

# The impact of online machine-learning methods on long-term investment decisions and generator utilization in electricity markets

Alexander J. M. Kell, A. Stephen McGough, Matthew Forshaw

*School of Computing, Newcastle University, Newcastle-upon-Tyne, United Kingdom*

---

## Abstract

Electricity supply must be matched with demand at all times. This is to maintain a high-quality service to customers, as well as for load-frequency control. To gain a better understanding of the load that may be required in the future, estimations under uncertainty are needed.

In this paper, we use different machine learning and statistical methods to predict electricity demand 24 hours ahead to simulate a day-ahead market. Once we have made the predictions, we sample from the residual distributions and perturb the electricity market in a long-term agent-based model, ElecSim. This enables us to understand the impact of errors on the long-term dynamics of a decentralized electricity market.

Our results show that we are able to reduce the mean absolute error by 30% using an online algorithm when compared to the best offline model, whilst reducing the required tendered national grid reserve required. We also show that large errors in prediction accuracy have a disproportionate error on investments made over a 17-year time frame, as well as electricity mix.

**Keywords:** Long-Term Energy Modelling, Online learning, Machine learning, Market investment, Climate Change

---

## 1. Introduction

The integration of higher proportions of intermittent renewable energy sources (IRES) in the smart grid will mean that the forecasting of electricity demand will become increasingly important. This is due to the fact that supply must mean demand at all times and that IRES are less predictable than dispatchable energy sources such as coal and combined-cycle gas turbines (CCGTs) [1].

Typically, peaker plants, such as reciprocal gas engines, are used to fill fluctuations in demand, that had not been previously planned for. These peaker plants are typically expensive to run and have higher greenhouse gas emissions than their non-peaker counterparts [2].

To reduce reliance on peaker plants, it is helpful to know how much electricity demand there will be in the future so that more efficient plants can be used to meet this demand. To aid in this, machine learning and statistical techniques have been used to accurately predict demand based on several different factors and data

sources, such as weather, day of the week and holidays [3, 4, 5, 6].

While various studies have looked at how to predict electricity demand at various time horizons with the highest accuracy [7], the impact that the differing methods used have on a long-term electricity market has been studied to a lesser degree. In this paper, we compare several machine learning and statistical techniques to forecast 24 hours ahead to simulate a day-ahead market.

In addition to this, we use the long-term agent-based model, ElecSim [8, 9], to simulate the impact of different forecasting methods on long-term investments, power plant usage and carbon emissions from the year 2018 to 2035 in the United Kingdom.

To compare the impact of different methods, we trial both online and offline machine learning and statistical techniques. Online learning methods can learn from novel data while maintaining what was learnt from previous data. This type of statistical modelling is useful for non-stationary datasets, and time-series data where recalculation of a model would take a prohibitive amount of time. We trial different algorithms and split times of the year into seasons, weekdays, weekends and

---

*Email address:* a.kell2@newcastle.ac.uk (Alexander J. M. Kell)

holidays.

Using these methods, we take the error distributions, or residuals, and fit a variety of distributions. The distribution that best fits the respective residuals is then used and sampled from to adjust the demand in the model ElecSim. We then observe the differences in carbon emissions, and power plants invested in and utilised with each of the different statistical and machine learning methods to a base case.

We show that online learning has a significant impact on reducing the error for predicting electricity consumption a day ahead when compared to traditional offline learning techniques.

We show that the forecasting algorithm has a non-negligible effect on carbon emissions and use of coal, onshore, photovoltaics, reciprocal gas engines and CCGT. Specifically, the amount of coal, photovoltaics, and reciprocal gas used from 2018 to 2035 was proportional to the median absolute error, while both onshore and offshore wind are inversely proportional to the median absolute error.

Total investments in coal, offshore and photovoltaics are proportional to the median absolute error, while investments in CCGT, onshore and reciprocal gas engines are inversely proportional.

In Section 2 we review the literature. We introduce the model ElecSim, and the methods used in Section 3. The Methods section is shown in Section 4, results in Section 5. We discuss and conclude our work in Sections 6 and 7 respectively.

## 2. Literature Review

Multiple papers have looked at demand-side forecasting [7]. These include both artificial intelligence [10, 11, 12] and statistical techniques [13, 14]. However, the impact of online learning has been discussed with less frequency. In addition to this, our research shows the impact of different algorithms on investments made, electricity sources contribution and carbon emissions over a 17 year period using the long-term electricity market agent-based model, ElecSim.

There have been several studies in diverse applications on the use of online machine learning to predict time-series data. Johansson *et al.* apply online machine learning algorithms for heat demand forecasting [15]. They find that their demand predictions display robust behaviour within acceptable error margins. They find that artificial neural networks (ANNs) provide the best forecasting ability of the standard algorithms and can handle data outside of the training set.

Baram *et al.* combine an ensemble of active learners by developing an active-learning master algorithm [16]. To achieve this, they propose a simple maximum entropy criterion that provides effective estimates in realistic settings. Their active-learning master algorithm is empirically shown to, in some cases, outperform the best algorithm in the ensemble on a range of classification problems.

An extension of the FLORA algorithm is proposed by Schmitt *et al.* in [17]. Their FLORA-MC enhances the FLORA algorithm for multi-classification and numerical input values. They use this algorithm for an ambient computing application. Ambient computing is where computing and communication merges into everyday life. They find that their model outperforms traditional offline learners by orders of magnitude.

Pindoriya *et al.* trial several different machine learning methods such as an adaptive wavelet neural network (AWNN). They find that AWNN has good prediction properties when compared to other forecasting techniques such as wavelet-ARIMA, multi-layer perceptron (MLP) and radial basis function (RBF) neural networks as well as the fuzzy neural network (FNN).

Goncalves Da Silva *et al.* show the effect of prediction accuracy on local electricity markets [18]. To this end, they compare forecasting of groups of consumers in comparison to single individuals. They trial the use of the Seasonal-Naïve and Holt-Winters algorithms and look at the effect that the errors have on trading in an intra-day electricity market of consumers and prosumers. They found that with a photovoltaic penetration of 50%, over 10% of the total generation capacity was uncapitalised and roughly 10, 25 and 28% of the total traded volume were unnecessary buys, demand imbalances and unnecessary sells respectively. This represents energy that the participant has no control.

## 3. Material

### 3.1. Machine Learning

Machine learning is a methodology for finding and describing structural patterns in data [19]. Typically machine learning algorithms are algorithms which are trained using data. These generate models which can make predictions or decisions with regards to classification of data [15]. Machine learning algorithms can be used for both static datasets as well as streaming data.

### 3.2. Online learning

Online machine learning is an extension of machine learning algorithms, that can be used on dynamic



Figure 1: A three layer feed forward neural network.

datasets, such as time-series data. In traditional machine learning algorithms, when new data is obtained, the historical data must be used to retrain the entire model with a new model. This can be costly both in terms of time and in computation power [20]. Online algorithms, therefore, avoids the repeated retraining of data and improves the learning efficiency [21].

Examples of such algorithms are Passive Aggressive (PA) Regressor [22], Linear Regression, Box-Cox Regressor [23], K-Neighbors Regressor [24], Softmax Regression, Multilayer perceptron regressor [25]. For our work, we trial the stated algorithms, in addition to a host of offline learning techniques. The online techniques trialled were Lasso regression [26], ridge regression [27], Elastic Net [28], least angle regression [29], Extra Trees Regressor [29], Random Forest Regressor [30], AdaBoost Regressor [31], Gradient Boosting Regressor [32], Support vector regression [33], and offline versions of Multilayer perceptron regressor, K-Neighbors regressor, linear regression.

We trial the previously mentioned statistical and machine learning algorithms and vary the parameters using grid search and cross-validation in the python package, scikit learn [34].

### 3.3. Algorithms

In this section, we give a brief outline of the most successful online learning algorithms in this work.

#### 3.3.1. Multilayer perceptron

Artificial Neural Networks are a model which can model non-linear relationships between input and output data [35]. A popular neural network is a feed-forward multilayer perceptron. Fig. 1 shows a three-layer feed-forward neural network with a single output unit,  $k$  hidden units,  $n$  input units.  $w_{ij}$  is the connection weight from the  $i$ th input unit to the  $j$ th hidden unit, and  $T_j$  is the connecting weight from the  $j$ th hidden unit to the output unit [36]. These weights transform the input

variables in the first layer to the output variable in the final layer using the training data.

Typically, a dataset is split into three sections, the test set, training set and validation set. The training set is used to find the connection weights of the network, whilst the test set is used to determine the accuracy of the models. The validation set allows for an unbiased evaluation of the model whilst tuning the hyperparameters, and can avoid overfitting by stopping training if the error begins to increase.

For a univariate time series forecasting problem, suppose we have  $N$  observations  $y_1, y_2, \dots, y_N$  in the training set,

$$y_{N+1}, y_{N+2}, \dots, y_{N+m} \quad (1)$$

in the test set and we are required to predict  $m$  periods ahead [36].

The training patterns are as follows:

$$y_{p+m} = f(y_p, y_{p-1}, \dots, y_1) \quad (2)$$

$$y_{p+m+1} = f(y_{p+1}, y_p, \dots, y_2) \quad (3)$$

$$\vdots$$

$$y_N = f(y_{N-m}, y_{N-m-1}, \dots, y_{N-p+1}) \quad (4)$$

where  $f$  is the function made up of weights and activation functions in the trained neural network.

The  $m$  testing patterns are

$$y_{N+1} = f(y_{N+1-m}, y_{N-m}, \dots, y_{N-m-p+2}) \quad (5)$$

$$y_{N+2} = f(y_{N+2-m}, y_{N-m+1}, \dots, y_{N-m-p+3}) \quad (6)$$

$$\vdots$$

$$y_{N+m} = f(y_N, y_{N-1}, \dots, y_{N-p+1}) \quad (7)$$

The training objective is to minimize the overall predictive error means (SSE) by adjusting the connection weights. For this network structure the SSE can be written as:

$$SSE = \sum_{i=p+m}^N (y_i - \hat{y}_i) \quad (8)$$

where  $\hat{y}_i$  is the output from the network. The number of input nodes corresponds to the number of lagged observations. Having too few or too many input nodes can affect the predictive ability of the neural network [36].

#### 3.3.2. Box-Cox regressor

The ordinary least squares regression assumes a normal distribution of residuals. However, when this is not the case, the Box-Cox Regression may be useful [23].

It transforms the dependent variable using the Box-Cox Transformation function and employs maximum likelihood estimation to determine the optimal level of the power parameter lambda. The Box-Cox Regression requires that no dependent variable has any negative values.

Variable selection and ordinary least squares output dialogues are identical to that of linear regression.

The Box-Cox regression will transform the dependent variable as follows:

$$y^{(\lambda)} = \frac{y^\lambda - 1}{\lambda} \text{ if } \lambda \neq 0 \quad (9)$$

$$y^{(\lambda)} = \ln(y) \text{ if } \lambda = 0 \quad (10)$$

and determine the optimal value of lambda by maximising the following log-likelihood function:

$$L^{(\lambda)} = -\frac{n}{2} \ln(\hat{\sigma}_{(\lambda)}^2) + (\lambda - 1) \sum_{i=1}^n \ln(y_i) \quad (11)$$

where  $\hat{\sigma}_{(\lambda)}^2$  is the estimate of the least squares variance using the transformed y variable.

### 3.3.3. Passive-Aggressive regressor

The goal of the Passive-Aggressive (PA) algorithm is to change the model as little as possible to correct for any mistakes and low-confidence predictions it encounters [22]. Specifically, with each example PA solves the following optimisation [37]:

$$\mathbf{w}_{t+1} \leftarrow \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w}_t - \mathbf{w}\|^2 \quad (12)$$

$$\text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1. \quad (13)$$

Updates occur when the inner product does not exceed a fixed confidence margin - i.e.,  $y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1$ . The closed-form update for all examples is as follows:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \alpha_t y_t \mathbf{x}_t \quad (14)$$

where  $\alpha_t = \max \left\{ \frac{1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)}{\|\mathbf{x}_t\|^2}, 0 \right\}$ . Full details of the derivation can be found in [22].

### 3.4. Long-term Energy Market Model

In order to test the impact of the different residual distributions, we used the model developed by Kell *et al.*, ElecSim [8, 9]. ElecSim is an agent-based model which mimic the behaviour of decentralised electricity markets. In this paper, we parametrised the model with data of the United Kingdom in 2018. This enabled us to create a digital twin of the UK electricity market and

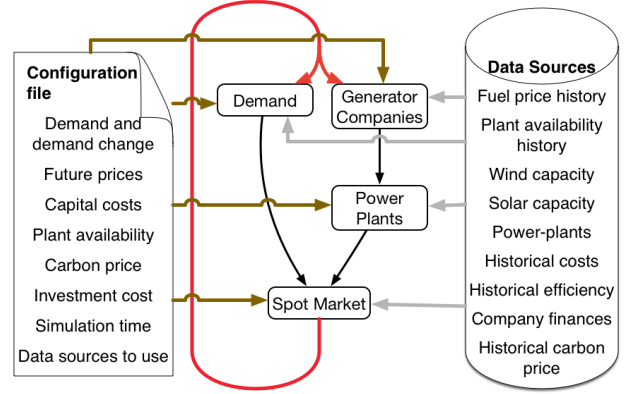


Figure 2: System overview of ElecSim [8].

project forward. The data used for this parametrisation included power plants in operation in 2018 and the funds available to the generation companies [38, 39].

ElecSim is made up of six fundamental components: 1) power plant data; 2) scenario data; 3) the time-steps of the algorithm; 4) the power exchange; 5) the investment algorithm and 6) the generation companies (GenCos) as agents. ElecSim uses a subset of representative days of electricity demand, solar irradiance and wind speed to approximate a full year. In this context, representative days are a subset of days which, when scaled up, represent an entire year [9]. We show how these components interact in Figure 2 [8].

The configuration file details the scenario which can be set by the user. This includes electricity demand, carbon price and fuel prices. The data sources parametrise the digital twin to a particular country, including information such as wind capacity and power plants in operation. Generation Companies own and invest in power plants. These power plants are then matched to electricity demand using a spot market.

The market runs a merit-order dispatch model and bids are made by the power plant's short-run marginal cost (SRMC). Investment in power plants are based upon a net present value (NPV) calculation. NPV is able to evaluate and compare investments with cash flows spread over many years. This is shown in Equation 15, where  $t$  is the year of the cash flow,  $i$  is the discount rate,  $N$  is the total number of years, or lifetime of power plant, and  $R_t$  is the net cash flow of the year  $t$ :

$$NPV(t, N) = \sum_{t=0}^N \frac{R_t}{(1+i)^t}. \quad (15)$$

Each of the GenCos estimate the yearly income for each prospective power plant by running a merit-order

dispatch electricity market ten years into the future. However, it is true that the expected cost of electricity ten years into the future is particularly challenging to predict. We, therefore, use a reference scenario projected by the UK Government Department for Business and Industrial Strategy (BEIS), and use the predicted costs of electricity calibrated by Kell *et al* [40, 9]. The agents predict the future carbon price by using a linear regression model.

## 4. Methods

### 4.1. Data preparation

Similarly to the work by Kell *et al.* [3], we selected a number of calendar attributes from the following dataset [41]. This dataset contained data between the years 2011-2018 for the United Kingdom. The calendar attributes used as predictors to the models were hour, month, day of the week, day of the month and year. These attributes allow us to take into account the periodicity of the data within each day, month and year.

It is also the case that electricity demand on a public holiday which falls on a weekday falls is dissimilar to load behaviours of ordinary weekdays [10]. We, therefore, passed a holiday variable to allow the model to account for this.

~~In addition to this,~~ as time-series demand data is correlated in the time-domain, we lagged the input demand data. This enabled us to take into account correlations on previous days, weeks and the previous month. Specifically, we used the previous 28 hours before the time step to be predicted for the previous 1st, 2nd, 7th and 30th day. We chose this as we believe that the previous two days were the most relevant to the day to be predicted, as well as the weekday of the previous week and the previous month. We could have increased the number of days provided to the algorithm. However, due to time and computational constraints, we used our previously described intuition for lagged data selection. Future work could be in selecting the optimum number of lagged inputs.

In addition to this, we gave the data information with regards to six different seasons. These seasons were defined by the National Grid Short Term Operating Reserve (STOR) Market Information Report [42]. Finally, to predict a full 24-hours ahead, we used 24 different models, 1 for each hour of the day.

The data is standardized and normalized using min-max scaling between -1 and 1 before training and predicting with the model.

### 4.2. Algorithm Tuning

~~Many machine learning and statistical models have different parameters to train, known as hyperparameters. To find the optimum hyperparameters, cross-validation was used. Cross validation is the process of training the model on a certain subset of data and testing it with the remaining subset.~~ As this time-series data were correlated in the time-domain, we took the first six years of data (2011-2017) for training and tested on the remaining year of data (2017-2018).

Each machine learning algorithm has a different set of parameters to tune. To tune the parameters in this paper, we used a grid search method. Grid search is effectively a brute force approach that trials each combination of parameters at our choosing.

Tables 1 and 2 display each of the models and respective parameters that were used in the grid search. Table 1 shows the offline machine learning methods, whereas Table 2 displays the online machine learning methods. Each of the parameters within the columns “Values” are trialled with every other parameter.

Whilst there is room to increase the total number of parameters, due to the exponential nature of grid-search, we chose a smaller subset of hyperparameters to choose, and a larger number of regressor types.

### 4.3. Prediction Residuals in ElecSim

Each of the previously mentioned models trialled will have a certain amount of errors. Prediction residuals are the difference between the estimated and actual values. We collect the prediction residuals to form a distribution for each of the models. We then trial 80 different closed-form distributions to see which of the distributions best fits the residuals from each of the models.

Once each of the prediction residual distributions are fit with a sensible closed-form distribution, we sample from this new distribution and perturb the demand for the electricity market at each time step within ElecSim.

By perturbing the market by the residuals, we can observe what the effects are of incorrect predictions of demand in an electricity market using the long-term electricity market model, ElecSim. We are able to understand the differences that prediction residuals have on long-term investment decisions as well as generators utilised.

## 5. Results

In this section, we detail the accuracy of the algorithms and statistical models to predict 24 hours ahead for the day-ahead market. In addition to this, we display

Regressor Type	Parameters	Values	Parameters	Values	Parameters	Values
Linear	N/A	N/A				
Lasso	N/A	N/A				
Elastic Net	N/A	N/A				
Least-Angle	N/A	N/A				
Extra Trees	# Estimators	[16, 32]				
Random Forest	# Estimators	[16, 32]				
AdaBoost	# Estimators	[16, 32]				
Gradient Boosting	# Estimators	[16, 32]	learning rate	[0.8, 1.0]		
Support Vector	Kernel	[linear, rbf]	C	[1, 10]	Gamma	[0.001, 0.0001]
Multilayer Perceptron	Activation function	[tanh, relu]	hidden layer sizes	[1, 50]	Alpha	[0.00005, 0.0005]
K-Neighbours	# Neighbours	[5, 20, 50]				

Table 1: Hyperparameters for offline machine learning regression algorithms

Regressor Type	Parameters	Values	Parameters	Values	Parameters	Values
Linear	N/A	N/A				
Box-Cox	Power	[0.1, 0.05, 0.01]				
Softmax	N/A	N/A				
Multilayer Perceptron	Hidden layer sizes	[(10, 50, 100), (10), (20), (50), (10, 50)]				
Passive Aggressive	C	[0.1, 1, 2]	Fit intercept?	[True, False]	Max iterations	[1, 10, 100, 1000]

Table 2: Hyperparameters for online machine learning regression algorithms



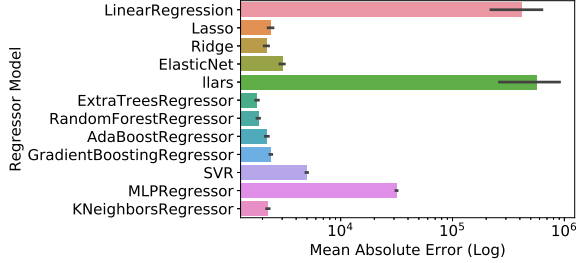


Figure 3: Offline models mean absolute error comparison, with 95% confidence interval for 5 runs of each model.

the impact of the errors on electricity generation investment and electricity mix from the years 2018 to 2035 using the agent-based model ElecSim.

### 5.1. Offline Machine Learning

Figure 3 displays the mean absolute error of each of the offline statistical and machine learning models on a log scale. It can be seen that the different models have varying degrees of success. The least accurate models were linear regression, the multilayer perceptron (MLP) model and the Least Angle Regression (lars). These all have mean absolute errors over 10,000MWh. This error would be prohibitively high in practice; the max tendered national grid reserve is 6,000MWh, while the average tendered national grid reserve is 2,000MWh [42].

A number of models, however, perform well, with low mean absolute error. These include the Lasso, gradient Boosting Regressor and K neighbours regressor. The best model, similarly to [3], was the decision tree-based model, Extra Trees Regressor, with a mean absolute error of 1604MWh. This level is well within the average national grid reserve of 2,000MWh.

Figure 4 displays the distribution of the best offline machine result (Extra Trees Regressor), as previously discussed. It can be seen that the max tendered national grid reserve falls well above the 5% and 95% percentiles. However, there are occasions where the errors are greater than the maximum tendered national grid reserve. In addition, the majority of the time, the model's predictions fall within the average available tendered national grid reserve.

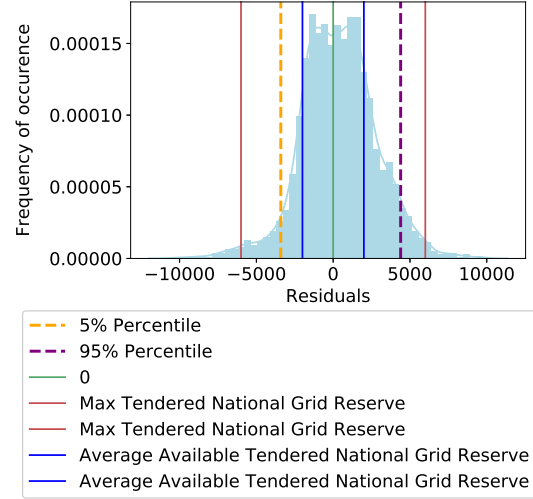


Figure 4: Best offline machine learning learning algorithm (Extra Trees Regressor) distribution.

### 5.2. Online Machine Learning

To see if we can improve on the predictions, we utilize an online machine learning approach. If we are successful, we should be able to reduce the national grid reserves, reducing cost and emissions.

Figure 5 displays the comparison of mean absolute errors for the different trialled online regressor models. To produce this graph, we showed various hyperparameter trials. Where the hyperparameters had the same results, we removed them. For the multilayer perceptron (MLP), we aggregated all hyperparameters, due to the similar nature of the predictions.

It can be seen that the best performing model was the Box-Cox regressor, with a MAE of 1100. This is an improvement of over 30% on the best offline model. The other models perform less well. However, it can be seen that the linear regression model improves significantly for the online case when compared to the offline case. The passive aggressive (PA) model improve significantly with the varying parameters, and the MLP performs poorly in all cases.

Figure 6 displays the best online model. We can see a significant improvement over the best online model distribution, shown in Figure 4. We remain within the max tendered national grid reserve for the vast majority of the time, and the average available tendered national grid reserve is close to the 5% and 95% percentiles.

Figure 7 displays the residuals for a model with poor predictive ability, the passive aggressive regressor. It

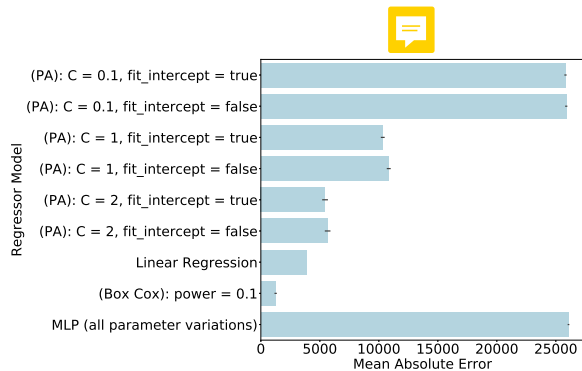


Figure 5: Comparison of mean absolute errors (MAE) for different online regressor models (identical MAEs removed).

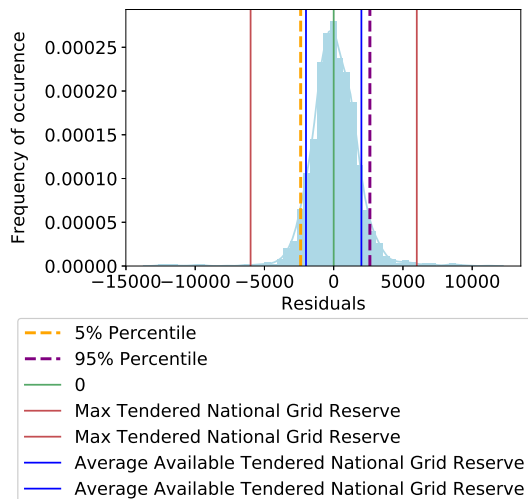


Figure 6: Best online model (Box-Cox Regressor) distribution.

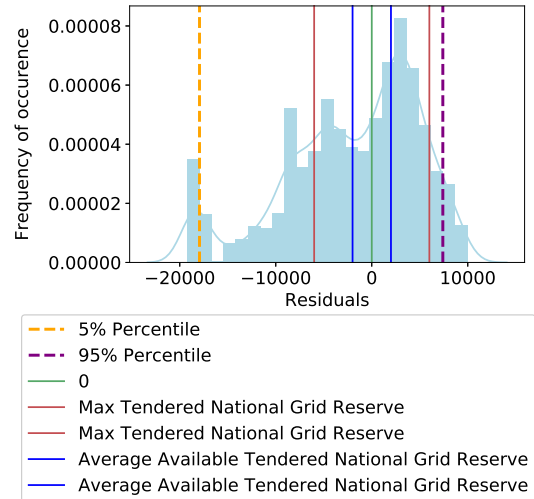


Figure 7: Online machine learning learning algorithm distribution. (Passive Aggressive Regressor (C=2, fit\_intercept = false, maximum iterations = 1000, shuffle = false, tolerance = 0.001)).

displays a large period of time of prediction errors at -20000MWh, and often falls outside of the national grid reserve. These results demonstrate the importance of trying a multitude of different models and parameters to improve prediction accuracy.

### 5.3. Scenario Comparison

In this section we explore the effect of these residuals on investments made and the electricity generation mix.

#### 5.3.1. Mean Contributed Energy Generation

Figure ?? displays the mean photovoltaic (PV) contributed between 2018 and 2035 vs. mean absolute error of the various online regressor models displayed in Figure 5. A positive correlation can be seen with PV contributed and mean absolute error. This is similar for coal and nuclear output, shown in Figures 8b and 8c respectively. However, as shown by Figure 8d, offshore wind reduces with mean absolute error. Figure 9 displays the mean reciprocal gas engine output vs mean absolute error between the same time period. Output for the reciprocal gas engine also increases with mean absolute error.

The reciprocal gas engine was expected to increase with times of high error. This is because, traditionally, reciprocal gas engines are peaker power plants. Peaker power plants provide power at times of peak demand,



which cannot be covered by other plants due to them being at their maximum capacity level or out of service.

It is hypothesised that coal and nuclear output increase to cover the predicted increased demands of the service. As these generation types are dispatchable, meaning operators can choose when they generate electricity, they are more likely to be used in times of higher predicted demand.

Photovoltaics may be used more with higher errors due to the times at which the errors were greatest. For example, during the day, where demand is higher, as is solar irradiance.

### 5.3.2. Total Energy Generation

In this section, we detail the difference in total technologies invested in over the time period between 2018 to 2035, as predicted by ElecSim.

CCGT, onshore, and reciprocal gas engines are invested in less over the time period, as shown by Figures 10a, 10d, 11c respectively. While coal, offshore, nuclear and photovoltaics all exhibit increasing investments.

It is hypothesised that coal and nuclear increase in investment due to their dispatchable nature. While onshore, non-dispatchable by nature, become a less attractive investment when compared to the other technologies.

CCGT and reciprocal gas engines may have decreased in capacity over this time, due to the increase in coal. This could be because of the large consistent errors in prediction accuracy that meant that reciprocal gas engines were perceived to be less valuable.

Figure 11d shows an increase in relative mean carbon emitted. This could be due to the increased output of utility of the reciprocal gas engine, coal, and decrease in offshore output.

## 6. Discussion

From our results, it can be seen that different algorithms yield differing prediction accuracies. Online models result in a decrease in 30% of prediction error on the best offline models. We, therefore, recommend the use of online machine learning for predicting electricity demand in a day-ahead market.

Additionally, the impact on the broader electricity market has been shown to be significant. Principally, the investment behaviours of generation companies change as well as the dispatched electricity mix. The relative mean carbon emitted over this time period increases, due to an increase in the utilisation of coal and reciprocal gas engines, at the expense of offshore wind.

## 7. Conclusion

In this paper, we trialled multiple different machine learning and statistical models to predict electricity demand in the UK for the day-ahead market. Specifically, we used both online and offline algorithms to predict electricity demand 24 hours ahead.

We then measured the errors and compared these to each model as well as the national grid reserve. We found that through the use of an online learning approach, we were able to significantly reduce error by 30% on the best offline algorithm. We were also able to reduce our errors to significantly below the national grid's mean and maximum tendered reserve.

In addition to this, we took these errors, or residuals, and perturbed the electricity market of the agent-based model ElecSim. This enabled us to see the impact of different error distributions on the long-term electricity market, both in terms of investment and in terms of electricity mix.

In future work, we would like to trial a different selection of algorithms and statistical models and trial different inputs to the models, for instance, by providing the model with two months worth of historical data.

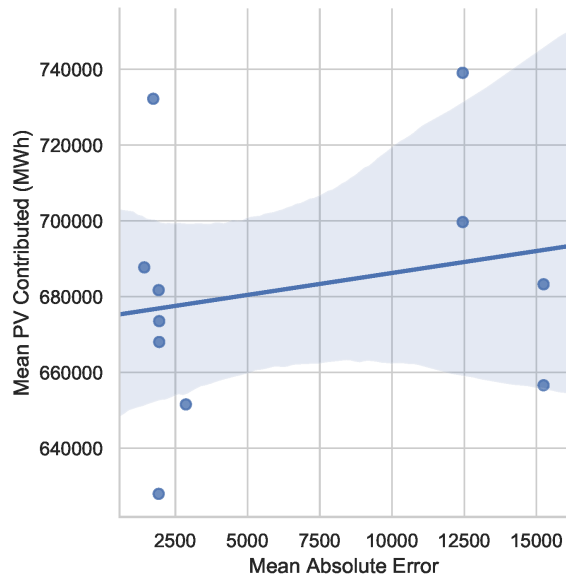
## 8. Funding Sources

This work was supported by the Engineering and Physical Sciences Research Council, Centre for Doctoral Training in Cloud Computing for Big Data [grant number EP/L015358/1].

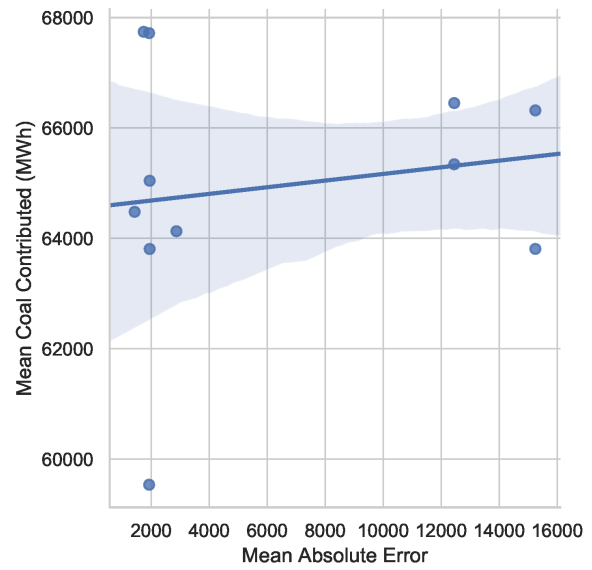
## References

### References

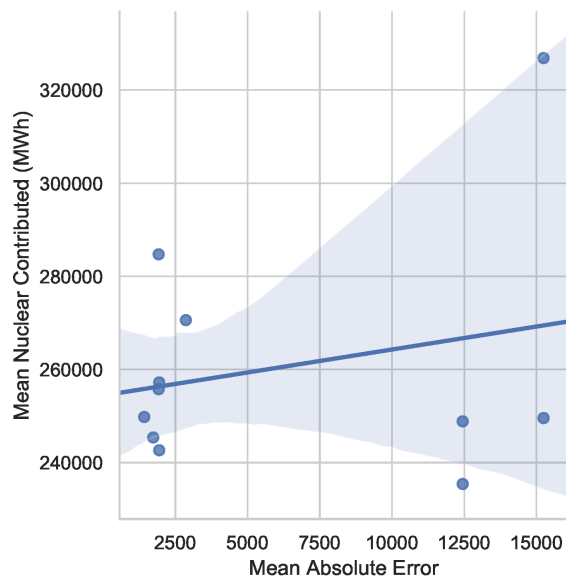
- [1] C. Lu, H. T. Wu, S. Vemuri, Neural network based short term load forecasting, *IEEE Transactions on Power Systems* 8 (1) (1993) 336–342. doi:10.1080/02533839.1995.9677697.
- [2] A. Mahmood, M. N. Ullah, S. Razzaq, A. Basit, U. Mustafa, M. Naeem, N. Javaid, A new scheme for demand side management in future smart grid networks, *Procedia Computer Science* 32 (2014) 477–484. doi:10.1016/j.procs.2014.05.450. URL <http://dx.doi.org/10.1016/j.procs.2014.05.450>
- [3] A. Kell, A. McGough, M. Forshaw, Segmenting residential smart meter data for short-Term load forecasting, in: *e-Energy 2018 - Proceedings of the 9th ACM International Conference on Future Energy Systems*, 2018.
- [4] M. Al-Musaylh, R. Deo, J. Adamowski, Y. Li, Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia, *Advanced Engineering Informatics* 35 (November 2017) (2018) 1–16. doi:10.1016/j.aei.2017.11.002. URL <https://doi.org/10.1016/j.aei.2017.11.002>



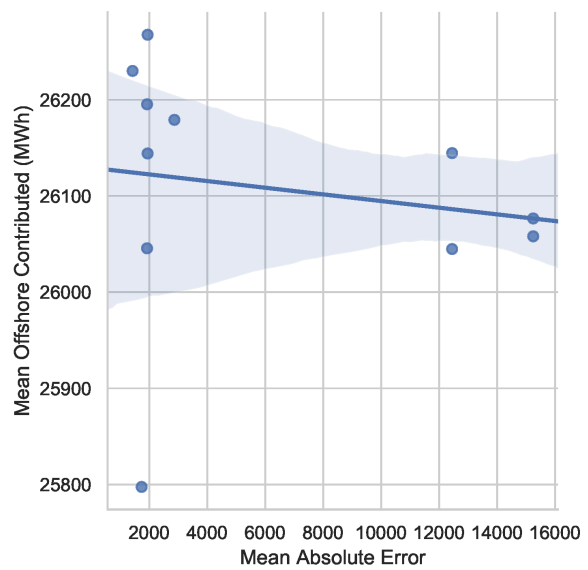
(a) Photovoltaic output.



(b) Coal output.



(c) Nuclear output.



(d) Offshore output.

Figure 8: Mean outputs of various technologies vs. mean absolute error from 2018 to 2035 in ElecSim.

- [5] P. Vrablecová, A. Bou Ezzeddine, V. Rozinajová, S. Šárik, A. K. Sangaiah, Smart grid load forecasting using online support vector regression, *Computers & Electrical Engineering* 0 (2017) 1–16. doi:10.1016/j.compeleceng.2017.07.006.
- [6] T. Hong, J. Wilson, J. Xie, A. Member, Long Term Probabilistic Load Forecasting and Normalization With Hourly Information 5 (1) (2014) 456–462.
- [7] A. K. Singh, Ibraheem, S. Khatoun, M. Muazzam, D. K. Chaturvedi, Load forecasting techniques and methodologies:

A review, ICPCES 2012 - 2012 2nd International Conference on Power, Control and Embedded Systems (2012). doi:10.1109/ICPCES.2012.6508132.

- [8] A. Kell, M. Forshaw, A. S. McGough, ElecSim : Monte-Carlo Open-Source Agent-Based Model to Inform Policy for Long-Term Electricity Planning, *The Tenth ACM International Conference on Future Energy Systems (ACM e-Energy)* (2019) 556–565.
- [9] A. J. M. Kell, M. Forshaw, A. S. McGough, Long-Term Elec-

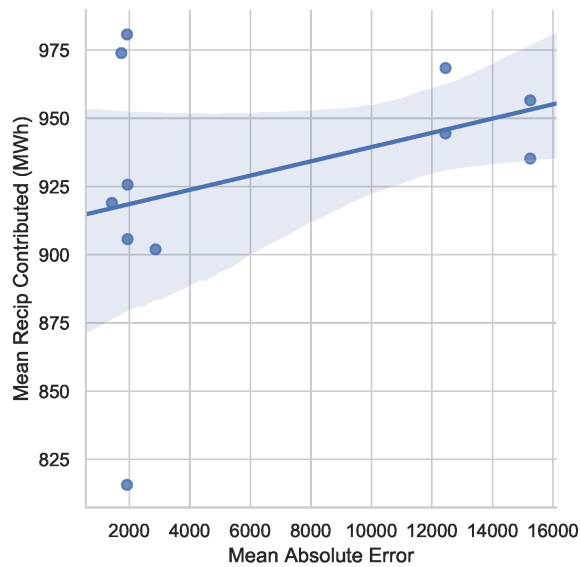
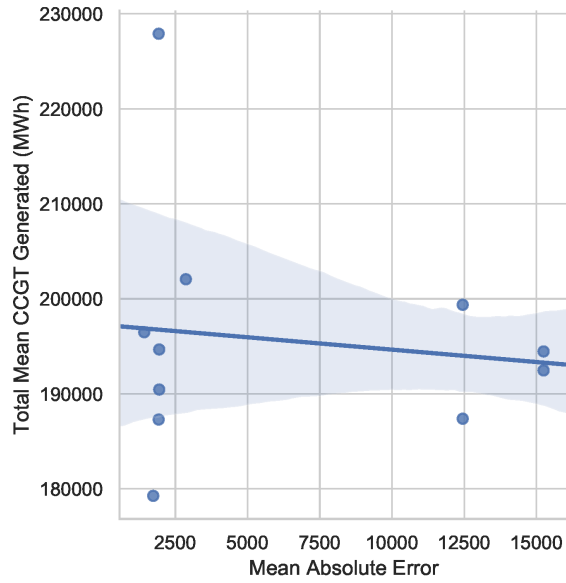
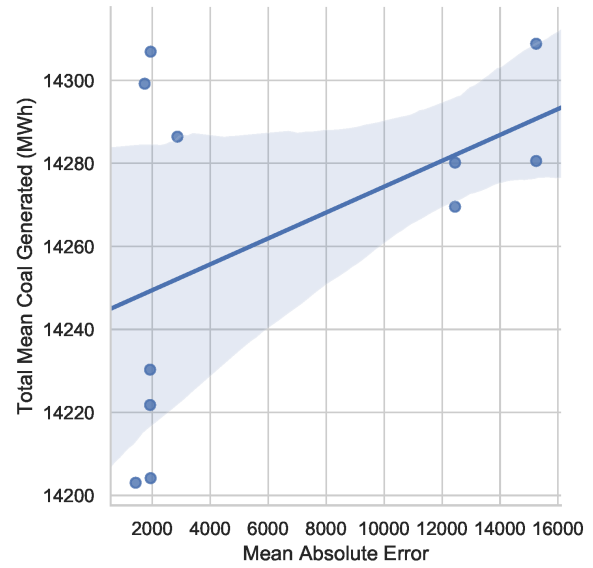


Figure 9: Mean reciprocal gas engine output vs. mean absolute error from 2018 to 2035 in ElecSim.

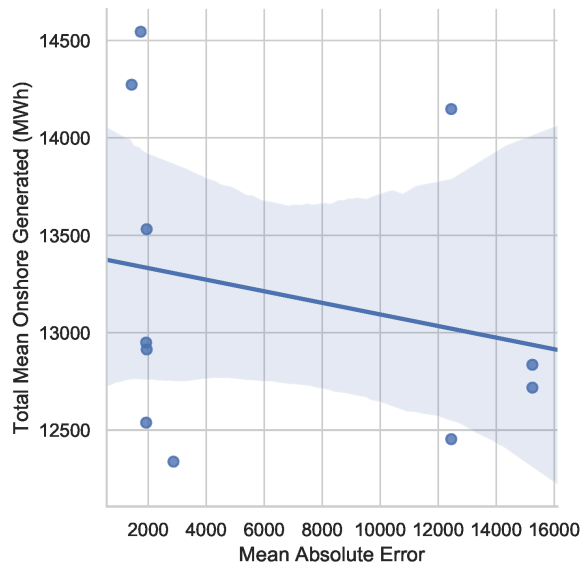
- tricity Market Agent Based Model Validation using Genetic Algorithm based Optimization, The Eleventh ACM International Conference on Future Energy Systems (e-Energy'20) (2020).
- [10] K.-h. Kim, H.-s. Youn, S. Member, Y.-c. Kang, Short-term load forecasting for special days in anomalous load conditions using neural networks, *IEEE Transactions on Power Systems* 15 (2) (2000) 559–565. doi:10.1109/59.867141.
  - [11] J. Nagi, S. K. Yap, S. K. Tiong, S. K. Ahmed, Electrical Power Load Forecasting using Hybrid Self-Organizing Maps and Support Vector Machines, *The 2nd International Power Engineering optimization Conference (PEOCO) (June) (2008)* 51 – 56.
  - [12] F. L. Quilumba, W.-j. Lee, H. Huang, D. Y. Wang, S. Member, R. L. Szabados, Using Smart Meter Data to Improve the Accuracy of Intraday Load Forecasting Considering Customer Behavior Similarities (2014) 1–8.
  - [13] S.-j. Huang, S. Member, K.-r. Shih, Short-Term Load Forecasting Via ARMA Model Identification Including Non-Gaussian 18 (2) (2003) 673–679.
  - [14] H. Nguyen, C. K. Hansen, Short-term electricity load forecasting with Time Series Analysis, *2017 IEEE International Conference on Prognostics and Health Management (ICPHM) (2017)* 214–221doi:10.1109/ICPHM.2017.7998331. URL <http://ieeexplore.ieee.org/document/7998331/>
  - [15] C. Johansson, M. Bergkvist, D. Geysen, O. D. Somer, N. Laveson, D. Vanhoudt, Operational Demand Forecasting in District Heating Systems Using Ensembles of Online Machine Learning Algorithms, *Energy Procedia* 116 (2017) 208–216. URL <http://dx.doi.org/10.1016/j.egypro.2017.05.068>
  - [16] Y. Baram, R. El-Yaniv, K. Luz, Online Choice of Active Learning Algorithms, *Proceedings, Twentieth International Conference on Machine Learning* 1 (2003) 19–26.
  - [17] J. Schmitt, M. Hollick, C. Roos, R. Steinmetz, Adapting the user context in realtime: Tailoring online machine learning algorithms to ambient computing, *Mobile Networks and Applications* 13 (6) (2008) 583–598. doi:10.1007/s11036-008-0095-8.
  - [18] P. Goncalves Da Silva, D. Ilic, S. Karnouskos, The Impact of Smart Grid Prosumer Grouping on Forecasting Accuracy and Its Benefits for Local Electricity Market Trading, *IEEE Transactions on Smart Grid* 5 (1) (2014) 402–410. doi:10.1109/TSG.2013.2278868.
  - [19] I. H. Witten, E. Frank, M. a. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 2011.
  - [20] Y. Li, P. Guo, X. Li, Short-Term Load Forecasting Based on the Analysis of User Electricity Behavior, *Algorithms* 9 (4) (2016) 1–14. doi:10.3390/a9040080.
  - [21] H. J. Rong, G. B. Huang, N. Sundararajan, P. Saratchandran, H. J. Rong, Online Sequential Fuzzy Extreme Learning Machine for Function Approximation and Classification Problems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 39 (4) (2009) 1067–1072. doi:10.1109/TSMCB.2008.2010506.
  - [22] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, Online Passive-Aggressive Algorithms, *Journal of Machine Learning Research* (2006). doi:10.1201/b15810-63.
  - [23] G. E. P. Box, D. Cox, An Analysis of Transformations **Author (s)** : G . E . P . Box and D . R . Cox Published by : Wiley for the Royal Statistical Society Stable URL : <http://www.jstor.org/stable/2984418> Accessed : 08-03-2016 06 : 53 UTC Your use of the JSTOR archive indicates, *Journal, Source Statistical, Royal Series, Society* 26 (2) (1964) 211–252.
  - [24] E. Forgy, Cluster analysis of multivariate data: Efficiency versus interpretability of classification, *Biometrics* 21 (3) (1965) 768–769.
  - [25] G. E. Hinton, Connectionist learning procedures, *Artificial Intelligence* 40 (1-3) (1989) 185–234. doi:10.1016/0004-3702(89)90049-0.
  - [26] R. Tibshirani, Regression Shrinkage and Selection Via the Lasso, *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1) (1996) 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.
  - [27] P. Geladi, Measurement, regression and calibration, philip brown, oxford statistical science series. vol. 12, oxford science publications, oxford, 1993, no of pages: 224. price: £27.50. isbn 0 19-852245-2, *Journal of chemometrics* 8 (5) (1994) 371–372.
  - [28] J. Friedman, T. Hastie, R. Tibshirani, Regularization Paths for Generalized Linear Models via Coordinate Descent, *Journal of Statistical Software* 33 (1) (2010) 1–22. arXiv:0908.3817, doi:10.1016/j.expneurol.2008.01.011.
  - [29] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, *The Annals of Statistics* 32 (2) (1988) 440–444. doi:10.1109/glocom.1988.25879.
  - [30] L. Breiman, Random forests, *Machine Learning* 45 (1) (2001) 5–32. arXiv:dx.doi.org/10.1023%2FA%3A1010933404324, doi:10.1023/A:1010933404324.
  - [31] Y. Freund, R. E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *Journal of Computer and System Sciences* 55 (1) (1997) 119–139. doi:10.1006/jcss.1997.1504.
  - [32] J. H. Friedman, Greedy Function Approximation: A Gradient Boosting Machine (316) 400.
  - [33] C. Cortes, V. Vapnik, Support-Vector Networks, *Machine Learning* 20 (3) (1995) 273–297. arXiv:arXiv:1011.1669v3, doi:10.1023/A:1022627411411.
  - [34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
  - [35] H. Akaike, A New Look at the Statistical Model Identification, *IEEE Transactions on Automatic Control* 19 (6) (1974) 716–723. arXiv:arXiv:1011.1669v3,



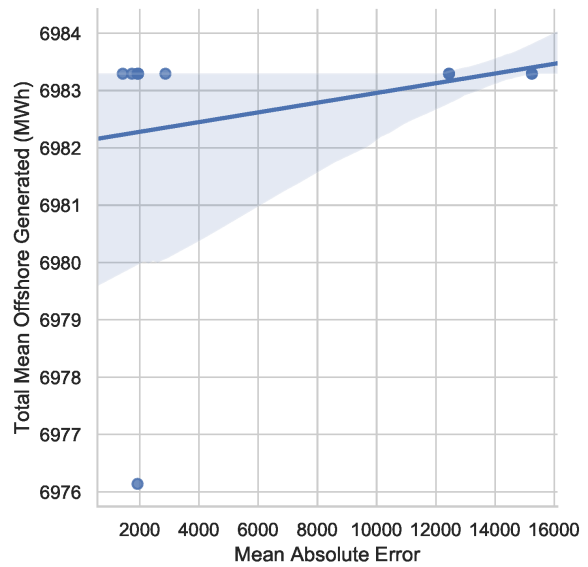
(a) Total CCGT.



(b) Total Coal.



(c) Total Onshore.



(d) Total Offshore.

Figure 10: Total technologies invested in vs. mean absolute error from 2018 to 2035 in ElecSim.

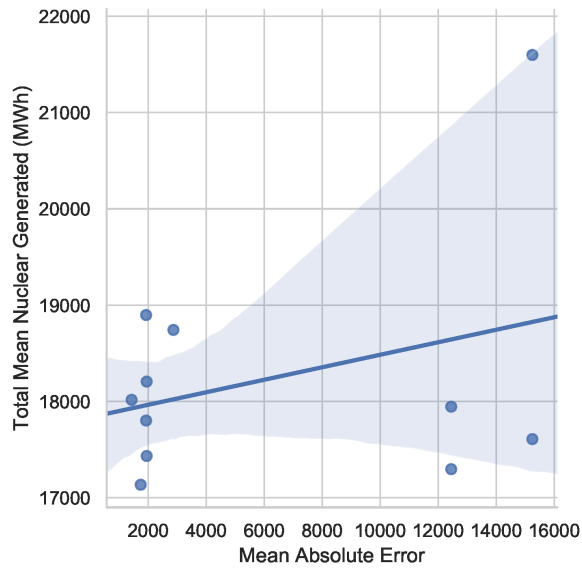
- doi:10.1109/TAC.1974.1100705.
- [36] H.-T. Pao, Forecasting electricity market pricing using artificial neural networks, *Energy Conversion and Management* 48 (3) (2007) 907–912. doi:10.1016/j.enconman.2006.08.016.
- [37] J. Ma, L. K. Saul, S. Savage, G. M. Voelker, Identifying suspicious URLs: An application of large-scale online learning, *Proceedings of the 26th International Conference On Machine Learning, ICML 2009* (2009) 681–688.
- [38] D. for Business Energy, I. Strategy, Power stations in the united

kingdom, may 2019, Digest of United Kingdom Energy Statistics (DUKES) (2019).

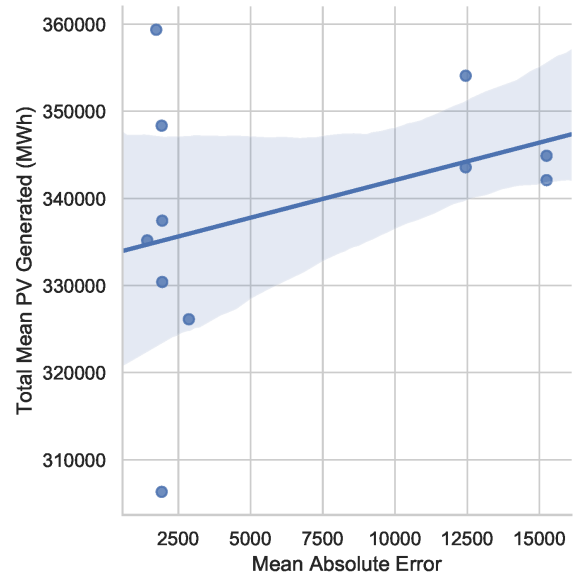
- [39] E. . I. S. Department for Business, Companies house - gov.uk, UK Government (2019).

- [40] Department for Business Energy & Industrial Strategy, Updated energy and emissions projections 2018, The Energy White Paper (April) (2019).

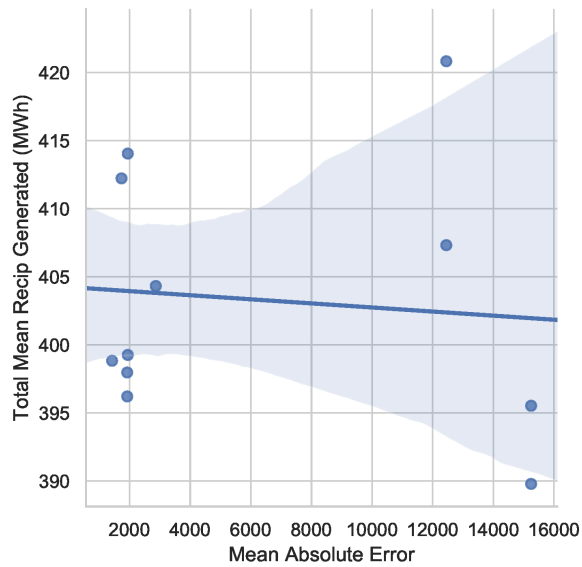
- [41] (2019). [link], URL <https://www.gridwatch.templar.co.uk/>



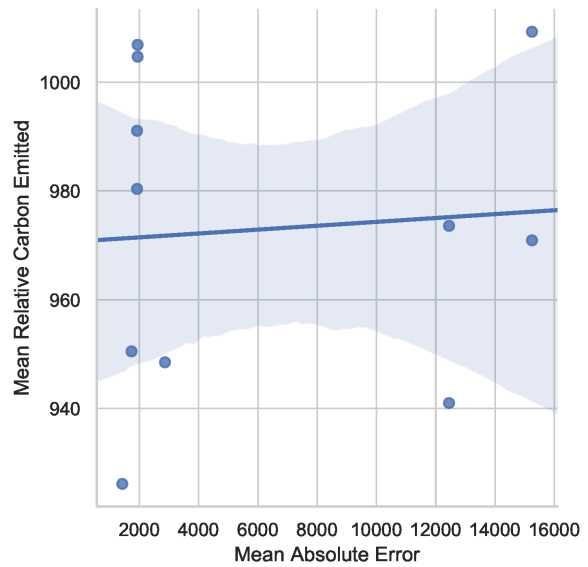
(a) Total nuclear.



(b) Total photovoltaics.



(c) Total Reciprocal gas engine.



(d) Mean carbon emitted.

Figure 11: a), b), c) Investments in various technologies vs. mean absolute error from 2018 to 2035 in ElecSim and d) mean carbon emissions between 2018 and 2035.

[42] **N. G. ESO**, STOR Market Information Report (October) (2019) 0–11.