Improving the efficiency of an energy management system with machine learning

Richard Wijers
Supervisors: Valentin Robu, Sho Cremers , Peter Zhang
EEMCS, Delft University of Technology, The Netherlands

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Improving the efficiency of an energy management system with machine learning

Richard Wijers
Supervisors: Valentin Robu , Sho Cremers , Peter Zhang
EEMCS, Delft University of Technology, The Netherlands

Abstract

This paper explores the possibility of using machine learning to improve the profits generated by an energy management system for so called prosumer households. Which are households with their own energy production, consumption, and storage. The commonly used deterministic algorithms only take the current data into account when deciding on how much power to buy or sell. But, if there are flexible energy costs and future energy consumption and production data is available, better choices can be made. For example, you could store energy at a cheaper time for later usage when the prices are higher. To make these decisions, a new algorithm was created. In this paper different machine learning models are evaluated for generating the future data. The final simulations will use the improved algorithm use the data predicted by these models. For a prosumer with a 10 kWh battery and which produces 70% of its own total energy consumption, the machine learning algorithm decreased the total energy cost by around 7%. While this does show that machine learning is a viable option to increase the efficiency of an energy management system, there is still much improvement possible to get closer to the theoretical 24% reduction.

1 Introduction

With renewables becoming increasingly popular [IEA, 2021], the demand for an efficient energy management system also grows. For a prosumer household, which is a household with its own energy production and storage, an energy management system monitors the energy being consumed and produced, and it makes decisions on when to buy and sell energy. By making sure that the system buys and sells energy on the most profitable times, we are not only able to increase the profits that the complete system generates, but we also increase the environmental benefits due to a more stable energy grid [Silva et al., 2020].

Most current system still make use of deterministic algorithms, but this research project explores the possibility of using machine learning to improve the efficiency. The baseline deterministic algorithm that is used for comparison is described in the following paper [Norbu et al., 2021]. It makes use of the basic approach of storing excess energy until the batteries are full and using this stored energy when there is more demand than production. If the battery is already completely full or empty, the extra energy will be sold or bought respectively. This approach works perfectly with the assumption of a consistent price for buying and selling energy, but when also considering flexible prices, it might be better to store the extra energy when the price is low and use it at a later time when the price is higher.

To research if machine learning can be used to improve the efficiency of an energy management system, the project will be divided into three parts. The first part of the research project will be about creating a more efficient energy management system which assumes perfect knowledge about the future energy demand, production, and prices. Depending on the circumstances, it must be able to decide how much energy to buy or sell energy to decrease the total energy bill of a household. Then the second part will be about how accurate machine learning methods are for predicting the future energy data. The machine learning models that will be evaluated for extending the timeseries data are ARIMA, ETS, K-neighbors and Neural networks. The final part will combine this generated future data with the improved energy management system and run multiple simulations on prosumers with the baseline and the improved algorithm. comparing these results will show if machine learning is a viable option to increase the efficiency of energy management systems.

The structure of the report is as follows. Firstly, section 2 will give a formal description of the research problem, then section 3 contains more information on the improved algorithm works and what machine learning models have been used. Section 4 shows how the experiment has been setup and what the results are, with section 5 discussing if the gathered results are valid. Then finally section 6 concludes the project and discusses future work.

2 Problem Description

The main aim of this research project is to determine if machine learning can improve the efficiency of an energy management system. These energy management systems are needed in households where there is not only energy consumption, but also energy storage and production. These so-called prosumer households need to make decisions on what to do with their energy at a certain time. For example, they can either store or sell excess energy and either buy energy or withdraw from the battery when there is more energy demand than production. The baseline algorithm from the paper [Norbu et al., 2021] uses a relatively simple deterministic approach of only selling or buying power when battery is already completely full or empty as shown in figure 1.

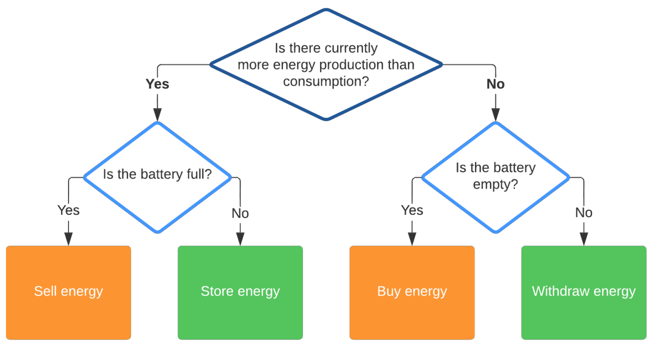

Figure 1: Flowchart of the baseline algorithm

However, this approach only works optimally when the buy and sell prices of energy are constant. And due to the energy prices fluctuating based on the total energy production and consumption on the grid, this is not always the case. If we use the baseline algorithm with these changing energy prices, it becomes apparent that there is room for improvement. Consider for example a prosumer with a consumption of 1000 watts for two hours and a battery 1 kWh of energy stored, then if the first hour the energy price is 20 cents per kWh and the second hour 30 cents, it is sensible to use the 1 kWh of stored power at the second hour and buy 1 kwh at the first hour to use directly. By making this decision the energy cost will be 20 cents instead of 30 cents.

There will be multiple problems that need to be answered to make a proper conclusion. The first problem will be to create an improved algorithm that is able to make these optimal decisions. This improved algorithm will need to be able to consider the future energy prices and monitor the battery levels to buy and sell energy at the most profitable times. Then the second problem will research the ways of getting the future data. For a given prosumer, the energy consumption, production, and prices must be predicted accurate enough for the final algorithm to use. The final problem will be assessing a combination of the improved algorithm and the predictions from the machine learning models, and the results will be compared against the baseline algorithm. Since this report will only show the differences between the two algorithms, the purchase prices for the energy systems like the battery and the wind generator are the same for both simulations and will thus not be included in the calculated energy costs.

3 Improved energy management system

To determine if machine learning can have an impact on the efficiency of an energy management system, there are multiple parts that will need to be researched. Firstly, an algorithm will need to be designed to make the optimal decisions based on future data. Then machine learning can be used for generating data. And finally, the improved algorithm will be compared against the baseline algorithm.

3.1 Improved algorithm

The main algorithm for deciding when to buy or sell energy uses the greedy approach. The simulation will start for a prosumer with the battery lever at 0% and using the future energy production and consumption data, the battery level for the next time points will be calculated. This will be continued until the algorithm finds a point n where the battery will run out of energy or overflow. Then the algorithm will go back in time from this point n searching for the optimal point m to buy or sell the required energy while keeping in mind the limits of the battery. When the optimal time has been found, the possible adjustment will be made at the point m and the battery level will be changed accordingly, now the algorithm can continue at point n until it reaches the end or needs to make another adjustment. If, when searching for the optimal price, the algorithm finds the battery already at its limit, it will use the best price found so far. The process of making an adjustment is shown in figure 2. When the algorithm finds the battery level negative (requires grid energy), it will search for the cheapest price. This price is at timepoint 3 and thus the algorithm will buy the extra energy at this timepoint and make the adjustment.

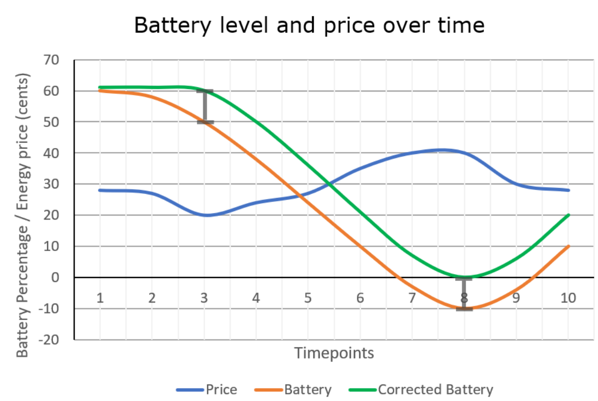

Figure 2: Improved algorithm buying energy at a cheaper timepoint

The algorithm keeps track of the amount of energy bought, sold, and stored at each timepoint. With this information the total energy costs of the system can be calculated. To take battery efficiency losses and battery degradation into account the prices used, when searching for the best price, will be altered to reflect this. If at a certain timepoint the energy price is 10 cents and we have a usage of 200 watts, these first 200 watts can be bought for 10 cents per kWh, but if more power is required it will be stored in the battery instead. When the battery is used to store extra power, there will be a so-called levelized cost of storage. This LCOS value is the extra cost per kWh when storing power. On current batteries this cost is still extremely high but is predicted to decline in the upcoming years [Vonsien and Madlener, 2020]. For this research project, the LCOS value is set to a relatively low 25 cents per kWh, resulting for the example that all the energy over 200 watts will cost 35 cents per kWh instead of 10 cents. This results in only rare cases where the battery will charge from the grid. For the simulation we assume that a surplus of energy for the wind generator will always be used to charge the battery. In conclusion, most of the time the charging of the battery at an earlier timepoint will instead come down to not withdrawing energy from the battery and letting it charge by the wind generator.

3.2 Machine learning

To generating accurate future data there are many possible methods. For this project two main methods have been used Firstly, there are the algorithms which find certain trends or averages in previous data. These models give good average predictions but are less sensitive for sudden peaks and take a lot of data to train. Examples of these models are:

- ETS: Exponential smoothing models are able to find the Error, Trend and Seasonal components of a dataset (Where error is the component that is left after removing the Trend and seasonal components). These components can then each be combined by adding, multiplying, or even not including it at all. By combining these components in various ways and with different parameters, the performance of the model can be improved. For this research project, the AutoETS model from [Löning et al., 2019] is used to minimize the amount of parameter tuning required.
- ARIMA: Instead of making predictions based on the actual values, this model uses the differences between concurrent values to predict if the next value will increase or decrease. While the default version is not able to effectively use seasonal trends, the AutoARIMA model from [Löning et al., 2019] also uses a seasonal component.

Then there are the algorithms which are able to recognize situations. These algorithms can take a n number of inputs and are trained to give a corresponding result. When predicting time series data, this property is utilized by using a so-called sliding window as seen in figure 3. A sliding window has a certain length n, and this window will move over one position at a time to go over a set of data. Using these n items as input for the machine learning algorithm, we can predict the next datapoint and move the sliding window over to include this new datapoint. Then using this predicted datapoint another future point can be predicted. This process can be continued until the wanted number of points has been predicted.

This type of algorithms is generally better at predicting sudden spikes and run faster because they can for example be trained on a subset of the available data and still produce good results. Examples of these models are the following models:

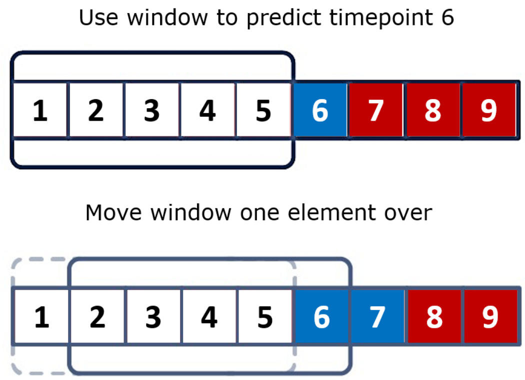

Figure 3: Sliding window with a window length of 5

- Neural network: This model makes use of layers of nodes connected to each other. Starting at the input layer, the nodes will have an activation value that corresponds with the inputs of the algorithm. These activations will then be passed on to the connected nodes of the next layer with different weights, giving the nodes in the next layer also a certain amount of activation. This process will be continued to the final output layer, which gives the corresponding output to the input. The model is able to train on previous data by changing the weights of the connections until it has the right results. By changing for example the number of nodes and layers in the model, better performance can gained.
- K-neighbors: Given a certain input, this model compares this to the previous gathered data and sorts the data on the closest matching situations. Then the model can take the K number of most similar situations and depending on the configuration, it takes the uniform average of those situations or calculates the relative distance to each and uses this to create a prediction.

Finally, for this research project there are also specific models created for the energy production predictions. Because the dataset contains production data based on a windmill, the prediction model is split into two parts. Firstly, there is a neural network that takes as input the last n datapoints for the wind speed, bearing, humidity, pressure, and temperature. As output the models has a k number of nodes, which correspond to the k number of next datapoints to predict. By training this model on past data, the neural network is able to make a direct prediction for the upcoming k datapoints of the wind speed. Then the second part is a function fitting algorithm that is able to find a correlation between given x and y values. This model is used for finding the corresponding energy production data for each wind speed. Thus, by first using the neural network, a prediction for the next k wind speed datapoints can be created. And those wind speeds can be converted to the right energy productions by using the function fitting algorithm.

3.3 Comparison

When comparing the baseline algorithm with the improved algorithm, a set of prosumers will be used with each their own energy production, demand, and storage. These prosumers can be customized to evaluate different setups. The ratio between their produced and consumed energy can be changed for each prosumer. Furthermore, the battery levels and the length of the simulation can be changed. The results of these simulations will be the total energy bills of the prosumers.

4 Experimental Setup and Results

- Demand: This dataset from [UK power networks, 2014] contains the power demand data of 5567 households in London over a period of 2.5 years. The datapoints are given at an interval of 30 minutes. The dataset has been processed by taking the average demand for a timepoint at a specific date and using linear interpolation to fill in missing entries. The resulting data only spans for one year.
- Price: The datasets for both the import and export energy prices are from [Octopus Energy, 2020]. The average import price is 18 cents with a 10th percentile of 7 cents and a 90th percentile of 40 cents. The average export price is 11 cents with a 10th percentile of 4 cents and a 90th percentile of 23 cents.
- Weather: The weather data is gathered from the Kirkwall airport weather station located in Orkney, Scotland and provided by [UKERC Energy Data Centre, 2020]. The wind measurements are taken at a height of 26 meters above ground.
- Production: The power production data is based on the weather data. To calculate the power production, the same sigmoid function as in the paper [Norbu et al., 2021] is used. The parameters of this sigmoid are a=0.756 s/m and b=8.424 m/s with the following equation, with u as the input wind speed.

$$f(u, a, b) = \frac{1}{1 + e^{-a(u-b)}}$$

Using these datasets, a list of prosumers can be created with different parameters. By setting different parameters we are able to change the battery size of a prosumer, the amount of energy production of a prosumer relative to its energy consumption and length of the simulation. The amount of energy production is set by a ratio variable. The energy production pattern will be based on the production dataset, but it will be scaled up or down according to the ratio. So, if a prosumer has a total energy consumption of 100 kWh for a simulation and a ratio of 70%, the total energy production of this prosumer will be 70 kWh.

After a n number of prosumers has been created from the datasets, there will be a simulation run with both the baseline algorithm and the improved machine learning algorithm. The improved machine learning algorithm is a variation on the algorithm described in section 3.1, instead of having an unlimited future vision range, now the range will be limited to length of the predictions. So, for each time point i the machine learning algorithms will generate future data of a length k, and then the improved algorithm determines the best timepoints to buy or sell energy. If the algorithm returns that there should be an adjustment at the current time i, the adjustment will be made. It will then continue to the next time point i and do the same again. The corresponding machine learning model for each dataset of the prosumer will be selected based on the performance of the different models. During the simulations the training data will be the past data, so for the first k timepoints the algorithm will only collect data and not yet make predictions. After all the models have gathered enough data, the predictions will start and the improved algorithm will be used. The prediction length in all the experimental simulations is 24 hours (48 timepoints)

The parameters of the models have mostly been chosen on intuition and trial and error, since parameter tuning a machine learning model perfectly for a dataset can be a research project on its own. Since the AutoETS and AutoARIMA models from sktime [Löning et al., 2019] do some parameter tuning themselves, only the seasonal period has been set. For this value, a full day is used (48 datapoints of 30 minutes). This is because most patterns repeat daily and if this value is set higher, there is also much more data needed for fitting the models. For the K-neighbors and neural network models, the window length has been set to 100. This value is again chosen to be a good tradeoff between using the largest amount of available data (the last 100 timepoints) and not requiring too much data to train the model. The k-neighbors model uses 24 neighbors and a uniform weight, which are chosen based on trial and error. Lastly, the neural network uses one hidden layer with 100 nodes and the maximum number of iterations has been increased to 500 to increase the training performance. There are many more parameters that could be altered, but for the scope of this project those have been left at the default settings. The performance of the models is measured by using the RMSE. When a model makes a prediction, the predicted data is compared to real values. The error for each timepoint is squared and these squared values are then summed together into a value x. By dividing x by the length of the prediction, we get the MSE score. Then by taking the square root of the MSE, the RMSE score is calculated. The RMSE score gives an indication of how far away the predictions are and it punishes models with higher variance more due to squaring the error. Since the models will be run multiple times, a normal distribution is created of the resulting RMSE scores. Next the performances of each of the prediction models on the different data sets will be discussed.

4.1 Demand

Since the demand of a prosumer often takes on a certain pattern (for example more usage in the morning and evening), all of the models perform well. However, the situation recognition models are better at adjusting to, or even predicting energy spikes, which is very important to make sure the simulation stays accurate enough to make good decisions.

As seen in table 1, the k-neighbors is has both the lowest RMSE score and the fastest runtime. Thus, for the total simulation this model will be used.

4.2 Prices

The patterns for the energy prices are similar to the energy demand, but possibly due to the much higher volume of dif-

ML algorithm	Mean	Std	Runtime (s)
ETS	196.6	85.8	283.7
ARIMA	169.2	68.1	16207.3
Neural network	196.3	94.7	42.9
K-neighbours	159.8	73.6	1.7

Table 1: RMSE scores of the demand in watts for 100 predictions

ferent prosumers and consumers having effect on the prices, they are relatively more stable, and thus the models tend to perform better. The models are evaluated on both the import and export prices datasets.

ML algorithm	Mean	Std	Runtime (s)
ETS	0.87	0.478	99.3
ARIMA	1.12	0.69	16405.8
Neural network	1.70	1.09	17.7
K-neighbours	1.54	0.94	0.9

Table 2: RMSE scores of the prices in cents for 50 predictions

While table 2 shows that the ETS model performs by far the best on the dataset, for the improved algorithm, it is more important that the timepoints with the highest and lowest prices are predicted accurately than the actual amounts to make good decisions. And, due to the much faster runtime of the k-neighbors model, this model is instead used for the final simulation.

4.3 Production

For the wind energy production it is by far the hardest to create an accurate prediction since there is not a real trend or reoccurring situation if we only look at the past wind speeds. Therefore, the extra weather dataset is used. Using the purpose build neural network described in section 3.2 with the last 3 weather timepoints as input, the model predicts the wind speed predictions for the next 24 hours (48 timepoints of 30 minutes). This neural network uses 3 hidden layers with 250 nodes, and the ReLu function as activation. The function fitting algorithm is then able convert the wind speeds to energy production. The RMSE score for the neural network is based on the error for the wind speed predictions and the RMSE score for the function fitting model is the error in energy predictions given perfect wind speed data.

In contrast to the other models, past wind data should already be available for training and thus create accurate predictions from the start. To mimic this effect, the wind model is trained once at the start of the simulation on 5% of the whole dataset and is then used for all the predictions.

ML algorithm	Mean	Std	Runtime (s)
*Neural network	0.495	0.283	84.6
Energy correlation	0.032	0.121	21.63

Table 3: RMSE scores of production in m/s and watts for 1000 predictions

In table 3 we can see the RMSE of the wind speed and the energy production models. The RMSE scores for the energy correlation appear to be extremely low. This is likely because of how the wind energy dataset is generated. Since there is a perfect sigmoid correlation between the wind speed and energy production, the model is easily able to find this correlation.

4.4 Total simulation

To determine if machine learning is a viable option to increase the efficiency of an energy management system, the baseline algorithm is compared against the improved algorithm with both perfect data and machine learning data. As stated before, the prediction length for all the experiments is set to 24 hours (48 timepoints). The final run uses the k-neighbors model for the energy demand and energy prices due to the good performance and faster runtimes. For the energy production the models in section 4.3 are used. The battery size for each prosumer is set to 10 kWh. Furthermore, the simulation is run on 100 different households for 1000 timepoints, so for 500 hours. Then the total cost per prosumer is calculated for both algorithms and compared, the final costs only include the energy costs and battery inefficiencies. The base cost for the whole setup is not included since they are the same for both algorithms.

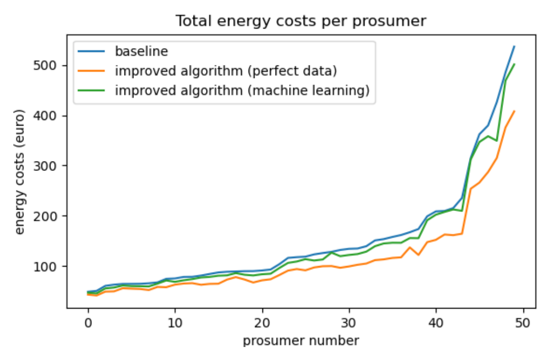

Figure 4: Comparison of the algorithms for 100 prosumers

As shown in figure 4, the improved algorithm using machine learning has a relative improvement on average of around 6.8%. While this does show a positive result, it is still only small amount compared to the possible 23.2% improvement.

5 Responsible Research

To show that the results are significant enough we can use a normal distribution of the results as shown in figure 5. This figure contains the relative improvements of the 100 prosumers. By integrating the normal distribution for all improvements below 0%, we get a chance of 0.9%. Therefore, we can conclude that there is a negligible chance that the results were based on a lucky run.

While the gathered results show a significant improvement, there are still some concerns that might arise. Firstly, the starting dates of the datasets used do not align properly, making the simulation not fully represent real world data. However, since this report is about the comparison between two

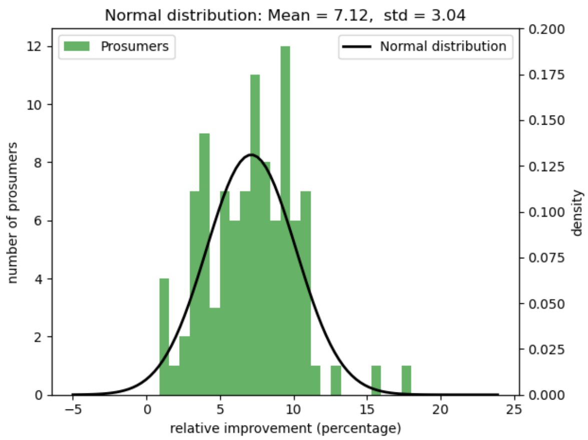

Figure 5: Normal distribution of the final results

algorithms which both run on the same data, the result is still valid. Secondly, the training data for the wind speed forecasting partially overlaps with the testing data. This increases the possibility that the neural network is overfitting for the testing data and thus the performance increases. Furthermore, the energy production of the wind generator is artificially created. However, since there are other papers [O'Brien and Ralph, 2015] which discuss forecasting wind generated power and achieved a 97% accuracy score for 24-hour predictions, it is clear that other researchers have even acquired better results and thus the results of this report are still valid.

6 Conclusions and Future Work

The main research question, if the efficiency of an energy management system can be improved with machine learning, is shown to have a positive result. Firstly, for each of the different datasets, it is shown that a machine learning model is able to make accurate predictions. Then combining these predictions with the improved algorithm, a relative improvement of 7% is calculated.

However, while this report does explore a few different timeseries prediction models, there are still many other methods out there and, with more in dept data analysis, better parameters can be selected. This is made obvious by the difference between the possible and achieved improvements. Another interesting future project would be to try and increase the efficiency of the energy management system in a deterministic way. The baseline algorithm in this report is relatively basic and there might be better performing options.

Finally, it will be interesting to see how battery technology will advance. By decreasing the degradation cost of storing energy, the improved algorithm will be able to make more use of the cheaper import prices and the relative improvements will increase drastically.

References

[IEA, 2021] IEA (2021). Renewable electricity growth is accelerating faster than ever worldwide, supporting the emergence of the new global energy economy.

[Löning et al., 2019] Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., and Király, F. (2019). *sktime: A Unified Interface for Machine Learning with Time Series*.

[Norbu et al., 2021] Norbu, S., Couraud, B., Robu, V., Andoni, M., and Flynn, D. (2021). *Modelling the redistribution of benefits from joint investments in community energy projects*. Applied Energy.

[O'Brien and Ralph, 2015] O'Brien, E. and Ralph, A. (2015). Operational evaluation of a wind-farm forecasting system. *Energy Procedia*, 76:216–222. European Geosciences Union General Assembly 2015 - Division Energy, Resources and Environment, EGU 2015.

[Octopus Energy, 2020] Octopus Energy (2020). Agile octopus tariff. https://octopus.energy/agile. Accessed: 2022-06-19.

[UK power networks, 2014] UK power networks (2014). Smartmeter energy consumption data in london households. https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households. Accessed: 2022-06-19.

[UKERC Energy Data Centre, 2020] UKERC Energy Data Centre (2020). Midas: Uk mean wind data. ncas british atmospheric data centre. https://catalogue.ceda.ac.uk/uuid/a1f65a362c26c9fa667d98c431a1ad38. Accessed: 2022-06-19.

[Silva et al., 2020] Silva, B. N., Khan, M., and Han, K. (2020). Futuristic sustainable energy management in smart environments: A review of peak load shaving and demand response strategies, challenges, and opportunities. *Sustainability*, 12(14).

[Vonsien and Madlener, 2020] Vonsien, S. and Madlener, R. (2020). Li-ion battery storage in private households with pv systems: Analyzing the economic impacts of battery aging and pooling. *Journal of Energy Storage*, 29:101407.