceedings of the 34th Annual Hawaii International Conference on System Sciences.* IEEE. 2001, 8–pp.

[21] Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Wil L Kling. “Deep learning for estimating building energy consumption.” In: *Sustainable Energy, Grids and Networks* 6 (2016), pp. 91–99.

[22] Shahzad Muzaffar and Afshin Afshari. “Short-term load forecasts using LSTM networks.” In: *Energy Procedia* 158 (2019), pp. 2922–2927.

[23] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. “Short-term residential load forecasting based on LSTM recurrent neural network.” In: *IEEE Transactions on Smart Grid* 10.1 (2017), pp. 841–851.

[24] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. “LSTM network: a deep learning approach for short-term traffic forecast.” In: *IET Intelligent Transport Systems* 11.2 (2017), pp. 68–75.

[25] Jian Cao, Zhi Li, and Jian Li. “Financial time series forecasting model based on CEEMDAN and LSTM.” In: *Physica A: Statistical mechanics and its applications* 519 (2019), pp. 127–139.

[26] Yue-Shan Chang, Hsin-Ta Chiao, Satheesh Abimannan, Yo-Ping Huang, Yi-Ting Tsai, and Kuan-Ming Lin. “An LSTM-based aggregated model for air pollution forecasting.” In: *Atmospheric Pollution Research* 11.8 (2020), pp. 1451–1463.

[27] Xuan-Hien Le, Hung Viet Ho, Giha Lee, and Sungho Jung. “Application of long short-term memory (LSTM) neural network for flood forecasting.” In: *Water* 11.7 (2019), p. 1387.

[28] Azar Rahimi, M Zarghami, M Vaziri, and S Vadhva. “A simple and effective approach for peak load shaving using Battery Storage Systems.” In: *2013 North American Power Symposium (NAPS).* IEEE. 2013, pp. 1–5.

[29] R Hanna, J Kleissl, A Nottrott, and M Ferry. “Energy dispatch schedule optimization for demand charge reduction using a photovoltaic-battery storage system with solar forecasting.” In: *Solar Energy* 103 (2014), pp. 269–287.

[30] Xiangjun Li, Rui Ma, Wei Gan, and Shijie Yan. “Optimal dispatch for battery energy storage station in distribution network considering voltage distribution improvement and peak load shifting.” In: *Journal of Modern Power Systems and Clean Energy* (2020).

[31] Roel JJ Brouwer, J Marjan van den Akker, FPM Dignum, and Wouter Vermeiden. “Forecast-based Optimization of Islanded Microgrids.” In: *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe).* IEEE. 2018, pp. 1–6.

[32] Sarmad Hanif, Md Jan E Alam, Kini Roshan, Bilal A Bhatti, and Juan C Bedoya. “Multi-service battery energy storage system optimization and control.” In: *Applied Energy* 311 (2022), p. 118614.

[33] Matthew P Johnson, Amotz Bar-Noy, Ou Liu, and Yi Feng. "Energy peak shaving with local storage." In: *Sustainable Computing: Informatics and Systems* 1.3 (2011), pp. 177–188.

[34] Souhaib Ben Taieb, Gianluca Bontempi, Amir F Atiya, and Antti Sorjamaa. "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition." In: *Expert systems with applications* 39.8 (2012), pp. 7067–7083.

[35] Navin Kumar Manaswi and Navin Kumar Manaswi. "Rnn and lstm." In: *Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras* (2018), pp. 115–126.

[36] Sakib Ashraf Zargar. "Introduction to sequence learning models: RNN, LSTM, GRU." 2021.

[37] Stephen Haben, Jonathan Ward, Danica Vukadinovic Greetham, Colin Singleton, and Peter Grindrod. "A new error measure for forecasts of household-level, high resolution electrical energy consumption." In: *International Journal of Forecasting* 30.2 (2014), pp. 246–256.

[38] Ángela Herraiz-Cañete, David Ribó-Pérez, Paula Bastida-Molina, and Tomás Gómez-Navarro. "Forecasting energy demand in isolated rural communities: A comparison between deterministic and stochastic approaches." In: *Energy for Sustainable Development* 66 (2022), pp. 101–116.

[39] Richard P Brent. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

[40] Scott Haney. "Practical applications and properties of the Exponentially Modified Gaussian (EMG) distribution." PhD thesis. Drexel University, 2011.

[41] Manuel Blum and Martin A Riedmiller. "Optimization of Gaussian process hyperparameters using Rprop." In: *ESANN*. Citeseer. 2013, pp. 339–344.

[42] Pushparaja Murugan. "Hyperparameters optimization in deep convolutional neural network/bayesian approach with gaussian process prior." In: *arXiv preprint* (2017).

[43] Miao Zhang, Huiqi Li, Juan Lyu, Sai Ho Ling, and Steven Su. "Multi-level CNN for lung nodule classification with Gaussian Process assisted hyperparameter optimization." In: *arXiv preprint* (2019).

[44] Henok Ayele Behabtu, Maarten Messagie, Thierry Coosemans, Maitane Berecibar, Kinde Anlay Fante, Abraham Alem Kebede, and Joeri Van Mierlo. "A review of energy storage technologies. Application potentials in renewable energy sources grid integration." In: *Sustainability* 12.24 (2020), p. 10511.

[45] Tesla. *Powerwall 2 Datasheet*. Accessed: 5-5-2023.

Appendix A

Additional Figures

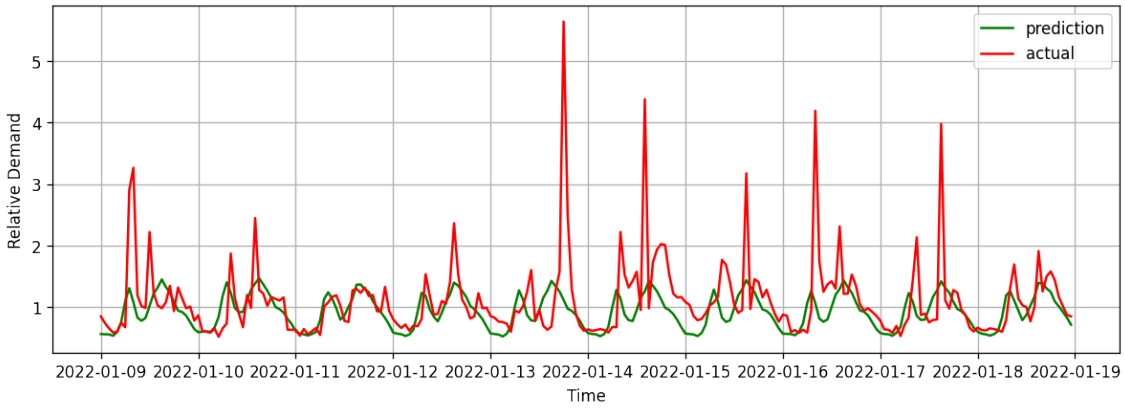


Figure A.1: Predicted and actual demand of one consumer for 10 days in January 2022 (test set).

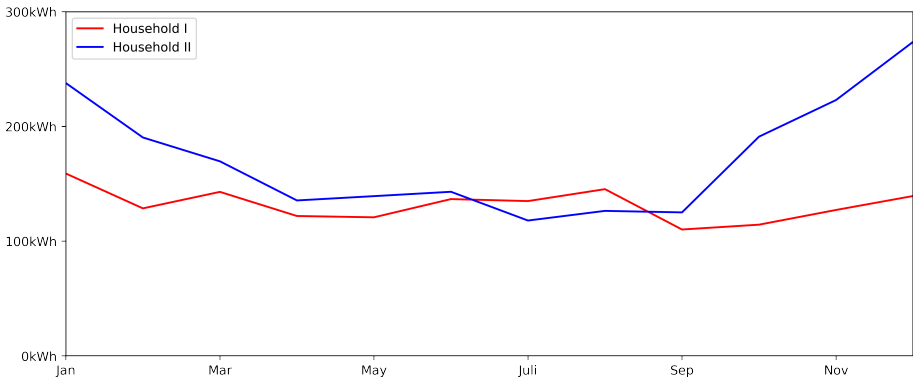


Figure A.2: Differences in seasonal effect between two households. One has a higher relative demand during the winter month, and the other does in the summer month.

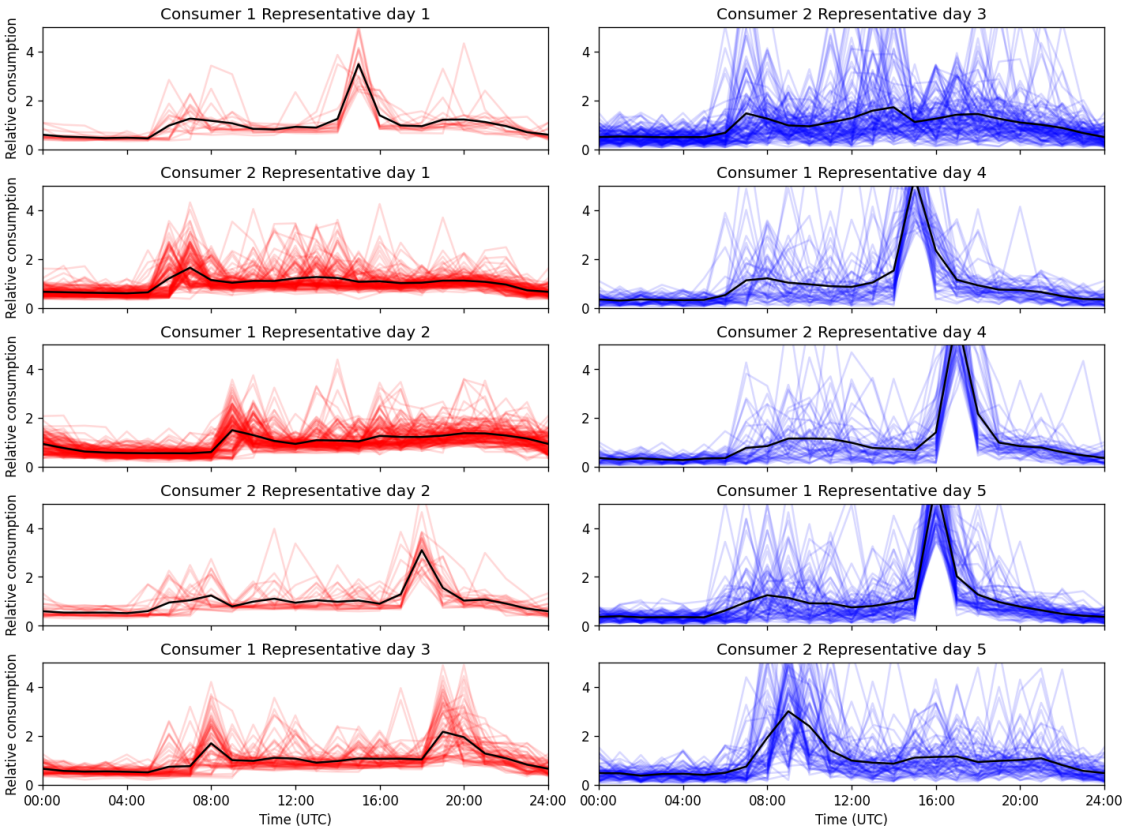


Figure A.3: Visualization of the result of representative days clustering for two consumer and five representative days per consumer. Days in a cluster are plotted in red and blue, and the resulting representative day is plotted with a black line.

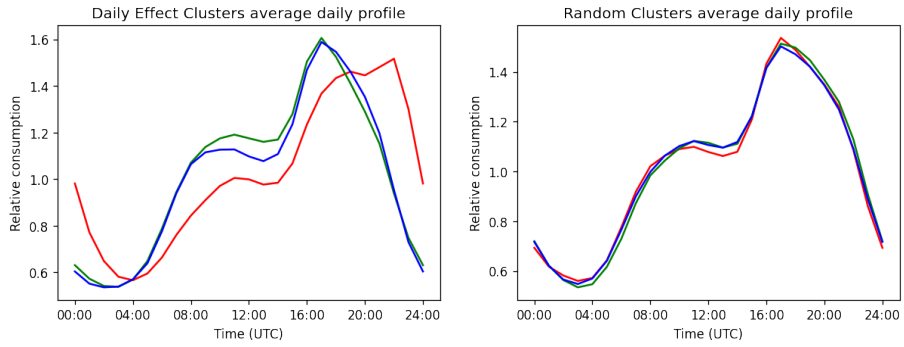


Figure A.4: Average demand per day of three clusters made on basis of representative days (left), and of three randomly formed clusters (right). The average of the randomly formed clusters look similar, but the averages of the clusters made with representative days are different.

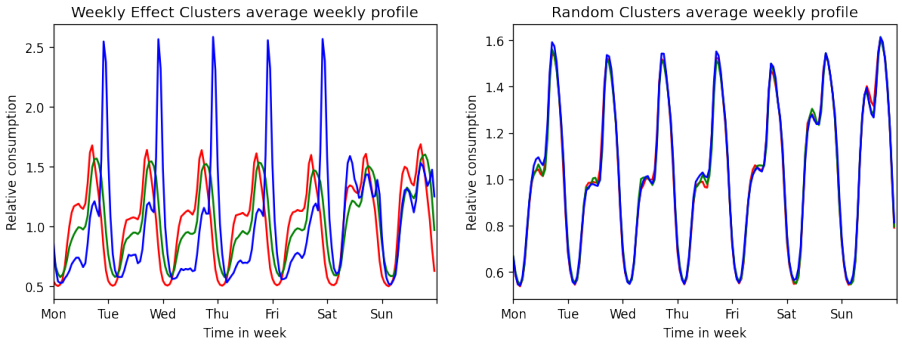


Figure A.5: Average weekly profiles of three clusters made on basis of the weekly profile (left), and of three randomly formed clusters (right). The average of the randomly formed clusters look similar, but the averages of the clusters made on basis of weekly profiles are different.

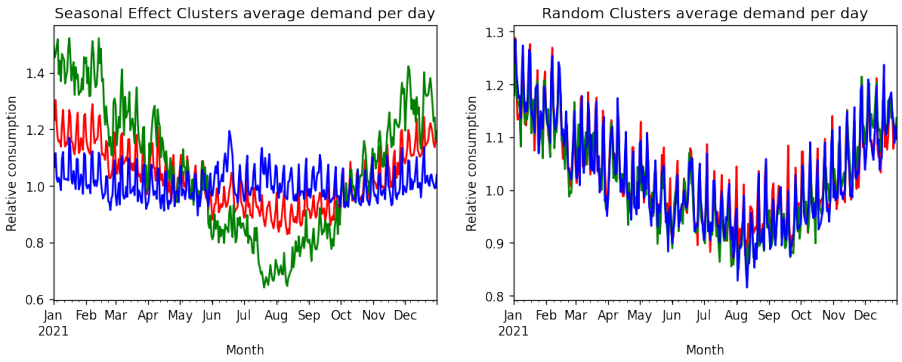


Figure A.6: Average total daily demand during the year of training data of three clusters made on basis of total weekly consumption (left), and of three randomly formed clusters (right). The average of the randomly formed clusters look similar, but the averages of the clusters made on basis of monthly energy usage are different.

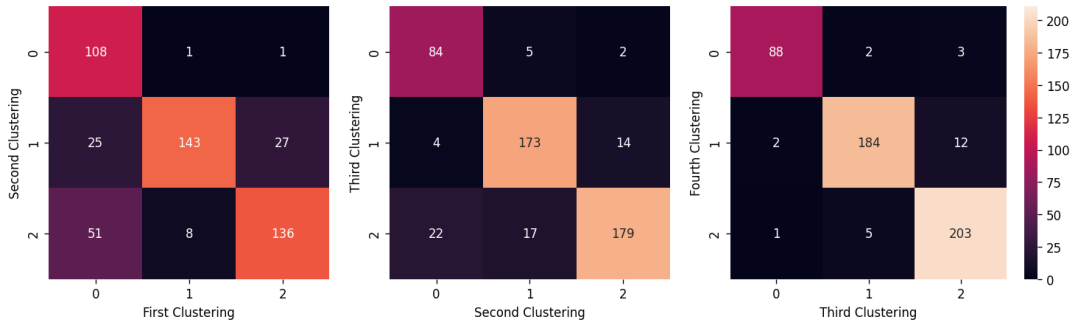


Figure A.7: Mutation matrices for the changes for three ReCluster steps for three clusters.

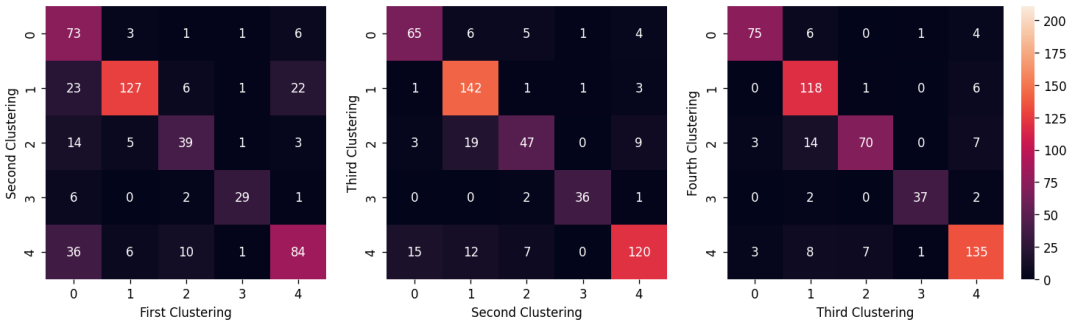


Figure A.8: Mutation matrices for the changes for three ReCluster steps for five clusters.

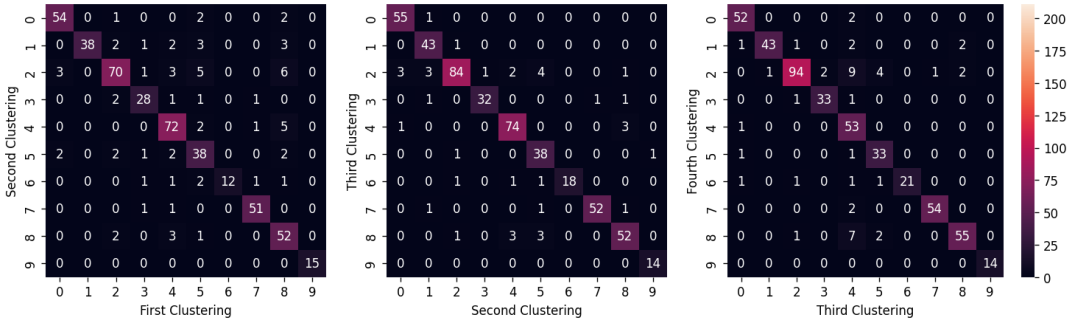


Figure A.9: Mutation matrices for the changes for three ReCluster steps for ten clusters.

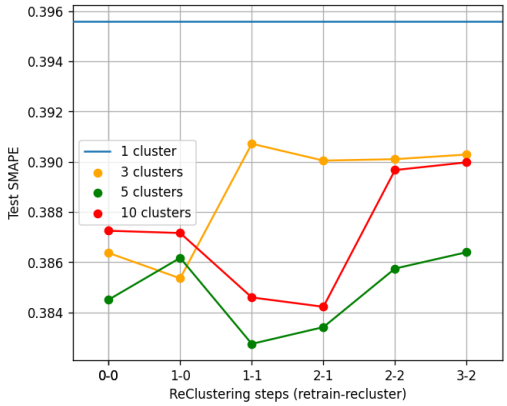


Figure A.10: Performance of the ReClustering approach on 1 (blue), 3 (yellow), 5 (green) and 10 (red) clusters, based on SMAPE on the Testing dataset. The x-axis notation has two numbers. The first represents the number of recluster steps, and the second represents the number of retraining steps.

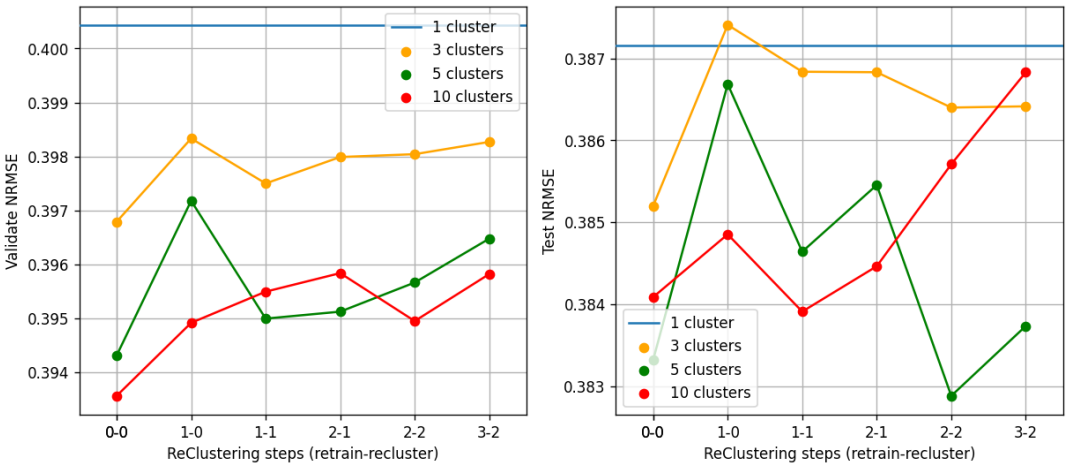
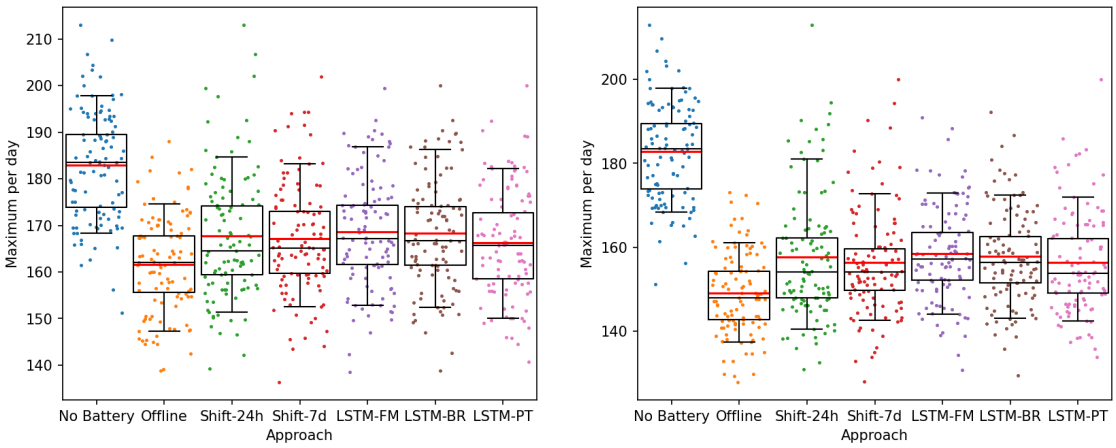


Figure A.11: Performance of the ReClustering approach on 1 (blue), 3 (yellow), 5 (green) and 10 (red) clusters, based on NRMSE on the Validation dataset (left) and the Testing dataset (right). The x-axis notation has two numbers. The first represents the number of recluster steps, and the second represents the number of retraining steps.



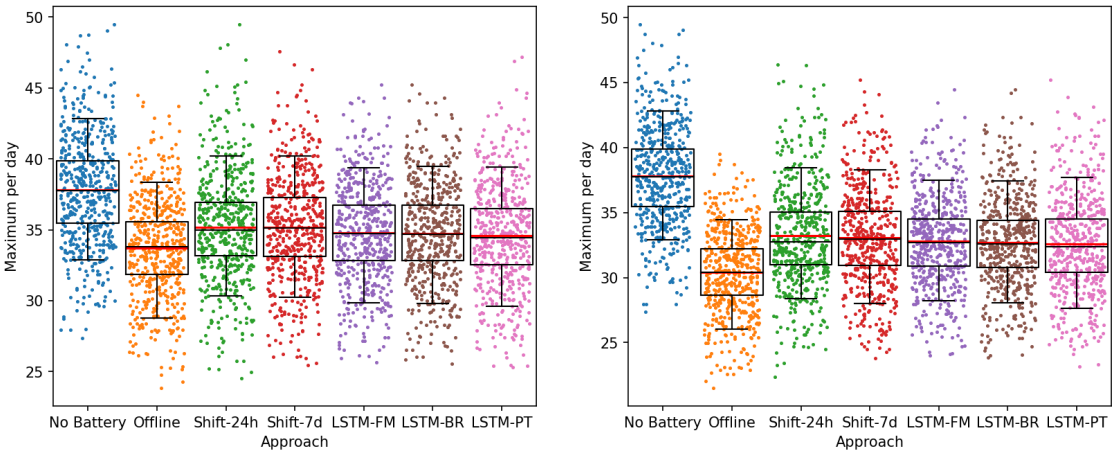
(a) Five batteries.

(b) Ten batteries.

Figure A.12: The test cases with all 500 consumers and five batteries (left) or ten batteries (right) for the case where no battery is used, the offline case and the five other approaches. Each dot represents the maximum energy demand of a day in the data. The mean daily bandwidth per approach is plotted in red lines, and boxplots are visualized for the 10%, 25%, 50%, 75% and 90% percentiles.

	All 5	Groups 5 Mean	Groups 5 Total	All 10	Groups 10 Mean	Groups 10 Total
Offline	11.6%	10.9%	9.4%	18.5%	19.6%	17.5%
Shift-24h	8.3%	6.9%	7.5%	13.7%	12.2%	13.6%
Shift-7d	8.6%	7.0%	7.3%	14.5%	12.6%	14.1%
LSTM-FM	7.8%	8.0%	7.9%	13.4%	13.3%	13.6%
LSTM-BR	8.0%	8.1%	8.0%	13.7%	13.5%	13.8%
LSTM-PT	**9.1**%	**8.6**%	**8.7**%	**14.5**%	**13.8**%	**14.4**%

Table A.1: Percentage of reduction in maximum energy demand per approach and per test case. The best online approach per test-case is written in bold.



(a) One battery.

(b) Two batteries.

Figure A.13: The test cases with 5 groups of 100 consumers and one batteries (left) or two batteries (right) for the case where no battery is used, the offline case and the five other approaches. Each dot represents the maximum energy demand of a day in the data for one of the five groups. The mean daily bandwidth per approach is plotted in red lines, and boxplots are visualized for the 10%, 25%, 50%, 75% and 90% percentiles.