Predictive Modelling of Low Tariff Energy Consumption in UK

MSc Research Project
MSc Data Analytics

Sanket Patne
Student ID: x22171746

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sanket Patne
Student ID:	x22171746
Programme:	MSc Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	14/12/2023
Project Title:	Predictive Modelling of Low Tariff Energy Consumption in UK
Word Count:	6529
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sanket Patne
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Predictive Modelling of Low Tariff Energy Consumption in UK

Sanket Patne
x22171746

Abstract

Deep learning applications are constantly evolving and it has seen more innovations in 2023 alone than ever in the past. In such an ever changing scenario its application in commercial sectors like load prediction can also not fall behind, but unlike traditional machine learning techniques these new techniques should be more widely applicable in scope for global adoption. This is a compact study towards determining which neural networks are best suited for load prediciton as compared to the ones explored in previous studies that include LSTM, Bidirectional LSTM (Bi-LSTM), Multi-output Gaussian processes (MGP), etc. With the custom data formulated, best results were achieved by the CNN model with a MAPE values of 28.0559 percent and Variance Score of 0.0389. The best model was compared against models developed from the RNN and LSTM algorithms.

Keywords: Load prediction, Neural Networks, Hyper Parameter Tuning, low tariff energy consumption

1 Introduction

Energy efficiency is more important than ever in the current geopolitical climate. Consumers need more information about their energy usage and pricing models to make informed decisions about their energy consumption. This information can help them reduce their energy bills and carbon footprint and help power providers optimize electricity delivery. One way to improve energy efficiency is to monitor consumption patterns over time. As observed in a previous study by (Chen et al.; 2023), this data can be used to identify peak and low usage periods, and to adjust the production of electricity from regular and sustainable sources accordingly.

Seasonal variances in usage can be caused by factors such as heating and cooling needs, and the adoption of solar alternatives. For example, in the winter, households need more heating and have shorter days, which can lead to higher energy consumption. In the summer, households may use more air conditioning and have longer days, which can also lead to higher energy consumption. However, the adoption of solar alternatives can offset some of this increased energy consumption.

Convolutional neural networks (CNN), recurrent neural networks (RNN), and artificial neural networks (ANN) are implemented and evaluated for load prediction in this study. It excels in comparison by offering an extensive analysis across multiple deep learning architectures. By introducing a hyperband parameter tuner, it improves upon previous studies that did not take hyperparameter tuning into account. Models for feature extraction and retraining are unique in that they exhibit iterative refinement and ongoing improvement. The study is improved by the model comparison, which sheds light on how well various deep learning architectures perform in comparison. Furthermore, the comparison with current models broadens contributions beyond research that just benchmark against conventional machine learning methods.

1.1 Motivation and Project Background

Electricity consumers use electricity based upon their necessity in any household or business setting. The usage of electricity is not influenced by any major habits or patterns normally. But the current events and global usage habits affecting carbon emissions, and a bunch of other factors have all come into play to create the need for more effective and efficient use of not just electricity but all the natural resources that are at our disposal, affect the surroundings and might even affect the environment in the long term to make the planet uninhabitable. In the more recent studies, it has been found out that if the global temperatures go up by over 2 degrees celsius above preindustrial levels people in certain regions of the world will have to face extended periods of heat that is beyond human tolerance as early as next year. All these concerns prompt us to use the resources available as efficiently as possible to avoid these dire scenarios. That brings us to using one of the most essential resources electricity which is used without most considerations and freely as and when needed. This gives us the most scope to improve the usage patterns in electricity and make the most difference in offsetting the carbon footprints. Overall, research into consumption patterns of consumers is essential for improving energy efficiency and reducing carbon emissions. By monitoring consumption patterns over time, researchers can identify areas where energy efficiency can be improved, and power providers can optimize electricity delivery.

1.2 Research Questions

In the current day and age there are highly complex deep learning models being employed globally for high accuracy load prediction. But the complexity of these models comes in it also becomes highly unlikely that the outcome is going to be legible to the end user and replicable by third parties to implement.Therefore, the following research question are formulated to meet the replicability and simplicity demands for more recent ground level studies:
Research Question: What machine learning methods are best for precisely predicting the amount of energy consumed in a given area? Power providers can use this data to optimize electricity delivery and reduce costs for both consumers and them. For example, they can use the data to plan for maintenance and upgrades, and to develop more efficient pricing models.
Sub Research Question: How can machine learning algorithms be effectively used to estimate short-term energy consumption load based on historical data? Overall, research into consumption patterns of consumers is essential for improving energy efficiency and reducing carbon emissions. By monitoring consumption patterns over time, researchers can identify areas where energy efficiency can be improved, and power providers can optimize electricity delivery.

1.3 Research Objectives and Contributions

Obj1: Extract features from data provided by energy providers
Obj2(a): Implement, evaluation and results of CNN models on House A
Obj2(b): Create a hyperband parameter tuner for finding best parameters for house A
Obj2(c): Extract best features for new model and retrain model
Obj3(a): Implement, evaluation and results of RNN models on House B
Obj3(b): Create a hyperband parameter tuner for finding best parameters for house B
Obj3(c): Extract best features for new model and retrain model
Obj4(a): Implement, evaluation and results of ANN models on House C
Obj4(b): Create a hyperband parameter tuner for finding best parameters for house C
Obj4(c): Extract best features for new model and retrain model
Obj5(c): Comparison of developed models
Obj6(d): Comparison of developed models with existing models

2 Related work

In smart grids, load prediction plays a critical role in helping utilities better manage their resources and give consumers more dependable service. An increasing amount of research is showing that machine learning (ML) is a highly effective method for load prediction. ML models are able to recognise patterns and trends that can be utilised to forecast future load demand by learning from past load data as well as other pertinent features. Because ML models can learn from data, they are a good fit for complicated, dynamic tasks like load prediction.

The following are some major arguments in favour of the need for machine learning in load prediction:
Complex and non-linear relationships between variables can be handled by ML models. The complicated issue of load forecast is impacted by numerous variables, such as the weather, day of the week, and time of day. These intricate interactions can be modelled by ML models, which can then produce precise load estimates. ML models are able to adjust to shifting load patterns. Climate change, technology breakthroughs, and economic expansion are some of the variables that cause load patterns to fluctuate regularly. ML models can learn from fresh data and adjust to these shifting trends. Load for various time horizons can be predicted using machine learning models. Different time horizons can be used for load prediction, including short-term (like intra-day), medium-term (like day-ahead), and long-term (like weekly or monthly). For each of these time frames, load can be predicted using machine learning models.

All things considered, machine learning (ML) is an essential technique for load prediction since it can handle the problem's complexity and dynamism and produce precise predictions for various time horizons. Several of the particular studies conducted in this field made use of the following models: Long short-term memory (LSTM) networks, Convolutional neural networks (CNNs), Recurrent neural networks (RNNs), Bidirectional LSTM (Bi-LSTM), Multi-output Gaussian processes (MGP), Attention mechanism, Artificial bee colony (ABC) and Transformer. The majority of the research trained their models through supervised learning. A model is trained using a collection of input data and matching output data through supervised learning. When it comes to load prediction, the input data usually consists of historical load data along with other pertinent features like weather, day of the week, and time of day. The load that the model is attempting to forecast is the output data.

Several studies employed ensemble learning techniques to enhance the models' performance. In ensemble learning, many models' predictions are combined to yield a prediction that is more accurate. This can be accomplished by utilising a weighted average, where the weights are established by how well each model performs on a held-out validation set, or simply averaging the forecasts of the several models.
The studies used a range of indicators to assess the models' performance, such as: Mean absolute error (MAE), Mean squared error (MSE), Root mean squared error (RMSE), Accuracy, F1 score.

2.1 Comparison of several time horizons (hourly, intraday, day-ahead):

This technique allowed the researchers (Huang; 2023) to show that the model can accurately anticipate load for a range of time horizons. As a result, the model becomes more flexible and useful for a larger variety of load prediction jobs. They were able to show that the model can accurately forecast load for various customer kinds by testing it on a variety of customer types. As a result, the model gains more adaptability and is suitable for a greater variety of load prediction applications.

The researchers(Li et al.; 2023) demonstrated that the model can accurately represent the dynamics of load patterns at various time scales by evaluating it over a number of time horizons. This is crucial for creating precise load prediction models, particularly for applications involving short-term load prediction.

Sun et al. (2023) and others assessing the model over several time horizons would have added to the study's high computing expense. This is because, for each time horizon, the researchers had to train and assess the model on a different dataset. They were able to determine which client kinds the model works best and worst for by testing the model on a variety of customer types. To create load prediction models that are more specifically tailored to the needs of various clientele, this data can be utilised.

By concentrating on residential clients Tomar (2023), the study's breadth and the findings' practicality were restricted.

2.2 Comparison of various weather conditions and seasons:

Huang (2023), by testing the model using data from various weather conditions and seasons, the researchers were able to show that the model is resilient to variations in the weather and the seasons. This is crucial for creating realistic load prediction models that work in the real world. The researchers demonstrated that the model can capture the seasonal and weather trends that impact load demand by testing it with data from various seasons and weather conditions. This is crucial for creating precise load prediction models that are applicable to both long- and short-term load forecasting.

Going back to the study by Li et al. (2023), by concentrating on cold load circumstances, the study's breadth and the findings' applicability were restricted.

The benefits of testing models across a variety of time horizons by Zou et al. (2023), clientele, and meteorological circumstances are generally greater than the drawbacks. On the other hand, it's critical to make sure the models are trained on representative data and to be conscious of the increased time and processing requirements.

2.3 Comparison of different types of customers (residential, commercial, industrial):

Revisiting the study by Huang (2023) the researchers were able to show that the model can accurately forecast load for various customer kinds by testing it on a variety of customer types. As a result, the model gains more adaptability and is suitable for a greater variety of load prediction applications.

The researchers Sun et al. (2023) again were able to determine which client kinds the model works best and worst for by evaluating the model on a variety of customer types. To create load prediction models that are more specifically tailored to the needs of various clientele, this data can be utilised.

By concentrating on residential clients by Tomar (2023), the study's breadth and the findings' practicality were restricted. In summary:

In smart grids, deep learning models have proven to be successful in predicting load. A range of deep learning models, including LSTM networks, CNNs, RNNs, Bi-LSTM, MGP, and attention mechanisms, were employed in the experiments examined in this literature review. While some studies employed ensemble learning to boost model performance, the majority of studies used supervised learning to train their models. The research used a range of metrics, such as accuracy, F1 score, M-statistic, MSE, RMSE, and MAE, to assess how well their models performed. According to the majority of the studies, Manju et al. (2023) their deep learning models could forecast load for a wide range of scenarios with accuracy.

All things considered, the research by Friansa et al. (2023) offers compelling proof that deep learning algorithms may be applied to precisely forecast load in smart grids. In addition to investigating the applicability of deep learning models to other smart grid applications, more research is required to build and enhance deep learning models for load prediction.

3 Modified CRISP DM Methodology for Load Prediction

The variation of CRISP DM methodology that was followed skips the business application layer as the study is not end user ready similar to (Kamilin et al.; 2023). Although the purpose for the research is to bring value for the customers to get the maximum out of their electricity consumption habits, it is not associated directly with any businesses. The information includes daily readings of the energy use over time for three houses (MAC000147, MAC000150, and MAC000151). Apart from this dataset, a third party dataset retrieved from the Darksky[1] API was also integrated to the original dataset.

[1]https://support.apple.com/en-us/102594

This API enabled us to add additional features to the data that could provide correlation between energy usage patters and seasonal trends in weather.

3.1 Data Acquisition

The energy consumption data was obtained from the open data source provided by the Government of London(Networks; 2023)[2]. The following variables are present in the data:
LCLid: A distinct identifier for every dwelling
day: Date of the reading recorded
energy_mean: The average daily energy usage
energy_max: The daily maximum energy use
energy_min: The daily minimum energy use

3.2 Data Preprocessing and Data Selection

To deal with missing values and inconsistent data, the data underwent preprocessing. After removing rows where energy_mean had missing values, the day variable was formatted as a datetime. The readings for energy usage were transformed into numerical numbers.

The houses with the most data points MAC000147, MAC000150, and MAC000151 were chosen for additional examination. Here on, the three houses are referred as A, B and C respectively in the report. Each of these houses contains more than 1,000 data points for statistical analysis. Preprocessing was done on the data to address missing values, data discrepancies, and to combine weather and energy consumption data. An outline of the preprocessing stages is provided below:
Eliminated rows that contained missing energy_mean values. The energy consumption readings were converted to numeric values and the day variable was formatted as a datetime. The day column was used to establish a datetime index for each consumption dataset. The resample() function was used to resample the weather data to a 30-minute period. For numerical columns, missing values were interpolated; for categorical columns, they were forward-filled.

Since the icon and summary columns were unnecessary given the other variables, they were removed from the weather data. The meteorological and consumption datasets' datetime indexes were reset.Using the day column to match timestamps, each consumption dataset was left-joined with the meteorological data.

3.3 Exploratory Data Analysis

Following the preprocessing of the data, an exploratory data analysis was carried out to identify the features of the energy use data. Appropriate visualisations, like box plots or histograms, were used to analyse the distribution of energy usage. To find any patterns or seasonality, the patterns of energy use across time were examined. In order to comprehend how the weather affects energy usage, the correlation between energy consumption and weather variables such as temperature and humidity was examined.

Apart from the preprocessing stages stated before, the subsequent operations were carried out: In order to describe and identify UK public holidays according to their unique occurrence criteria, a custom calendar class called UKHolidayCalendar was developed.

[2]https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households

To indicate if the associated date was a holiday, an isholiday column was added to each consumption dataset. Each consumption dataset now has a season column that assigns the date to one of the four seasons—winter, spring, summer, or autumn. Compare energy use on holiday days to non-holiday days to determine how holidays affect patterns of energy consumption. To find possible connections and underlying trends, look into the correlations between energy use and calendar features such the season and holiday indicator.

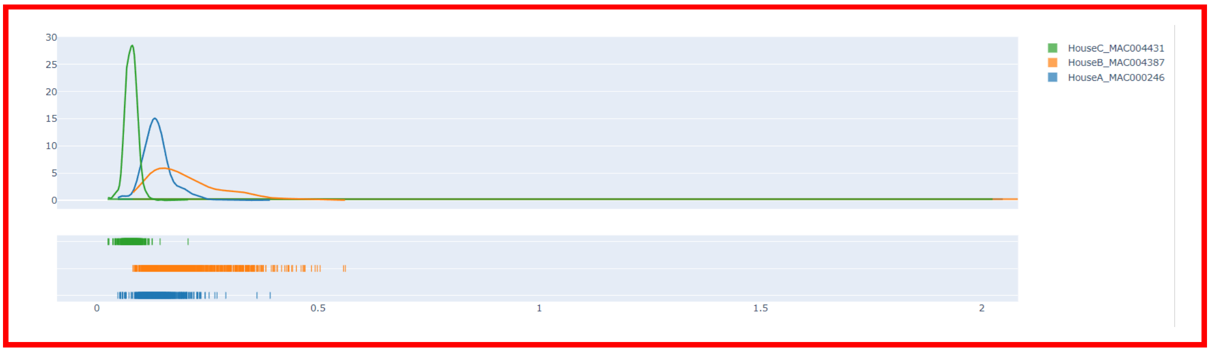

Figure 1: Distribution plot for energy consumption

The distribution of energy consumption over the course of the seasons and holidays was examined as part of the exploratory data analysis to determine any potential effects on patterns of energy usage. Figure 1 displays the three houses' (House A, House B, and House C) distribution of energy use (energy_mean). For all three residences, the distribution is skewed to the right and has a longer tail pointing towards greater energy usage levels. This suggests that while most days have lower energy usage, there are a few days with relatively high energy consumption. The energy consumption of each dwelling is displayed as the minimum, maximum, median, and interquartile range (IQR) in a box plot. The interval between the data's 25th and 75th percentiles is known as the IQR. House C has the greatest median energy usage, followed by House B and House A, according to the box plot. House C has the broadest IQR as well, suggesting that this home's energy use is more erratic.

For all three dwellings, the energy consumption distribution is biased to the right, with a longer tail pointing towards greater energy consumption values. This suggests that while most days have lower energy usage, there are a few days with relatively high energy consumption. The houses with the greatest median energy use are House C, House B, and House A in that order. House C has the broadest IQR, indicating that this home's energy use is more variable.

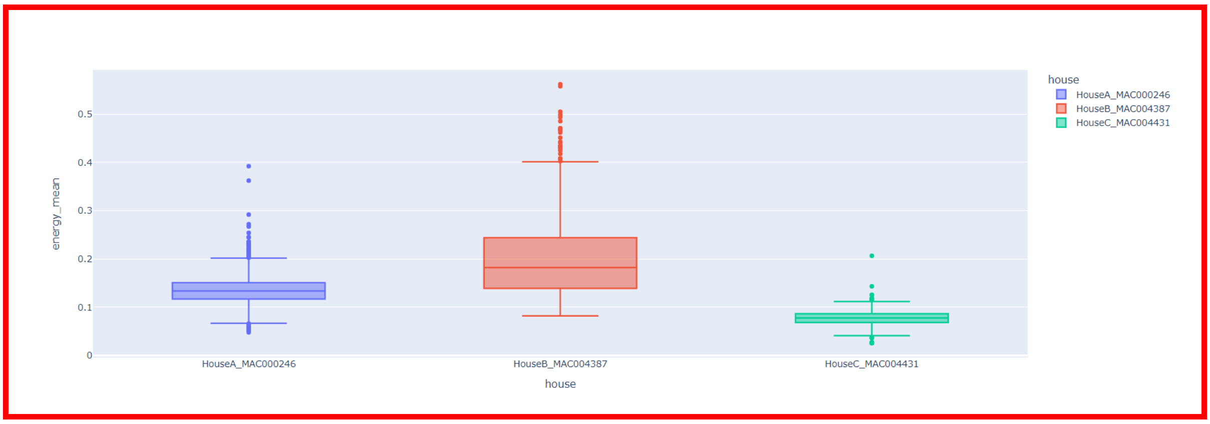

Figure 2: Energy consumption for top 3 consumers House A, B and C respectively

Figure 2 illustrates how the three households' energy usage fluctuates with the seasons. Nevertheless, each residence experiences the seasonal change to a different extent. In House A, for instance, there is comparatively minor seasonal variation in energy consumption, and the median energy consumption is consistent throughout all four seasons. House C, on the other hand, exhibits a significant seasonal fluctuation in energy consumption, with the median energy consumption peaking in the summer and falling in the winter.

Seasons and dwellings also have an impact on the IQR. In all four seasons, for instance, House A has a comparatively narrow IQR, suggesting that its energy consumption is less variable. On the other hand, House C's summer time IQR is comparatively wide, suggesting that this is the season with more fluctuations in energy usage.

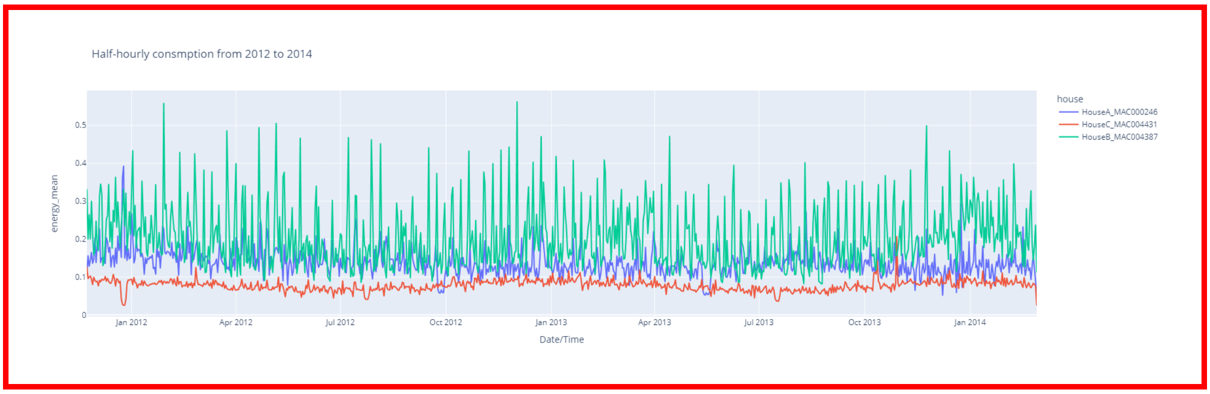

Figure 3: Skewness patterns for the 3 houses A, B and C

Three houses (House A, House B, and House C) are represented in Figure 3 with their half-hourly energy use from 2012 to 2014. Each house's unique energy consumption trends are displayed in the plot. Over the course of the period, House A shows a generally stable pattern of energy usage with a few small variations. This study (Han et al.; 2023) implies that the residents of this home have rather consistent energy-use patterns.

Energy use in House B varies more noticeably, with peaks and troughs happening on a regular basis. The use of appliances like washing machines and dishwashers or the presence of people at specific hours of the day or week could be to blame for this.

There are notable variations in House C's energy usage as well, with summertime seeing the largest usage. The usage of air conditioning during this period could be the cause of this.

The houses with the greatest median energy use are House C, House B, and House A, in that order. Numerous variables, like the size of the home, the number of people, and the kind of appliances utilised, could be to blame for this. All things considered, the plot offers a useful summary of the three residences' patterns of energy use. Additional research into the variables influencing energy consumption can be guided by the identified variances in consumption patterns and median usage. All three of the homes use more energy in the winter, most likely because they are heating their homes. During the evening, the three houses' energy use drops. The house with the most summertime energy usage increase is House C. The house with the least fluctuation in energy use over the course of the week and day is House A.

4 Design Specification

For this study the dataset not just consisted of the data provided from the original source but also a custom python class that helped us map out UK holidays(Colman-Goff; 2023). The study quite similar to the one by (MA et al.; 2023) combined these two aspects of the data with the third element being the weather data which was aggregated from a third party. This gave more scope to work, than just the time series data from energy providers. This provided a greater context into the load prediction domain and also expands the scope for the neural networks implemented to learn from. The abstract holiday class helped us define all the possible holidays that had been declared for the UK which helped us in insights further.

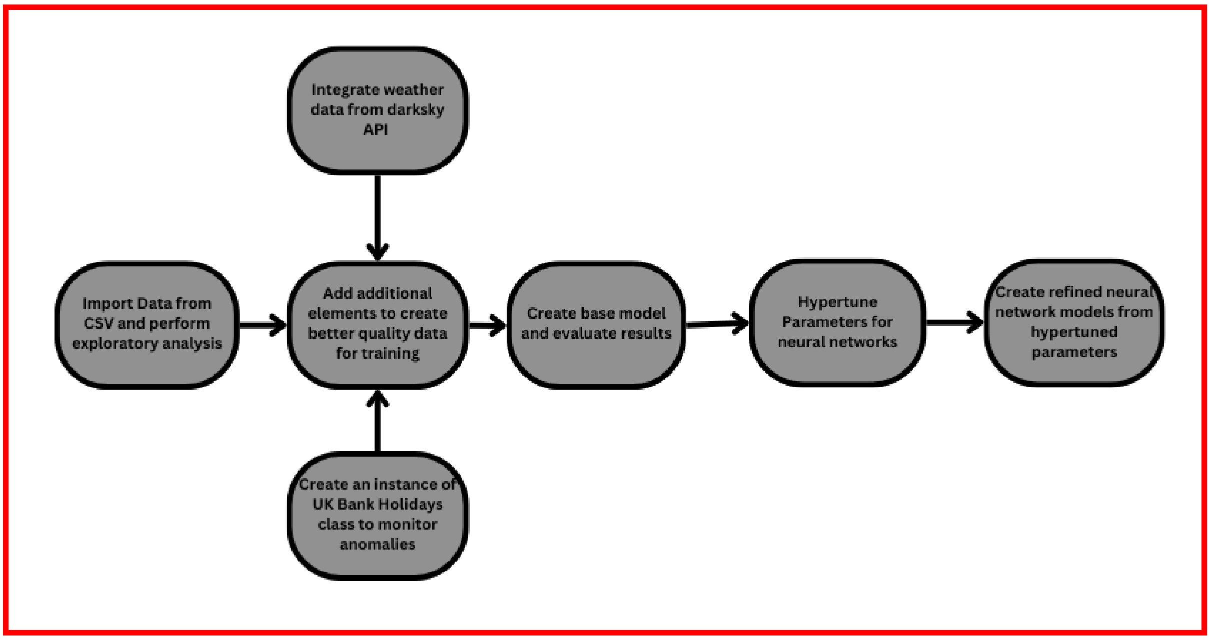

Figure 4: Project Design Process Flow

5 Implementation

The timeseries data that was available since the beginning was not sufficient for quality predictions by the neural network models in place. Thus, additional features from the weather API darksky(Support; 2023) were added to the dataset to provide an additional layer of depth to the existing features. This established some correlations between existing data in its own time frames.

5.1 Feature Engineering

Feature engineering approaches were used on the data to improve the statistical models' prediction power. This involved representing temporal features as sinusoidal and cosinusoidal functions and encoding category features using one-hot encoding.

Encoding of Categorical Features: With one-hot encoding, categorical features like season and precipType were encoded. By producing distinct binary features for every category, this method by Zhang et al. (2022) efficiently converts the categorical data into numerical representations that the statistical models can use.

Temporal Feature Representation: The temporal features 'hour','minute', 'day','month', and 'week_day' were converted to cosinusoidal and sinusoidal representations. This method encodes the features as periodic functions of time, capturing the cyclical structure of temporal data.

Rationale behind Feature Engineering: In this analysis, feature engineering accomplishes multiple goals:

Feature engineering can increase the statistical models' prediction accuracy by changing categorical features and better reflecting temporal features. Simple linear features might not be able to sufficiently capture non-linear interactions between temporal features and energy usage. These relationships can be captured via sinusoidal and cosinusoidal representations. By choosing a subset of pertinent binary features, one-hot encoding has the ability to both raise and decrease the dimensionality of the data. By making temporal aspects easier to understand, sinusoidal and cosinusoidal representations help to better understand how these features affect energy use.

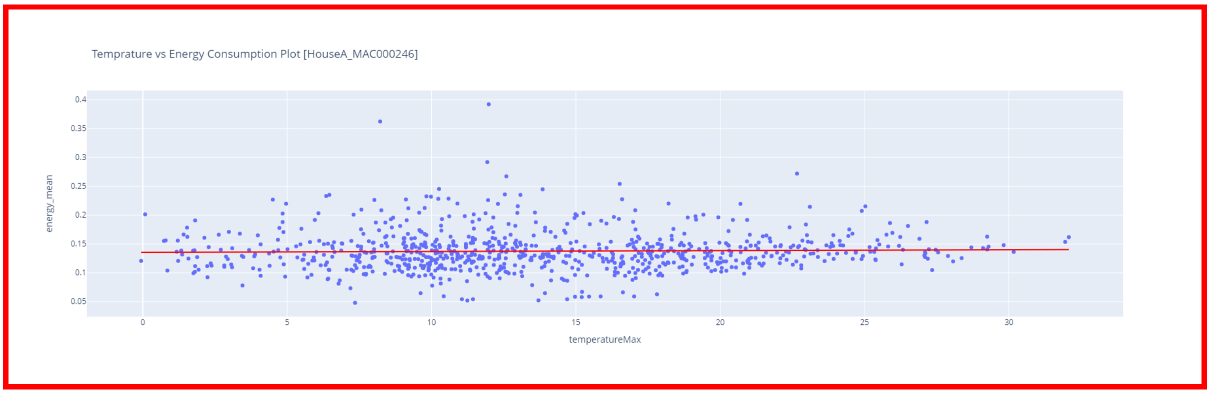

Figure 5: Relation between Energy Consumption and Temperature for House A

The following conclusions can be drawn from Figure 5:

The energy use and temperature are negatively correlated. The red trend line's declining slope serves as an indicator of this and The amount of energy used drops with increasing temperature. The irregularities in the data are indicated by the dispersed data points around the trend line. This shows that energy usage is influenced by a variety of factors in addition to the general trend of reduced energy consumption with rising temperatures. The data does not seem to include any noteworthy outliers. This implies that the information is largely trustworthy and consistent. The negative association in similar studies (Zhao et al.; 2022) may indicate that during colder months, the household utilises more energy for heating. Lower energy usage results from less demand for heating as the temperature rises.

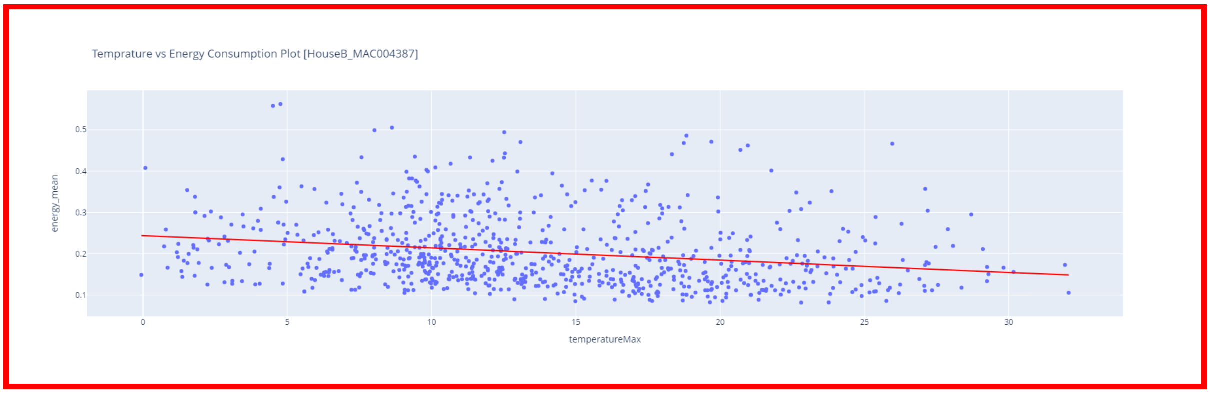

Figure 6: Relation between Energy Consumption and Temperature for House B

Following conclusions are derived from Figure 6: The figure clearly displays a negative trend, meaning that energy consumption falls with rising temperatures. This implies that when the temperature is more agreeable, the household might use fewer heating or cooling appliances. The plot's dispersed data points indicate that the data are variable. Other factors that affect energy consumption, like variations in usage patterns or appliance efficiency, may be the cause of this. The line that best fits the data is indicated by the red trend line. The trend line indicates a steady decline in energy use with rising temperature in spite of the dispersed data points.

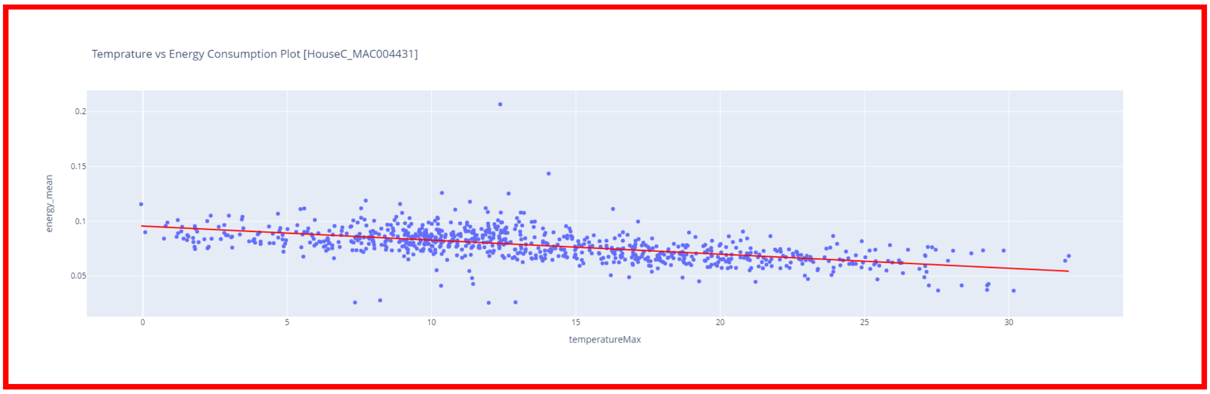

Figure 7: Relation between Energy Consumption and Temperature for House C

The scatterplot for house C as seen in Figure 7 depicts the following observations: The graphic clearly displays a negative trend, meaning that energy consumption falls with rising temperatures. This implies that when the temperature is more agreeable, the household might use fewer heating or cooling appliances. The plot's dispersed data points indicate that the data are variable. Other factors that affect energy consumption, like variations in usage patterns or appliance efficiency, may be the cause of this. The data's linear regression line is shown by the red trend line. The trend line indicates a steady decline in energy use with rising temperature in spite of the dispersed data points. The main variables affecting energy usage in House C are humidity, seasonality, cloud cover, precipitation, and wind speed.Since there is a strong positive correlation between humidity and energy usage, controlling indoor humidity levels may result in significant energy savings as observed. The findings suggest that energy conservation efforts should prioritise optimising heating and lighting throughout the autumn, winter, and cloudier months due to the positive associations with seasonality and cloud cover. The favourable relationships found between wind speed and precipitation suggest that weatherproofing techniques may be able to lower the energy used to prevent heat loss. It appears that energy usage tends to decrease with rising temperatures based on the negative correlations found with temperature factors.

5.2 Splitting the Data into Training and Testing Sets

The data for each house (A, B, and C) was initially divided into training and testing sets. The train_test_split() method from the scikit-learn(Fabian Pedregosa and Michel; 2023) library is used for this. The testing set is used to assess the models' performance, and the training set is used to train the machine learning models. To prevent the data from being shuffled prior to splitting, the shuffle argument is set to False. This guarantees that in both the training and testing sets, the temporal relationships between the data points are maintained. Eighty percent of the data is utilised for training and twenty percent is used for testing, according to the test_size parameter, which is set to 0.2.

Scaling the Data: Next, a MinMaxScaler was used to scale the data for each house (A, B, and C). The purpose of this is to normalise the data to a shared range of 0.01 to 1. The machine learning models' performance may be enhanced as a result. Both the training and testing data are transformed using the fit_transform() method, which fits the scaler to the training set.

Reshaping the Output Variable: In the end, a 2D array was created out of the output variable for each house (A, B, and C). The reason for this is that the machine learning models anticipate that the output variable will have this structure. The output variable is reshaped into a 2D array with one row representing each data point and one column representing the output value using the reshape method.

Dimensionality Reduction using (PCA)

The training and testing data for each house (A, B, and C) were then subjected to Principal Component Analysis (PCA). Using as much of the original data as feasible, PCA is a dimensionality reduction approach that converts a high-dimensional dataset into a lower-dimensional representation. By lowering the number of features the models must take into account, this can help machine learning applications by increasing the models' performance and lowering their computational complexity.

The PCA constructor has the n_components parameter set at 0.95. This indicates that 95 percent of the variance in the original data will be captured by the lower-dimensional representation created by the PCA. This method of applying PCA is popular because it strikes a compromise between information preservation and dimensionality reduction.

The PCA model is fitted to the training set and both the training and testing sets are transformed using the fit_transform method. In other words, the training and testing data are converted into the lower-dimensional representation by means of the PCA model, which has been trained on the training data.

A data preparation approach called the sliding_window method transforms a time-series sequence into a format that may be used for time series forecasting. It accomplishes this by dividing the input sequence into overlapping windows, each of which serves as a forecasting model input sample. As a result, the time series' temporal patterns and correlations can be learned by the model. For every house (A, B, and C), the data was preprocessed using the sliding window technique. In time series forecasting, the sliding window technique is a popular method for preparing data. It entails splitting the input sequence into overlapping windows, each of which serves as the forecasting model's single input sample. To create a single DataFrame, the input features and output values for each house were first concatenated. This makes it possible to manipulate the data during the sliding window procedure more easily. Next, each DataFrame is subjected to the sliding_window function, which creates the input and output sequences needed for testing and training.

The function sliding_window requires two arguments:

data_array: The DataFrame for a particular house that has the input features and output values.

window_size: The total number of time steps that every window contains. Since the window size is set to 12 in this instance, each input sample is made up of the data from the preceding 12 time steps.

The function sliding_window removes overlapping panes of data by iterating through the data_array. It generates an output target (the value of the output variable at the end of the window) and corresponding input sample (containing the input features) for each window. Following that, these input targets and samples are appended to distinct lists, which are ultimately transformed into NumPy arrays and returned.

The time series forecasting models are then trained and assessed using the final input and output sequences for every home. By capturing temporal patterns and relationships within the time series data, the sliding window technique can assist forecasting models become more accurate.

5.3 Defining the Multivariate CNN Model for House A

A function called multivariateCNNmodel_A, which builds a convolutional neural network (CNN) model with a sequential architecture—that is, layers are stacked one on top of the other—that is especially designed for House A. The temporal patterns and linkages present in the time series data for House A are intended to be captured by the model architecture.

An HyperParameters object called hp is the input for the multivariateCNN_model_A function. This object supports the use of methods such as random search and Bayesian optimisation for hyperparameter tuning.

The function defines four hyperparameters: Filter_Size: The convolutional layer's number of filters is managed by this hyperparameter. The 64 and 128 filter options are investigated by the function. Pooling_Size: The max-pooling layer's pooling window's dimensions are set by this hyperparameter. Four alternatives are evaluated by the function: 2, 3, 4, and 5. Kernel_Size: The convolution kernel size utilised in the convolutional layer is specified by this hyperparameter. Four possibilities are evaluated by the function: 2, 4, 6, and 8. Number_of_Neurons: The number of neurons in the dense layers is managed by this hyperparameter. Three choices are investigated by the function: 30, 50, and 80 neurons.

The model architecture consists of the following layers:

Convolutional Layer: This layer extracts local patterns and features from the input data by applying convolutions. Hyperparameters include the number of filters, pooling size, and kernel size.

Max Pooling Layer: By using max pooling and choosing the maximum value within each pooling window, this layer lowers the dimensionality of the data. The size of the pooling is a hyperparameter.

In order to prepare the data for the dense layers, the flatten layer converts it into a one-dimensional vector.

Dense Layers: These layers add nonlinearity and change the data to fit the output layer's appropriate representation. A hyperparameter is the quantity of neurons in each dense layer.

The output layer generates the final predicted number, which is House A's energy consumption at a specific time step.

The model's performance is assessed by compilation using the Adam optimizer, a mean squared error loss function, and the mean squared error measure.

House A Hyperparameter Tuning Results

Using the Hyperband tuner, the CNN model was adjusted for House A's hyperparameters. The tuner uses the mean_squared_error measure to determine the optimal set of hyperparameter values. Thirty trials totaling a maximum of fifty epochs were completed by the tuner. During the search process, the best mean squared error obtained was 0.00017605. This shows that the tuner was successful in locating a set of hyperparameter values that considerably lower the model's prediction error.The hyperparameter tweaking procedure took 00h 01m 17s in total. This suggests that, in a fair amount of time, the Hyperband algorithm effectively examined a large range of hyperparameter values.Overall, the results of the hyperparameter tuning show how well the Hyperband method optimised the CNN model for House A. The model's ability to predict House A's energy consumption is supported by the attained mean squared error of 0.00017605.

Chosen Hyperparameters for Residence A:

The tunerA object's recommended hyperparameter setup was obtained. During the hyperparameter search procedure, all assessed configurations' performance and hyperparameter values are stored in the tunerA object. It is then determined whether the tunerA object has finished at least one trial if the best_hps_list is not empty. The best hyperparameter configuration (best_hps) is taken from the list and printed to the terminal if it is not empty. The code is able to successfully extract the optimal hyperparameter configuration in this instance. The hyperparameters found are:

Size of filter: 128 Size of Pooling: 2 There are fifty neurons. Size of Kernel: 6

The optimal set of values discovered by the tunerA object to reduce the mean squared error metric for the CNN model of House A is represented by these hyperparameters. With the use of these data, one can better comprehend the CNN model's ideal configuration and train a new model with the optimised hyperparameters for enhanced functionality.

The CNN model for House A was trained during the optimal epoch, which is the one in which the model gets the highest validation accuracy. The optimal epoch in this instance is 47. This indicates that training the model for 47 epochs yields the best results. By using this data, overfitting can be avoided and the model's ability to generalise can be enhanced. The model will perform better on unseen data and is less likely to memorise the training set if training is stopped at the optimal epoch.

5.4 Assessing Performance and Retraining the Model with Optimal Hyperparameters:

Next, the best hyperparameters found by the tunerA object (best_hps) are used to construct a new CNN model. By doing this, the mean squared error measure is guaranteed to be minimised throughout model training. The model is then retrained by the code using the given best_epoch, in this case 47. This guarantees that in order to avoid overfitting and enhance generalisation performance, the model is trained for the ideal amount of epochs.

The test data for House A is then predicted using the retrained model, and its performance is assessed by calculating the mean squared error, or MSE, between the predicted and actual values. A very low prediction error is achieved by the retrained model with appropriate hyperparameters, as indicated by the observed RMSE value of 0.0370. This illustrates how well the hyperparameter tweaking procedure worked and how well the model could forecast House A's energy usage.

6 Evaluation

Considering the neural network models in place they were evaluated based on the following metrics: R squared: R squared indicates how well the actual results can be explained by the model's predictions. A greater explanatory ability is indicated by a larger value.
MAPE (Mean Absolute Percentage Error): The average percentage error in predicting the target variable is provided by MAPE. Greater precision is suggested by a lower MAPE.
Explained Variance Score: Explained variance score indicates the extent to which the model explains the variability observed in the data. Greater score indicates that more variance is explained by the model.
Root Mean Squared Error (RMSE): The average magnitude of the model's errors is measured by RMSE. Better precision is indicated by a smaller RMSE.
Mean Absolute Error (MAE): The mean absolute error (MAE) yields the average error magnitude without directionality. Greater precision is suggested by a lower MAE.

6.1 Comparision of Developed Models

Based on the data for the three houses three different neural network models were developed to determine which one is the most effective for load prediction. As the data

House	MAPE (%)	RMSE	MAE	Explained Variance Score	R squared
House A with CNN	28.0559	0.0370	0.02940	0.0389	-0.0911
House B with RNN	31.8411	0.0945	0.0657	0.0835	-0.209
House C with LSTM	108.438	0.0813	0.0766	-3.0634	-35.305

Table 1: Performance Metrics for Different Houses

training and testing split is randomized the results can vary on every iteration the code runs. That is why the results in the report and the results upon the latest execution may not always be the same. The most recent results at the time of report can be seen in the Table 1 below:

7 Conclusion

This study explored 3 neural network feed forward models namely CNN(Convoluted Neural Network), RNN(Recurrent Neural Network) and the LSTM(Long Short Term Memory) as mentioned in the research objectives section. From the evaluation metrics it can safely be concluded that the CNN model performed best with the limited categorical data that was worked upon. The other models did not perform satisfactorily and were not suited for the task. Hence this study concludes that the CNN deep learning model is the best suited for load prediction with limited features that do not encroach upon the end user's privacy.

7.1 Future Work

The scope of this research can further be expanded by working on more diverse data which is not limited to time series features. The limitations of this research are such that it only explores time series features in load prediction apart from the custom categorical features. Thus with the load prediction methods currently in place being not completely secure in terms of data privacy more focus could be given to diverse features that cannot be traced back to the end users and which can help in creating better models in deep learning.

References

Chen, G., Zhang, H. and Song, H. (2023). A novel neural network structure for electricity load prediction, *2023 3rd Power System and Green Energy Conference (PSGEC)*, pp. 439–443.

Colman-Goff, L. (2023). How to automate bank holidays using python, `https://medium.com/analytics-vidhya/python-how-to-automate-bank-holidays-767de0d42d44`. [Online; accessed December 14th, 2023].

Fabian Pedregosa, Gael Varoquaux, A. G. and Michel, V. (2023). scikit-learn: Machine learnign in python, `https://scikit-learn.org/stable/`. [Online; accessed December 14th, 2023].

Friansa, K., Pradipta, J., Haq, I. N., Utama, P. H. K., Wasesa, M. and Leksono, E. (2023). Enhancing hvac electricity load prediction accuracy using bi-lstm method based on daily dataset, *2023 8th International Conference on Instrumentation, Control, and Automation (ICA)*, pp. 92–96.

Han, B., Xie, Z., Luo, X., Wu, M., Wang, Y. and Hong, L. (2023). Lstm-based strategy for customer directrix load prediction, *2023 6th International Conference on Energy, Electrical and Power Engineering (CEEPE)*, pp. 1433–1437.

Huang, J. (2023). Research on load prediction method based on deep learning in smart grid, *2023 International Conference on Smart Electrical Grid and Renewable Energy (SEGRE)*, pp. 270–275.

Kamilin, M. H. B., Yamaguchi, S. and Ahmadon, M. A. B. (2023). Radian scaling: A novel approach to preventing concept drift in electricity load prediction, *2023 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 1–4.

Li, S., Zhang, W. and Wang, P. (2023). Ts2arcformer: A multi-dimensional time series forecasting framework for short-term load prediction.
URL: *https://research.ebsco.com/linkprocessor/plink?id=7e2972b6-a8a6-3b8f-a6a0-06de08021e6e*

MA, W., Duan, X., Li, L., Sun, Z., Li, H. and Yang, S. (2023). Short-term load prediction of cnn-bilstm based on gwo optimization, *2023 International Conference on Power System Technology (PowerCon)*, pp. 1–5.

Manju, J., Manjula, R. B. and Dash, R. (2023). An efficient artificial bee colony based optimized model for load prediction in iot enabled smart grid, *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, pp. 129–136.

Networks, U. P. (2023). London data store, `https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households`. [Online; accessed 14th December, 2023].

Sun, J., Wang, T., Ding, L. and Jing, C. (2023). Short-term electricity load prediction based on machine learning and improved lstm model, *2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)*, pp. 1003–1007.

Support, A. (2023). Darksky users apple support, `https://support.apple.com/en-us/102594`. [Online; accessed 14th December, 2023].

Tomar, A. (2023). Optimized multi-model machine learning approach for residential load prediction, *2023 5th International Conference on Energy, Power and Environment: Towards Flexible Green Energy Technologies (ICEPE)*, pp. 1–6.

Zhang, X., Lin, Y., Wang, M., Yang, D. and Zhou, X. (2022). Research on power load prediction and application based on load derivation, *2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, pp. 73–76.

Zhao, J., Zhang, X., Wang, H. and Mu, D. (2022). Load trend prediction analysis based on lstm and sliding time window, *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Vol. 5, pp. 1239–1242.

Zou, Q., Wang, L., Xue, H., Feng, X., Qiao, B. and Dong, Y. (2023). Random forest algorithm based dynamictraining set for cold load prediction, *2023 8th Asia Conference on Power and Electrical Engineering (ACPEE)*, pp. 165–172.