



Topic:

**Using Machine Learning to Predict Building Energy Consumption**

Usman Kolo Mohammed

P2649855

*This thesis was submitted in partial fulfilment of the requirements for the degree of Masters of  
Intelligent Systems and Robotics*

May 2022

# Abstract

To predict the energy consumption of three De Montfort University buildings, this project uses and compares five novel machine learning algorithms: Long Short-Term Memory (LSTM) networks, Bidirectional Long Short-Term Memory (BiLSTM), Extreme Gradient Boosting (XGBoost), Light Gradient Boost Mechanism (LightGBM), and Facebook Prophet. Using a persistence approach and daily resampling, a baseline prediction was established. All of the systems were able to outperform the baseline and accurately anticipate consumption.

Here is a link to the [GitHub](#) repository

# Table of Contents

Abstract.....	2
Table of Figures .....	5
List of Tables .....	6
Introduction.....	7
Background .....	7
Organisation of thesis .....	8
Literature Review .....	9
Introduction .....	9
Regression Models .....	9
Machine Learning Introduction.....	10
Long Short-Term Memory (LSTM).....	10
Bidirectional-LSTM (BLSTM) .....	12
Gradient Boosting Regression Trees .....	12
XGBoost .....	12
LightGBM .....	13
Facebook Prophet .....	13
Methodology .....	15
Model .....	15
RNN .....	15
LSTM .....	16
Bidirectional LSTM .....	17
XGBoost.....	18
Prophet.....	19
LightGBM.....	20
Loss Functions .....	21
Mean Squared Error.....	21
Mean Absolute Error.....	22
Root Mean Squared Error .....	22
R-squared.....	23
Data Analysis .....	24
Queens Building Dataset .....	24
Preprocess/Cleaning .....	24
Visualize .....	25
Check for stationarity.....	26

Correlation .....	27
Feature Importance .....	28
Hugh Aston Building Dataset.....	29
Preprocess/Cleaning .....	29
Visualize .....	30
Check for stationarity.....	31
Correlation .....	31
Feature Importance .....	33
Gateway House Building Dataset .....	33
Preprocess/Cleaning .....	33
Visualize .....	34
Check for stationarity.....	35
Correlation .....	35
Feature Importance .....	37
Weather Dataset .....	37
Preprocessing.....	38
Visualization .....	40
Feature Selection .....	42
Normalization.....	42
Min-Max Scaler .....	43
Standard Scaler .....	43
Experimentation .....	45
Hardware.....	45
Software .....	45
Settings.....	45
Training .....	45
Experiments and Results .....	46
Queen's Building .....	46
Hugh Aston .....	53
Gateway House .....	60
Summary of Results .....	67
Conclusion .....	68
Bibliography.....	70

## List of Figures

FIGURE 1: RECURRENT NEURAL NETWORK .....	15
FIGURE 2: SIMPLE RNN STRUCTURE WITH A HIDDEN LAYER.....	16
FIGURE 3: RNN WITH ONE HIDDEN LAYER EXPANDED OVER TIME.....	16
FIGURE 4: SINGLE LSTM NEURON.....	17
FIGURE 5: BIDIRECTIONAL LSTM STRUCTURE .....	18
FIGURE 6:EVOLUTION OF XGBOOST FROM DECISION TREES.....	19
FIGURE 7: EXAMPLE OF THE PROPHET ALGORITHM OUTPUT.....	20
FIGURE 8: DIAGRAMMATIC REPRESENTATION OF LEAF-WISE TREE GROWTH.....	21
FIGURE 9: QUEEN’S BUILDING ELECTRICITY READING BEFORE CLEANING.....	24
FIGURE 10: QUEEN’S BUILDING ELECTRICITY READING AFTER CLEANING.....	24
FIGURE 11: ELECTRICITY DAILY DATASET .....	25
FIGURE 12:ELECTRICITY MONTHLY DATASET .....	26
FIGURE 13: HEATMAP BETWEEN QUEEN’S BUILDING DAILY DATASET.....	27
FIGURE 14: EXAMPLE SHOWING A CORRELATION BETWEEN ELECTRICITY AND WATER DATASETS.....	28
FIGURE 15: EXAMPLE SHOWING A CORRELATION BETWEEN ELECTRICITY AND GAS DATASETS.....	28
FIGURE 16: FEATURE IMPORTANCE BASED ON ELECTRICITY .....	29
FIGURE 17: HUGH ASTON ELECTRICITY READING BEFORE CLEANING .....	29
FIGURE 18: HUGH ASTON ELECTRICITY READING AFTER CLEANING .....	30
FIGURE 19: ELECTRICITY MONTHLY DATASET.....	30
FIGURE 20: HEATMAP BETWEEN HUGH ASTON DAILY DATASET .....	31
FIGURE 21: ELECTRICITY AND GAS CORRELATION.....	32
FIGURE 22: ELECTRICITY AND WATER CORRELATION .....	32
FIGURE 23: FEATURE IMPORTANCE.....	33
FIGURE 24: GATEWAY HOUSE ELECTRICITY DATASET BEFORE CLEANING.....	33
FIGURE 25: GATEWAY HOUSE ELECTRICITY DATASET AFTER CLEANING.....	34
FIGURE 26: ELECTRICITY MONTHLY DATASET.....	34
FIGURE 27: HEATMAP BETWEEN GATEWAY HOUSE DAILY DATASET.....	35
FIGURE 28: CORRELATION BETWEEN ELECTRICITY AND GAS.....	36
FIGURE 29: CORRELATION BETWEEN ELECTRICITY AND WATER .....	36
FIGURE 30: FEATURE IMPORTANCE.....	37
FIGURE 31: RAW WEATHER DATA STATISTICS .....	38
FIGURE 32: NULL VALUES REMOVED.....	38
FIGURE 33: ENCODED CATEGORICAL FEATURES.....	39
FIGURE 34: UNNECESSARY VALUES REMOVED.....	40
FIGURE 35: HEATMAP BETWEEN FEATURES IN WEATHER DATASET .....	40
FIGURE 36: EXAMPLE SHOWING THE NEGATIVE CORRELATION BETWEEN TEMPERATURE AND HUMIDITY .....	41
FIGURE 37: POSITIVE CORRELATION BETWEEN TEMPERATURE AND DEWPOINT .....	41
FIGURE 38: DATAFRAME BEFORE FEATURE SELECTION .....	42
FIGURE 39: FEATURES SCALED USING MIN-MAX .....	43
FIGURE 40: FEATURES SCALED USING STANDARD SCALER.....	44
FIGURE 41: EXPANDING WINDOW.....	46
FIGURE 42: THE PLOT OF QUEEN’S BUILDING GAS HOURLY DATASET .....	47
FIGURE 43: QUEEN’S BUILDING FORECAST USING PROPHET .....	53
FIGURE 44: HUGH ASTON FORECAST USING PROPHET.....	60
FIGURE 45: GATEWAY HOUSE FORECAST USING PROPHET .....	66
FIGURE 46: RMSE VS BASELINE.....	67

## List of Tables

TABLE 1: QUEEN'S BUILDING ADF TEST .....	26
TABLE 2: HUGH ASTON ADF TEST .....	31
TABLE 3: GATEWAY HOUSE ADF TEST .....	35
TABLE 4: QUEENS BUILDING HOURLY BASELINE .....	46
TABLE 5: QUEENS BUILDING DAILY BASELINE.....	47
TABLE 6: QUEEN'S BUILDING LSTM HOURLY PREDICTION.....	47
TABLE 7: QUEEN'S BUILDING LSTM DAILY PREDICTIONS .....	48
TABLE 8: QUEEN'S BUILDING BLSTM PREDICTIONS.....	49
TABLE 9: QUEEN'S BUILDING XGBOOST PREDICTIONS .....	50
TABLE 10: QUEEN'S BUILDING LIGHTGBM PREDICTIONS .....	51
TABLE 11: QUEEN'S BUILDING PROPHET PREDICTIONS .....	52
TABLE 12: HUGH ASTON HOURLY BASELINE.....	53
TABLE 13: HUGH ASTON DAILY BASELINE .....	54
TABLE 14: HUGH ASTON LSTM PREDICTIONS .....	54
TABLE 15: HUGH ASTON BLSTM PREDICTIONS .....	55
TABLE 16: HUGH ASTON XGBOOST PREDICTIONS .....	56
TABLE 17: HUGH ASTON LIGHTGBM PREDICTIONS .....	57
TABLE 18: HUGH ASTON PROPHET PREDICTIONS.....	58
TABLE 19: GATEWAY HOUSE HOURLY BASELINE.....	60
TABLE 20: GATEWAY HOUSE DAILY BASELINE.....	60
TABLE 21: GATEWAY HOUSE LSTM PREDICTIONS .....	60
TABLE 22: GATEWAY HOUSE BLSTM PREDICTIONS .....	61
TABLE 23: GATEWAY HOUSE XGBOOST PREDICTIONS .....	62
TABLE 24: GATEWAY HOUSE LIGHTGBM PREDICTIONS .....	63
TABLE 25: GATEWAY HOUSE PROPHET PREDICTIONS .....	65

# Introduction

## Background

Buildings are the UK's greatest energy consumer, accounting for 40% of overall consumption [1]. The main source of carbon emissions in the academic sector is power use in Higher Education (HE) institutions [2]. Energy management solutions swiftly gained popularity in the UK HE sectors when the Higher Education Funding Council of England (HEFCE) stipulated a 43 percent reduction in carbon emissions by 2020 for all participating institutions compared to the baseline year 2005 [3]. The majority of England's member institutions now have a dedicated energy management team working toward this goal. The largest contributor is electricity, which accounts for 63 percent, followed by natural gas, which accounts for 33.3 percent [1,4,5].

The type of building, its age, occupancy, working hours, the type of equipment installed, and weather patterns all have an impact on energy consumption in university buildings. On a typical university campus in England, academic buildings (42 percent) and administrative buildings (26 percent) take up around 68 percent of the space [6]. Non-residential facilities at English universities consumed roughly 80% of total energy consumption in 2014/15, while residential structures consumed 20% of total energy consumption [4].

With shares of 63 percent and 33 percent, respective, the previously stated data shows clearly how gas and electricity use are the two biggest carbon-emitting sources in England's HE sectors [4]. The modest decline in carbon emissions in the industry implies that universities ought to investigate more energy-saving alternatives as well as improve their energy management systems.

Tracking and evaluating building energy usage trends, which assists in assessing the facility's operational behaviour under various conditions, is an important aspect of a good energy management system. Under some settings, it can also help detect accidental energy waste. When energy consumption data is retrieved and saved, the patterns in the data and other variables such as temperature, humidity, and the number of inhabitants may be investigated, and future energy estimates can be made utilising all of these variables. Another key benefit is that data on projected energy usage can be utilised to anticipate accurate future energy budgets. Energy management teams at universities are in charge of monitoring, assessing, and storing data on energy consumption in their buildings. To create their energy budget estimates for the years ahead, they are in charge of building up accurate energy consumption forecasts and identifying possibilities for energy reduction.

The ability to accurately estimate energy use is important to the effective application of energy management systems. Due to a major lack of forecasting and the adoption of less efficient forecasting methodologies for planning, many businesses have struggled to control energy use and budgets [7,8]. Forecasting aids in assessing present and future economic conditions to guide the organization's policies and decisions. It's a technique that uses past and current data to forecast future information. A credible prediction system can assist financial and energy management teams at universities in establishing objectives and strategic goals and can even be used as part of its annual budgetary process [9].

Researchers have extensively employed diverse predictive techniques including MR, ANN, and GA for energy consumption prediction for various building types in various areas [10]. MR is an easy, dependable, and rapid technique among such [11,12]. Several researchers have employed the MR approach in their studies [11,12,15], but all such MR models anticipate the energy consumption of a single building or an

area and require a large amount of input data. Energy managers and their staff are often busy, thus a single, trustworthy, and rapid model for diverse building categories would be preferable to multiple forecasting methods.

This study intends to assist the university's energy management teams by creating a fast and easy predictive model that employs a variety of methodologies. Every manager should be able to anticipate the hourly, daily, and monthly energy usage of the facilities using these methodologies. Three buildings on DE Montfort University's main campus were chosen for this purpose, and their past energy usage data was used to generate the forecasting models.

### Organisation of thesis

This thesis is divided into six chapters and begins with an abstract. The abstract provides a concise overview of the work that went into producing this dissertation. In the first chapter, the dissertation's foundation is thoroughly discussed. The current difficulties in the energy business are also discussed, as well as the aim of the energy forecast. The second chapter examines various approaches to predicting energy consumption in buildings. After a brief introduction, the chapter delves into how each approach performed in terms of energy estimation. The third chapter describes the methodology used in this thesis explaining how LSTM, BiLSTM, XGBoost, LightGBM, and Prophet work, as well as the metrics they employ. Each variable in the dataset used for prediction is explained in the fourth chapter. The chapter also contains the various equations used in the thesis and discusses how to apply the technique from the previous chapter to the dataset. The model's performance is examined in the fifth chapter, which also describes the data received from the model built in the fourth chapter. The chapter also looks at some of the data's conclusions. The dissertation ends in the concluding chapter, which also includes recommendations for further research.



# Literature Review

## Introduction

The goal of this chapter is to provide a thorough examination of the efforts undertaken in the field of energy prediction. This chapter will cover a variety of subjects all connected to energy forecasting models and the results that determine which method is the best. In the past, best-fit regression equations were used for prediction and analysis [14]. This can be thought of as a "go-to" strategy for constructing equations from historical data and predicting future key variables. To better forecast the future, modern machine learning techniques are replacing or collaborating with traditional methodologies such as regression. This chapter will discuss previous work on regression, and machine learning techniques such as decision trees, Facebook Prophet, and RNNs specifically for energy prediction.

## Regression Models

[10, 15] described how regression can be used to predict energy use. According to [10], the amount of energy consumed by buildings supports the need for thorough research and modelling in this field. According to the article, quantitative methods are a particularly convenient option for constructing energy models when its user only has access to historical data but not the multiple values necessary for technical equations. When compared to other quantitative methodology, linear regression was determined to be a comparatively simple and accurate application. Hourly and daily records from a residence dedicated to research were used in this study. Because of the time difference, the researchers were able to investigate the impact of data collection frequency on the model's precision. External temperature and sun radiation were significant variables. The dependent variable was energy consumption. Simple and multiple linear regression, as well as quadratic linear regression, were all examined. It was conducted to check if the quadratic regression's extra depth was supported by a higher quality of performance.

The time interval was discovered to be a critical aspect in determining the model's quality. The model's performance improves with the length of time. To clarify this, the researchers state that when data is obtained over a shorter period, energy usage anomalies reveal large disparities. When a larger amount of data is acquired over a longer period, the errors average out over time. Using a multiple linear regression model, the coefficient of determination was enhanced. However, the RMSE suffered because of this method. When both criteria were considered, it was discovered that multiple linear regression provided the best overall quality of energy prediction. Daily time intervals also yielded the most accurate model parameters.

A multiple regression model for predicting heating energy demand was presented in [15]. Heating energy demand, according to the authors, is a crucial estimate. It is used during the design phase of a building to anticipate how much energy will be required for space conditioning over its lifetime. The global energy loss coefficient of a building, south equivalent surface, and temperature difference were used as independent variables. [15] is a simpler model than [10] because only three variables and a single prediction model were used. [15] also adds that in the event of big datasets, such as the one used in this work, a regression can be employed with reasonable success, even though the fact that it is simpler and easier to construct. The model in this article was developed using the "black box" idea. When input and output variables are known, and the user is required to fit in the best curve possible (also known as "black-box" owing to the unknown nature) to build a generalised relationship between dependent and independent factors, this approach is applied. Though least squares estimation is the most popular method, it can occasionally result in errors that are not normally distributed, which is a strong validation

of any curve fitting model [15]. As a result, we employed an incrementally reweighted least-squares method. To lessen the impact of a residual anomaly, this approach modifies the weight of the coefficients in the regression model. This method of determining the best fit curve yields a better least square estimate.

The model was evaluated on 17 blocks of apartments once it had been trained. With an  $R^2$  of 0.9744, the model was determined to be quite efficient. Also, it was discovered that 90% of the calculated results exhibited relative errors of 20% or less. [15] also analyses the model against several dynamic solutions, claiming that such a proposed approach is quicker and produces similar outcomes. Regression is the optimum strategy in terms of model quality and efficiency when constructing models for datasets with a lesser set of variables, according to [15].

### Machine Learning Introduction

Machine learning is described as the application of computational methods that have been trained on historical data to assist in the decision-making process for a specific system. These tools are typically used to increase performance or create precise forecasts. Machine learning approaches are currently being used in a variety of industries to make predictions using historical data, either as a substitute or in conjunction with traditional regression/statistical models [16]. [16] is a good place to start reading about machine learning's real-world applications. Several machine-learning techniques are now being used to estimate energy usage, including but not restricted to RF [17], SVMs [18, 19], and NNs [20, 21, 22]. Several studies have found that ML algorithms are at least as effective as traditional methods. This chapter delves into such publications in-depth.

### Long Short-Term Memory (LSTM)

Given the availability of studies that use alternative networks, the far more effective methods for forecasting electricity usage are LSTM networks. As a result, the research presented in [27] investigated the usage of multiple LSTM designs for projecting electricity use in the near to medium term. The ideal number of hidden layers with time lags was determined using a GA. The approach's appropriateness was assessed using data from France's usage. Bedi et al. [26] provided a methodology for analysing past data's long-term connections as well as short-term trends in segmented data. Later, utilizing electric load data from India, LSTM was implemented with a shifting frame. DRNN, ANN, and SVR were all surpassed by the model constructed. [28] presents the analysis of power prediction using temperature as an independent factor. They combined power demand data with some other factors such as temperature, air pressure, and humidity to predict power demand usage data. For the short to medium range (24 and 48 hours, 7 and 30 days), the prediction is used. Performance measurements like RSME and MAPE were used to compare the proposed model to other conventional methods. The analysis indicated that LSTM surpasses competing algorithms when it comes to boosting predictive performance. Using a MATLAB toolkit, the LSTM system was autonomously tuned. For many prediction horizons, the results are then compared to that of ARMA, SARIMA, and ARMAX.

External variables were also given to the LSTM by Kwon et al. [29]. These hyperparameters were configured using a heuristic methodology. The electricity system administrator in Korea used 2 years to analyse the model, with only an error of 1.5%. The LSTM presented by [30], on the other hand, provided a data dimension reduction to reduce computing costs. To evaluate the approach's effectiveness, the authors devised two groups of trials. The suggested technique outperformed ANN, ARMA, and ARFIMA in comparisons. The suggested LSTM technique was shown to have an RMSE of 19.7% lesser than that of the

benchmark feed-forward neural network. Subsequently, [31] suggested the CVOA, which has been used to improve the hyper-parameters of an LSTM network. The findings provided surpassed a large variety of deep learning techniques combined with well-known optimization strategies. As a baseline, data on Spanish power use has been employed. [32] proposed a multi-layer bi-directional RNN centred on LSTM and GRU to estimate electricity usage. The researchers beat the findings of ANN and SVR when they looked at peak usage and seasonality individually. Pegalajar et al. introduced in recent years three variants of RNN to anticipate Spanish energy demands, comparing the results against a wide range of ML models and finding that they outperformed every one of them [33].

Utilizing the LSTM algorithm and raw data from a housing complex, Kong et al. [34] established a framework for estimating short-term load. For most datasets, the suggested LSTM architecture had the best prediction performance. Based on the LSTM method, Jiao et al. [35] created a model for forecasting the demand for non-residential buildings. The suggested framework was built using real data from forty-eight non-residential buildings, which included energy consumption statistics. Kim and Cho [36] used a mixture of LSTM and CNNs to design a model for precisely estimating the power usage of a housing complex for a steady electrical supply. CNNs were designed to obtain the features of variables that affect power use in the research, which were initially designed for image analysis. LSTM may detect details in time-series data that have an abnormal pattern. Through this work, a CNN–LSTM model was created by merging the properties of the two algorithms, and electricity consumption was forecasted accurately, which was formerly hard to forecast. Even with minute, hourly, daily, and weekly data, a prominent level of predictive performance was discovered.

Khafaf et al. [37] suggested an LSTM neural network model for forecasting energy consumers' 3-day energy use clusters. Wang et al. [38] used the LSTM neural network to estimate power usage and identify grid irregularities. Regarding energy consumption forecasting in residential and commercial buildings, Khan et al. [39] used a hybrid CNN with an LSTM autoencoder. Regarding everyday natural gas consumption forecast, Wei et al. [40] developed a hybrid single spectrum analysis and an LSTM model. Singaravel et al. [41] examined the effectiveness of LSTM in the planning phase of a building. With four alternative versions of the LSTM model, 201 design scenarios were analysed. LSTM models are found to have superior accuracy and processing time than ANN models. To estimate the power usage of air-conditioning systems, Zhou et al. [42] presented an LSTM model. The number of iterations, the size of the time series feeding the LSTM, and the learning rate were all assessed repeatedly to identify the ideal hyper-parameters.

Furthermore, to increase LSTM's predictive performance, hybrid evolutionary optimisation approaches such as GA and PSO are merged with it. To find the optimum weight matrix or component of the LSTM hyper-parameters, the majority of GA and PSO are used. He et al. [43] presented a hybrid short-load forecasting system combining variational mode decomposition and LSTM networks, with the LSTM network's hyper-parameters tuned by using the Bayesian Optimisation algorithm. For home energy usage forecasting, Kim et al. [44] developed an LSTM network. To determine the best hyperparameters, including learning rate, layer size, and dropout rate, the PSO is used. Yang et al. [45] suggested a hybrid prediction model that combined extreme learning, RNNs, and SVMs. To choose the best weight matrix between these networks, PSO is used. Guo et al. [46] presented an LSTM neural network-based short-term prediction method that captures real-time responses into account. To get the appropriate weight matrix for LSTM, the enhanced GA is applied. The LSTM network's architecture, on the other hand, is set based upon that developer's expertise. Regarding hourly natural gas demand forecasting, Su et al. [47]

developed a hybrid wavelet transform and an LSTM model. Trial-and-error is used to determine the amount of LSTM layers, whereas GA is used to optimise the number of neurons for each LSTM layer.

### Bidirectional-LSTM (BLSTM)

Deep neural networks based on BLSTM have been extensively used in speech recognition and text classification, but they are seldom used in time series prediction or stock market forecasting. RNNs aid in the detection of relationships in sequential data. Bi-LSTMs are a type of RNN that can model data forward as well as backwards. This incorporates several previous stock prices as well as potential future stock values. M. Jia et al. [25] established a methodology for predicting the future price of a stock using a BLSTM neural network. The writers used the GREE stock's statistical data. From January 2017 to May 2019, they compiled data for 568 days. There were fourteen distinctive features in the data, which included open, high, close, volume, and so on. The data was pre-processed and standardised. For the forecast, the close price was used as a baseline. A one-way and two-way LSTM were used to process the pre-processed data. Overfitting can be avoided by using a technique called dropout. The neural network was subjected to the dropout approach. The suggested BLSTM was assessed to both the ARIMA and a standard LSTM model. The RMSE, MAE, and deviation result of the proposed system were determined to assess its accuracy. The suggested approach surpassed both the ARIMA and the LSTM models, as evidenced by the findings. The RMSE and MAE also were lowered by 24.2% and 19.4%, correspondingly.

[23], on the other hand, adopted BLSTM for energy load prediction. The performance of various architectures was examined, including BLSTM, multilayer LSTM, and decoder-encoder architecture. The BLSTM structure was found to have the best results. [24] also used BLSTM for network-wide traffic speed forecast. Both linear SLSTM and BLSTM neural networks were examined in this study. It investigated several LSTM structures and found that two layers of stacked BLSTM surpass other LSTM-based designs. Employing multiple tests to assess BLSTM with SLSTM, we study and assess the integration of a suggested methodology using BLSTM into financial time series forecasting.

### Gradient Boosting Regression Trees

GBRT variants have been pitted against a slew of newer models. Random Forest [56], XGBoost [56][57], CatBoost [58], AdaBoost [59], SVM [60], MLP [61], and CNN and GRU framework [62] have all outperformed GBRT. [63] advocated evaluating seven algorithms per day during the forecast, including GBRT and XGBoost, and then using the most effective for the next day's forecast. One suggested incorporation of the model is to use it as a foundation for a stacking model that includes RF, XGBoost, SVR, and kNN to increase performance and generalisation [65]. Investigating feature selection approaches such as Mutual Information, F-regression, Elastic Net, and Recursive Feature Elimination in tandem with the GBRT model [64] is another field of research that is beyond the scope of this paper.

### XGBoost

[65] developed a new method for anticipating electricity demand. They took daily power load data and translated it to weekly power demand data. This expands the number of features that can be used to forecast load for a lag parameter. For selecting features from the transformed data, the XGBoost method was used. After that, the model was trained, yielding a minimum MAPE of 10%, 97% accuracy, and an MAE of 88.90%.

When used in the literature, the XGBoost approach has had a lot of success. XGBoost or XGBoost hybrid systems surpassed other methods [57], [66], [67], [68]. With its capacity to execute parallel processing, XGBoost is incredibly effective [67]. Before training, XGBoost organises the data, stores it in a block structure, and then uses it in later phases. This increases performance by drastically decreasing processing, as grouping the values of the attributes are amongst the most time-consuming phases in decision tree learning. As a result, in the Wang et al. [56] analysis, XGBoost had the optimum performance. Another aspect of XGBoost is its built-in capacity to choose and save notable features; as a result, additional feature selection algorithms aren't always necessary by the XGBoost algorithm and can even degrade XGBoost efficiency [70].

XGBoost was used in conjunction with other techniques by several academics. CEEDMAN would be used initially to de-noise the original data by dividing it into twelve samples, according to Lu et al. XGBoost surpassed SVM with PSO, Least-squares SVM, CEEMDAN Random Forest, and Radial basis function NN in the 12 samples, and the forecast results were then averaged and denormalized. [73]. Pairing EMD and ARIMA with XGBoost [74], along with solely using ARIMA [71], and merging XGBoost with k-means on Similar Days selection [69] are two further instances. All the XGBoost-based models in the prior articles surpassed other algorithms in contrast. Different architectures such as the CNN and GRU frameworks [62], as well as an ANN [72], were able to best XGBoost in a few articles.

#### LightGBM

A concurrent IDS built on a PID method and LightGBM was proposed by Jin et al. [78]. To accomplish response time outside of reducing attack detection performance, the suggested framework employs two main methods. To begin, the intrusion sensor is a LightGBM. Next, traffic data is properly analysed using PID. Rapid IDS is built on PID methods that have expenses in terms of collaboration and communication. Furthermore, with a connection rate of up to 1.26 Gbit/s, the suggested framework is stable.

When dealing with non-linear traffic data, machine learning is more precise than mathematical statistics. Nonetheless, the lack of understandability of prediction outputs limits ML. A variety of boosting techniques inspired by the GBDT have recently been discovered. XGBoost [75] and LightGBM [76] are two examples. Due to its parallel learning, robustness to anomalous values, and adaptability to missing values, they have been used and coupled with other models in a variety of domains. To predict ultra-short-term wind energy, Ju et al. [77] suggested a hybrid model based on CNN and LightGBM, and the findings confirmed that the inclusion of LightGBM surpassed effectiveness and precision.

#### Facebook Prophet

Prophet is a new prediction technique that has a lot of potential for use in power demand forecasting, according to [48]. In recent years, various applications of this theory have been investigated. Yenidogan et al. [49] investigated two methods for predicting Bitcoin prices: ARIMA and Prophet. The Prophet model appears to have a precision of 94.5 percent, significantly higher than ARIMA's 68 percent. Furthermore, Ashwini Chaudhari [50] employed ARIMA, Prophet, and LSTM networks to predict the costs of cryptocurrencies like Bitcoin, Litecoin, and Ethereum. According to the findings, utilising LSTM and Prophet gave 100% accurate predictions for the three cryptocurrencies, ranging from 93 to 99 percent, whereas using the ARIMA model produced just 82 to 66 percent accuracy. Furthermore, Bianchi et al. [51] used raw data from an Italian utility firm to assess thermal short-term demand forecasting methodologies using the ARM, NARM, and Prophet. In terms of short-term prediction, the ARM outperformed the other models. Das [52] used five different forecasting models to estimate wind speed in two Indian states (SES,

Dynamic Harmonic Regression, NN, ARIMA, and Prophet) (Tamil Nadu and Maharashtra). The best results came from the neural network. The Prophet framework, on the other hand, produced good results and was recommended for future usage.

The usefulness of random forests and Facebook's Prophet in predicting daily flow in a river in the United States is investigated by Papacharalampous & Tyrallis [53]. These prediction algorithms employ historical streamflow data, and random forests also use historical rainfall data. They employ a simple approach based on historical streamflow observations, as well as an MLR model that employs the same data as random forests. Random forests outperform the naive technique overall, but Prophet outperforms it for prediction timescales longer than three days, according to the findings.

Previous research [54] employed it to forecast sales, with the findings given utilising the MAPE level for sales forecasting of several categories of items. On a quarterly estimate, they were able to attain a MAPE of slightly less than 30% for 70% of the commodities. In this study, standard seasonality trends were used. The WMS is in charge of warehousing, a complex collection of procedures. The concept of smart WMS is described in [54]. The implementation of Facebook's Prophet technology for sales forecasting as part of the smart WMS idea and supply business optimization was discussed by Zunic et al. [55]. In several of Bosnia and Herzegovina's important facilities, the concept of smart WMS and sales forecast has been verified in real scenarios and with real data. No study has been undertaken on the Prophet model's effectiveness in projecting long-term power demand, as per prior papers.

# Methodology

## Model

First, we'll go through the RNN and its LSTM structure. Following that, the LSTM will be completely discussed, including its bi-directional properties. Then we'll go into XGBoost and its capabilities in tackling this problem in greater depth. The Facebook Prophet algorithm will be explored, followed by an examination of the LightGBM.

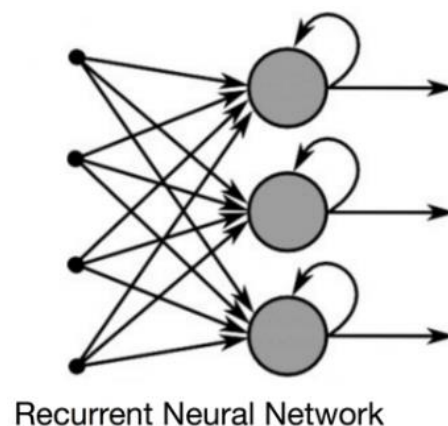
## RNN

Recurrent Neural Networks are robust and effective classes of neural networks ranking among the top algorithms due to them being one with internal memory.

They are an old deep learning algorithm like many others developed in the eighties, but the true potential was discovered only recently. RNNs have been propelled to the forefront through the increased processing power and a large amount of data available today also with the advent of LSTM in the nineties.

Because of the internal memory, they retain vital information about the inputs received, making it very precise in predictions. Therefore, it is the trusted algorithm for time series data, speech, text, weather, video, and much more sequential data. Unlike, other algorithms RNNs have a more comprehensive grasp of the sequence and its context.

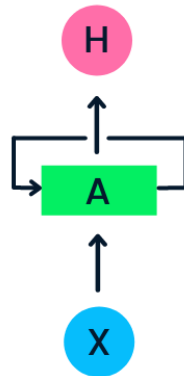
How do they work? The data is passed via a loop. After a choice has been made, it accounts for both the current and past inputs. This is shown in the figure below.



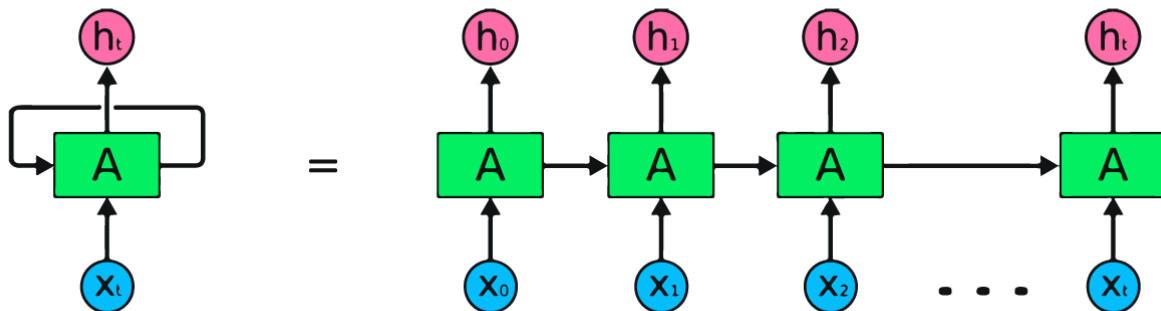
**Figure 1: Recurrent Neural Network**

Unlike a feed-forward neural network, RNNs add the past to the present. This is explained when we feed to the word “apple” as an input; the feed-forward would forget the previous information as it progresses, making it impossible to predict what letter comes next. RNNs remember information by making duplicates of the output and feeding it to the network.

The recurrence is a feedback link that has been created to allow past information to be sent back and received again. Figure 2 illustrates this point. RNNs can be layered and enlarged to better issue modelling while maintaining the previously improved time-series forecasts. Figure 3 depicts a more expanded form of RNN.



**Figure 2: Simple RNN structure with a hidden layer**



**Figure 3: RNN with one hidden layer expanded over time**

## LSTM

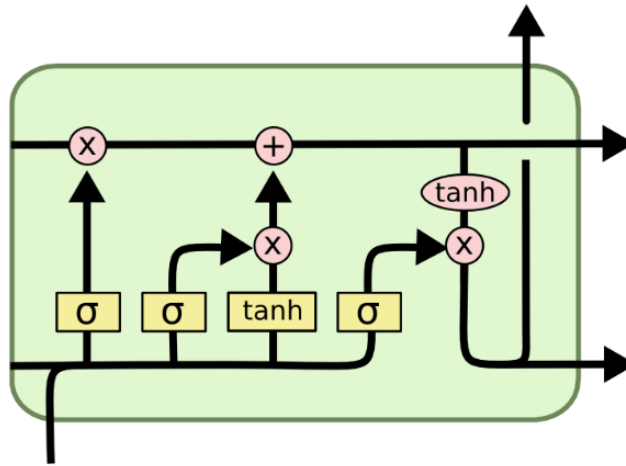
LSTMs are a type of Deep Learning network. It is a class of RNN that can learn long-term connections, which is useful for solving sequence prediction issues. Apart from single data points like photos, LSTM has backpropagation, which means it can analyse the complete sequence of data. This is useful in natural language processing, machine learning, and other areas. The LSTM is a type of RNN that performs exceptionally well on a wide range of issues.

An LSTM model's leading role is played by a memory cell called a 'cell state,' which maintains its state across time. The horizontal line which passes through the top of the diagram below represents the cell state. It can be compared to a conveyor system on which data just passes, unmodified.

In an LSTM, information can be provided to or withdrawn from the cell state, which is controlled by gates. These gates allow information to move in and out of the cell if desired. The method is aided by a pointwise multiplication operation as well as a sigmoid neural - network layer.

The sigmoid layer outputs integers between 0 and 1, with zero indicating that "nothing should be let through" and one indicating that "everything should be let through". This is represented in Figure 4.





**Figure 4: Single LSTM neuron**

LSTM can address a variety of issues that prior learning algorithms, such as RNNs, couldn't. Long-term temporal connections can be efficiently overseen by LSTM without the need for a lot of optimization work. This is used to deal with high-end issues. The LSTM equation is seen below:

$$f_t = \sigma_g(W_f * x_t + U_f * h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + b_o)$$

$$c'_t = \sigma_g(W_c * x_t + U_c * h_{t-1} + b_c)$$

$$c_t = f_t * c_{t-1} + i_t * c'_t$$

$$h_t = o_t * \sigma_c(c_t)$$

Where.

$f_t$  = forget gate,  $i_t$  = input gate,  $o_t$  = output gate

$c_t$  = cell state,  $h_t$  = hidden state

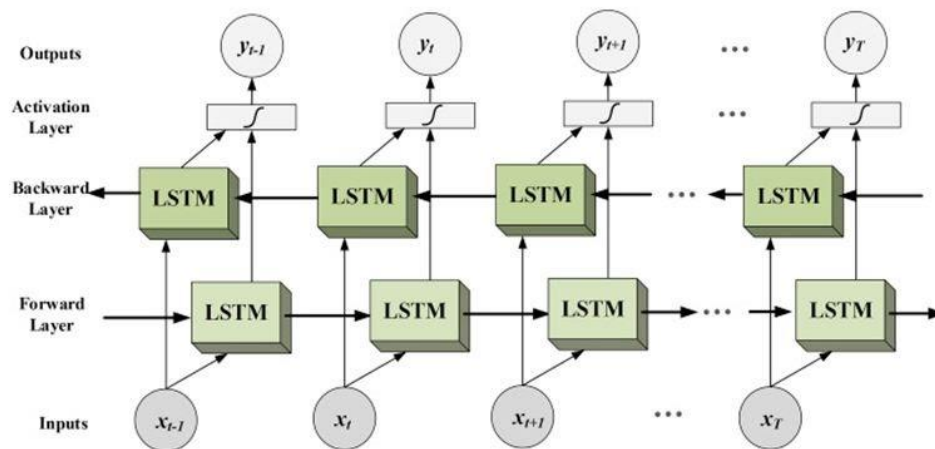
These formulas are only valid for one timestep. This indicates that for the subsequent timestep, it must be recalculated. If there are ten timesteps in a sequence, it will be evaluated ten times. The weights (W), as well as biases (b), aren't time-dependent, which means they don't vary through one phase to the next.

**Bidirectional LSTM**

Bidirectional-LSTM allows any neural network to store sequence information both in forwarding and backward directions.

The data runs in two channels in a bi-lstm, which distinguishes it from a conventional LSTM. We can have input move in one way, either forwards or backwards, with a normal LSTM. However, with bi-directional input, we can have the flow of information in both channels, preserving both futures as well as the past. Let's consider the sentence for a deeper understanding.

The vacant area in the line "guys visit..." cannot be filled. However, if we have an upcoming sentence like "the guys came out of the pub," we can readily anticipate that previous vacant space and then have the model do the same thing, and bi-LSTM enables the network to do so.

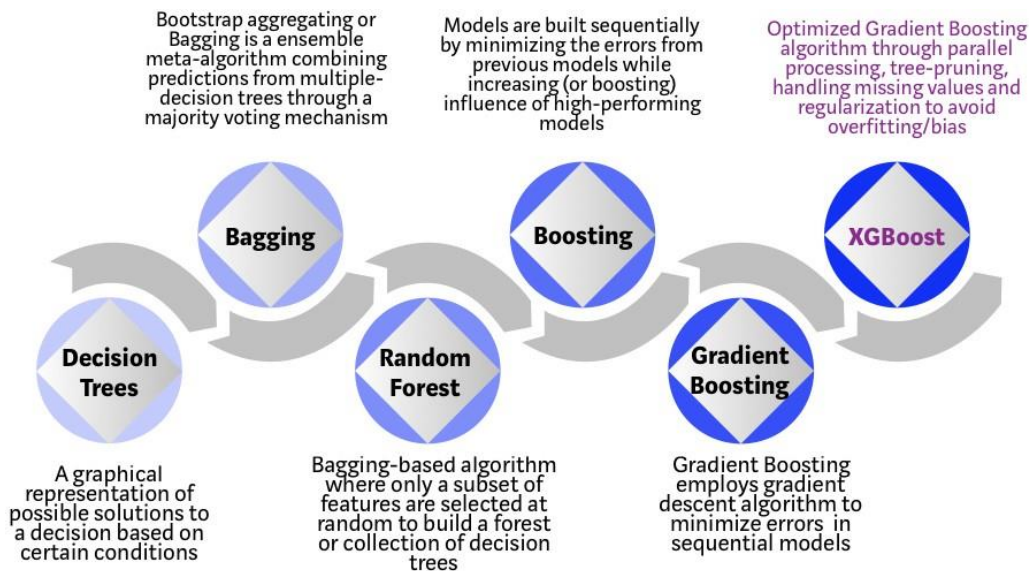


**Figure 5: Bidirectional LSTM Structure**

The information flow from forward and backward levels can be seen in the figure. When jobs requiring sequence to sequence are required, BI-LSTM is typically used. Text categorization, natural language processing, and predictive models can all benefit from this type of network.

## XGBoost

This is a gradient boosting application that utilizes a decision-tree-based group ML model. Artificial NNs surpass most other algorithms or methodologies in prediction issues involving complex data (pictures, text, and so on). Decision tree-based algorithms, on the other hand, are now rated best-in-class for small-to-medium structured/tabular data.



**Figure 6: Evolution of XGBoost from Decision Trees**

The XGBoost algorithm was created as part of a University of Washington research study. In 2016, Tianqi Chen and Carlos Guestrin submitted their article at the SIGKDD Conference, which ignited the ML industry. Since its inception, this algorithm has been acknowledged for not only topping a slew of Kaggle contests but also serving as the brains behind several innovative industrial applications. As an outcome, the open-source projects have a robust ecosystem of data scientists collaborating with them, with 350 members and 3,600 contributions on GitHub. It sets itself apart through this means:

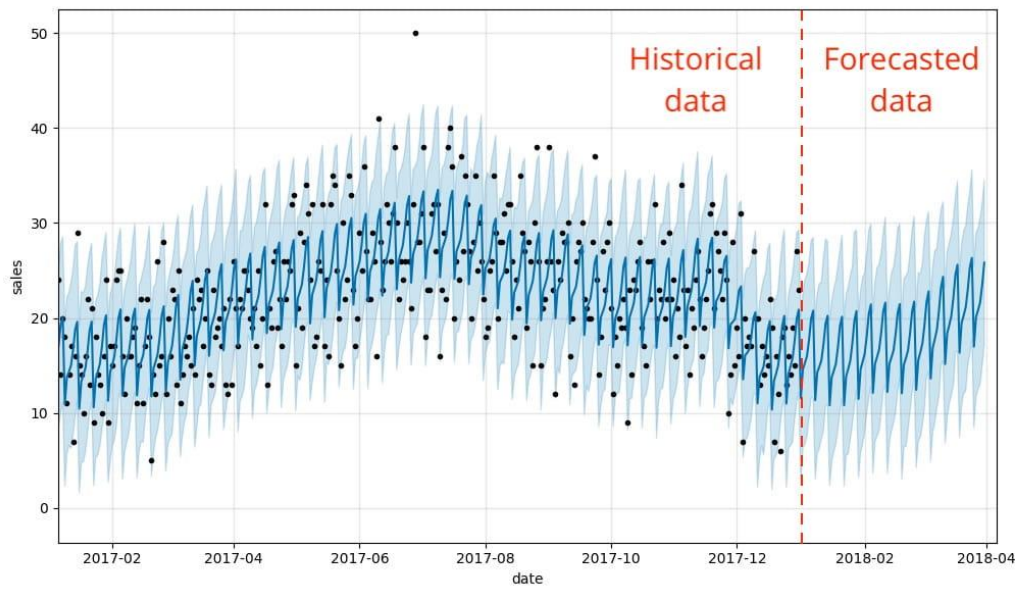
- Regression, classification, ranking, and user-defined prediction issues can all be solved using this tool.
- Very portable i.e., runs on popular Operating Systems (Windows, Linux, etc).
- Supports all major programming languages
- Supports cloud integration i.e., Azure, AWS, and other ecosystems.

With these facts, it does not mean the XGBoost algorithm should be used for all ML problems. A good rule of thumb is to explore both new and old frameworks to tackle the desired problems. It will only be a matter of time before a new model framework emerges that outperforms XGBoost in respect of predictive accuracy, adaptability, explainability, and practicality. XGBoost, on the other hand, will continue to rule the Machine Learning field till a serious opponent emerges.

## Prophet

Prophet is a time-series forecasting method which is based on an additive model that predicts non-linear patterns including annual, weekly, and daily seasonality as well as vacation effects. It performs effectively with time series with substantial seasonal influences and past data from multiple seasons. Prophet is forgiving of missing information and trend changes, and it usually manages anomalies efficiently.

Facebook's Core Data Science team published Prophet as open-source software. It may be downloaded from [CRAN](#) and [PyPI](#).



**Figure 7: Example of the Prophet algorithm output**

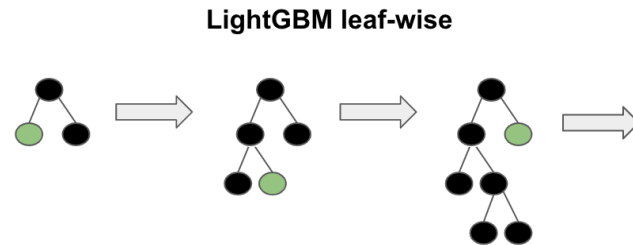
Prophet is utilized in a variety of Facebook apps to generate accurate predictions for strategy and goal setting. In most circumstances, we've found it to outperform any other strategy. We use Stan to fit models so that you may get forecasts in a matter of seconds. With no effort, you can get a fair prediction on sloppy data. Anomalies, missing information, and significant improvements in your time-series data are not a problem for Prophet. The Prophet method gives customers a lot of options for tweaking and adjusting predictions. By combining your subject expertise with human-interpretable variables, you can enhance your predictions. The Prophet process has been coded in R and Python, however, the fundamental Stan code for fitting is the same across both. To get predictions, use any language you're most confident with.

### LightGBM

This is a gradient boosting concept that includes tree-based learning methods, which are regarded to be quite a strong processing algorithm. This is thought to be a quick-processing algorithm.

Whilst trees of other algorithms develop horizontally, the LightGBM algorithm develops vertically, which means it expands leaf-wise while many other algorithms develop level-wise. To expand, LightGBM selects the leaf with the greatest loss. When expanding the same leaf, this can reduce loss more than a level-wise method.

Existing algorithms are finding it increasingly difficult to provide quick results as the number of data grows exponentially. For its computational power and ability to deliver findings quickly, LightGBM is dubbed "Light." It uses less bandwidth to operate and can oversee massive volumes of data. Since the goal of the method is to achieve high accuracy while also bracing GPU leaning, this makes it the most often utilized technique in Hackathons.



**Figure 8: Diagrammatic representation of Leaf-Wise Tree Growth**

Unlike many other boosting techniques that develop trees level-by-level, LightGBM breaks the tree leaf-by-leaf. It grows the leaf with the greatest delta loss. The leaf-wise approach has a smaller loss than that of the level-wise algorithm because the leaf is stationary.

LightGBM isn't suited for small samples. Because of its sensitivity, it can quickly overfit little data. It's suitable for data with more than 10,000 entries. There are no predetermined criteria for considering whether to use LightGBM. This can be utilized for enormous amounts of data, particularly when great accuracy is required.

#### Loss Functions

MAE, MSE, RMSE, and R2 are some of the evaluation metrics employed. The MAE indicates how well the model performs on most data by describing the mean error of the model's prediction. Because RMSE is derived from MSE, which gives significant weights to outliers, it is a good measure for predicting outliers. R2 is a metric that describes how effectively a model can represent and fit data. Naturally, the MAE, MSE, and RMSE should be as near to zero as feasible, and R2 should be as close to one as possible.

#### Mean Squared Error

The degree of inaccuracy in statistical models is measured by the mean squared error (MSE). It is determined the average squared discrepancy between experimental and predicted values. When a program has no mistakes, the MSE constitutes zero. Its value rises as the model inaccuracy rises. The mean squared deviation (MSD) is another name for the mean squared error.

$$\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}$$

Where:

$n$  = number of data points

$y_i$  = observed values

$\hat{y}_i$  = predicted values

## Mean Absolute Error

Mean Absolute Error (MAE) is the most basic measure of forecast accuracy. It is just the average of the absolute mistakes, as the name implies. The absolute error is defined as the absolute value of the difference between the predicted and actual value. It is a metric that quantifies the precision of continuous variables. This shows us how large of an error we may expect on average from the forecast. It is a statistic that assesses the average magnitude of errors in a set of predictions without taking their direction into account. The Mean Absolute Error is the average of the absolute differences between prediction and actual observation over the test sample, assuming that all individual deviations are equally weighted. The average model prediction error is expressed in units of the variable of interest in both Mean Absolute Error and Root Mean Square Error. The Mean Absolute Error and the Root Mean Square Error both have a range of 0 to 1 and are unaffected by error direction.

$$\frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

Where:

$y_i$  = prediction

$x_i$  = true value

$n$  = total number of data points

## Root Mean Squared Error

The root mean square error (RMSE) is the residuals' standard deviation (prediction errors). Residuals are a representation of how far observed values deviate from the trendline, and RMSE is a representation of how far these results are spread out. To put it another way, it shows how closely the data is grouped around the fitted line. In climatology, forecasting, and regression analysis, root mean square error is widely used to check experimental results. When normalized observations and forecasts are utilized as RMSE inputs, the correlation coefficient has a direct link. For example, if the Pearson correlation equals one, the Relative error is zero because all the observations are on the trendline (and therefore no errors are detected).

$$\sqrt{\frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{n}}$$

Where:

$i$  = variable  $i$

$n$  = number of non-missing data points

$x_i$  = actual observations

$\hat{x}_i$  = predicted observations

R-squared

The coefficient of determination, or  $R^2$ , is a statistic that indicates how well a model fits the data. It is a statistical tool about how well the regression line resembles the actual findings in the context of regression. When a statistical approach is used to forecast future outcomes or test hypotheses, it is critical to remember this. The maximum displacement of the sample is away from the mean, all squared, is the optimum sum of squares, and the sum squared prediction is the aggregate of the residuals squared. It will take the value of 0 or 1 because it is a percentage.

$$1 - \frac{\text{sum of squared regression (SSR)}}{\text{total sum of squares (SST)}}$$

$$1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2}$$

## Data Analysis

There are three data sets, each for every building, namely the electricity data set, water data set, and the gas data set collected from the University energy manager. Each dataset contains forty-eight columns of half-hourly readings being recorded and stored for later analysis. This chapter would discuss the analysis and preprocess conducted before model training can be conducted. 4 steps are necessary to complete the analysis process are the cleaning/preprocess phase where the data is reformatted and any null values are removed, the decomposition phase where we take a closer look at the data and denote any trends occurring, checking if the data is stationary (without trends) or not which is expected in the case of this project, and finally prepare the dataset for training. The next subsections will go through analysing the building datasets and noting any findings.

### Queens Building Dataset

#### Preprocess/Cleaning

	Date	0:30	1:00	1:30	2:00	2:30	3:00	3:30
0	3/3/2008	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	3/4/2008	50.975	50.997	51.000	54.482	54.996	55.877	55.097
2	3/5/2008	55.865	56.000	55.105	55.837	56.023	56.872	56.105
3	3/6/2008	53.877	53.114	52.996	53.757	54.060	54.013	53.871
4	3/7/2008	55.638	55.140	54.144	54.000	53.974	53.144	53.851

5 rows × 49 columns

Figure 9: Queen's Building Electricity Reading before cleaning

As the figure above shows, the raw data collected is recorded every 30 minutes from the assigned meter and is not comprehensive to understand, so a preprocessing is essential.

	value
Datetime	
2008-03-03 00:00:00	0.0
2008-03-03 00:30:00	0.0
2008-03-03 01:00:00	0.0
2008-03-03 01:30:00	0.0
2008-03-03 02:00:00	0.0
2008-03-03 02:30:00	0.0
2008-03-03 03:00:00	0.0
2008-03-03 03:30:00	0.0
2008-03-03 04:00:00	0.0
2008-03-03 04:30:00	0.0

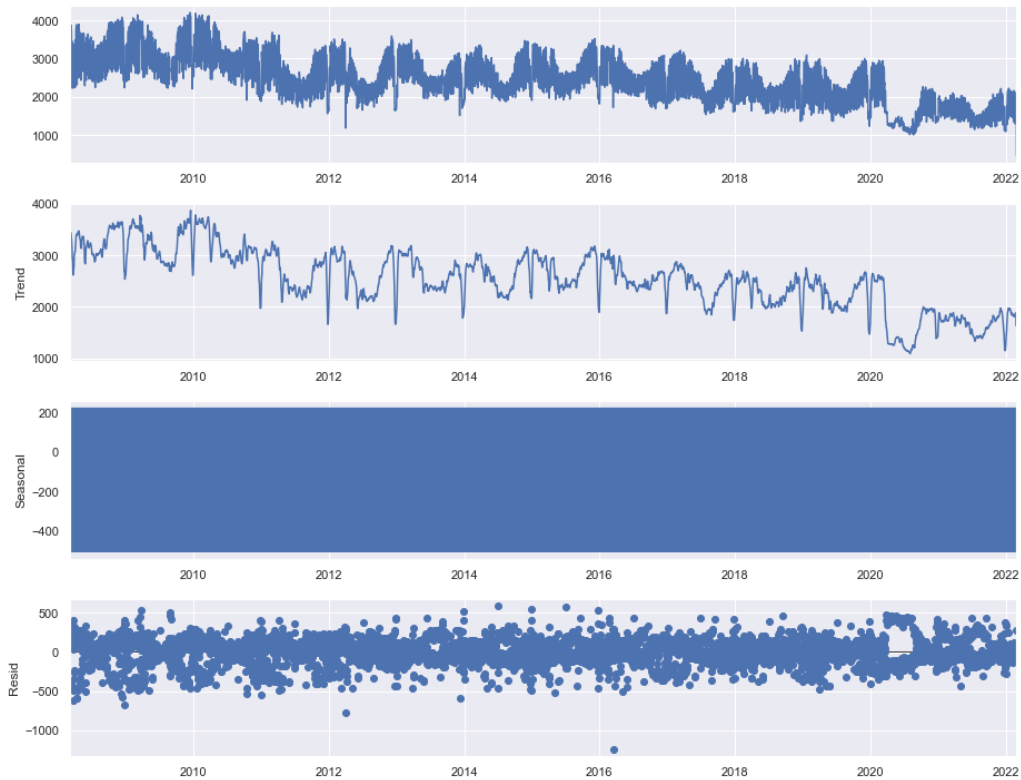
Figure 10: Queen's Building Electricity Reading after cleaning

After cleaning the data, it is now easy to work with and we can move to the next step in our analysis.



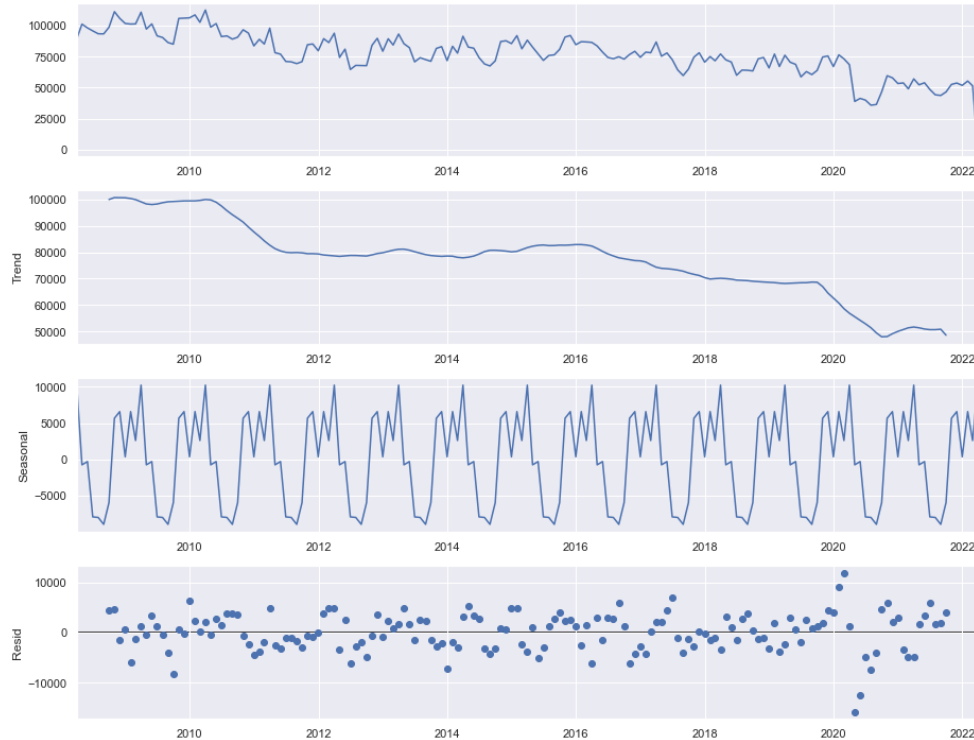
## Visualize

Visualization of the data can be achieved in numerous ways but to save time we will call upon the statsmodel library which is used in conducting statistical tests. To use the library, we must resample or data into a daily format giving us the figure.



**Figure 11: Electricity Daily Dataset**

From the figure above we can see that the library was able to display four different plots, the original data, the trend of the data, the seasonality, and the residuals (leftovers). The seasonality fails to get detected by the library; a further resampling is essential to observe the seasonal trend.



**Figure 12:Electricity Monthly Dataset**

From this new figure, we can see a more defined seasonal trend and it is exhibiting a clear downward trend as the year increases. The seasonality also shows more usage in the winter and spring months and significantly less usage in the summer months.

Check for stationarity

Using the above figure, we can see that our data is non-stationary which increases the desired model's predictive accuracy. Another way of testing is by performing an Augmented Dickey-Fuller (ADF) test, a unit root test, which is a statistical test. This test is conducted under certain conditions:

- Null Hypothesis: Data is non-stationary
- Alternate Hypothesis: Data is stationary

This test may show that the series is non-stationary if the null hypothesis fails to be rejected. The condition to reject the null hypothesis is:

- If the test statistic < critical value and p-value < 0.05

**Table 1: Queen's Building ADF Test**

	Test Statistic	P-Value	Critical Values
Electricity	-2.983	0.036552	1% -> -3.432 5% -> -2.862 10% -> -2.567
Gas	-5.183	0.000009	1% -> -3.432 5% -> -2.862 10% -> -2.567
Water	-4.54	0.0002	1% -> -3.432

			5% -> -2.862
			10% -> -2.567

From the table, we can see that all the datasets for the Queen's Building show that it is non-stationary (has trends & seasonality)

### Correlation

We need to see how the three datasets are correlated with each other and to achieve this we use a Pearson correlation function:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Where

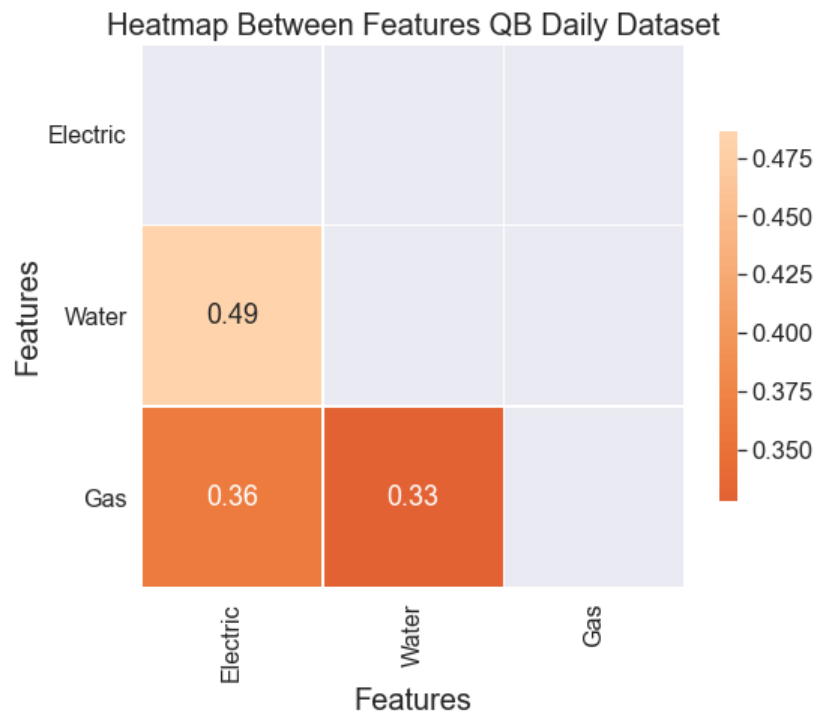
$r$  – correlation coefficient

$x_i$  - values of the x-variable in the data

$\bar{x}$  - mean of the values of the x

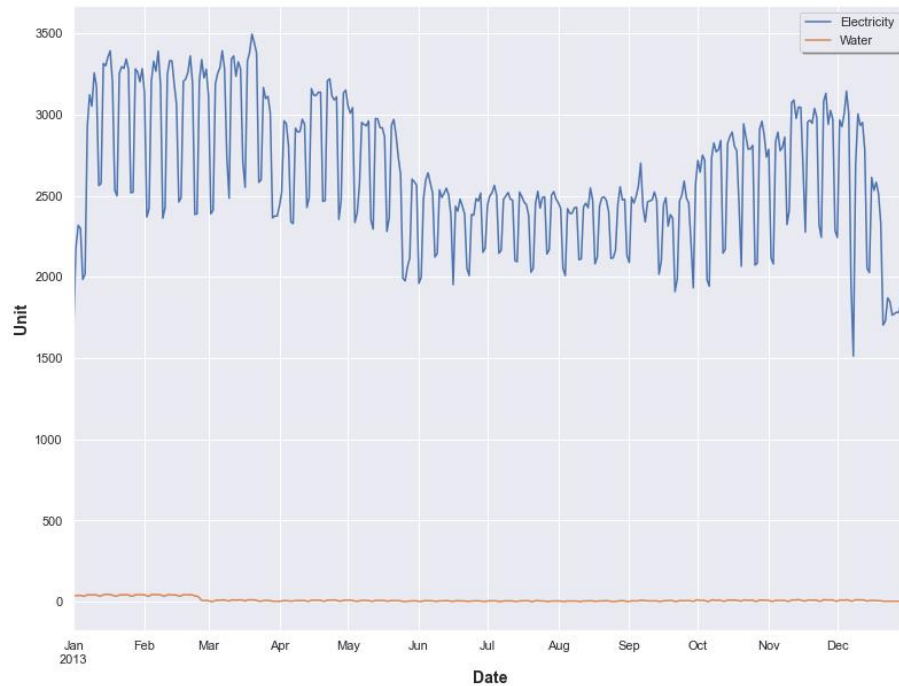
$y_i$  – values of the y-variable in the data

$\bar{y}$  – mean of the values in y

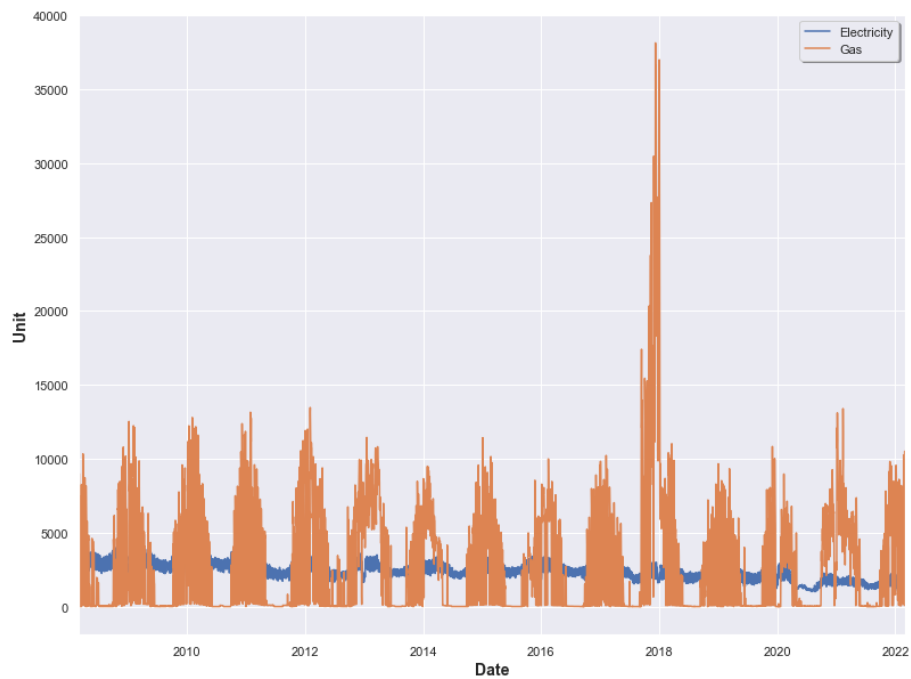


**Figure 13: Heatmap between Queen's Building Daily Dataset**

Figure 13, all datasets are positively correlated with the electricity and water datasets having the highest correlation. Further examining the correlation, we can see that this is not the case as the values never intersect but Electricity and Gas have the same pattern



**Figure 14: Example showing a correlation between Electricity and Water Datasets**

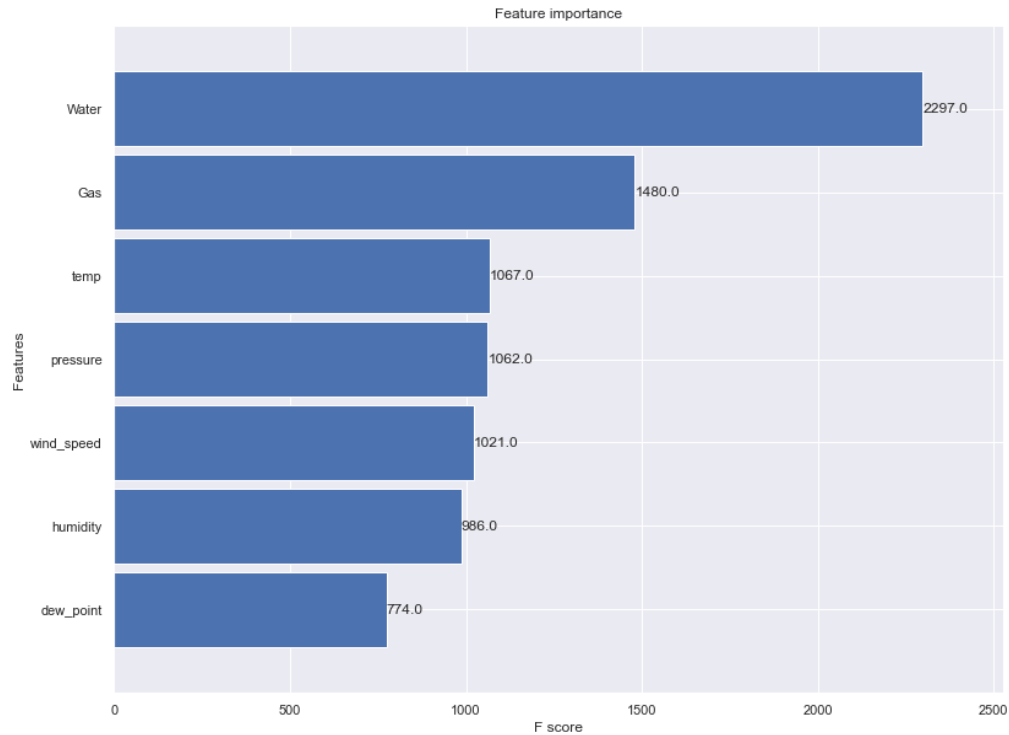


**Figure 15: Example showing a correlation between Electricity and Gas Datasets**

## Feature Importance

XGBoost has a feature importance function which determines how each feature contributes to the selected target variable during regression.

Target Variable – Electricity



**Figure 16: Feature importance based on electricity**

The two most notable features are Gas and Water which is expected, and temperature plays an equal role in electricity usage.

Hugh Aston Building Dataset

Preprocess/Cleaning

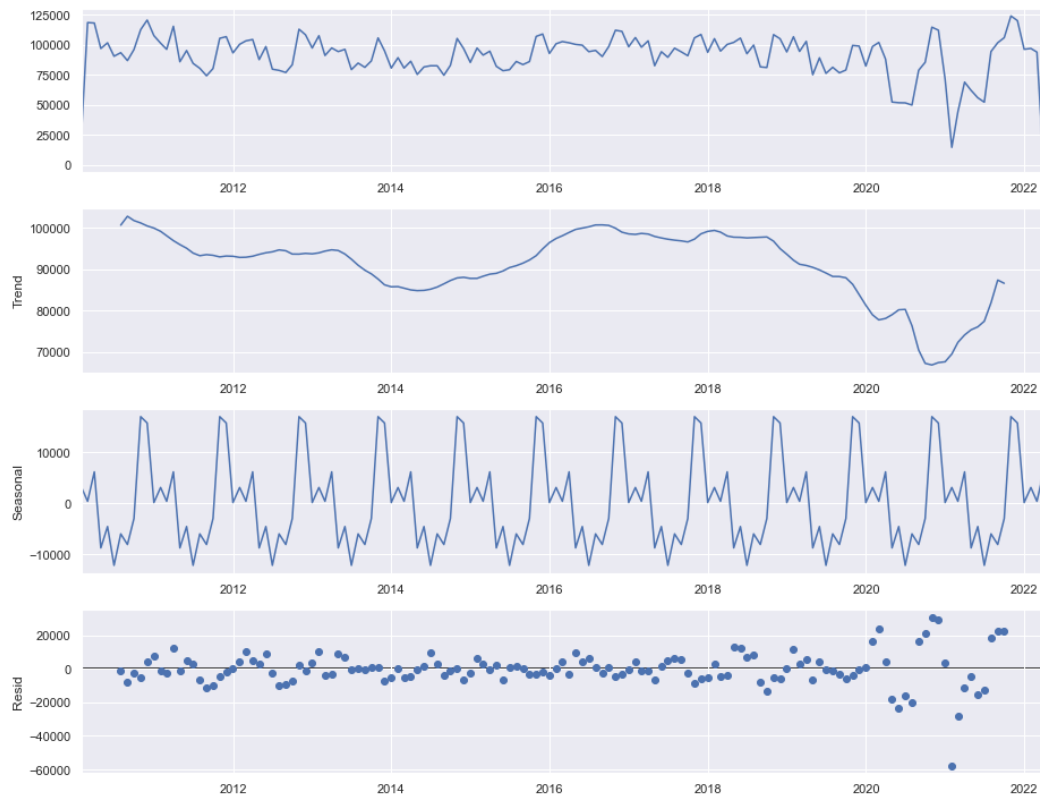
	Date	0:30	1:00	1:30	2:00	2:30	3:00	3:30	4:00
0	1/25/2010	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	1/26/2010	38.420	39.369	39.801	38.259	43.921	40.821	39.303	37.411
2	1/27/2010	40.800	39.213	40.976	42.198	40.277	40.303	42.819	41.755
3	1/28/2010	39.911	38.658	39.931	38.606	40.285	39.607	40.999	39.786
4	1/29/2010	37.091	42.577	37.711	37.919	40.732	35.796	39.858	37.781
5 rows × 49 columns									

**Figure 17: Hugh Aston Electricity Reading Before Cleaning**

	value
Datetime	
2010-01-25 00:00:00	0.0
2010-01-25 00:30:00	0.0
2010-01-25 01:00:00	0.0
2010-01-25 01:30:00	0.0
2010-01-25 02:00:00	0.0
2010-01-25 02:30:00	0.0
2010-01-25 03:00:00	0.0
2010-01-25 03:30:00	0.0
2010-01-25 04:00:00	0.0
2010-01-25 04:30:00	0.0

**Figure 18: Hugh Aston Electricity Reading After Cleaning**

Visualize



**Figure 19: Electricity Monthly Dataset**

We can see from [Figure 19](#) above that there is a clear trend occurring. There was a downward trend happening from 2013 before picking back up in 2015 and maintaining that momentum till 2019 when it starts to drop again which might be an indicator of better energy-saving tactics employed in the building.

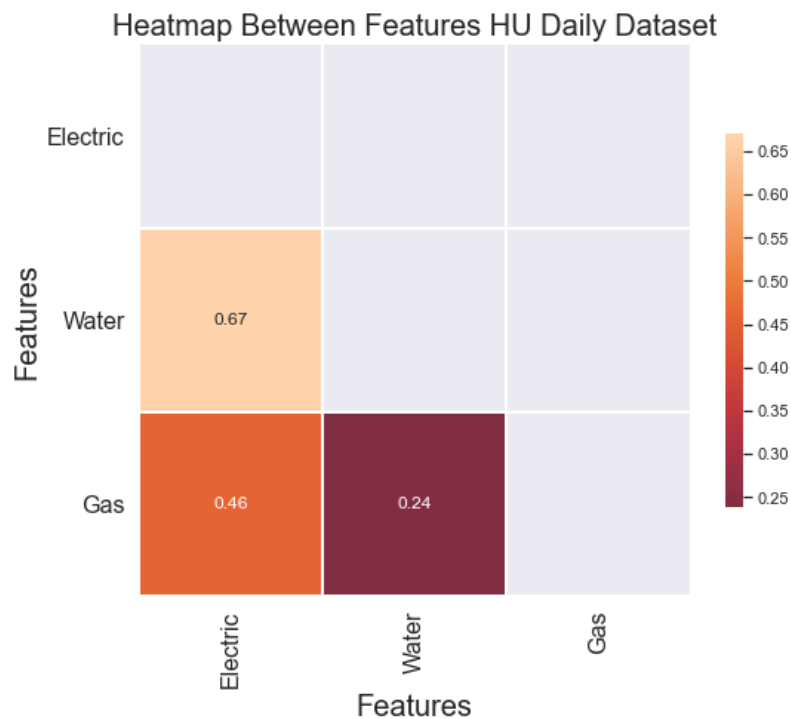
Check for stationarity

**Table 2: Hugh Aston ADF Test**

	Test Statistic	P-Value	Critical Values
Electricity	-6.193937	0.000000	1% -> -3.432 5% -> -2.862 10% -> -2.567
Gas	-5.006579	0.000022	1% -> -3.432 5% -> -2.862 10% -> -2.567
Water	-6.168843	0.000000	1% -> -3.432 5% -> -2.862 10% -> -2.567

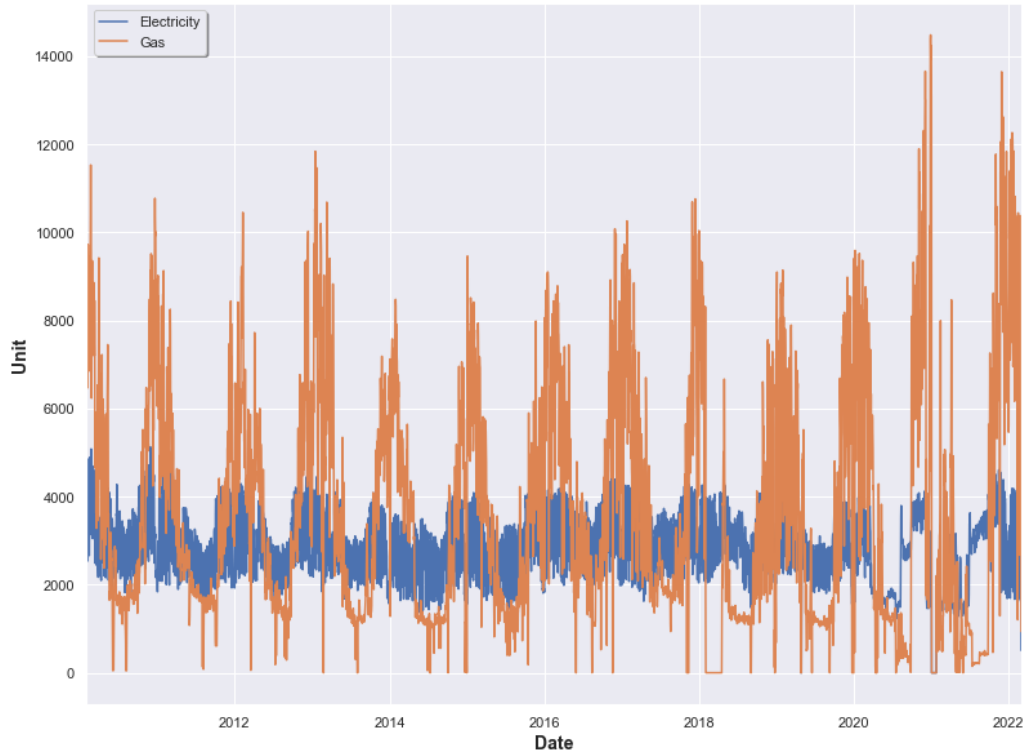
From the table, we can see that all the datasets for the Hugh Aston Building show that it is non-stationary (has trends & seasonality)

Correlation



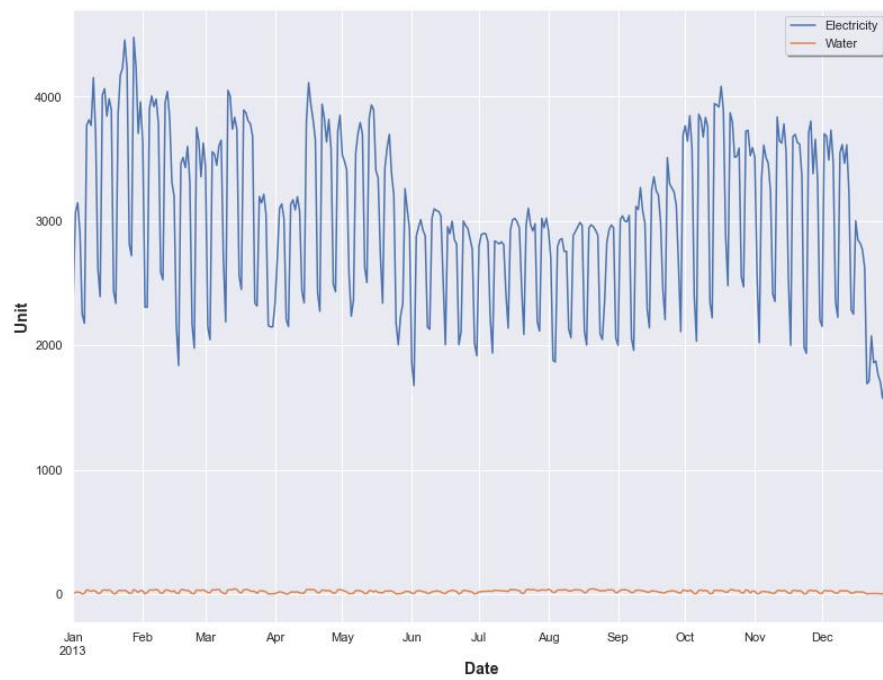
**Figure 20: Heatmap between Hugh Aston Daily Dataset**

We see the same pattern occurring also within the Hugh Aston dataset as the Queen's Building dataset where the highly correlated features do not show any correlation visually.



**Figure 21: Electricity and Gas Correlation**

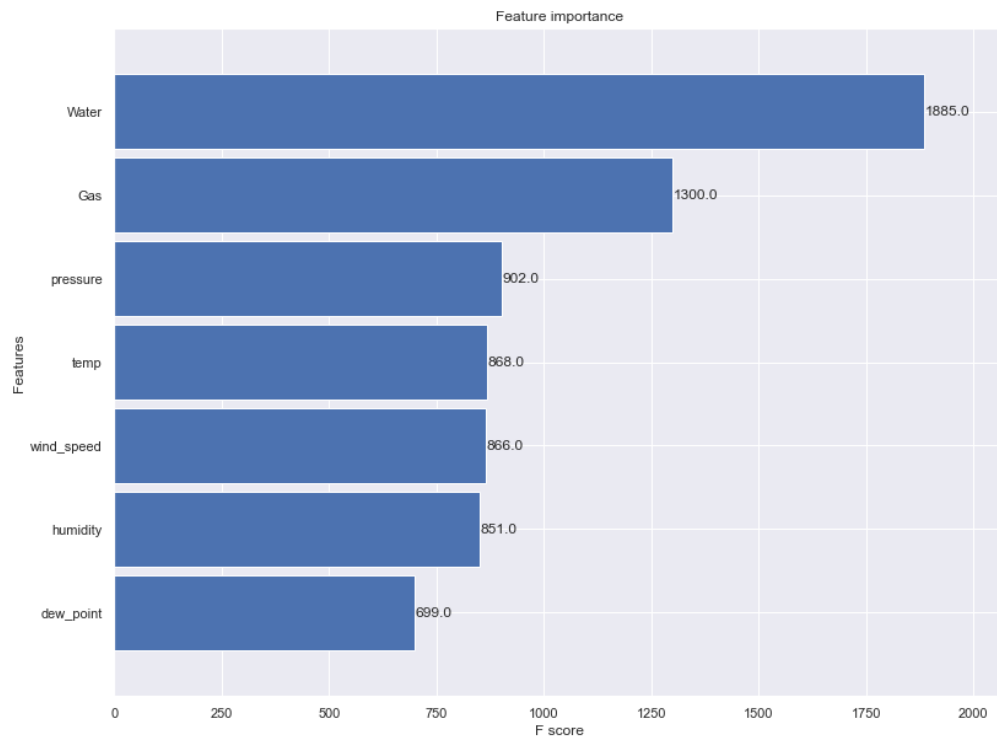
As the electricity consumption rises so does the gas consumption which is expected during warm and cold months.



**Figure 22: Electricity and Water Correlation**



## Feature Importance



**Figure 23: Feature Importance**

By applying XGBoost regressors, the important feature of the dataset is shown above. Similarly, with the previous dataset, Water and Gas features show to be important to electricity consumption. The only difference is that the air pressure shows to be more important than the temperature which would need further examination.

## Gateway House Building Dataset

### Preprocess/Cleaning

	Date	0:30	1:00	1:30	2:00	2:30	3:00	3:30	4:00
0	5/7/2008	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	5/8/2008	11.593	11.033	11.285	14.218	10.644	11.374	11.047	11.375
2	5/9/2008	12.168	11.206	10.073	10.997	14.464	11.528	12.849	11.070
3	5/10/2008	10.383	10.415	10.602	10.293	10.283	10.958	10.753	10.824
4	5/11/2008	11.721	8.893	9.132	10.808	11.878	11.038	10.061	11.020

5 rows × 49 columns

**Figure 24: Gateway House Electricity Dataset before cleaning**

	value
Datetime	
2008-05-07 00:00:00	0.0
2008-05-07 00:30:00	0.0
2008-05-07 01:00:00	0.0
2008-05-07 01:30:00	0.0
2008-05-07 02:00:00	0.0
2008-05-07 02:30:00	0.0
2008-05-07 03:00:00	0.0
2008-05-07 03:30:00	0.0
2008-05-07 04:00:00	0.0
2008-05-07 04:30:00	0.0

Figure 25: Gateway House Electricity Dataset after cleaning

Visualize

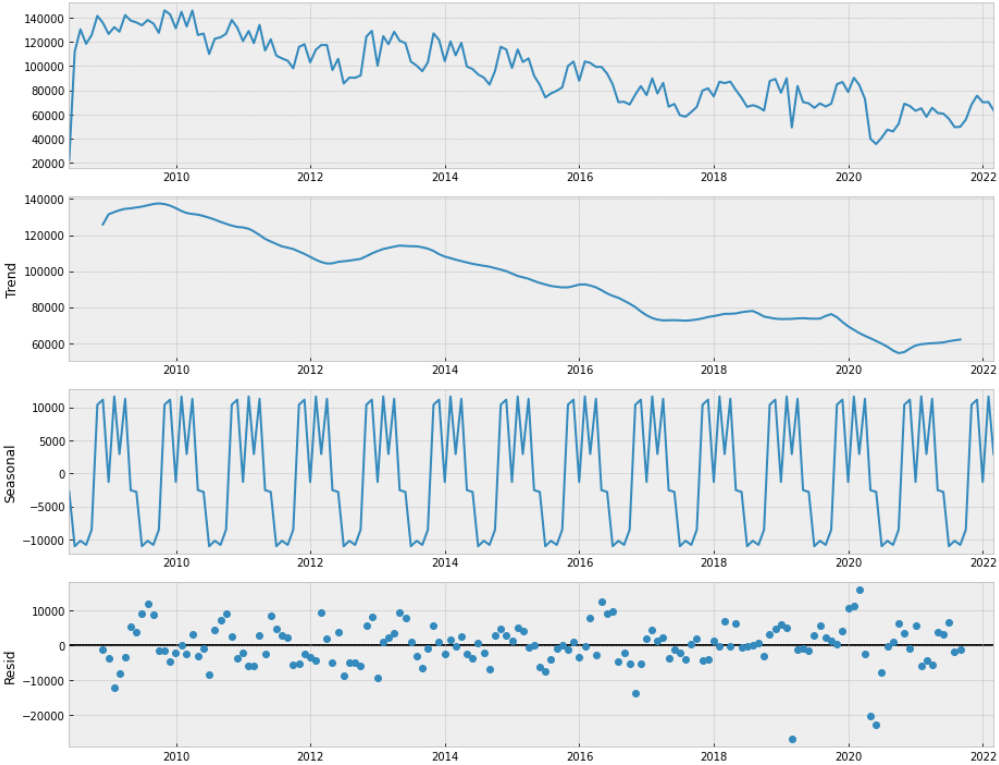


Figure 26: Electricity Monthly Dataset

There is a very distinct downward trend occurring which shows that the Gateway House electricity consumption rates are increasingly getting better through excellent energy-saving tactics. This shows that even without the pandemic happening the rate of consumption will always improve.

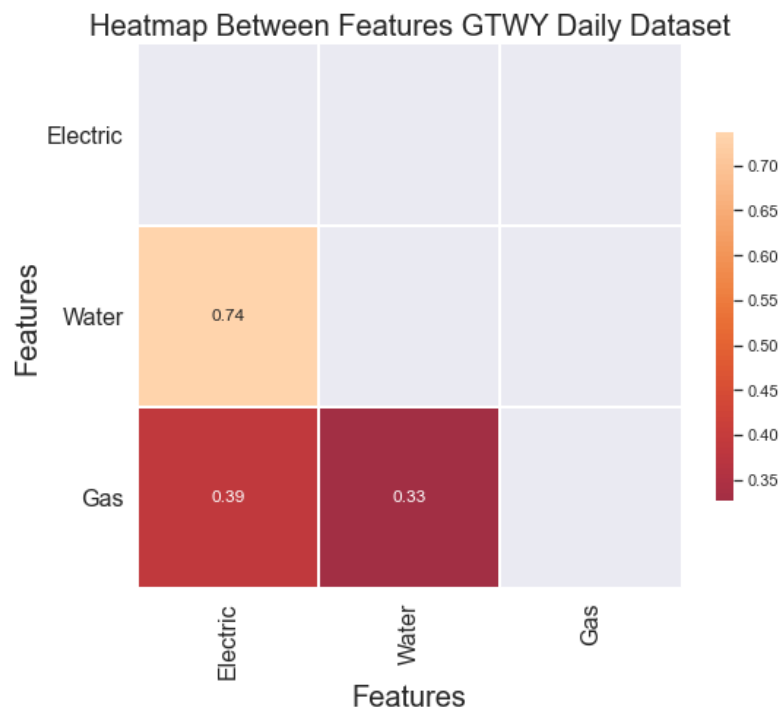
Check for stationarity

**Table 3: Gateway House ADF Test**

	Test Statistic	P-Value	Critical Values
Electricity	-3.446605	0.009469	1% -> -3.432 5% -> -2.862 10% -> -2.567
Gas	-4.618122	0.000120	1% -> -3.432 5% -> -2.862 10% -> -2.567
Water	-5.722182	0.000001	1% -> -3.432 5% -> -2.862 10% -> -2.567

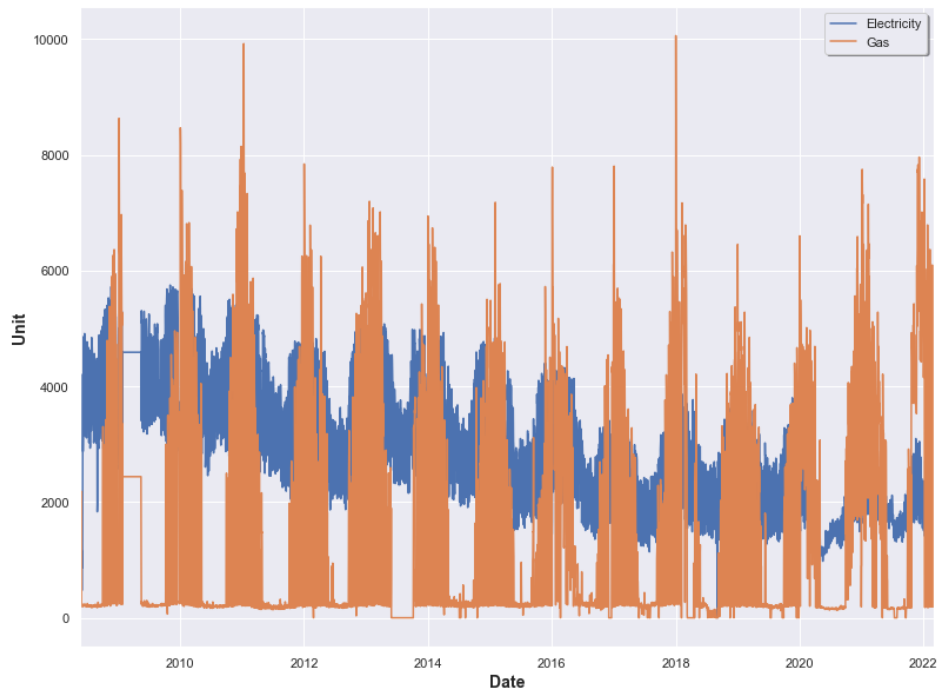
From the table, we can see that all the datasets for the Gateway House Building show that it is non-stationary (has trends & seasonality)

Correlation



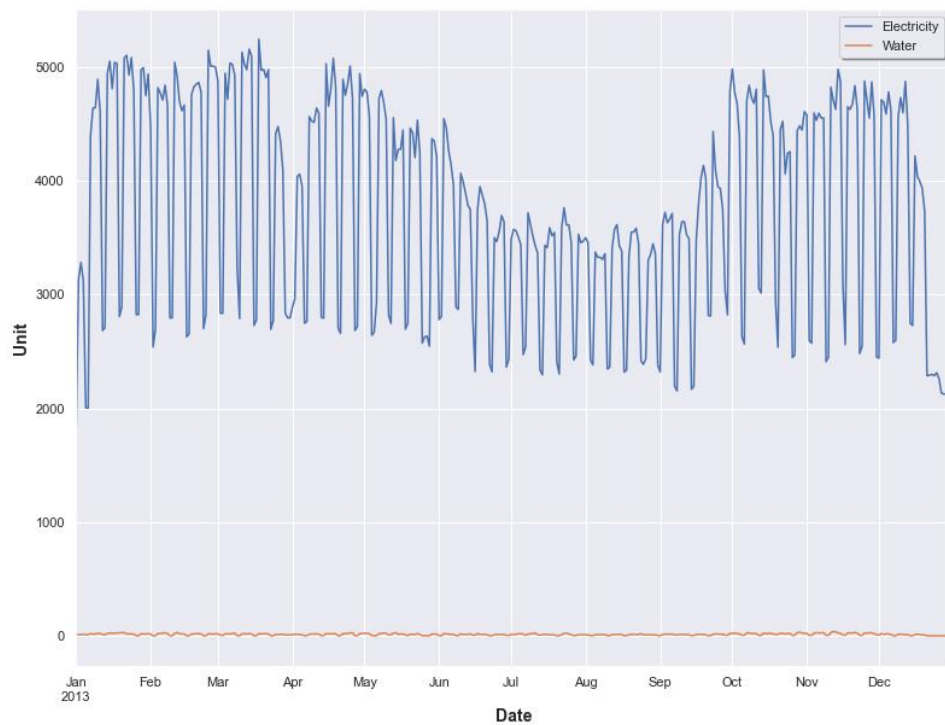
**Figure 27: Heatmap between Gateway House Daily Dataset**

We see the same pattern occurring also within the Hugh Aston dataset as the Queen's Building dataset where the highly correlated features do not show any correlation visually.



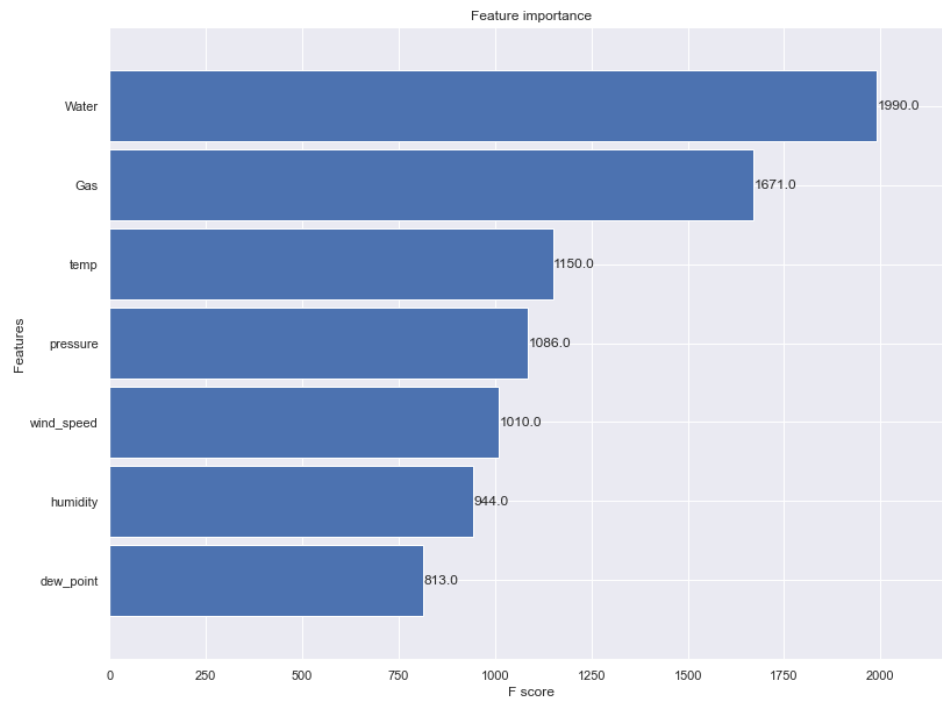
**Figure 28: Correlation between Electricity and Gas**

At the start of the dataset, the electricity and gas features seem to struggle but they started exhibiting normal behaviours afterwards till 2020 when the sudden drop in usage happened.



**Figure 29: Correlation between Electricity and Water**

## Feature Importance



**Figure 30: Feature Importance**

By applying XGBoost regressors, the important feature of the dataset is shown above. Similarly, with the previous dataset, Water and Gas features show to be important to electricity consumption. This is extremely like the Queen's Building dataset and shows that temperature impacts the rate of consumption.

## Weather Dataset

The analysis of the weather dataset has been broken into three sections; Preprocessing, Visualization and Feature Selection. The last section will be the final data which will then be added to our building data for training as a regressor.

## Preprocessing

### View Data

	count	mean	std	min	25%	50%	75%	max
dt	124176.0	1.422661e+09	1.290480e+08	1.199146e+09	1.310903e+09	1.422661e+09	1.534418e+09	1.646176e+09
timezone	124176.0	2.079706e+03	1.778142e+03	0.000000e+00	0.000000e+00	3.600000e+03	3.600000e+03	3.600000e+03
lat	124176.0	5.263688e+01	1.421091e-14	5.263688e+01	5.263688e+01	5.263688e+01	5.263688e+01	5.263688e+01
lon	124176.0	-1.139759e+00	0.000000e+00	-1.139759e+00	-1.139759e+00	-1.139759e+00	-1.139759e+00	-1.139759e+00
temp	124176.0	1.040077e+01	5.906019e+00	-1.072000e+01	6.140000e+00	1.023000e+01	1.447000e+01	3.488000e+01
visibility	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
dew_point	124176.0	7.248109e+00	4.999126e+00	-3.523000e+01	3.550000e+00	7.480000e+00	1.104000e+01	2.221000e+01
feels_like	124176.0	8.762749e+00	7.195326e+00	-1.294000e+01	3.160000e+00	9.560000e+00	1.413000e+01	3.729000e+01
temp_min	124176.0	9.365455e+00	5.912856e+00	-1.375000e+01	5.130000e+00	9.230000e+00	1.354000e+01	3.323000e+01
temp_max	124176.0	1.146047e+01	6.006574e+00	-8.410000e+00	7.210000e+00	1.125000e+01	1.552000e+01	3.743000e+01
pressure	124176.0	1.014386e+03	1.138366e+01	9.590000e+02	1.008000e+03	1.015000e+03	1.022000e+03	1.049000e+03
sea_level	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
grnd_level	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
humidity	124176.0	8.200810e+01	1.260884e+01	2.000000e+00	7.400000e+01	8.500000e+01	9.200000e+01	1.000000e+02
wind_speed	124176.0	4.431663e+00	2.150646e+00	3.000000e-02	2.790000e+00	4.150000e+00	5.740000e+00	1.597000e+01
wind_deg	124176.0	2.001849e+02	9.208754e+01	0.000000e+00	1.400000e+02	2.200000e+02	2.660000e+02	3.600000e+02
wind_gust	3114.0	4.353953e+00	2.498245e+00	5.100000e-01	2.570000e+00	4.110000e+00	5.650000e+00	1.593000e+01
rain_1h	18995.0	7.528097e-01	1.940067e+00	1.000000e-01	1.700000e-01	3.300000e-01	7.300000e-01	4.826000e+01
rain_3h	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
snow_1h	498.0	7.510843e-01	1.603744e+00	1.000000e-01	1.800000e-01	3.100000e-01	6.275000e-01	1.600000e+01
snow_3h	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clouds_all	124176.0	7.033787e+01	3.450556e+01	0.000000e+00	4.300000e+01	8.800000e+01	1.000000e+02	1.000000e+02
weather_id	124176.0	7.557358e+02	1.093496e+02	5.000000e+02	8.000000e+02	8.030000e+02	8.040000e+02	8.040000e+02

Figure 31: Raw Weather Data Statistics

From the above figure, we can see that there are a lot of features that could be used and some showing null values. The next step is to remove any null and incomplete data.

### Remove null value

This can be achieved by using the built-in `dropna()` function that is available in python and specifying the method of removal which in this case would be 'any' null values found in the columns. The resulting data then becomes:

	count	mean	std	min	25%	50%	75%	max
dt	124176.0	1.422661e+09	1.290480e+08	1.199146e+09	1.310903e+09	1.422661e+09	1.534418e+09	1.646176e+09
timezone	124176.0	2.079706e+03	1.778142e+03	0.000000e+00	0.000000e+00	3.600000e+03	3.600000e+03	3.600000e+03
lat	124176.0	5.263688e+01	1.421091e-14	5.263688e+01	5.263688e+01	5.263688e+01	5.263688e+01	5.263688e+01
lon	124176.0	-1.139759e+00	0.000000e+00	-1.139759e+00	-1.139759e+00	-1.139759e+00	-1.139759e+00	-1.139759e+00
temp	124176.0	1.040077e+01	5.906019e+00	-1.072000e+01	6.140000e+00	1.023000e+01	1.447000e+01	3.488000e+01
dew_point	124176.0	7.248109e+00	4.999126e+00	-3.523000e+01	3.550000e+00	7.480000e+00	1.104000e+01	2.221000e+01
feels_like	124176.0	8.762749e+00	7.195326e+00	-1.294000e+01	3.160000e+00	9.560000e+00	1.413000e+01	3.729000e+01
temp_min	124176.0	9.365455e+00	5.912856e+00	-1.375000e+01	5.130000e+00	9.230000e+00	1.354000e+01	3.323000e+01
temp_max	124176.0	1.146047e+01	6.006574e+00	-8.410000e+00	7.210000e+00	1.125000e+01	1.552000e+01	3.743000e+01
pressure	124176.0	1.014386e+03	1.138366e+01	9.590000e+02	1.008000e+03	1.015000e+03	1.022000e+03	1.049000e+03
humidity	124176.0	8.200810e+01	1.260884e+01	2.000000e+00	7.400000e+01	8.500000e+01	9.200000e+01	1.000000e+02
wind_speed	124176.0	4.431663e+00	2.150646e+00	3.000000e-02	2.790000e+00	4.150000e+00	5.740000e+00	1.597000e+01
wind_deg	124176.0	2.001849e+02	9.208754e+01	0.000000e+00	1.400000e+02	2.200000e+02	2.660000e+02	3.600000e+02
clouds_all	124176.0	7.033787e+01	3.450556e+01	0.000000e+00	4.300000e+01	8.800000e+01	1.000000e+02	1.000000e+02
weather_id	124176.0	7.557358e+02	1.093496e+02	5.000000e+02	8.000000e+02	8.030000e+02	8.040000e+02	8.040000e+02

Figure 32: Null Values removed

### Encode Categorical features

There are still some features to be discussed which are weather id, main, description and icon. These were found to be categorical data and so will be encoded. One of the features (weather\_id) had no sensible data that could be useful and was dropped from the table.

weather_main	weather_description	weather_icon	weather_main	weather_description	weather_icon
			1	7	7
Clouds	overcast clouds	04n	1	7	7
			1	7	7
Rain	light rain	10n	1	7	7
			1	7	7
Rain	light rain	10n	...	...	...
			1	7	7
Rain	light rain	10n	2	4	9
			2	4	9
Rain	light rain	10n	2	4	9
			2	4	9

Figure 33: Encoded categorical features

### Remove unnecessary features

This is the final stage of the weather dataset preprocess, and we looked at the features that bore no significance to the data. By using the drop function, we selected the following features:

- Latitude and Longitude: these are just coordinates and do not serve well as regressors since they cannot be categorically encoded.
- dt: this is the time of data calculation which is a duplicate of the dt\_iso column that serves as the index.
- timezone: these are just shifts in seconds so there is no significance
- city\_name: this is a weather dataset of Leicester.
- clouds\_all: shows the percentage of cloudiness
- wind\_deg: wind direction does not affect building energy consumption, so it was dropped.

Our final dataset before proceeding to the last step is now:

	temp	dew_point	feels_like	temp_min	temp_max	pressure	humidity	wind_speed	weather_main	weather_description	weather_icon
dt_iso											
2008-01-01 00:00:00	7.38	6.79	5.53	6.04	8.21	1026	96	2.76	1	7	7
2008-01-01 01:00:00	6.49	6.05	4.51	6.04	7.13	1026	97	2.70	1	7	7
2008-01-01 02:00:00	6.45	6.01	4.54	5.94	7.06	1025	97	2.60	1	7	7
2008-01-01 03:00:00	6.74	6.45	5.05	5.04	7.25	1025	98	2.39	1	7	7
2008-01-01 04:00:00	6.64	6.35	5.08	5.59	7.21	1024	98	2.22	1	7	7
...	...	...	...	...	...	...	...	...	...	...	...
2022-03-01 19:00:00	6.67	4.83	3.32	5.75	7.23	1027	88	5.24	1	7	7
2022-03-01 20:00:00	6.30	4.62	2.91	5.75	7.09	1027	89	5.12	2	4	9
2022-03-01 21:00:00	6.10	4.43	2.52	5.16	7.09	1027	89	5.43	2	4	9
2022-03-01 22:00:00	5.87	4.36	2.59	5.12	6.62	1026	90	4.65	2	4	9
2022-03-01 23:00:00	5.63	4.43	2.03	5.04	6.23	1026	92	5.21	2	4	9

Figure 34: Unnecessary Values removed

## Visualization

We now must calculate the correlation between the features and plot it as a heatmap to show the feature's importance. From the figure below we can see that there is a negative correlation between the humidity and temperature features. The features hardly intersect as shown below. We can see a high correlation between the dew point and temperature features. As the temperature rises, so does the dewpoint, making it similar

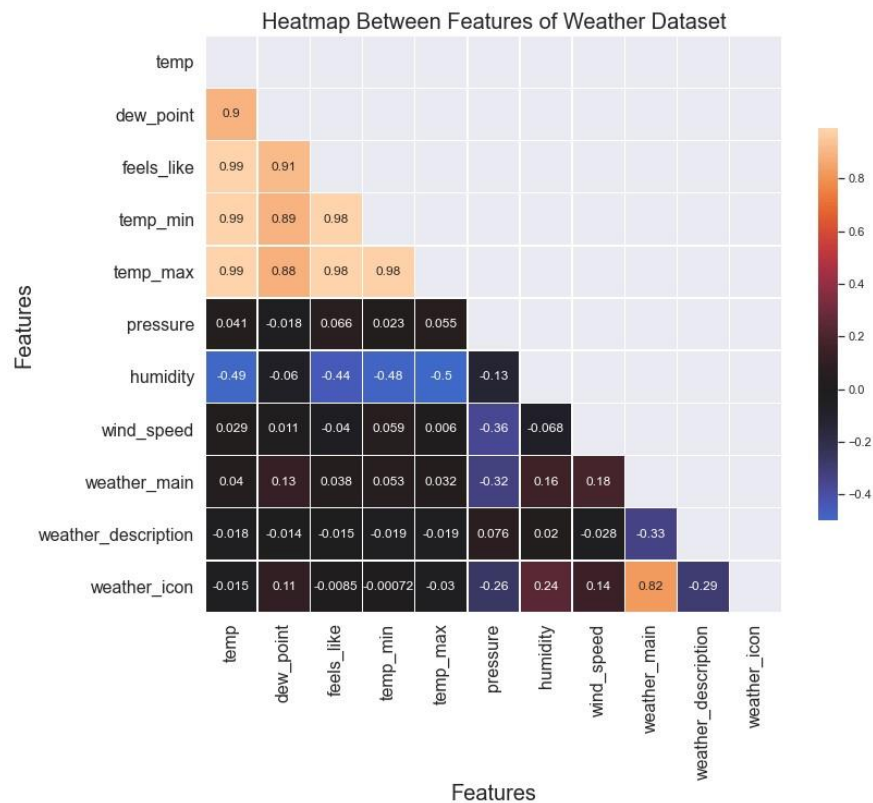
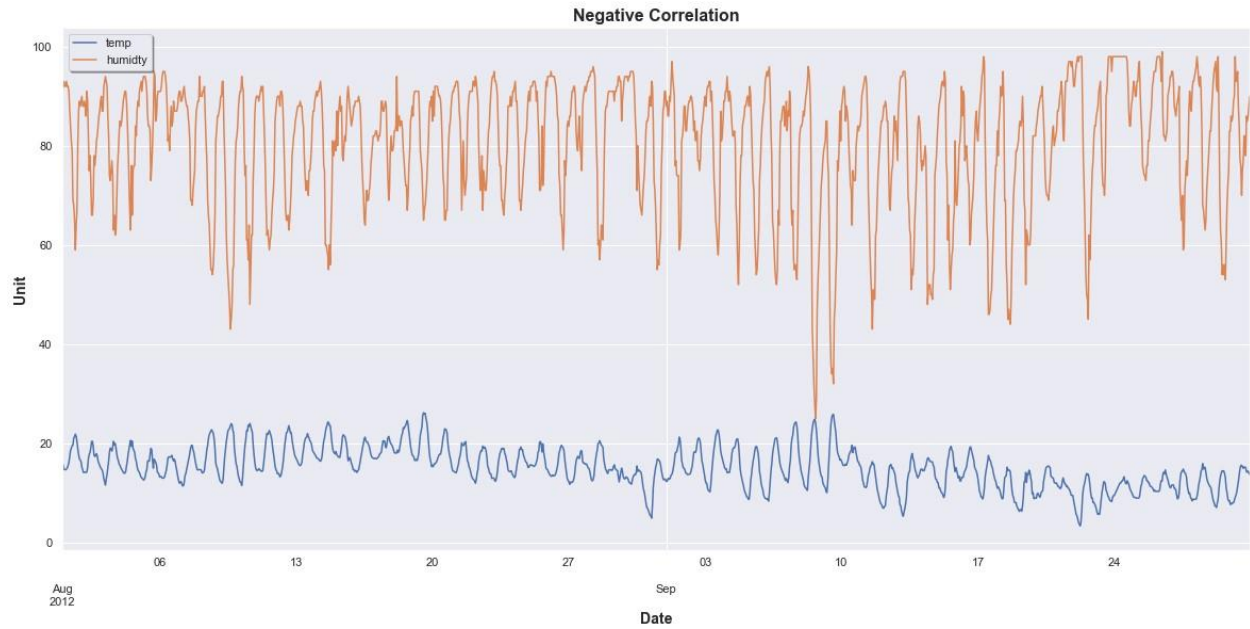
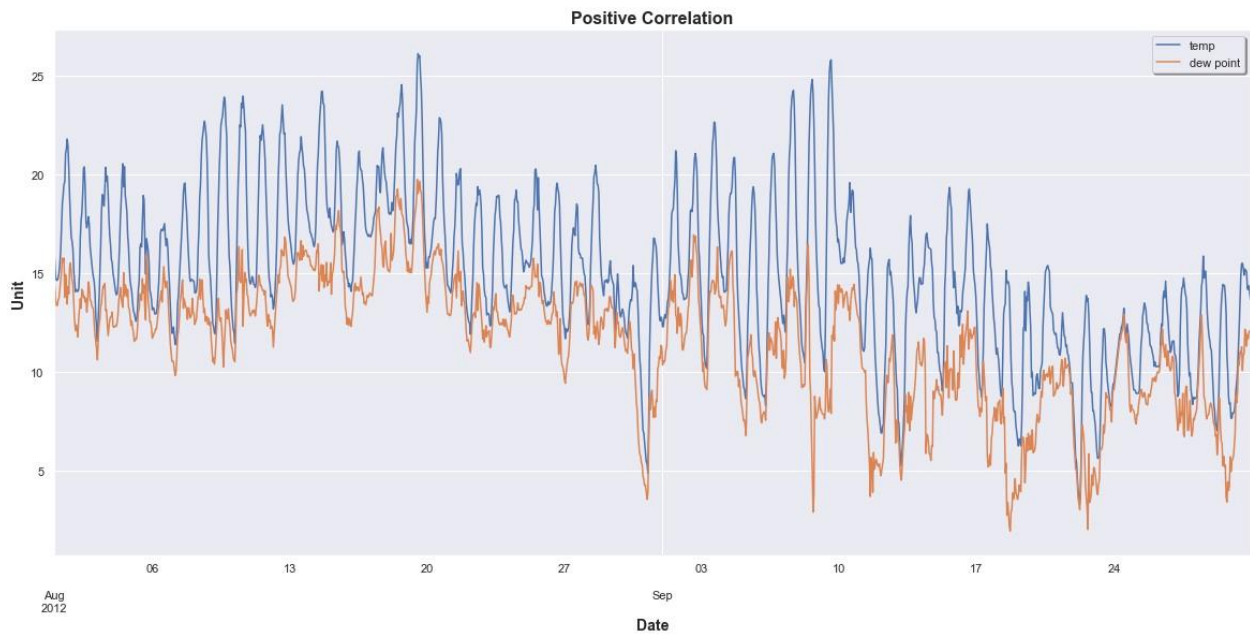


Figure 35: Heatmap between features in Weather Dataset





**Figure 36: Example showing the negative correlation between Temperature and Humidity**



**Figure 37: Positive correlation between Temperature and Dewpoint**

Below shows the final dataframe with the features and respective attributes. The count shows the total length of the dataset, mean, maximum value and the rest of the variables are also given. As shown the values are all in different ranges which would make training less accurate and difficult.

	count	mean	std	min	25%	50%	75%	max
temp	124176.0	10.400765	5.906019	-10.72	6.14	10.23	14.47	34.88
dew_point	124176.0	7.248109	4.999126	-35.23	3.55	7.48	11.04	22.21
feels_like	124176.0	8.762749	7.195326	-12.94	3.16	9.56	14.13	37.29
temp_min	124176.0	9.365455	5.912856	-13.75	5.13	9.23	13.54	33.23
temp_max	124176.0	11.460467	6.006574	-8.41	7.21	11.25	15.52	37.43
pressure	124176.0	1014.386307	11.383661	959.00	1008.00	1015.00	1022.00	1049.00
humidity	124176.0	82.008101	12.608843	2.00	74.00	85.00	92.00	100.00
wind_speed	124176.0	4.431663	2.150646	0.03	2.79	4.15	5.74	15.97
weather_main	124176.0	1.060189	0.515905	0.00	1.00	1.00	1.00	3.00
weather_description	124176.0	5.409218	3.054708	0.00	4.00	7.00	7.00	11.00
weather_icon	124176.0	5.732179	2.278740	0.00	5.00	6.00	7.00	11.00

Figure 38: Dataframe before feature selection

### Feature Selection

For this project, the temp, dew\_point, pressure, humidity, and wind\_speed was chosen as the features that would be used in training and evaluating the algorithms. The decision on these features as they have the most relevance when putting energy consumption into consideration. For example, during cold days electricity usage is expected to reduce and vice versa.

### Normalization

Feature scaling is the process of converting the values of different numeric objects to be within similar ranges. Scaling is used to prevent overfitting and biased results of supervised learning models. For example, if a model uses linear regression and the features are not scaled, some features end up having a higher impact than others, affecting the predictive performance. Therefore, it is important to scale your features. The only algorithms that do not need scaling are random forests and Decision trees since they are scaling invariant.

What is normalization? This refers to scaling features into specified ranges [0,1] in the case of a min-max scaler. Normalization is important when the data is needed in bounded intervals. Below is a formula for normalizing based on the min-max scaler:

$$x_{norm}^{(i)} = \frac{x^{(i)} - x_{min}}{x_{max} - x_{min}}$$

Where:

$x^{(i)}$  – represents the values

$x_{min}$  – minimum values

$x_{max}$  – maximum values

## Min-Max Scaler

Min-Max Scaler is a normalisation class from the sklearn.preprocessing package. When scaling the training and test data sets, the Min-Max Scaler estimate will match the training data set, and the same estimate would be used to convert both training and test data sets.

	count	mean	std	min	25%	50%	75%	max
temp	124176.0	0.463175	0.129518	0.0	0.369737	0.459430	0.552412	1.0
dew_point	124176.0	0.739521	0.087032	0.0	0.675139	0.743558	0.805536	1.0
feels_like	124176.0	0.432067	0.143248	0.0	0.320526	0.447939	0.538921	1.0
temp_min	124176.0	0.492028	0.125859	0.0	0.401873	0.489144	0.580885	1.0
temp_max	124176.0	0.433474	0.131033	0.0	0.340750	0.428883	0.522033	1.0
pressure	124176.0	0.615403	0.126485	0.0	0.544444	0.622222	0.700000	1.0
humidity	124176.0	0.816409	0.128662	0.0	0.734694	0.846939	0.918367	1.0
wind_speed	124176.0	0.276139	0.134921	0.0	0.173149	0.258469	0.358218	1.0
weather_main	124176.0	0.353396	0.171968	0.0	0.333333	0.333333	0.333333	1.0
weather_description	124176.0	0.491747	0.277701	0.0	0.363636	0.636364	0.636364	1.0
weather_icon	124176.0	0.521107	0.207158	0.0	0.454545	0.545455	0.636364	1.0

Figure 39: Features scaled using Min-Max

## Standard Scaler

Unlike the min-max scaler that normalises the values around a certain range, the feature columns are centred at a mean of zero with a standard deviation of one using the standardisation approach, giving them the same properties as a typical normal distribution. It keeps relevant information about outliers and makes the algorithm less sensitive to them. The formula for standardization is:

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

Where:

$x^{(i)}$  – represents the values

$\mu_x$  – mean of values

$\sigma_x$  – standard deviation of values

	count	mean	std	min	25%	50%	75%	max
temp	124176.0	-2.380378e-17	1.000004	-3.576157	-0.721430	-0.028914	0.689001	4.144811
dew_point	124176.0	-1.345829e-16	1.000004	-8.497141	-0.739754	0.046387	0.758514	2.992913
feels_like	124176.0	-1.171878e-16	1.000004	-3.016241	-0.778668	0.110802	0.745939	3.964708
temp_min	124176.0	4.394543e-17	1.000004	-3.909371	-0.716316	-0.022909	0.706015	4.036060
temp_max	124176.0	-2.188116e-16	1.000004	-3.308133	-0.707639	-0.035040	0.675851	4.323536
pressure	124176.0	1.657109e-15	1.000004	-4.865441	-0.561009	0.053910	0.668829	3.040659
humidity	124176.0	4.669202e-17	1.000004	-6.345421	-0.635120	0.237287	0.792455	1.426933
wind_speed	124176.0	8.239769e-18	1.000004	-2.046679	-0.763338	-0.130967	0.608348	5.365078
weather_main	124176.0	3.021249e-17	1.000004	-2.055017	-0.116667	-0.116667	-0.116667	3.760032
weather_description	124176.0	2.706535e-17	1.000004	-1.770788	-0.461328	0.520766	0.520766	1.830226
weather_icon	124176.0	1.280597e-16	1.000004	-2.515514	-0.321310	0.117531	0.556372	2.311735

**Figure 40: Features scaled using Standard Scaler**

# Experimentation

## Hardware

The tests were conducted on a PC with a Windows OS. The computer's features are described in the following:

- GPU: Intel Nvidia RTX 2070
- CPU: Intel Core i7 8<sup>th</sup> Generation
- RAM: 16GB

## Software

Python is the programming language utilised. The experiment is conducted in a virtual environment generated with Visual Studio Code and Python 3.9.0. The relevant packages are used in the experiment. A Jupyter Notebook was used to write and run the code. The network was trained using the Nvidia RTX 2070 GPU, which has 16 GB VRAM and CUDA version 11.6.

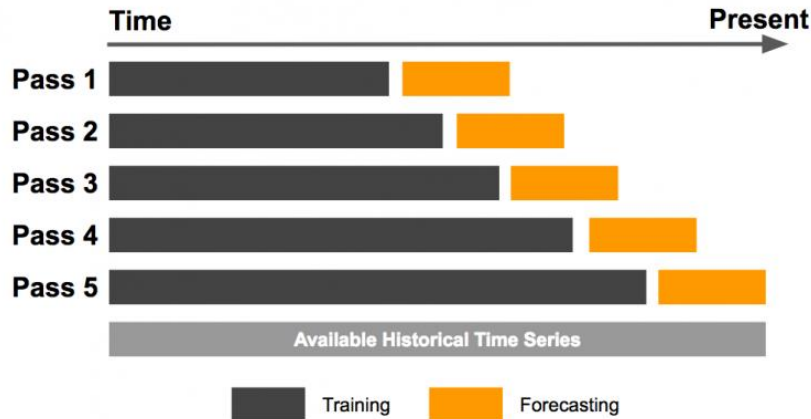
- TensorFlow
- Keras
- Scikit-learn
- Pandas
- Prophet
- Matplotlib
- NumPy
- Seaborn

## Settings

The challenge is to create a model using the S2S LSTM as well as BLST technique for correctly predicting a structure's energy usage using Machine Learning, ensemble, and even the Facebook Prophet methodologies. Predictions of energy use across various periods and train/test splits are among the experiments. This time-series regression issue, if successful, can be used as proof for employing a learning algorithm in energy consumption prediction by applying past data of energy (Electricity, Water, and Gas) usage and weather conditions.

## Training

A typical cross-validation method, in which random points are used in each fold, cannot be performed in this situation due to the time constraint. The time series must be preserved; thus, the folds do not include random selection; rather, every fold increases the volume of training data. Each fold has its own test set, and the preceding fold's test set is appended to the current fold's training set for the next fold. An example of the expanding window is below:



**Figure 41: Expanding window**

## Experiments and Results

This chapter will go about performing ‘black-box’ testing to evaluate or created models and afterwards a visualization of the models will be shown ranking by the model with the best performance. All the models will be assessed against a baseline prediction calculated using the simple persistence algorithm. The Zero Rule Algorithm is by far the most frequent baseline method for monitoring ML. In the instance of classifications, the algorithm correctly forecasts most classes. With regression cases, it predicts the expected average. This can be utilized with time-series data, but the sequence correlation architecture in the data set is not considered. The persistence algorithm is a similar technique for time series data sets. The persistence method predicts the expected outcome for the next step ( $t + 1$ ) based on the last step ( $t - 1$ ). The following baseline forecast conditions must be fulfilled:

- **Simple** : A technique that does not need extensive training or understanding.
- **Fast** : Predicting in a quick and computationally simple manner.
- **Repeatable** : A deterministic technique gives the expected output when given the same input.

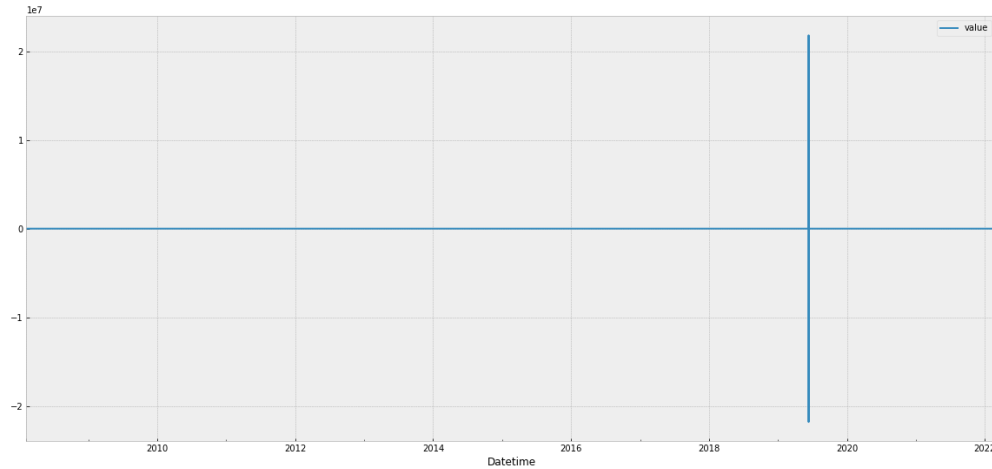
Queen’s Building

Baseline

**Table 4: Queens Building Hourly Baseline**

	Electric	Gas	Water
RMSE	4.265	152110.65	0.057
MSE	18.190	23137650036.14	0.003
MAE	2.4481	726.80	0.034
R2	0.944	-2.000	0.895

From the table above we can see the gas dataset metrics could not accurately be measured which means a further investigation of the plot is needed.



**Figure 42: The plot of Queen's Building Gas Hourly Dataset**

Based on the figure above we can see that the hourly dataset could not be plotted due to the fact we are dealing with data lower than the program can compute. The electric and water datasets produced exceptionally good metrics that could help train the desired algorithm, but with the gas dataset producing such results, it could degrade the performance.

**Table 5: Queens Building Daily Baseline**

	Electric	Gas	Water
RMSE	8.217	53.023	0.071
MSE	67.521	2811.472	0.005
MAE	4.996	26.286	0.043
R2	0.644	0.540	0.744

The daily baseline shows exceptionally satisfactory results and a perfect univariate baseline to try to beat for the chosen algorithms. These are all baseline metrics from a univariate model in hopes that adding regressors would improve it.

LSTM

**Table 6: Queen's Building LSTM Hourly Prediction**

Layers	Neurons	Epochs	Lag Variable	Metrics	Beaten?
2	50	50	6 hours	RMSE: 4.089 MSE: 16.723 MAE: 2.639 $R^2$ : 0.944	Yes
	50	200	6 hours	RMSE: 4.180 MSE: 17.473 MAE: 2.595 $R^2$ : 0.942	Yes
	150	50	6 hours	RMSE: 5.066 MSE: 25.664 MAE: 3.940 $R^2$ : 0.914	No
6	50	50	6 hours	RMSE: 9.510	No

				MSE: 90.433 MAE: 7.489 $R^2$ : 0.698	
8	32	50	6 hours	RMSE: 11.859 MSE: 140.642 MAE: 9.564 $R^2$ : 0.531	No

We can observe from the table above that if the LSTM model contains two layers, it improves the baseline RMSE. Another interesting fact is that the number of neurons is capped at 50; much greater and the RMSE begins to exceed the baseline. Due to time constraints, the remaining tests will be evaluated using the Daily dataset.

**Table 7: Queen's Building LSTM Daily Predictions**

Layers	Neurons	Epochs	Training	Metrics	Beaten?
2	50	50	4 years	RMSE: 6.864 MSE: 47.117 MAE: 5.471 $R^2$ : 0.662	Yes
	50	500	4 years	RMSE: 6.039 MSE: 36.465 MAE: 4.404 $R^2$ : 0.738	Yes
	150	500	5 years	RMSE: 5.619 MSE: 31.576 MAE: 4.162 $R^2$ : 0.772	Yes
	300	1000	5 years	RMSE: 6.707 MSE: 44.980 MAE: 4.972 $R^2$ : 0.675	Yes
	1000	500	5 years	RMSE: 5.546 MSE: 30.761 MAE: 4.247 $R^2$ : 0.778	Yes
6	50	50	5 years	RMSE: 10.842 MSE: 117.539 MAE: 9.111 $R^2$ : 0.150	No
	50	500	5 years	RMSE: 5.923 MSE: 35.081 MAE: 4.442 $R^2$ : 0.746	Yes
	150	500	5 years	RMSE: 7.003 MSE: 49.041 MAE: 5.424 $R^2$ : 0.645	Yes
	300	1000	5 years	RMSE: 6.623	Yes



				MSE: 43.861 MAE: 5.032 $R^2$ : 0.683	
8	50	500	5 years	RMSE: 7.200 MSE: 51.838 MAE: 5.470 $R^2$ : 0.625	Yes

Based on the extensive tests conducted, we can see that the LSTM performs very well and improves the RMSE. The LSTM with two layers and one thousand input neurons had the better metric. From this, we can assume that the BLSTM would also show related results or even better.

BLSTM

**Table 8: Queen's Building BLSTM Predictions**

Layers	Neurons	Epochs	Train Variable	Metrics	Beaten?
2	50	50	5 years	RMSE: 5.480 MSE: 30.025 MAE: 4.173 $R^2$ : 0.783	Yes
	50	200	5 years	RMSE: 4.817 MSE: 23.206 MAE: 3.508 $R^2$ : 0.832	Yes
	200	200	5 years	RMSE: 4.807 MSE: 23.104 MAE: 3.591 $R^2$ : 0.833	Yes
6	50	50	5 years	RMSE: 7.200 MSE: 51.839 MAE: 5.744 $R^2$ : 0.625	Yes
	150	500	5 years	RMSE: 6.624 MSE: 43.883 MAE: 5.012 $R^2$ : 0.683	Yes
8	50	200	5 years	RMSE: 11.280 MSE: 127.247 MAE: 9.271 $R^2$ : 0.080	No

Based on the extensive tests conducted, we can see that the LSTM performs very well and improves the RMSE. The BLSTM with two layers and two hundred input neurons had the better metric. This proves the statement made in the literature that for a bidirectional LSTM, only two layers are needed for it to work effectively. Therefore, the rest of the test would only be conducted using only two layers of both the LSTM and BLSTM.

XGBoost

**Table 9: Queen's Building XGBoost Predictions**

Iterations	Learning Rate	Metrics	Beaten?
50	0.001	RMSE: 51.521 MSE: 2654.385 MAE: 49.731 $R^2$ : -12.798	No
	0.01	RMSE: 33.547 MSE: 1125.394 MAE: 31.791 $R^2$ : -4.850	No
	0.1	RMSE: 7.542 MSE: 56.885 MAE: 5.843 $R^2$ : 0.704	Yes
200	0.001	RMSE: 44.606 MSE: 1989.701 MAE: 42.860 $R^2$ : -9.343	No
	0.1	RMSE: 7.575 MSE: 57.374 MAE: 5.791 $R^2$ : 0.702	Yes
500	0.001	RMSE: 33.618 MSE: 1130.203 MAE: 31.863 $R^2$ : -4.875	No
	0.1	RMSE: 7.701 MSE: 59.306 MAE: 5.851 $R^2$ : 0.692	Yes
2000	0.001	RMSE: 10.989 MSE: 120.751 MAE: 9.065 $R^2$ : 0.372	No
	0.01	RMSE: 7.552 MSE: 57.028 MAE: 5.772 $R^2$ : 0.704	Yes
5000	0.001	RMSE: 7.607 MSE: 57.871 MAE: 5.916 $R^2$ : 0.699	Yes
	0.1	RMSE: 7.819 MSE: 61.134 MAE: 5.931	Yes

		$R^2$ : 0.682	
25000	0.001	RMSE: 7.528 MSE: 56.678 MAE: 5.757 $R^2$ : 0.705	Yes
	0.1	RMSE: 7.819 MSE: 61.134 MAE: 5.931 $R^2$ : 0.682	Yes

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50, 200 and 500, the best learning rate is 0.1 but in the later iterations especially 25,000, it was discovered that the learning was instead 0.001 producing the lowest RMSE.

LightGBM

**Table 10: Queen's Building LightGBM Predictions**

Iterations	Learning Rate	Metrics	Beaten?
50	0.001	RMSE: 13.443 MSE: 180.715 MAE: 10.954 $R^2$ : 0.061	No
	0.01	RMSE: 10.676 MSE: 113.977 MAE: 8.554 $R^2$ : 0.408	No
	0.1	RMSE: 7.576 MSE: 57.400 MAE: 5.907 $R^2$ : 0.702	Yes
200	0.001	RMSE: 12.322 MSE: 151.829 MAE: 9.977 $R^2$ : 0.211	No
	0.1	RMSE: 7.637 MSE: 58.325 MAE: 5.885 $R^2$ : 0.697	Yes
500	0.001	RMSE: 10.689 MSE: 114.254 MAE: 8.565 $R^2$ : 0.406	No
	0.1	RMSE: 7.848 MSE: 61.598 MAE: 6.050 $R^2$ : 0.680	Yes
2000	0.001	RMSE: 8.116 MSE: 65.873 MAE: 6.374	Yes

		$R^2$ : 0.658	
	0.01	RMSE: 7.647 MSE: 58.477 MAE: 5.886 $R^2$ : 0.696	Yes
5000	0.001	RMSE: 7.554 MSE: 57.061 MAE: 5.861 $R^2$ : 0.703	Yes
	0.1	RMSE: 8.047 MSE: 64.761 MAE: 6.202 $R^2$ : 0.663	Yes
25000	0.001	RMSE: 7.617 MSE: 58.014 MAE: 5.833 $R^2$ : 0.698	Yes
	0.1	RMSE: 8.048 MSE: 64.763 MAE: 6.202 $R^2$ : 0.663	Yes

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50, 200 and 500, the best learning rate is 0.1 but in the later iterations especially 5,000, it was discovered that the learning was instead 0.001 producing the lowest RMSE.

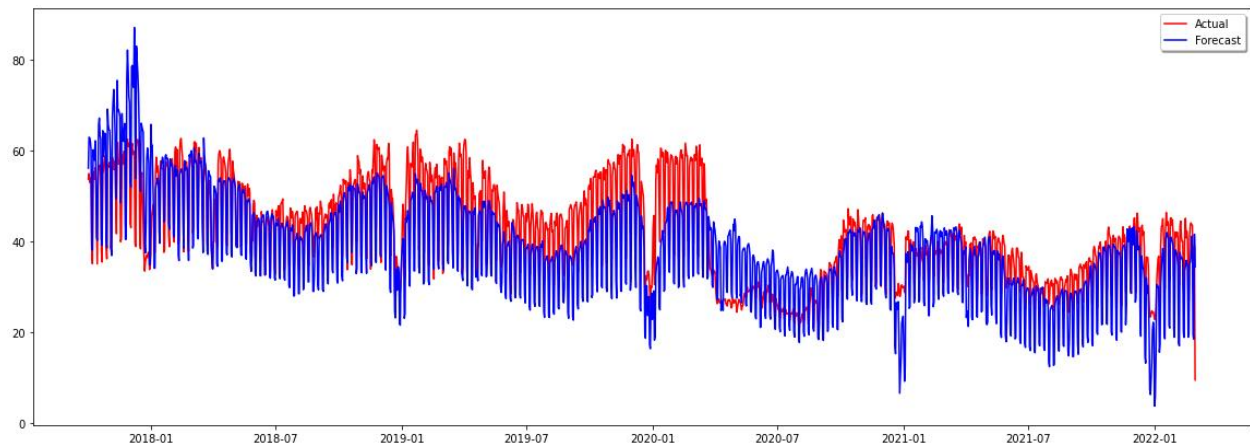
Prophet

**Table 11: Queen's Building Prophet Predictions**

Trainable Days	Parameter Added	Metrics	Beaten?
4700	Interval width: 0.9	RMSE: 4.900 MSE: 24.011 MAE: 3.811 $R^2$ : 0.330	Yes
	Interval width: 0.9 UK holidays	RMSE: 5.136 MSE: 26.382 MAE: 3.930 $R^2$ : 0.264	Yes
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 5.117 MSE: 26.188 MAE: 3.935 $R^2$ : 0.269	Yes
4000	Interval width: 0.9	RMSE: 7.209 MSE: 51.964 MAE: 5.597 $R^2$ : 0.514	Yes
	Interval width: 0.9 UK holidays	RMSE: 7.263 MSE: 52.753 MAE: 5.639	Yes

		$R^2$ : 0.507	
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 7.268 MSE: 58.829 MAE: 5.650 $R^2$ : 0.506	Yes
3500	Interval width: 0.9	RMSE: 7.329 MSE: 53.179 MAE: 5.928 $R^2$ : 0.547	Yes
	Interval width: 0.9 UK holidays	RMSE: 7.295 MSE: 53.211 MAE: 5.907 $R^2$ : 0.551	Yes
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 7.267 MSE: 52.814 MAE: 5.880 $R^2$ : 0.555	Yes
3000	Interval width: 0.9	RMSE: 21.462 MSE: 460.623 MAE: 18.605 $R^2$ : -2.748	No

We can see from the extensive test that the Prophet algorithm performed extremely well in terms of RMSE and shows that it could be used for energy forecasting. Training for 3500 days (10 years) yielded lower RMSE but an  $R^2$  score of 55% (Fig. 43).



**Figure 43: Queen's Building Forecast using Prophet**

Hugh Aston

Baseline

**Table 12: Hugh Aston Hourly Baseline**

	Electric	Gas	Water
RMSE	7.713	27.48	0.131
MSE	59.483	755.156	0.017
MAE	4.646	12.024	0.083

R2	0.917	0.827	0.853
----	-------	-------	-------

**Table 13: Hugh Aston Daily Baseline**

	Electric	Gas	Water
RMSE	13.573	25.191	0.172
MSE	184.23	634.58	0.03
MAE	8.354	14.7	0.115
R2	0.34	0.803	0.36

LSTM

**Table 14: Hugh Aston LSTM Predictions**

Layers	Neurons	Epochs	Training	Metrics	Beaten?
2	50	50	4 years	RMSE: 8.774 MSE: 76.976 MAE: 6.613 $R^2$ : 0.731	Yes
	50	500	4 years	RMSE: 8.408 MSE: 70.702 MAE: 5.981 $R^2$ : 0.752	Yes
	150	500	4 years	RMSE: 8.739 MSE: 76.368 MAE: 6.174 $R^2$ : 0.739	Yes
	300	1000	4 years	RMSE: 8.132 MSE: 66.137 MAE: 6.015 $R^2$ : 0.774	Yes
	1000	500	4 years	RMSE: 7.862 MSE: 61.816 MAE: 5.953 $R^2$ : 0.788	Yes
6	50	50	4 years	RMSE: 16.893 MSE: 285.381 MAE: 15.113 $R^2$ : 0.023	No
	50	500	4 years	RMSE: 11.091 MSE: 123.015 MAE: 7.805 $R^2$ : 0.579	Yes
	150	500	4 years	RMSE: 9.345 MSE: 87.331 MAE: 6.707 $R^2$ : 0.701	Yes

	300	1000	4 years	RMSE: 10.099 MSE: 101.982 MAE: 7.738 $R^2$ : 0.651	Yes
8	50	500	4 years	RMSE: 9.602 MSE: 92.192 MAE: 6.738 $R^2$ : 0.684	Yes

Based on the test conducted, this building's LSTM statistics show that the 2-layered structure provided the best RMSE results. Unlike the Queen's Building LSTM results, it only had a single instance of the network not outperforming the baseline results. On the 2-layered structure, the higher the input neurons the better the results with a computation time trade-off.

BLSTM

**Table 15: Hugh Aston BLSTM Predictions**

Layers	Neurons	Epochs	Training	Metrics	Beaten?
2	50	50	4 years	RMSE: 7.873 MSE: 61.977 MAE: 5.692 $R^2$ : 0.783	Yes
	50	500	4 years	RMSE: 7.582 MSE: 57.484 MAE: 5.581 $R^2$ : 0.799	Yes
	150	500	4 years	RMSE: 7.413 MSE: 54.955 MAE: 5.354 $R^2$ : 0.808	Yes
	300	1000	4 years	RMSE: 8.783 MSE: 77.142 MAE: 6.449 $R^2$ : 0.730	Yes
	1000	500	4 years	RMSE: 8.268 MSE: 68.360 MAE: 5.910 $R^2$ : 0.761	Yes
6	50	50	4 years	RMSE: 12.031 MSE: 144.750 MAE: 9.379 $R^2$ : 0.505	Yes
	50	500	4 years	RMSE: 11.579 MSE: 134.073 MAE: 8.638 $R^2$ : 0.541	Yes
	150	500	4 years	RMSE: 8.535 MSE: 72.843 MAE: 5.893	Yes

				$R^2$ : 0.751	
	300	1000	4 years	RMSE: 9.362 MSE: 87.643 MAE: 6.998 $R^2$ : 0.700	Yes
8	50	500	4 years	RMSE: 9.738 MSE: 94.829 MAE: 7.050 $R^2$ : 0.668	Yes

Based on the test conducted, this building's BLSTM statistics show that the 2-layered structure provided the best RMSE results. On the 2-layered structure, the input neuron is capped at 150 as any higher the RMSE starts to trail off.

XGBoost

**Table 16: Hugh Aston XGBoost Predictions**

Iterations	Learning Rate	Metrics	Beaten?
50	0.001	RMSE: 62.211 MSE: 3870.216 MAE: 60.174 $R^2$ : -13.611	No
	0.01	RMSE: 40.385 MSE: 1630.946 MAE: 38.390 $R^2$ : -5.127	No
	0.1	RMSE: 9.326 MSE: 86.973 MAE: 7.073 $R^2$ : 0.672	Yes
200	0.001	RMSE: 53.812 MSE: 2895.756 MAE: 51.827 $R^2$ : -9.932	No
	0.1	RMSE: 9.361 MSE: 87.622 MAE: 7.093 $R^2$ : 0.669	Yes
500	0.001	RMSE: 40.477 MSE: 1638.348 MAE: 38.400 $R^2$ : -5.185	No
	0.1	RMSE: 9.518 MSE: 90.600 MAE: 7.214 $R^2$ : 0.658	Yes
2000	0.001	RMSE: 13.015 MSE: 169.388 MAE: 10.837	Yes



		$R^2$ : 0.361	
	0.1	RMSE: 9.617 MSE: 92.492 MAE: 7.309 $R^2$ : 0.651	Yes
5000	0.001	RMSE: 9.270 MSE: 85.926 MAE: 7.063 $R^2$ : 0.676	Yes
	0.1	RMSE: 9.621 MSE: 92.554 MAE: 7.312 $R^2$ : 0.651	Yes
25000	0.001	RMSE: 9.270 MSE: 85.938 MAE: 7.034 $R^2$ : 0.676	Yes
	0.1	RMSE: 9.621 MSE: 92.554 MAE: 7.312 $R^2$ : 0.651	Yes

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50, 200 and 500, the best learning rate is 0.1 but in the later iterations especially 25,000, it was discovered that the learning was instead 0.001 producing the lowest RMSE.

LightGBM

**Table 17: Hugh Aston LightGBM Predictions**

Iterations	Learning Rate	Metrics	Beaten?
50	0.001	RMSE: 15.796 MSE: 249.509 MAE: 13.255 $R^2$ : 0.06	No
	0.01	RMSE: 12.537 MSE: 157.167 MAE: 10.338 $R^2$ : 0.407	Yes
	0.1	RMSE: 9.268 MSE: 85.891 MAE: 7.039 $R^2$ : 0.676	Yes
200	0.001	RMSE: 14.475 MSE: 209.528 MAE: 12.059 $R^2$ : 0.209	No
	0.1	RMSE: 9.283 MSE: 86.177 MAE: 7.034	Yes

		$R^2$ : 0.675	
500	0.001	RMSE: 12.548 MSE: 157.457 MAE: 10.349 $R^2$ : 0.406	Yes
	0.1	RMSE: 9.573 MSE: 91.635 MAE: 7.253 $R^2$ : 0.654	Yes
2000	0.001	RMSE: 9.639 MSE: 92.903 MAE: 7.519 $R^2$ : 0.649	Yes
	0.1	RMSE: 9.779 MSE: 95.625 MAE: 7.411 $R^2$ : 0.639	Yes
5000	0.001	RMSE: 9.227 MSE: 85.147 MAE: 7.021 $R^2$ : 0.679	Yes
	0.1	RMSE: 9.791 MSE: 95.868 MAE: 7.420 $R^2$ : 0.638	Yes
25000	0.001	RMSE: 9.298 MSE: 86.458 MAE: 7.079 $R^2$ : 0.674	Yes
	0.1	RMSE: 9.791 MSE: 95.869 MAE: 7.420 $R^2$ : 0.638	Yes

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50 and 200, the best learning rate is 0.1 but in the later iterations especially 5,000, it was discovered that the learning was instead 0.001 producing the lowest RMSE.

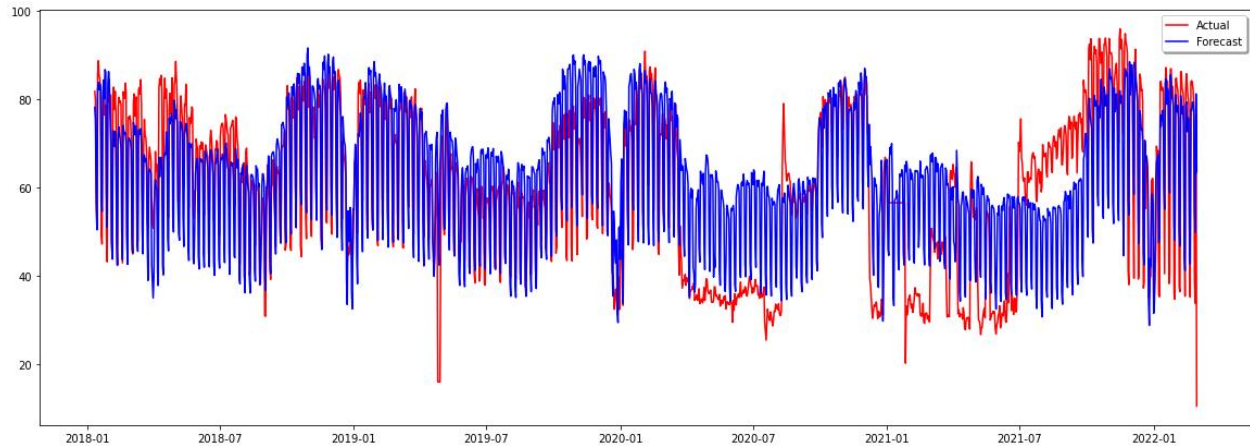
Prophet

**Table 18: Hugh Aston Prophet Predictions**

Trainable Days	Parameter Added	Metrics	Beaten?
3500	Interval width: 0.9	RMSE: 17.288 MSE: 298.868 MAE: 14.134 $R^2$ : 0.220	No
	Interval width: 0.9 UK holidays	RMSE: 17.005 MSE: 289.160 MAE: 13.950	No

		$R^2$ : 0.245	
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 17.010 MSE: 289.344 MAE: 13.972 $R^2$ : 0.245	No
2900	Interval width: 0.9	RMSE: 12.882 MSE: 165.940 MAE: 9.857 $R^2$ : 0.486	Yes
	Interval width: 0.9 UK holidays	RMSE: 12.859 MSE: 165.365 MAE: 9.663 $R^2$ : 0.488	Yes
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 12.912 MSE: 166.719 MAE: 9.683 $R^2$ : 0.484	Yes
2100	Interval width: 0.9	RMSE: 16.271 MSE: 264.751 MAE: 11.897 $R^2$ : 0.119	No
	Interval width: 0.9 UK holidays	RMSE: 16.284 MSE: 265.174 MAE: 11.734 $R^2$ : 0.118	No
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 16.524 MSE: 273.052 MAE: 11.907 $R^2$ : 0.091	No
2200	Interval width: 0.9	RMSE: 16.222 MSE: 263.169 MAE: 11.944 $R^2$ : 0.122	No

We can see from the extensive test that the Prophet algorithm did not perform as expected and could only surpass the baseline by a hair. The number of trainable days that were successful is 2900 days and the RMSE achieved was through the addition of the in-built Prophet function of country holidays as regressors. (See diagram below). This shows the forecast in blue and the actual dataset in red; the algorithm seemed to struggle from April 2020 due to the country going into sudden lockdown, but Prophet shows to an extent what the normal consumption rate would have been.



**Figure 44: Hugh Aston forecast using Prophet**

Gateway House

Baseline

**Table 19: Gateway House Hourly Baseline**

	Electric	Gas	Water
RMSE	16753.53	34.104	0.134
MSE	280680631.65	1163.092	0.018
MAE	83.353	10.998	0.047
R2	-1.999	0.702	0.741

**Table 20: Gateway House Daily Baseline**

	Electric	Gas	Water
RMSE	16.602	31.317	0.124
MSE	275.639	980.762	0.015
MAE	9.609	14.89	0.072
R2	0.54	0.411	0.271

LSTM

**Table 21: Gateway House LSTM Predictions**

Layers	Neurons	Epochs	Training	Metrics	Beaten?
2	50	50	7 years	RMSE: 10.537 MSE: 111.028 MAE: 8.845 $R^2$ : 0.582	Yes
	50	500	7 years	RMSE: 12.892 MSE: 166.216 MAE: 10.957 $R^2$ : 0.375	Yes
	150	500	7 years	RMSE: 9.995 MSE: 99.891	Yes

				MAE: 8.245 $R^2$ : 0.624	
	300	1000	7 years	RMSE: 7.710 MSE: 59.437 MAE: 5.942 $R^2$ : 0.776	Yes
	1000	500	7 years	RMSE: 7.196 MSE: 51.777 MAE: 5.720 $R^2$ : 0.805	Yes
6	50	50	7 years	RMSE: 13.081 MSE: 171.117 MAE: 10.644 $R^2$ : 0.356	Yes
	50	500	7 years	RMSE: 13.872 MSE: 192.421 MAE: 11.322 $R^2$ : 0.276	Yes
	150	500	7 years	RMSE: 10.423 MSE: 108.649 MAE: 8.039 $R^2$ : 0.591	Yes
	300	1000	7 years	RMSE: 13.464 MSE: 181.281 MAE: 10.404 $R^2$ : 0.318	Yes
8	50	500	7 years	RMSE: 25.333 MSE: 641.757 MAE: 21.502 $R^2$ : -1.415	No

Based on the test conducted, this building's LSTM statistics show that the 2-layered structure once again provided the best RMSE results. On the 2-layered structure, it showed that the best result occurred with one thousand input neurons and five hundred epochs. Three hundred input neurons were also assessed and showed the second-best results. After 8 layers, the model fails to train and produces bad results. This suggests more fine-tuning is needed.

BLSTM

**Table 22: Gateway House BLSTM Predictions**

Layers	Neurons	Epochs	Training	Metrics	Beaten?
2	50	50	7 years	RMSE: 9.369 MSE: 87.778 MAE: 7.550 $R^2$ : 0.670	Yes
	50	500	7 years	RMSE: 9.395 MSE: 88.271 MAE: 7.769	Yes

				$R^2$ : 0.688	
	150	500	7 years	RMSE: 10.064 MSE: 101.280 MAE: 8.375 $R^2$ : 0.619	Yes
	300	1000	7 years	RMSE: 11.185 MSE: 125.098 MAE: 9.606 $R^2$ : 0.529	Yes
6	50	50	7 years	RMSE: 10.866 MSE: 118.065 MAE: 9.052 $R^2$ : 0.556	Yes
	50	500	7 years	RMSE: 14.330 MSE: 205.358 MAE: 12.438 $R^2$ : 0.227	Yes
8	50	50	7 years	RMSE: 14.745 MSE: 217.404 MAE: 12.782 $R^2$ : 0.182	Yes

Based on the test conducted, this building's BLSTM statistics show that the 2-layered structure provided the best RMSE results. On the 2-layered structure, the input neuron is capped at 50 as any higher the RMSE starts to increase till a sudden drop at 150 neurons.

XGBoost

**Table 23: Gateway House XGBoost Predictions**

Iterations	Learning Rate	Metrics	Beaten?
50	0.001	RMSE: 65.773 MSE: 4326.104 MAE: 61.483 $R^2$ : -6.415	No
	0.01	RMSE: 43.137 MSE: 1860.776 MAE: 39.142 $R^2$ : -2.190	No
	0.1	RMSE: 12.981 MSE: 168.502 MAE: 9.833 $R^2$ : 0.711	Yes
200	0.001	RMSE: 57.034 MSE: 3252.834 MAE: 52.909 $R^2$ : -4.576	No
	0.1	RMSE: 12.882 MSE: 165.958 MAE: 9.651	Yes

		$R^2$ : 0.716	
500	0.001	RMSE: 43.226 MSE: 1868.479 MAE: 39.232 $R^2$ : -2.203	No
	0.1	RMSE: 13.089 MSE: 171.322 MAE: 9.788 $R^2$ : 0.706	Yes
2000	0.001	RMSE: 16.102 MSE: 260.089 MAE: 12.626 $R^2$ : 0.554	Yes
	0.1	RMSE: 13.153 MSE: 173.003 MAE: 9.833 $R^2$ : 0.703	Yes
5000	0.001	RMSE: 13.012 MSE: 169.305 MAE: 9.899 $R^2$ : 0.710	Yes
	0.1	RMSE: 13.155 MSE: 173.042 MAE: 9.836 $R^2$ : 0.703	Yes
25000	0.001	RMSE: 12.894 MSE: 166.250 MAE: 9.679 $R^2$ : 0.715	Yes
	0.1	RMSE: 13.155 MSE: 173.042 MAE: 9.836 $R^2$ : 0.703	Yes

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50, 200 and 500, the best learning rate is 0.1. The iterations with the best RMSE are when the model is trained on two hundred iterations.

LightGBM

**Table 24: Gateway House LightGBM Predictions**

Iterations	Learning Rate	Metrics	Beaten?
50	0.001	RMSE: 23.350 MSE: 545.244 MAE: 19.682 $R^2$ : 0.065	No
	0.01	RMSE: 18.126 MSE: 328.546 MAE: 15.167	No

		$R^2$ : 0.437	
	0.1	RMSE: 12.873 MSE: 165.706 MAE: 9.576 $R^2$ : 0.716	Yes
200	0.001	RMSE: 21.242 MSE: 451.229 MAE: 17.849 $R^2$ : 0.227	No
	0.1	RMSE: 12.913 MSE: 166.783 MAE: 9.668 $R^2$ : 0.714	Yes
500	0.001	RMSE: 18.142 MSE: 329.149 MAE: 15.184 $R^2$ : 0.436	No
	0.1	RMSE: 13.306 MSE: 177.046 MAE: 9.999 $R^2$ : 0.697	Yes
2000	0.001	RMSE: 13.549 MSE: 183.568 MAE: 10.764 $R^2$ : 0.685	Yes
	0.1	RMSE: 13.660 MSE: 186.609 MAE: 10.339 $R^2$ : 0.680	Yes
5000	0.001	RMSE: 12.873 MSE: 165.714 MAE: 9.765 $R^2$ : 0.716	Yes
	0.1	RMSE: 13.681 MSE: 187.171 MAE: 10.357 $R^2$ : 0.679	Yes
25000	0.001	RMSE: 12.952 MSE: 167.762 MAE: 9.694 $R^2$ : 0.712	Yes
	0.1	RMSE: 13.681 MSE: 187.174 MAE: 10.357 $R^2$ : 0.679	Yes



From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. Increasing iterations from 50 to 500 can see that the RMSE improves as the learning rate increases. The opposite is seen for iterations in the 1000s.

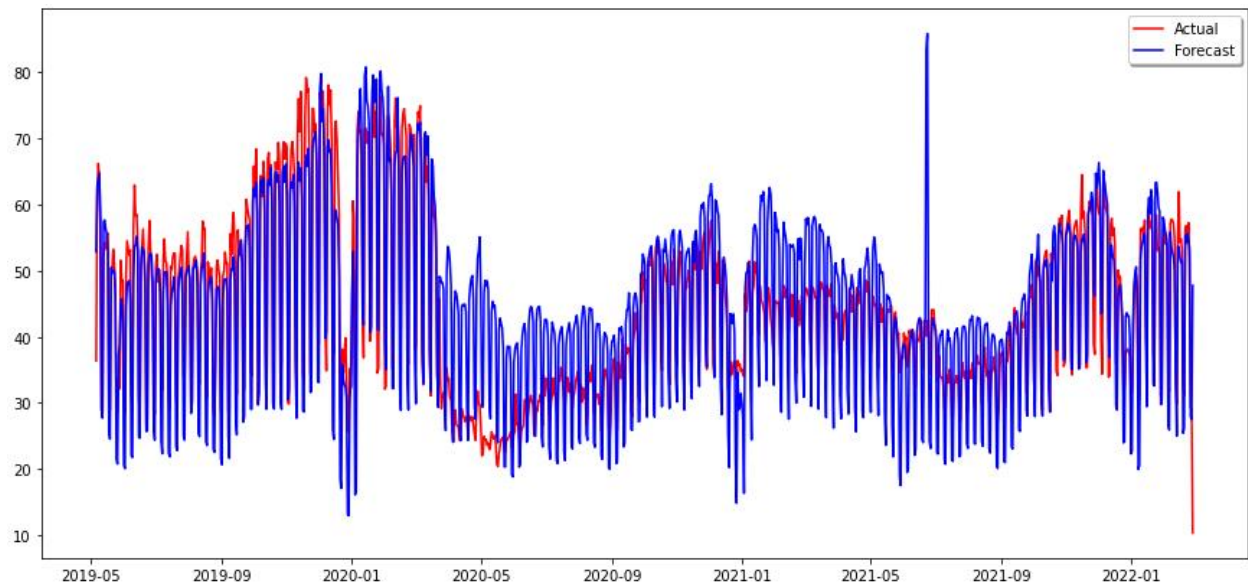
Prophet

**Table 25: Gateway House Prophet Predictions**

Trainable Days	Parameter Added	Metrics	Beaten?
4000	Interval width: 0.9	RMSE: 7.718 MSE: 59.564 MAE: 5.864 $R^2$ : 0.666	Yes
	Interval width: 0.9 UK holidays	RMSE: 8.572 MSE: 73.480 MAE: 6.592 $R^2$ : 0.588	Yes
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 8.662 MSE: 75.025 MAE: 6.670 $R^2$ : 0.579	Yes
4500	Interval width: 0.9	RMSE: 7.249 MSE: 52.525 MAE: 5.436 $R^2$ : 0.316	Yes
	Interval width: 0.9 UK holidays	RMSE: 7.649 MSE: 59.247 MAE: 5.829 $R^2$ : 0.229	Yes
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 7.736 MSE: 59.851 MAE: 5.846 $R^2$ : 0.221	Yes
3900	Interval width: 0.9	RMSE: 9.761 MSE: 95.271 MAE: 7.210 $R^2$ : 0.507	Yes
	Interval width: 0.9 UK holidays	RMSE: 10.625 MSE: 112.894 MAE: 7.998 $R^2$ : 0.416	Yes
	Interval width: 0.9 UK holidays Monthly seasonality	RMSE: 10.859 MSE: 117.929 MAE: 8.190 $R^2$ : 0.390	Yes

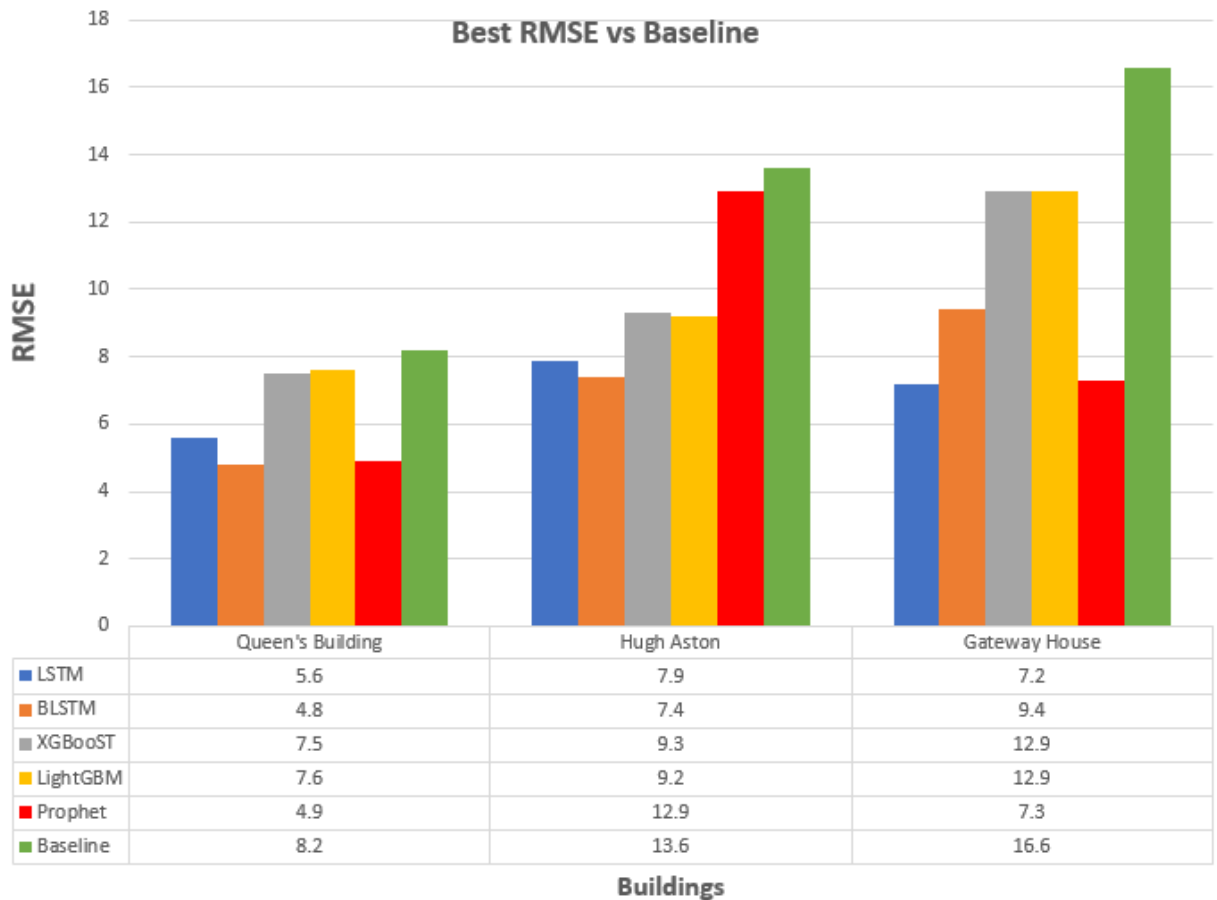
We can see from the extensive test that the Prophet algorithm performed extremely well in terms of RMSE and shows that it could be used for energy forecasting. Training for 4500 days (12 years) yielded lower RMSE but an  $R^2$  score of 31.6% (depicted below). Also found was that adding regressors yields

higher RMSE for the Gateway House dataset. From the figure, it is observed that the algorithm finds an issue with data found in July 2021 which needs investigating.



**Figure 45: Gateway House forecast using Prophet**

## Summary of Results



**Figure 46: RMSE vs Baseline**

Figure 46 shows all the chosen algorithms compared to the baseline scores. We can see that:

- In two out of three buildings, the bidirectional LSTM performed significantly better than all other algorithms as stated in the literature. Gateway House was not able to fully use the BLSTM which calls for further fine-tuning.
- In the case of Hugh Aston, the Prophet algorithm struggled to make sense of the chaotic data pattern occurring after the coronavirus outbreak. More testing needs to be undertaken and some hyper-parameter tuning.
- The means squared error (MSE) and the root mean squared error (RMSE) both share the same characteristics in terms of creating a bar chart.
- Lastly, this is by far not the end of as in ML, we are always testing to improve.

## Conclusion

In a building, gathering valuable data is a time-consuming operation. As a result, this research investigated a method that used machine learning to reduce the amount of time spent observing a target facility. The objective is to train the system on previously gathered building datasets before fine-tuning them to predict the energy consumption for the future. This work provides several algorithms for predicting energy usage in higher education facilities that employ an LSTM, Bidirectional LSTM, Ensemble algorithms (XGBoost and LightBoost) and Facebook's Prophet. For the technique to be implemented in everyday circumstances, the model leverages adequate learning using a mix of high to low correlation as they can prove useful to the model. Because multiple datasets can be utilized in the training step of a model and still generate satisfactory results, the above advances allow for a reduction in the time required to monitor energy consumption.

By locking the variables of the training set, machine learning is used in the manner of parametric learning. To extend to the validation strategy with varying time limitations, a scalable system that can understand patterns is required. As a result, the model employs **S2S** prediction to manage additional information and be more adaptable. The findings demonstrate that, according to various research, bidirectional long short-term memory is greater and the preferred option in terms of power use. All but one of the building datasets yielded a poor result, indicating that the BLSTM was unable to successfully train on the dataset, resulting in a poor RMSE. This demonstrates that the proposed model is trained well in terms of generality. It is worth noting that prior energy consumption research had a small amount of information both in training and validation, which could justify the poor results.

The findings of the study in Chapter Data Analysis are noteworthy; the numerous correlation visualizations reveal various relationships between the datasets, indicating that they are similar. As a result, although the validation step conveys a different tale, the transferability between datasets appears to remain a mystery. Such that, the system discovered a "secret" association in the datasets with numerous attributes. The findings of the experiments suggest that ml algorithms perform much better when the training data covers a wider range of the target characteristic (energy consumption).

In terms of the average RMSE score, it was a surprise that the LSTM outperformed the bidirectional-LSTM as this was meant to be an improvement to the LSTM according to the work "Short term power load forecasting based on multi-layer bidirectional recurrent neural network "[32]. The LSTM achieved an average RMSE score of 6.9 following the 'less is best' rule of the metric. Bidirectional LSTM came in second which was not a surprise due to its forward and backwards network traversal. The ensemble methods, XGBoost and LightGBM showed similar scores which means there needs to be more fine-tuning else a hybrid version of the algorithm needs to be explored. From the testing, the more the iterations there the better the predictive performance. Finally, the second algorithm that produce surprising results was the Facebook Prophet as it achieved an average RMSE score of 8.4 surpassing both ensemble methodologies. There is little research utilizing this algorithm for energy consumption prediction. The algorithm was very lightweight and fast to implement which serves as a great start for beginners and experts. The only reason for this score was due to the Hugh Aston dataset including lots of anomalies which could only be forward filled as we are dealing with a time-series data. Nonetheless, we foresee many studies to be conducted using Prophet.

In the coming years, it will be crucial to enhance the model's possible advantages to overcome the limitations of using more distinct features in training data. A further exciting project in the long term is to

combine data from multiple facilities to see whether efficiency can be retained or enhanced. It could make for improved generalization. One other potentially intriguing study is to use data about utility consumption to forecast building occupation. Extra information on the number of individuals in the facility must be acquired for such a forecast to be achievable. And it would be fascinating to see if occupancy data may help with utility usage forecast.

As a last note on this dissertation, we urge utilizing the developed framework with prudence since it performs differently than traditional research. Within issues with sufficient data throughout the training process, it generalizes successfully. Even while forecasts are satisfactory for the entire quantity of data, fine-tuning is required to improve precision for great quantities. Also, with greater training data, the model's accuracy for extended durations will improve, as the quantity of data currently available is insufficient. We feel that the presented model has promise and can be developed to the point of applicability and commercialization with more study and experimentation.

## Bibliography

- [1] D. Hawkins, S. M. Hong, R. Raslan, D. Mumovic, and S. Hanna, "Determinants of Energy Use in UK Higher Education Buildings Using Statistical and Artificial Neural Network Methods," *International Journal of Sustainable Built Environment*, vol. 1, no. 1, pp. 50–63, Jun. 2012, doi: [10.1016/j.ijsbe.2012.05.002](https://doi.org/10.1016/j.ijsbe.2012.05.002).
- [2] HESA, "HESA - Experts in Higher Education Data and Analysis," *Hesa.ac.uk*, 2022. [https://www.hesa.ac.uk/?option=com\\_heicontacts&Itemid=87](https://www.hesa.ac.uk/?option=com_heicontacts&Itemid=87)
- [3] K. P. Amber and J. Parkin, "Barriers to the Uptake of Combined Heat and Power Technology in the UK Higher Education Sector," *International Journal of Sustainable Energy*, vol. 34, no. 6, pp. 406–416, Nov. 2013, doi: [10.1080/14786451.2013.858158](https://doi.org/10.1080/14786451.2013.858158).
- [4] Higher Education Statistics Agency, "Environmental Information, Estates Management Statistics Tables," *Higher Education Statistics Agency*, 2022. <https://www.hesa.ac.uk/data-and-analysis/estates/table-1>
- [5] J. Elmes, "English Universities Fall Further behind Carbon Reduction Target," *Times Higher Education (THE)*, Sep. 24, 2015. <https://www.timeshighereducation.com/news/english-universities-fall-further-behind-carbon-reduction-target>
- [6] K. P. Amber, M. W. Aslam, and S. K. Hussain, "Electricity consumption forecasting models for administration buildings of the UK higher education sector," *Energy and Buildings*, vol. 90, pp. 127–136, Mar. 2015, doi: [10.1016/j.enbuild.2015.01.008](https://doi.org/10.1016/j.enbuild.2015.01.008).
- [7] S. Chand, "The Importance of Forecasting in the Operations of Modern Management," *Your Article Library*, Jun. 25, 2013. <https://www.yourarticlelibrary.com/management/the-importance-of-forecasting-in-the-operations-of-modern-management/3504>
- [8] Government Finance Administrative Association, "Financial Forecasting in the Budget Preparation Process," *Gfoa.org*, 2021. <https://www.gfoa.org/materials/financial-forecasting-in-the-budget-preparation-process>
- [9] A. R. Khan, A. Mahmood, A. Safdar, Z. A. Khan, and N. A. Khan, "Load forecasting, Dynamic Pricing and DSM in Smart grid: a Review," *Renewable and Sustainable Energy Reviews*, vol. 54, pp. 1311–1322, Feb. 2016, doi: [10.1016/j.rser.2015.10.117](https://doi.org/10.1016/j.rser.2015.10.117).
- [10] N. Fumo and M. A. Rafe Biswas, "Regression Analysis for Prediction of Residential Energy Consumption," *Renewable and Sustainable Energy Reviews*, vol. 47, pp. 332–343, Jul. 2015, doi: [10.1016/j.rser.2015.03.035](https://doi.org/10.1016/j.rser.2015.03.035).
- [11] M. R. Braun, H. Altan, and S. B. M. Beck, "Using Regression Analysis to Predict the Future Energy Consumption of a Supermarket in the UK," *Applied Energy*, vol. 130, pp. 305–313, Oct. 2014, doi: [10.1016/j.apenergy.2014.05.062](https://doi.org/10.1016/j.apenergy.2014.05.062).
- [12] N. Kaza, "Understanding the Spectrum of Residential Energy consumption: a Quantile Regression Approach," *Energy Policy*, vol. 38, no. 11, pp. 6574–6585, Nov. 2010, doi: [10.1016/j.enpol.2010.06.028](https://doi.org/10.1016/j.enpol.2010.06.028).
- [13] A. Lomet, F. Suard, and D. Chèze, "Statistical Modeling for Real Domestic Hot Water Consumption Forecasting," *Energy Procedia*, vol. 70, pp. 379–387, May 2015, doi: [10.1016/j.egypro.2015.02.138](https://doi.org/10.1016/j.egypro.2015.02.138).

- [14] H. Zhao and F. Magoulès, "A Review on the Prediction of Building Energy Consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, Aug. 2012, doi: [10.1016/j.rser.2012.02.049](https://doi.org/10.1016/j.rser.2012.02.049).
- [15] T. Catalina, V. Iordache, and B. Caracaleanu, "Multiple Regression Model for Fast Prediction of the Heating Energy Demand," *Energy and Buildings*, vol. 57, pp. 302–312, Feb. 2013, doi: [10.1016/j.enbuild.2012.11.010](https://doi.org/10.1016/j.enbuild.2012.11.010).
- [16] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of Machine Learning*. Cambridge, Massachusetts: The MIT Press, 2018.
- [17] A. Tsanas and A. Xifara, "Accurate Quantitative Estimation of Energy Performance of Residential Buildings Using Statistical Machine Learning Tools," *Energy and Buildings*, vol. 49, pp. 560–567, Jun. 2012, doi: [10.1016/j.enbuild.2012.03.003](https://doi.org/10.1016/j.enbuild.2012.03.003).
- [18] B. Dong, C. Cao, and S. E. Lee, "Applying Support Vector Machines to Predict Building Energy Consumption in Tropical Region," *Energy and Buildings*, vol. 37, no. 5, pp. 545–553, May 2005, doi: [10.1016/j.enbuild.2004.09.009](https://doi.org/10.1016/j.enbuild.2004.09.009).
- [19] K. Kavaklioglu, "Modeling and Prediction of Turkey's Electricity Consumption Using Support Vector Regression," *Applied Energy*, vol. 88, no. 1, pp. 368–375, Jan. 2011, doi: [10.1016/j.apenergy.2010.07.021](https://doi.org/10.1016/j.apenergy.2010.07.021).
- [20] B. B. Ekici and U. T. Aksoy, "Prediction of Building Energy Consumption by Using Artificial Neural Networks," *Advances in Engineering Software*, vol. 40, no. 5, pp. 356–362, May 2009, doi: [10.1016/j.advengsoft.2008.05.003](https://doi.org/10.1016/j.advengsoft.2008.05.003).
- [21] J. Yang, H. Rivard, and R. Zmeureanu, "On-line Building Energy Prediction Using Adaptive Artificial Neural Networks," *Energy and Buildings*, vol. 37, no. 12, pp. 1250–1259, Dec. 2005, doi: [10.1016/j.enbuild.2005.02.005](https://doi.org/10.1016/j.enbuild.2005.02.005).
- [22] S. Karatasou, M. Santamouris, and V. Geros, "Modeling and Predicting building's Energy Use with Artificial Neural networks: Methods and Results," *Energy and Buildings*, vol. 38, no. 8, pp. 949–958, Aug. 2006, doi: [10.1016/j.enbuild.2005.11.005](https://doi.org/10.1016/j.enbuild.2005.11.005).
- [23] L. Di Persio and O. Honchar, "Analysis of Recurrent Neural Networks for short-term Energy Load Forecasting," *AIP Conference Proceedings*, 2017, doi: [10.1063/1.5012469](https://doi.org/10.1063/1.5012469).
- [24] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Forecasting network-wide Traffic State with Missing Values," *Transportation Research Part C: Emerging Technologies*, vol. 118, p. 102674, Sep. 2020, doi: [10.1016/j.trc.2020.102674](https://doi.org/10.1016/j.trc.2020.102674).
- [25] M. Jia, J. Huang, L. Pang, and Q. Zhao, "Analysis and Research on Stock Price of LSTM and Bidirectional LSTM Neural Network," *Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)*, 2019, DOI: [10.2991/iccia-19.2019.72](https://doi.org/10.2991/iccia-19.2019.72).
- [26] J. Bedi and D. Toshniwal, "Deep Learning Framework to Forecast Electricity Demand," *Applied Energy*, vol. 238, pp. 1312–1326, Mar. 2019, doi: [10.1016/j.apenergy.2019.01.113](https://doi.org/10.1016/j.apenergy.2019.01.113).
- [27] S. Bouktif, A. Fiaz, A. Ouni, and M. Serhani, "Optimal Deep Learning LSTM Model for Electric Load Forecasting Using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches †," *Energies*, vol. 11, no. 7, p. 1636, Jun. 2018, doi: [10.3390/en11071636](https://doi.org/10.3390/en11071636).

- [28] S. Muzaffar and A. Afshari, "Short-Term Load Forecasts Using LSTM Networks," *Energy Procedia*, vol. 158, pp. 2922–2927, Feb. 2019, doi: [10.1016/j.egypro.2019.01.952](https://doi.org/10.1016/j.egypro.2019.01.952).
- [29] B.-S. Kwon, R.-J. Park, and K.-B. Song, "Short-Term Load Forecasting Based on Deep Neural Networks Using LSTM Layer," *Journal of Electrical Engineering & Technology*, vol. 15, no. 4, pp. 1501–1509, May 2020, DOI: [10.1007/s42835-020-00424-7](https://doi.org/10.1007/s42835-020-00424-7).
- [30] J. Q. Wang, Y. Du, and J. Wang, "LSTM Based long-term Energy Consumption Prediction with Periodicity," *Energy*, vol. 197, p. 117197, Apr. 2020, doi: [10.1016/j.energy.2020.117197](https://doi.org/10.1016/j.energy.2020.117197).
- [31] F. Martínez-Álvarez *et al.*, "Coronavirus Optimization Algorithm: a Bioinspired Metaheuristic Based on the COVID-19 Propagation Model | Big Data," *Big Data*, 2021. <https://www.liebertpub.com/doi/abs/10.1089/BIG.2020.0051>
- [32] X. Tang, Y. Dai, T. Wang, and Y. Chen, "Short-term Power Load Forecasting Based on multi-layer Bidirectional Recurrent Neural Network," *IET Generation, Transmission & Distribution*, vol. 13, no. 17, pp. 3847–3854, Sep. 2019, doi: [10.1049/iet-gtd.2018.6687](https://doi.org/10.1049/iet-gtd.2018.6687).
- [33] M. C. Pegalajar, L. G. B. Ruiz, M. P. Cuéllar, and R. Rueda, "Analysis and Enhanced Prediction of the Spanish Electricity Network through Big Data and Machine Learning Techniques," *International Journal of Approximate Reasoning*, vol. 133, pp. 48–59, Jun. 2021, doi: [10.1016/j.ijar.2021.03.002](https://doi.org/10.1016/j.ijar.2021.03.002).
- [34] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019, DOI: [10.1109/tsg.2017.2753802](https://doi.org/10.1109/tsg.2017.2753802).
- [35] R. Jiao, T. Zhang, Y. Jiang, and H. He, "Short-Term Non-Residential Load Forecasting Based on Multiple Sequences LSTM Recurrent Neural Network," *IEEE Access*, vol. 6, pp. 59438–59448, 2018, DOI: [10.1109/access.2018.2873712](https://doi.org/10.1109/access.2018.2873712).
- [36] T.-Y. Kim and S.-B. Cho, "Predicting Residential Energy Consumption Using CNN-LSTM Neural Networks," *Energy*, vol. 182, pp. 72–81, Sep. 2019, doi: [10.1016/j.energy.2019.05.230](https://doi.org/10.1016/j.energy.2019.05.230).
- [37] N. Al Khafaf, M. Jalili, and P. Sokolowski, "Application of Deep Learning Long Short-Term Memory in Energy Demand Forecasting," *Engineering Applications of Neural Networks*, pp. 31–42, 2019, DOI: [10.1007/978-3-030-20257-6\\_3](https://doi.org/10.1007/978-3-030-20257-6_3).
- [38] X. Wang, T. Zhao, H. Liu, and R. He, "Power Consumption Predicting and Anomaly Detection Based on Long Short-Term Memory Neural Network," *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Apr. 2019, doi: [10.1109/icccbda.2019.8725704](https://doi.org/10.1109/icccbda.2019.8725704).
- [39] Z. Khan, T. Hussain, A. Ullah, S. Rho, M. Lee, and S. Baik, "Towards Efficient Electricity Forecasting in Residential and Commercial Buildings: a Novel Hybrid CNN with a LSTM-AE Based Framework," *Sensors*, vol. 20, no. 5, p. 1399, Mar. 2020, doi: [10.3390/s20051399](https://doi.org/10.3390/s20051399).
- [40] N. Wei, C. Li, X. Peng, Y. Li, and F. Zeng, "Daily Natural Gas Consumption Forecasting via the Application of a Novel Hybrid Model," *Applied Energy*, vol. 250, pp. 358–368, Sep. 2019, doi: [10.1016/j.apenergy.2019.05.023](https://doi.org/10.1016/j.apenergy.2019.05.023).
- [41] S. Singaravel, J. Suykens, and P. Geyer, "Deep-learning neural-network Architectures and methods: Using component-based Models in building-design Energy Prediction," *Advanced Engineering Informatics*, vol. 38, pp. 81–90, Oct. 2018, doi: [10.1016/j.aei.2018.06.004](https://doi.org/10.1016/j.aei.2018.06.004).



- [42] C. Zhou *et al.*, "Using Long short-term Memory Networks to Predict Energy Consumption of air-conditioning Systems," *Sustainable Cities and Society*, vol. 55, p. 102000, Apr. 2020, doi: [10.1016/j.scs.2019.102000](https://doi.org/10.1016/j.scs.2019.102000).
- [43] F. He, J. Zhou, Z. Feng, G. Liu, and Y. Yang, "A Hybrid short-term Load Forecasting Model Based on Variational Mode Decomposition and Long short-term Memory Networks considering Relevant Factors with Bayesian Optimization Algorithm," *Applied Energy*, vol. 237, pp. 103–116, Mar. 2019, doi: [10.1016/j.apenergy.2019.01.055](https://doi.org/10.1016/j.apenergy.2019.01.055).
- [44] T.-Y. Kim and S.-B. Cho, "Particle Swarm Optimization-based CNN-LSTM Networks for Forecasting Energy Consumption," *2019 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2019, doi: [10.1109/cec.2019.8789968](https://doi.org/10.1109/cec.2019.8789968).
- [45] Y. Yang, Z. Shang, Y. Chen, and Y. Chen, "Multi-Objective Particle Swarm Optimization Algorithm for Multi-Step Electric Load Forecasting," *Energies*, vol. 13, no. 3, p. 532, Jan. 2020, DOI: [10.3390/en13030532](https://doi.org/10.3390/en13030532).
- [46] X. Guo, Q. Zhao, S. Wang, D. Shan, and W. Gong, "A Short-Term Load Forecasting Model of LSTM Neural Network considering Demand Response," *Complexity*, vol. 2021, pp. 1–7, Apr. 2021, doi: [10.1155/2021/5571539](https://doi.org/10.1155/2021/5571539).
- [47] H. Su, E. Zio, J. Zhang, M. Xu, X. Li, and Z. Zhang, "A Hybrid Hourly Natural Gas Demand Forecasting Method Based on the Integration of Wavelet Transform and Enhanced Deep-RNN Model," *Energy*, vol. 178, pp. 585–597, Jul. 2019, doi: [10.1016/j.energy.2019.04.167](https://doi.org/10.1016/j.energy.2019.04.167).
- [48] S. J. Taylor and B. Letham, "Forecasting at Scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, Sep. 2017, doi: [10.1080/00031305.2017.1380080](https://doi.org/10.1080/00031305.2017.1380080).
- [49] I. Yenidoğan, A. Çayır, O. Kozan, T. Dağ, and Ç. Arslan, "Bitcoin Forecasting Using ARIMA and PROPHET," *IEEE Xplore*, 2018. <https://ieeexplore.ieee.org/abstract/document/8566476>
- [50] A. Chaudhari, "Forecasting Cryptocurrency Prices Using Machine Learning," *norma.ncirl.ie*, 2020. <http://norma.ncirl.ie/id/eprint/4272>
- [51] F. Bianchi, A. Castellini, P. Tarocco, and A. Farinelli, "Load Forecasting in District Heating Networks: Model Comparison on a Real-World Case Study," *Machine Learning, Optimization, and Data Science*, pp. 553–565, 2019, DOI: [10.1007/978-3-030-37599-7\\_46](https://doi.org/10.1007/978-3-030-37599-7_46).
- [52] S. Das, "Forecasting the Generation of Wind Power in the Western and Southern Regions of India: Comparative Approach," *norma.ncirl.ie*, 2020. <http://norma.ncirl.ie/id/eprint/4267>
- [53] G. A. Papacharalampous and H. Tyrallis, "Evaluation of Random Forests and Prophet for Daily streamflow forecasting," *Advances in Geosciences*, vol. 45, pp. 201–208, Aug. 2018, DOI: [10.5194/adgeo-45-201-2018](https://doi.org/10.5194/adgeo-45-201-2018).
- [54] E. Žunić, K. Korjenić, K. Hodžić, and D. Đonko, "Application of Facebook's Prophet Algorithm for Successful Sales Forecasting Based on Real-world Data," *International Journal of Computer Science and Information Technology*, vol. 12, no. 2, pp. 23–36, Apr. 2020, doi: [10.5121/ijcsit.2020.12203](https://doi.org/10.5121/ijcsit.2020.12203).
- [55] E. Zunic, S. Delalic, K. Hodzic, A. Besirevic, and H. Hindija, "Smart Warehouse Management System Concept with Implementation," *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, Nov. 2018, DOI: [10.1109/neurel.2018.8587004](https://doi.org/10.1109/neurel.2018.8587004).

- [56] R. Wang, S. Lu, and Q. Li, "Multi-criteria Comprehensive Study on Predictive Algorithm of Hourly Heating Energy Consumption for Residential Buildings," *Sustainable Cities and Society*, vol. 49, p. 101623, Aug. 2019, doi: [10.1016/j.scs.2019.101623](https://doi.org/10.1016/j.scs.2019.101623).
- [57] S. Seyedzadeh, F. Pour Rahimian, P. Rastogi, and I. Glesk, "Tuning Machine Learning Models for Prediction of Building Energy Loads," *Sustainable Cities and Society*, vol. 47, p. 101484, May 2019, doi: [10.1016/j.scs.2019.101484](https://doi.org/10.1016/j.scs.2019.101484).
- [58] Y. Pan and L. Zhang, "Data-driven Estimation of Building Energy Consumption with multi-source Heterogeneous Data," *Applied Energy*, vol. 268, p. 114965, Jun. 2020, doi: [10.1016/j.apenergy.2020.114965](https://doi.org/10.1016/j.apenergy.2020.114965).
- [59] T. Pinto, I. Praça, Z. Vale, and J. Silva, "Ensemble Learning for Electricity Consumption Forecasting in Office Buildings," *Neurocomputing*, May 2020, DOI: [10.1016/j.neucom.2020.02.124](https://doi.org/10.1016/j.neucom.2020.02.124).
- [60] H. Deng, D. Fannon, and M. J. Eckelman, "Predictive Modeling for US Commercial Building Energy use: a Comparison of Existing Statistical and Machine Learning Algorithms Using CBECS Microdata," *Energy and Buildings*, vol. 163, pp. 34–43, Mar. 2018, doi: [10.1016/j.enbuild.2017.12.031](https://doi.org/10.1016/j.enbuild.2017.12.031).
- [61] N. G. Paterakis, E. Mocanu, M. Gibescu, B. Stappers, and W. van Alst, "Deep Learning versus Traditional Machine Learning Methods for Aggregated Energy Demand Prediction," *IEEE Xplore*, Sep. 01, 2017.  
[https://ieeexplore.ieee.org/abstract/document/8260289?casa\\_token=qxOEwgQXoF0AAAAA:cx5Pc\\_vwK5fzk41pqAnvWrNx6Bfff191dcx6jinPp\\_CeMUAJ2G1sm3CMkYeYSUhiORE-VJrOfHY](https://ieeexplore.ieee.org/abstract/document/8260289?casa_token=qxOEwgQXoF0AAAAA:cx5Pc_vwK5fzk41pqAnvWrNx6Bfff191dcx6jinPp_CeMUAJ2G1sm3CMkYeYSUhiORE-VJrOfHY)
- [62] M. Sajjad *et al.*, "A Novel CNN-GRU-Based Hybrid Approach for Short-Term Residential Load Forecasting," *IEEE Access*, vol. 8, pp. 143759–143768, 2020, DOI: [10.1109/access.2020.3009537](https://doi.org/10.1109/access.2020.3009537).
- [63] S.-V. Oprea and A. Bâra, "Machine Learning Algorithms for Short-Term Load Forecast in Residential Buildings Using Smart Meters, Sensors and Big Data Solutions," *IEEE Access*, vol. 7, pp. 177874–177889, 2019, DOI: [10.1109/ACCESS.2019.2958383](https://doi.org/10.1109/ACCESS.2019.2958383).
- [64] A. M. Pirbazari, A. Chakravorty, and C. Rong, "Evaluating Feature Selection Methods for Short-Term Load Forecasting," *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Feb. 2019, DOI: [10.1109/bigcomp.2019.8679188](https://doi.org/10.1109/bigcomp.2019.8679188).
- [65] R. Wang, S. Lu, and W. Feng, "A Novel Improved Model for Building Energy Consumption Prediction Based on Model Integration," *Applied Energy*, vol. 262, p. 114561, Mar. 2020, doi: [10.1016/j.apenergy.2020.114561](https://doi.org/10.1016/j.apenergy.2020.114561).
- [66] M. Al-Rakhami, A. Gumaiei, A. Alsanad, A. Alamri, and M. M. Hassan, "An Ensemble Learning Approach for Accurate Energy Load Prediction in Residential Buildings," *IEEE Access*, vol. 7, pp. 48328–48338, 2019, DOI: [10.1109/access.2019.2909470](https://doi.org/10.1109/access.2019.2909470).
- [67] C. Fan, F. Xiao, and Y. Zhao, "A short-term Building Cooling Load Prediction Method Using Deep Learning Algorithms," *Applied Energy*, vol. 195, pp. 222–233, Jun. 2017, doi: [10.1016/j.apenergy.2017.03.064](https://doi.org/10.1016/j.apenergy.2017.03.064).
- [68] Z. Wang, T. Hong, and M. A. Piette, "Building Thermal Load Prediction through Shallow Machine Learning and Deep Learning," *Applied Energy*, vol. 263, p. 114683, Apr. 2020, doi: [10.1016/j.apenergy.2020.114683](https://doi.org/10.1016/j.apenergy.2020.114683).

- [69] H. Zheng, J. Yuan, and L. Chen, "Short-Term Load Forecasting Using EMD-LSTM Neural Networks with a Xgboost Algorithm for Feature Importance Evaluation," *Energies*, vol. 10, no. 8, p. 1168, Aug. 2017, DOI: [10.3390/en10081168](https://doi.org/10.3390/en10081168).
- [70] D. Chakraborty and H. Elzarka, "Advanced Machine Learning Techniques for Building Performance simulation: a Comparative Analysis," *Journal of Building Performance Simulation*, vol. 12, no. 2, pp. 193–207, Jul. 2018, doi: [10.1080/19401493.2018.1498538](https://doi.org/10.1080/19401493.2018.1498538).
- [71] P. Li and J.-S. Zhang, "A New Hybrid Method for China's Energy Supply Security Forecasting Based on ARIMA and XGBoost," *Energies*, vol. 11, no. 7, p. 1687, Jun. 2018, doi: [10.3390/en11071687](https://doi.org/10.3390/en11071687).
- [72] Y. Zhang, X. Zhou, Y. Teng, J. Fang, and W. Zheng, "Resource Allocation for Multi-User MEC System: Machine Learning Approaches," *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, DOI: [10.1109/csci46756.2018.00159](https://doi.org/10.1109/csci46756.2018.00159).
- [73] H. Lu, F. Cheng, X. Ma, and G. Hu, "Short-term Prediction of Building Energy Consumption Employing an Improved Extreme Gradient Boosting model: a Case Study of an Intake Tower," *Energy*, vol. 203, p. 117756, Jul. 2020, doi: [10.1016/j.energy.2020.117756](https://doi.org/10.1016/j.energy.2020.117756).
- [74] W. Yucong and W. Bo, "Research on EA-Xgboost Hybrid Model for Building Energy Prediction," *Journal of Physics: Conference Series*, vol. 1518, no. 1, p. 012082, Apr. 2020, doi: [10.1088/1742-6596/1518/1/012082](https://doi.org/10.1088/1742-6596/1518/1/012082).
- [75] T. Chen and C. Guestrin, "XGBoost: a Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016, DOI: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).
- [76] G. Ke *et al.*, "LightGBM: a Highly Efficient Gradient Boosting Decision Tree," 2017. [Online]. Available: <https://papers.nips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>
- [77] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman, "A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting," *IEEE Access*, vol. 7, pp. 28309–28318, 2019, DOI: [10.1109/access.2019.2901920](https://doi.org/10.1109/access.2019.2901920).
- [78] D. Jin, Y. Lu, J. Qin, Z. Cheng, and Z. Mao, "SwiftIDS: Real-time Intrusion Detection System Based on LightGBM and Parallel Intrusion Detection Mechanism," *Computers & Security*, vol. 97, p. 101984, Oct. 2020, doi: [10.1016/j.cose.2020.101984](https://doi.org/10.1016/j.cose.2020.101984).