CAPSTONE PROJECT BY TEAM P

SHORT-TERM ELECTRICITY DEMAND FORECASTING IN NSW UNDER HIGH TEMPERATURES

Jamie Cryan (z5376188), Nasrin Kiani (z5370222), Jane Melencio (z5374932), David Rovere (z5388545)

School of Mathematics and Statistics
UNSW Sydney

October 2023

Submitted in partial fulfilment of the requirements of the capstone course ZZSC9020

Plagiarism statement

I declare that this thesis is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere.

I acknowledge that the assessor of this thesis may, for the purpose of assessing it:

- Reproduce it and provide a copy to another member of the University; and/or,
- Communicate a copy of it to a plagiarism checking service (which may then retain a copy of it on its database for the purpose of future plagiarism checking).

I certify that I have read and understood the University Rules in respect of Student Academic Misconduct, and am aware of any potential plagiarism penalties which may apply.

By signing this declaration I am agreeing to the statements and conditions above.

Signed: ______________________ Date: 7 October 2023

Signed: ______________________ Date: 6 October 2023

Signed: Kiani Date: 6 October 2023

Signed: ______________________ Date: 6 October 2023

Abstract

Accurate forecasting of demand is vital for a reliable and economically viable electricity network. With temperatures in NSW predicted to rise due to El Niño, there will be increased importance in the future of accurate demand forecasting during high temperatures. Accurate forecasting under high temperatures is especially important from a network reliability perspective, since it is when demand is at its peak. Despite this, the short-term demand values in NSW that are published by AEMO (Autralia's electricity regulator) perform poorly under these conditions. While there is extensive research on demand forecasting generally, there is little research that focuses specifically in the domain of high temperature observations.

This report outlines the development of a model that can be used to accurately forecast electricity demand during high temperatures. The model is built for forecasting in NSW and provides both 30-minute and 1-hour forecasts. Data exploration and feature selection plots were used to find the most influential model features, which included demand lags, temperature lags and datetime features. XGBoost, neural networks and LSTM networks were all tuned and compared. Loss functions weighted by temperature were explored as a strategy to make up for the small amount of training data available for high temperatures. The final model selected was a 4-layer fully connected neural network with 50 nodes in each layer. The model was trained with a mean squared error function weighted by temperature squared, which successfully improved forecasting accuracy during high temperatures. This model had much less error than the AEMO model for temperatures above 30°C, with the mean average percentage error dropping from 2.07% to 1.30% for 30-minute forecasts, and from 2.88% to 1.99% for 1 hour forecasts. The model also generalised well, outperforming the AEMO model under most other temperature conditions as well.

Contents

Chapter 1
Introduction

Accurate short-term load forecasting (STLF) models are important for reducing the costs of over or under-bidding supply contracts. This keeps the energy markets efficient and ensures that the network reliability is maintained. There is a chain of effects here - demand for energy drives prices and prices drive generation decisions and so forecasts are also essential for energy supply systems to plan their generation capacity and stability.

Intraday temperatures are strongly correlated with energy demand. With the Australian Bureau of Meteorology forecasting a particularly severe El Niño this year high temperatures are to be expected across the whole of the Australian continent and are expected to become more frequent in the future. High temperatures reduce transmission capabilities due to turbine inefficiencies at high ambient air temperatures. They impact the performance of transmission lines, transformers, circuit breakers, and insulation. Energy demand for air cooling applications increase at the same time. Although energy market operators commonly include climate related variables in their forecasting models, the danger of interruptions to power supply that continue across the world are an indication that these forecast models suffer a drop in reliability during high temperatures similarly to the physical components of the supply network. A review of the load forecast models for the Australian Energy Market Operator (AEMO) reveal an increase in inaccuracies in high temperature conditions. In fact analysis of research into forecast models reveals that more than half of forecast models fail to meet forecast demand under extreme climate events.

This study attempts to provide a model that can be used to better forecast load demand during times of high temperatures. Research indicates that high temperatures can be considered a time when the temperature is greater than or equal to 30^oC. Improving the reliability of short term forecasts during high temperatures will improve pricing efficiency, generation, transmission and network stability. This model would be of use to registered participants in the National Electricity Market (NEM) to assist with accurate pricing of generation time slots for traders or reallocators and to improve capacity planning for network service providers. The model is primarily focused on providing an accurate forecast 30 minutes ahead but it can be used to forecast 1 hour ahead or slightly further with greater accuracy than the existing publicly available forecasts.

Chapter 2

Literature Review

2.1 Overview of Load Forecasting

Electric load forecasting has a long history (De Gooijer and Hyndman, 2006). Statistical models - whether exponential smoothing or auto-regressive integrated moving average (ARIMA) models were used for most of the 20th century. Pattern recognition techniques started to be proposed in the early 1980's (Dehdashti et al., 1982) and although machine learning models were being used in the mid 1990's they were not consistently outperforming ARIMA models for over a decade. More recently, ensemble methods like Extreme Gradient Boosting models (XGBoost) and Artificial Neural Networks (ANN) provide the most accurate forecasts (Raza and Khosravi, 2015) and form the basis of this study.

2.2 XGBoost Models

XGBoost has proven to be a highly effective model for load forecasting in recent years as can be seen from the many studies that have sought to refine the model or to test against when investigating new models(Abbasi et al., 2019), (Behera et al., 2022) . The advantage it has over other boosting ensemble methods is that it avoids over-fitting and complexity by adding a regularisation item to the loss function. It uses a quadratic Taylor expansion to improve convergence and it performs column sampling to speed up operations. The most important hyper parameters of the model can be improved using grid search as shown by Tran et al. (2023).

The XGBoost Regression model has been used by many researchers for load forecasting either on its own (Dong et al., 2023), (Jiang et al., 2023) or in combination with other algorithms – Li et al. (2023) combined using linear regression with a Holt-Winters method. Wu et al. (2022) researched a Particle Swarm Optimisation – XGBoost model. Although XGBoost is a model that is easily deployed with efficient computing time and memory resource usage, it does not always outperform other models. Gokce and Duman (2022) compared the performance of simple regression models, random forest and XGBoost and found that XGBoost achieved similar accuracy to the Random Forest models in their study. However Ibrar et al. (2022) found that combining oversampling with their XGBoosting provided the most accurate forecasts of any models they tried.

In this study the XGBoost model serves as a base model to benchmark the performance of the AEMO forecast and the ANN models defined.

2.3 Artificial Neural Networks and Long Short-Term Models

Neural networks have been shown in many studies to offer improved forecasting accuracy for scenarios where domain knowledge is limited (Bianchi et al., 2017). Recurrent neural networks such as long short-term models (LSTMs) are particularly popular given their ability to retain knowledge from successive timestamps and have been shown to achieve an accuracy of over 99% for ultra short-term electricity demand forecasting (Tan et al., 2020). LSTMs are a special type of recurrent neural network (RNN) which address the gradient disappearance problem that RNN models can easily suffer from but as mentioned above, artificial neural networks had not been used much until the past decade due to their computational demands. The benefits of Moore's Law together with network refinements have made LSTM's very popular models for load forecasting in recent times. Whereas other deep neural networks with multiple layers may require expert knowledge in order to tune their hyper-parameters Manandhar et al. (2023), LSTMs have proven themselves to be flexible and easy to train.

Since 2018 (Bouktif et al., 2018), (Bedi and Toshniwal, 2018) Long Short-Term Memory (LSTM) models have been found to provide more accurate forecasts than other Recurrent Neural Networks (RNN). Other refinements were earlier tried like fuzzy neural networks, wavelet neural networks and fuzzy wavelet neural networks but LSTM models tested by Bedi and Toshniwal (2018) yielded better results. There have been many other studies of the efficacy of LSTM models in the past 5 years which indicate that the model trains well across a broad range of circumstances (Bashari and Rahimi-Kian, 2020), (Son and Kim, 2020), (Adewuyi et al., 2020), (Chen et al., 2022), (Chung and Jang, 2022).

2.4 Data Features

Many studies relied purely on the time series data of load demand itself, but constructing features within the time series helps to identify the non-stationary factors of the demand load patterns. Tan et al. (2020) trained an LSTM using the hour of the day, the month of the year and the day of the week. Fan et al. (2014) also found that for the purpose of forecasting less than one day ahead that time lags up of beyond 15 days were not significant. Further, they found that in addition to lag data, variables of month, day, weekday and meteorological data was needed for successful forecasting. Removal of seasonality from the data has been an essential first step in data preparation for many years whether using principal component analysis, exponential smoothing or an auto-correlation function (Taylor et al., 2006).

Temperatures have the strongest direct but non-linear impact on load demand. After temperatures, working day information seems to be the next most key factor in driving load demand. Importantly Zhang et al. (2013) noted the need to train the model based on working days vs non-working days and to train them based on the temperature, studying the correlation between load from day to day separately for hot and cold seasons. Other weather factors such as rainfall rate, or relative humidity or wind speed were found to have a significant correlation in some geographies but not in others (Aisyah and Simaremare, 2021). In Sydney, Ahmed (Ahmed et al., 2018) noted that the mean daily air temperature is the best indicator relating to energy demand. In addition to this correlation Ahmed found that demand for electricity had a linear relation to the temperature variation from a "balance point". Deviation from this point defined a Cooling Degree Day or a Heating Degree Day. Mean daily air temperature has been used as an indicator of energy demand in other studies too (Jovanović et al., 2015), (Vu et al., 2014).

McCulloch and Ignatieva (McCulloch and Ignatieva, 2017) used a time weighted temperature model which showed excellent forecast results based off just a single temperature for the entire state of NSW. The temperature they used was in Homebush. For our research we are using temperatures based on Bankstown which in the past 5 years is a closer approximation to the population centre of Greater Sydney, the largest urban area in NSW representing about 61% of the population of the NSW electricity demand area.

2.5 Data Sources

Ahmad et al. (2020) noted how the increase in smart meters and monitoring appliances has increased data availability to an unprecedented degree and are a core component of the new smart grids that are increasingly common . However, this data whilst helpful in achieving accurate forecasts that may be useful to building owners or local infrastructure, and they have been used to accurately forecast demand for individual households (Kong et al., 2019). For our purposes we have used data from the Australia Energy Market Operator (AEMO) who are a semi government institution in Australia responsible for the national market as it is hoped that this data will generalise better. Using AEMO data also allows a better comparison against other models as this has been used broadly for forecast models for many years (Liu et al., 2022), (Al-Musaylh et al., 2018), (Zhang et al., 2013), (Ismail et al., 2023), (Clements et al., 2016) among others.

2.6 Performance Measurement

Mean absolute percentage error is used as one of the key metrics because it can be easily compared to other models regardless of the scale of units being measured. This is a very commonly used metric in load forecasting studies (Hong and Fan, 2016) and suggested by Hyndman and Koehler (2006). We also use the root mean square error (RMSE) and R^2. RMSE is scale-dependent and R^2 is commonly used to determine goodness of fit. As a metric however MAPE can tend to favour models that under-forecast rather than over-forecast (Lewinson, 2020). RMSE on the other hand is highly affected by outliers and as such can be compared to MAPE to determine whether the forecast contains large but infrequent errors (Manandhar et al., 2023).

2.7 Conclusions

In summary, the key points that we took from our review are:

- Model flexibility to handle the nonlinear pattern of demand under extreme conditions (Both LSTM's and XGBoost exhibit these capabilities).
- Removal of the cyclical nature of demand data.
- Consideration of non-working days to affect demand.
- Single temperature reading and State-wise load demand data is sufficient for our modelling purposes.
- MAPE and RMSE as standard measures of error rates to allow easy comparison.

Chapter 3

Material and Methods

3.1 Software

In this analysis, **Python** served as the primary programming language since the machine learning libraries were best suited for the model types selected for the study. The following Python libraries were leveraged:

- **Pandas** for data manipulation and processing,
- **NumPy** for performing mathematical operations on our data,
- **Matplotlib** and **Seaborn** for creating a wide range of plots and charts,
- **Pmdarima** for examining stationarity in the data,
- **Scikit-Learn** for machine learning algorithms and model evaluation,
- **Keras** and **keras tuner** for building and training neural networks, and
- **XGBoost** for ensemble learning techniques.

The Python codes were written and executed interactively in **Jupyter Notebooks**, and the platform used to run Jupyter Notebooks was **Google Colab**. These platforms were used since they offered useful features for collaborative programming.

To track changes to the codes in the Jupyter Notebooks over time, the version control system used was **Git**, and the platform where the code repository was hosted is **GitHub**.

3.2 Description of the Data

There were three main data sets used to train and evaluate the models:

- `totaldemand_nsw.csv`, contains electricity demand values (in MegaWatts) for NSW (43.915 MB);
- `temperature_nsw.csv`, contains temperature data from Bankstown airport (8.195 MB); and
- `forecastdemand_nsw.csv`, contains AEMO short term forecasts for power demand in NSW, typically ranging from half an hour to 1.5 days prior. (769.555 MB)

AEMO uses the term 'Total Demand' for various types of demand. In this analysis, the field 'Total Demand' is referred to as the forecasted electricity demand at the Regional Reference Node (RRN). It includes local generation, interconnector imports, and wholesale demand response but excludes local scheduled loads and allocated interconnector losses. The NEM Dispatch Engine (NEMDE) calculates

Total Demand, serving as the starting point for central dispatch, which determines regional prices and dispatch targets for generating units.

A more detailed description of the data is available in the data dictionaries in Appendix B – Data Summaries, and the process of transforming the data so that they become use-able is discussed in the next section.

3.3 Pre-processing Steps

3.3.1 Demand Data

Historical electricity demand data sets were sourced and utilised. The Demand data set comprised of historical half-hourly data spanning from January 2010 to July 2022. This data set underwent a thorough analysis to identify potential data entry errors, including missing, implausible, or duplicate entries. No missing or implausible values were identified. Thirty-nine duplicate entries were discovered and were subsequently removed. Additionally, the number of observations per date was examined to ensure consistency with the expected 48 half-hourly periods. Data for the date 2022-08-01 was removed as only one period (midnight) was recorded.

All data were verified to be in their correct data types. Datetime variables were converted to the datetime data type to ensure proper handling by Python.

Eleven lag variables were created for the forecasting model that represent the last 10 prior values of demand ('DEM - 1' to 'DEM - 10') and the demand recorded 24 hours prior ('DEM - 48'). One lead variable was created ('TOTALDEMAND + 1') that represents the value of total demand an hour into the future, so that both 30-minute and 1-hour forecasts could be made.

3.3.2 Temperature Data

The temperature data set consisted of datetime entries at 30-minute intervals spanning from January 2010 to July 2022, featuring temperature measurements in degrees Celsius along with corresponding measurement locations. The data set included 19 temperature outliers recorded at -9999°C, necessitating removal or replacement. Additionally, upon inspection, it was determined that although there are no missing values (NaN), several expected datetime indexes are absent, suggesting the absence of certain data rows. Linear interpolation methods were utilised to fill in these missing values before model training.

Four lag variables were created that represent the last 4 prior values of temperature ('TEMP - 1' to 'TEMP - 4').

3.3.3 Forecast Data

The forecast data set consisted of datetime entries at 30-minute intervals spanning from January 2010 to July 2022. Each datetime entry was repeated 71 times, corresponding to forecasts made at different time horizons. No null values or duplicates values were present in the data set.

The forecast data set was filtered for forecasts that had a period ID of 1 or 2, since these corresponded to the 30-minute and 1-hour predictions that could be used for comparison against the models developed during this study. This information was then processed so that it was in the same form as the demand dataset, having a 'FORECASTDEMAND' column containing the 30-minute forecasts, and the 'FORECASTDEMAND + 1' column containing the one hour forecasts.

3.4 Data Cleaning

3.4.1 Missing Values

To verify that data was fully recorded every half hour, a datetime series was created and joined with our data sets. This analysis revealed no missing total demand values (in 30-minute intervals), but there were 3,737 missing forecast demand values and 689 missing temperature values.

To ensure consistent datetime records, these missing values were populated using linear interpolation with the following steps:

1. A sequence of datetime values at 30-minute intervals starting from 1 January 2010 to 1 August 2022 was generated.
2. The forecast demand data was merged with the datetimes dataframe using a right join ensuring timestamps for all periods. Missing demand data was assigned NaN values in the merged dataframe.
3. Datetimes with missing forecast demand data were identified.
4. Temperature data was merged with the datetimes dataframe using a right join.
5. Datetimes with missing temperature data were identified.
6. Missing temperature values were populated in the new temperature data dataframe using linear interpolation based on the surrounding temperature values.

3.4.2 Outliers

Summary statistics and data visualisation (both univariate and bivariate) were used to identify outliers in the data set and to gain insights into the distribution of the data. The only significant outliers found were in the temperature data set. Nineteen temperatures recorded at -9999^{o}C were found and removed from the temperature data set.

3.5 Assumptions

Key assumptions in the analysis based on the available data were:

- Temperature data gathered from Bankstown airport was representative of the temperature conditions for the NSW region.
- Missing temperature values could be reasonably estimated through linear interpolation.
- Forecast demand data provided was reliable and complete. Information from this forecast demand data set was only considered when comparing the models' performance against AEMO's forecasts.

3.6 Modelling Methods

XGBoost and ANN / LSTM were found to exhibit flexibility to handle the nonlinear pattern of demand under extreme conditions. Hence, these models were selected for this study.

XGBoost is an ensemble learning method that combines the predictions of multiple weaker models (decision trees) to create a stronger, more robust model. This ensemble approach was selected to reduce bias and variance in predictions. It also provides a feature importance score, which helps highlight the factors or features which are most influential in predicting electricity demand.

LSTM networks are designed to handle sequential data, making them well-suited for time series forecasting tasks. This type of recurring neural network was selected as electricity demand data is inherently sequential, with dependencies on historical time steps, and LSTMs can capture these temporal patterns effectively. ANNs, while not specifically designed for sequential data, offered a simpler neural network structure that could be trained much faster than LSTMs.

More details on the analysis and evaluation of these modelling methods are discussed in Section 5 – Analysis and Results.

Chapter 4

Exploratory Data Analysis

After the initial data cleaning and pre-processing, an exploratory data analysis (EDA) was conducted on all primary data sets used in this research. The objective of this exercise was to acquire a deeper understanding of the underlying data structure, their distributions, and the potential presence of errors or outliers. Additionally, the EDA facilitated the identification of trends and relationships between variables across different data sets, enabling the formulation of appropriate and well-justified assumptions before proceeding with modelling.

The source code for the results in this section can be accessed on our GitHub Repository. Please refer to Appendix 7.1 Codes – Data Cleaning and Exploration. In addition to the EDA shown in chapter 4, this file also outlines the data visuals and summaries that were used to clean and process the data as described in sections 3.3 and 3.4.

4.1 Inspecting the Processed Data

After the data was cleaned, raw data was visualised to confirm that there were no longer any univariate or bivariate outliers in the processed data. Figure 4.1 demonstrates there are no significant outliers remaining in the temperature or demand columns of the data and also confirms that there is a significant relationship between temperature and demand that needs to be considered when modelling.

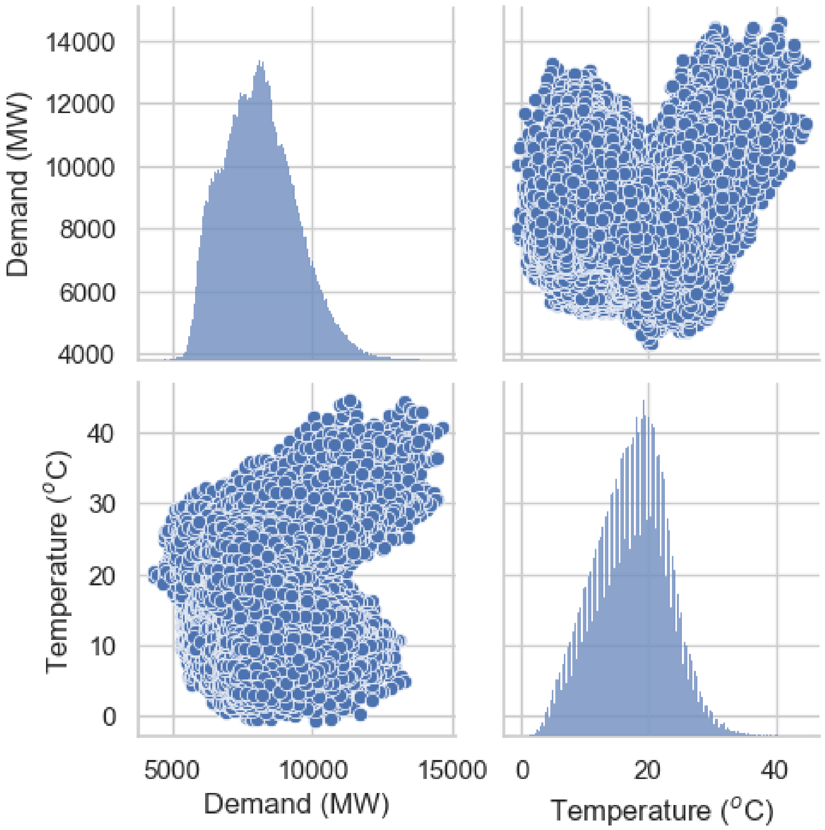


Figure 4.1: Pairwise Relationship between Demand and Temperature

Figure 4.2 shows that there are no obvious missing values across the date range of the processed data. The periodic nature of the plot shows that datetime features would have to be considered in the model.

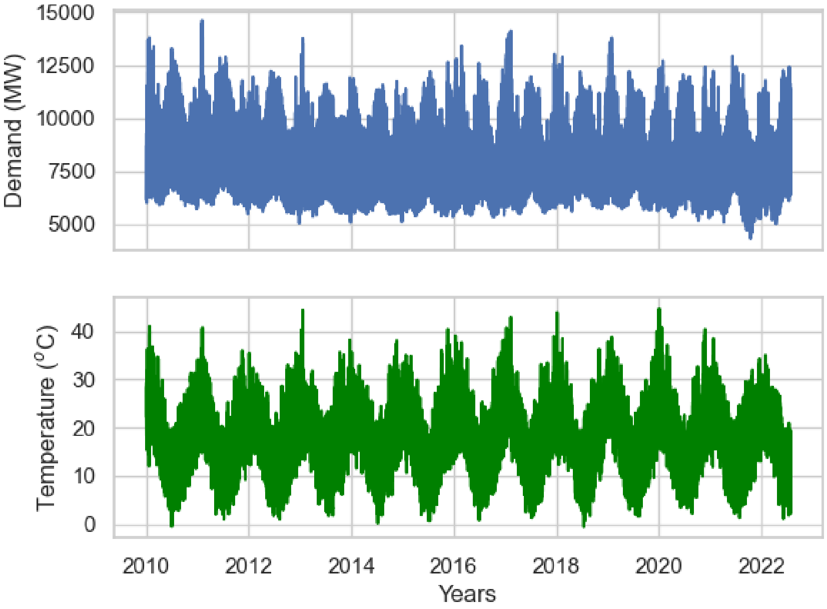


Figure 4.2: Data Spread of Total Demand and Temperature

4.2 Exploring the Relationship between Features and Demand

4.2.1 Demand Throughout the Day

Figure 4.3 shows how electricity demand varies throughout the day. There is a clear pattern demonstrated that indicates that it would be an important feature to include in the model. The dip in demand in the middle of the day is especially important, since it is primarily caused by solar PV usage which is not available in our training data.

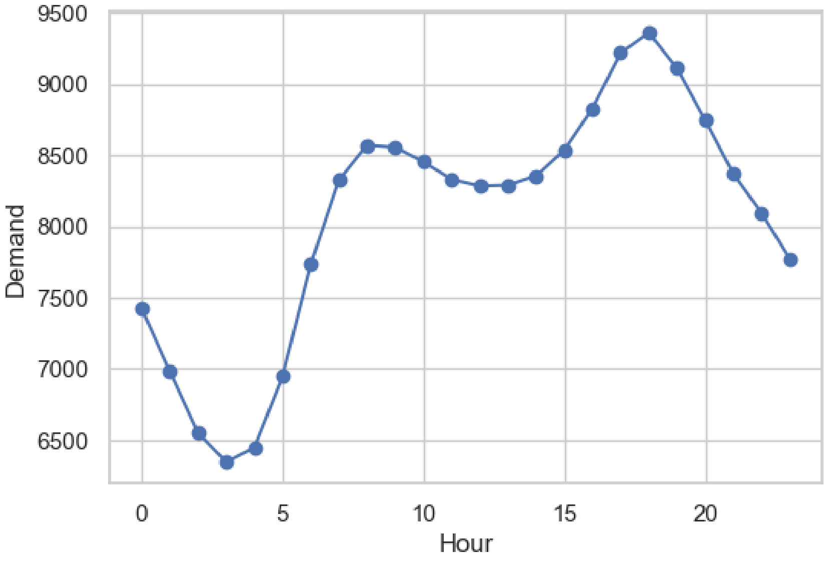


Figure 4.3: Hourly Mean of Demand

4.2.2 Demand on Different Months of the Year

Figure 4.4 shows the average electricity demand by month using a box plot. This helps to understand how electricity demand varies throughout the year, and that there is a seasonal pattern as seen by the increase in demand during the summer and winter months.

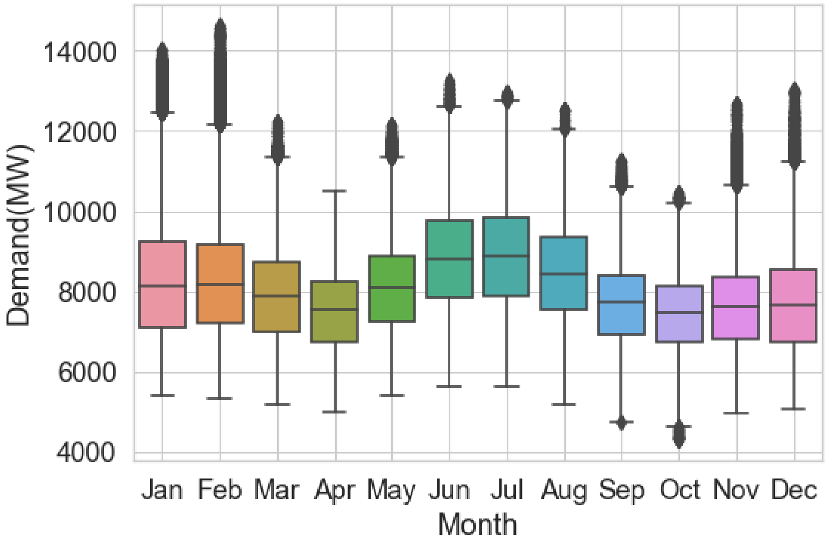


Figure 4.4: Average Yearly Demand by Month

4.2.3 Demand on Working Days and Non-Working Days (Weekends and Public Holidays)

Electricity demand may be influenced by weekends and public holidays due to distinct usage patterns that deviate from the normal working hours. This variation was considered a potential feature for the models.

Figure 4.5 shows a box plot depicting demand levels on working days and non-working days, which consists of both weekends and public holidays. Notably, workday demand consistently surpasses that of non-workdays throughout all years. This figure also indicates that the year has a noticeable effect on demand; a topic that is explored more robustly in section 4.2.5.

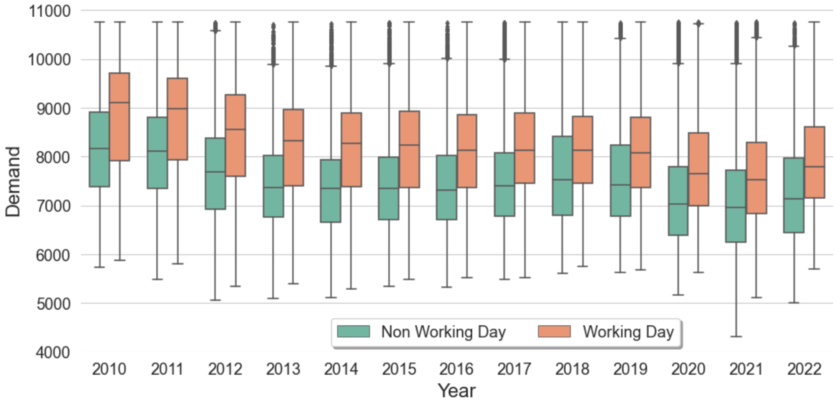


Figure 4.5: Box Plot of Demand: Workday vs Non-Workday

4.2.4 Demand Autocorrelation over One Day

The autocorrelation plot (ACF plot) in Figure 4.6 shows how demand is correlated with itself at different time intervals or lags. The x-axis of the plot represents the 48 30-minute time lags at which the autocorrelation is calculated, while the y-axis represents the the degree of correlation. The autocorrelation values decay until the middle of the series where it starts to increase again, which suggests a periodic behavior in the demand data from day to day. On the basis of this figure, it would be important to include multiple of the most recent lag features for demand, since these are most strongly correlated with current demand. It would also potentially be useful to include a demand lag from a day (48 periods) prior due to its strong correlation.

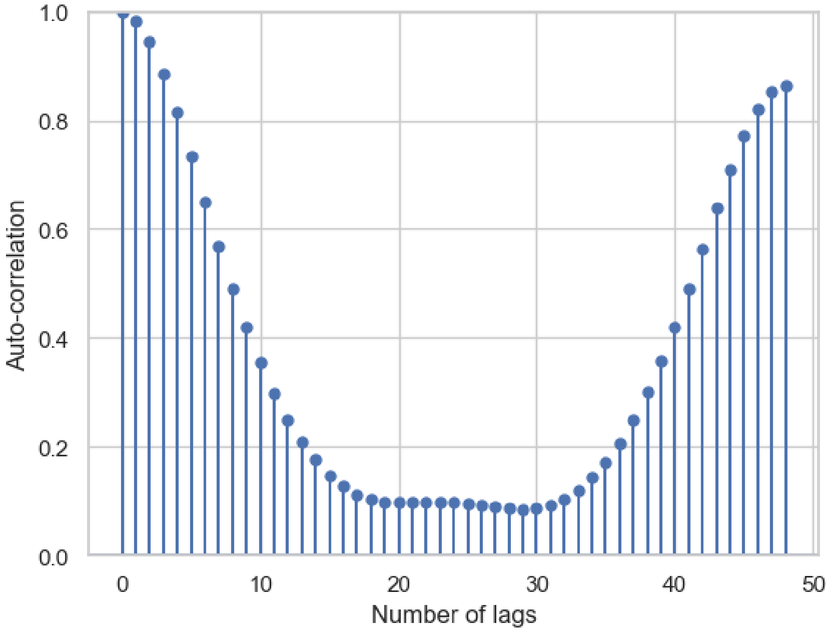


Figure 4.6: Auto-correlation between 30 min lags across 24 hours

4.2.5 Seasonal Decomposition

Figure 4.7 shows a seasonal decomposition of the Total Demand time series data to identify patterns, trends, and seasonal effects. The year on year decline shown in the trendline is contrary to expectations but is likely to be caused by the increase in rooftop solar supply and alternative energy sources that reduce the total demand at the NEM level.

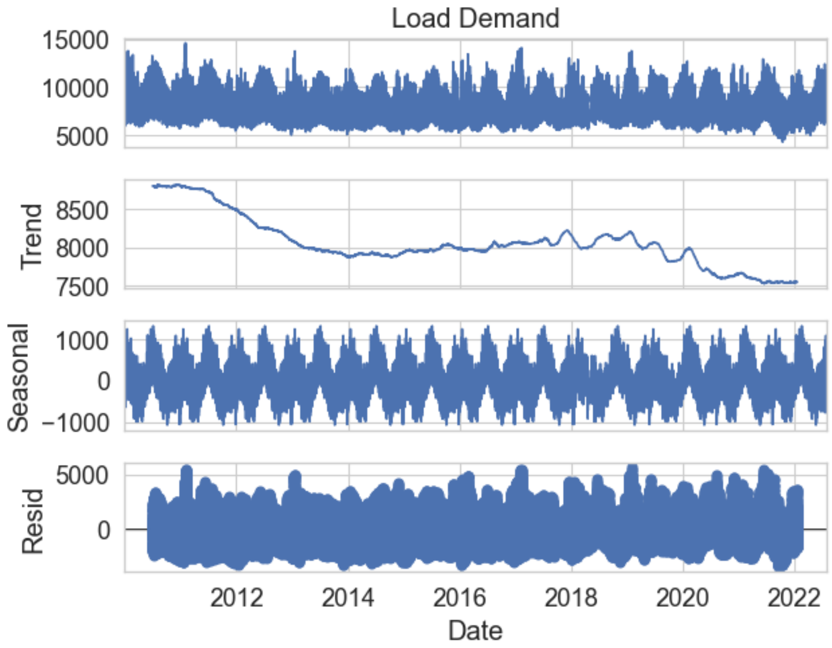


Figure 4.7: Patterns extracted from observed values showing year by year trend, seasonal cycles and remaining residuals

4.3 AEMO Forecasting Accuracy in High-temperature Conditions

The forecasting accuracy from AEMO's model was explored to see how accurate it was at demand forecasting in high temperature conditions. Figure 4.8 shows that the model performs worse for high-temperature observations, with temperatures >= 30^{o}C having significantly higher errors (on average) than more normal temperatures. Figure 4.9 demonstrates that the AEMO model generally under-estimates forecasts during high temperatures, whereas the bias for normal temperatures is close to zero. It was found that only 1.4% of the observations from 2010 to 2022 were in this high-temperature range. This could explain why AEMO's predictions are worse - it is favouring more accurate forecasts for temperature ranges which are most prevalent in the data (the 10^{o}C - 30^{o}C range).

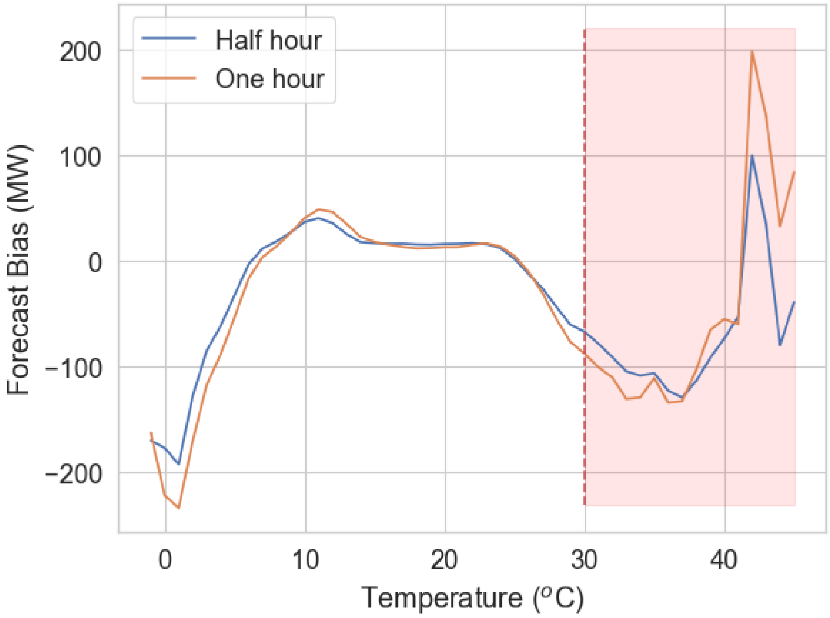


Figure 4.8: Mean Absolute Error of AEMO Demand Forecasts by Temperature

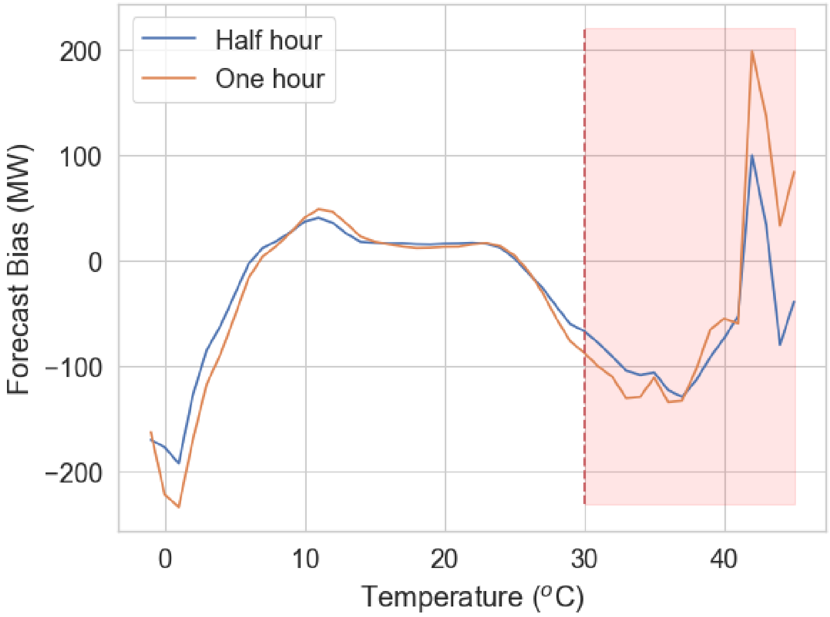


Figure 4.9: AEMO Demand Forecasting Bias by Temperature

Chapter 5
Analysis and Results

5.1 Modelling Details

There were a number of characteristics each model shared in common to ensure accurate comparisons could be made. Each model shared the same train-test splits, with the final two years of the data being used for testing, and the prior observations being used as training data, as shown in Figure 5.1.

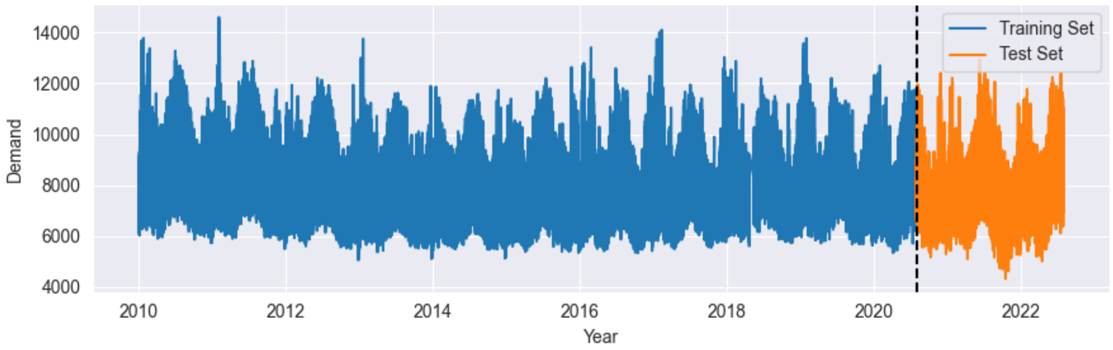


Figure 5.1: Train/Test split that was used for the demand forecasting models

Each model had 19 input features:

- The 10 previous demand values (in half-hourly intervals)
- The demand observed 24 hours ago
- The 4 previous temperature values (in half-hourly intervals)
- Hour of the day
- Workday
- Month
- Year

There were two target variables for each models:

- TOTALDEMAND: the demand 30 minutes after the most recently observed demand and temperature.
- TOTALDEMAND + 1: the demand one hour after the most recently observed demand and temperature.

Finally, each model type shared the same loss function with mean squared error being used as the loss function for all baseline models.

5.2 XGBoost Model

The first baseline model was created using XGBoost, an ensemble learning method capable of capturing non-linear relationships as in electricity demand. As XGBoost has the ability to provide a feature importance score, this was leveraged to determine the importance of the features within the data set. This prioritises features that hold the highest importance for both training and testing.

5.2.1 Feature Importance

Figure 5.2 represent the feature importance based on the F-score metric. The F-score is a statistical measure that evaluates the significance of a feature in predicting the target variable. It is calculated based on the analysis of variance (ANOVA) between different feature categories.

The features listed on the y-axis represent the variables or attributes used in the data set. Each feature has an associated F-score. Features with higher F-scores are more influential in predicting the target variable and contribute significantly to the model's decision-making process. Conversely, features with lower F-scores are less relevant or informative.

As shown by the bars in graph below, the feature that contributed most to the model's performance is the HOUR of the day, followed very closely by DEM - 1, or the energy demand captured half an hour ago, with one period being equivalent to half an hour. The scores also showed that none of the features had a significantly low importance that would necessitate their removal from the model.

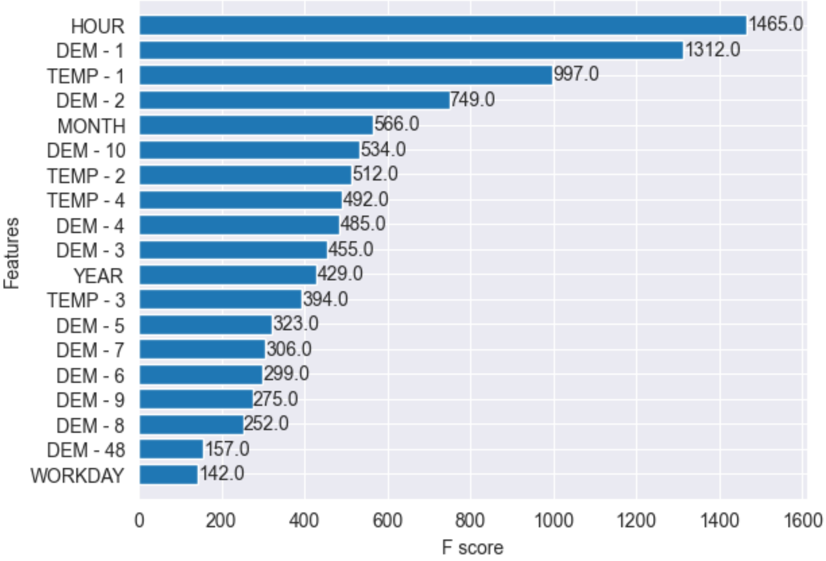


Figure 5.2: Plot showing the relative importance of features in training the XGBoost model

5.2.2 Hyperparameters Tuning

In forecasting with XGBoost, hyperperameters need to be fine tuned to achieve the best model performance. In this study, the algorithms that were used to determine these optimal hyperparameters were RandomizedSearch CV and GridSearch CV.

RandomizedSearchCV efficiently explores the hyperparameter space without requiring an exhaustive search. It helps manage computational resources by randomly sampling a subset of hyperparameter combinations, making it feasible to conduct hyperparameter tuning even with limited resources.

For this XGBoost model, a parameter distribution was used to sample the number of estimators (100 to 1000). During hyperparameter tuning, the code randomly selected values from this range. An XGBoost regressor object was then created with GPU acceleration (i.e., device is set to "cuda") and a K-Fold cross-validation was setup with 5 splits. The scoring metric used for model evaluation was the negative mean squared error.

Figure 5.3 shows how improvements in accuracy decrease with an increasing number of estimators. This helps select a level that will avoid over-fitting the model.

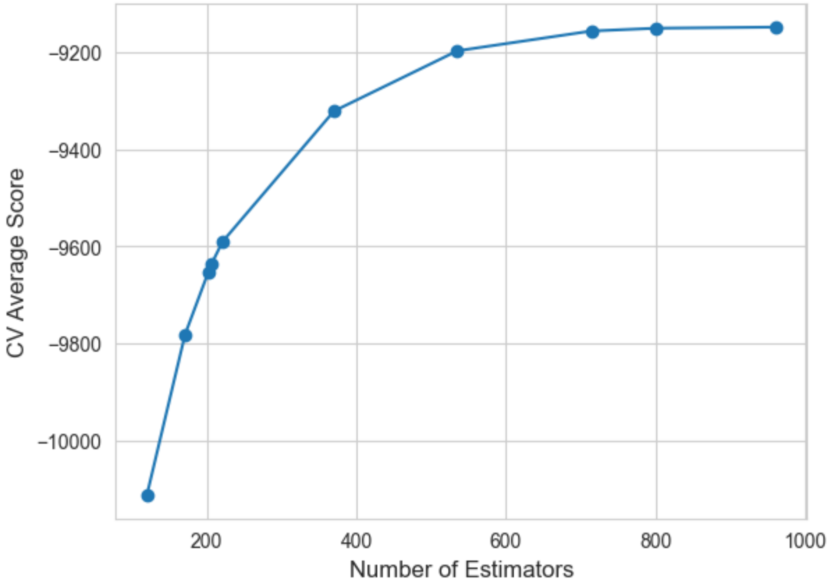


Figure 5.3: Random Search Cross Validated Error Scores

GridSearchCV offers a more comprehensive and systematic way to explore various hyperparameter combinations. This is particularly important in forecasting because time-series data can have unique patterns and dependencies that require specific model configurations.

For this XGBoost model, a range of values were specified for the hyperparameters 'max depth' and 'number of estimators'. A K-Fold cross-validation object (cv) was created with 5 splits, and a grid search was initialised to perform hyper-parameter tuning. The grid search aimed to minimise the negative mean squared error.

Figure 5.4 shows how increasing the depth the model trees decreases the learning curve of the model to prevent over-fitting.

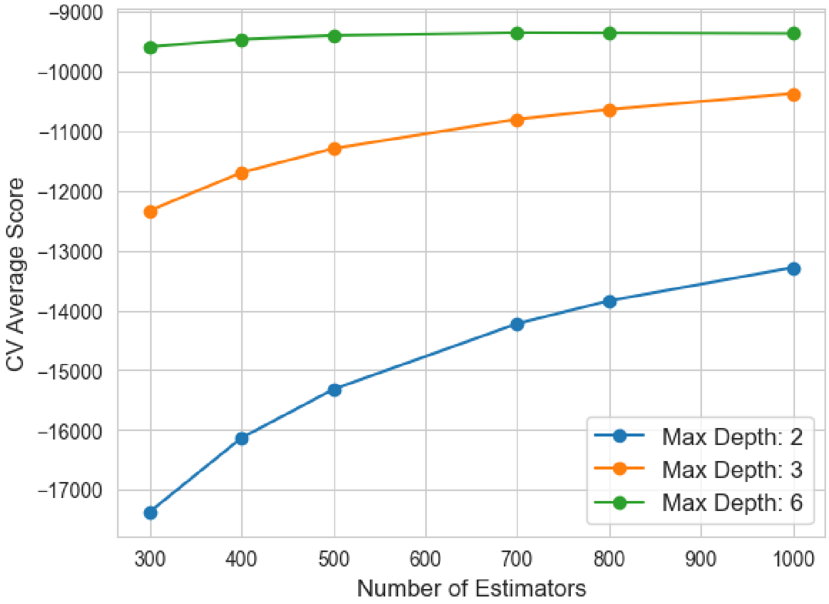


Figure 5.4: Grid Search Cross Validated Error Scores

The source code for the results of this XGBoost model can be accessed on our GitHub Repository. Please refer to Appendix 7.2 Codes - XGBoost Model.

5.3 Neural Network and LSTM Models

Two different neural network models were created to forecast electricity demand. The first model was a standard sequential neural network. The second model featured LSTM cells. LSTM cells are purpose-built for analysing sequential data given they have gates that can help to remember and forget information. This model would treat the demand and temperature lags as sequences, whereas the standard neural network treats them as independent parameters.

Values in both models were standardised with normal scaling. Through testing, this was found to produce more accurate results than using unscaled data or min-max scaled data.

Hyperparameters for both models were selected with the Keras tuner using the hyperband method, which aims to find the best combination of parameters without testing every single combination possible. It offers the time benefits of a randomised search along with the optimsing power of a grid search. The parameters that were tuned were:

- Number of hidden layers: 2, 3, 4. For the LSTM model only 2 was selected due to training time constraints.
- Learning rate: 1e-4, 1e-3, 1e-2
- Number of nodes per hidden layer: 20, 50, 100
- Regularization parameter: 1e-5, 1e-4, 1e-3

The parameters selected as a result of tuning are shown in Table 5.1.

Table 5.1: Hyperparameters chosen for neural networks as a result of tuning.

Parameter	Neural Network	LSTM Network
Hidden layers	4	2
Learning rate	1e-3	1e-3
Nodes per hidden layer	50	50
L2 regularisation constant	1e-5	1e-5

Some parameters were pre-selected to minimise training time. The most notable of these were:

- Activation functions: ReLu activations were chosen for all hidden layers.
- Optimizer: Adam was chosen.
- Batch size: The default setting was used.
- Max epochs: 20 was selected as the maximum number of epochs, with early stopping implemented in the event the validation error began to increase.

The source code for the results of this Neural Network and LSTM model can be accessed on our GitHub Repository. Please refer to Appendix 7.3 Codes – Neural Network Model.

5.4 Model Comparisons

A comparison between the base models created: the XGBoost model, the standard neural network and the LSTM network, is found in Tables 5.2 and 5.3. The results show that all three models were considerably better at forecasting than the AEMO model - both generally and for high temperature conditions. The difference was more drastic for high temperatures, indicating that the models created did a better job than AEMO of catering for hot conditions. There was still a noticeable increase in error across all models when moving from general performance to high temperature performance.

Comparing the three base models, the two neural networks had comparable performance, with the LSTM having slightly better accuracy for half-hour predictions, while the standard neural network has better accuracy for one hour forecasts. The XGBoost model was slightly less accurate than the neural networks. When focusing on high-temperature test data, the LSTM model had slightly better results than the neural network and significantly better results than the XGBoost model.

Table 5.2: Comparison of Model Performance for Test Data

Forecast	Metric	AEMO	XGBoost	Neural Network	LSTM
Half hour	MAE	120.66	96.16	90.36	86.46
	MAPE	1.56	1.30	1.20	1.14
	RMSE	153.60	136.36	121.25	117.16
	R^2	0.9848	0.9880	0.9905	0.9911
Hour	MAE	139.00	139.99	129.87	135.93
	MAPE	1.79	1.87	1.70	1.79
	RMSE	181.10	196.82	180.81	189.68
	R^2	0.9788	0.9750	0.9789	0.9767

Table 5.3: Comparison of Model Performance for Test Data where Temperature >= 30^{o}C

Forecast	Metric	AEMO	XGBoost	Neural Network	LSTM
Half hour	MAE	193.30	176.33	124.00	124.40
	MAPE	2.07	1.95	1.34	1.34
	RMSE	230.38	255.45	196.51	170.83
	R^2	0.9762	0.9708	0.9827	0.9869
Hour	MAE	272.91	322.95	205.48	196.01
	MAPE	2.88	3.35	2.18	2.10
	RMSE	350.61	471.47	324.20	304.77
	R^2	0.9415	0.8941	0.9499	0.9558

5.5 Improving Models for Demand Forecasts during High Temperatures

The modelling so far has focused on improving forecasting accuracy generally. The models have offered improved forecasting that the AEMO forecasts: both generally and during high temperatures. The models are all predicting demand under high-temperatures more poorly than they do generally - most likely because of the small sample size of high-temperature observations in the training data.

A common strategy for dealing with a class imbalance problem is to apply a weighted loss function, giving a higher emphasis on minimising the loss of certain observations in the data. In this case, a MSE loss function weighted by temperature was implemented.

The form of the loss function was:

$$\frac{1}{n}\sum_{i=1}^{n}\left[f(T) \times (y_{pred} - y_{obs})^2\right]$$

where $f(T)$ is a function of temperature which is always non-negative. If the weighting was ever negative, then the model would reduce loss by increasing the error, which is very undesirable. A simple way to ensure non-negative weightings was to subtract the minimum temperature from each temperature.

$$f(T) = T - T_{min}$$

This could be left as is, or it could also be exponentiated to increase the effect of the weightings (e.g. could square the weightings).

The loss functions that were linearly weighted by temperature were found to have a minor effect on model performance, but in most cases resulted in slight improvement at forecasting during high temperatures. Results for these are presented in Figures 5.5 and 5.6.

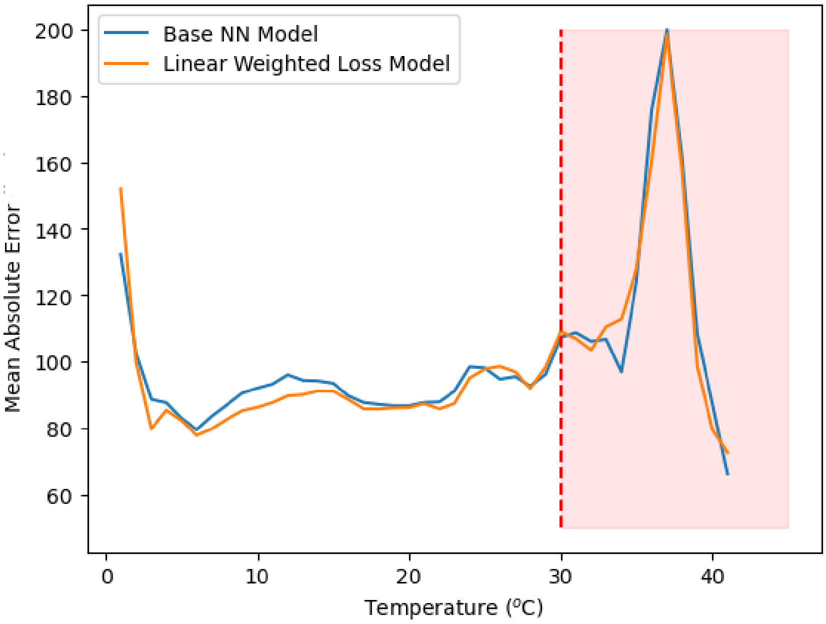


Figure 5.5: Half-hour forecasting error comparison between base NN model and the same model with a loss weighted by temperature

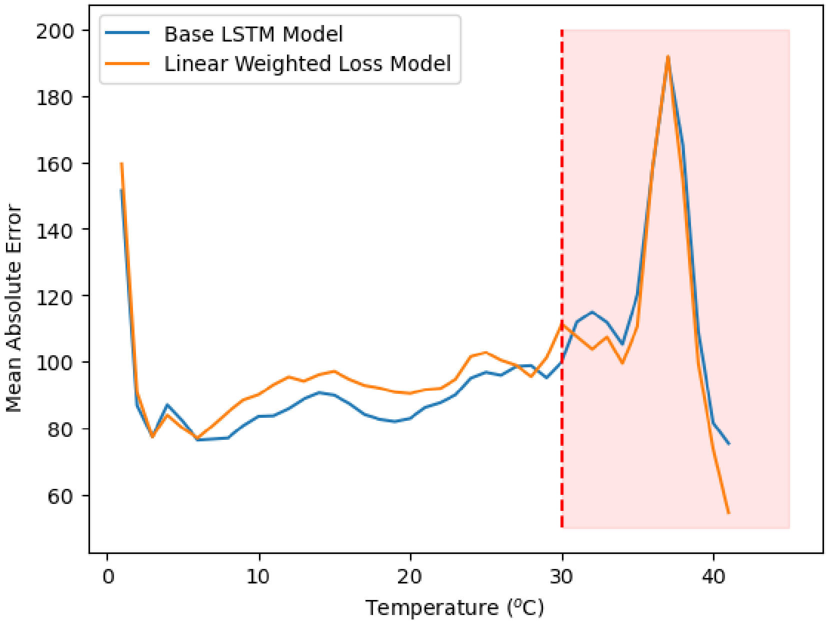


Figure 5.6: Half-hour forecasting error comparison between base LSTM model and the same model with a loss weighted by temperature

The loss functions that weighted the loss by temperature squared had a more significant effect on reducing forecasting error under high temperatures, with the trade-off of having worse forecasting performance for low temperature observations. This can be seen in Figures 5.7, and 5.8.

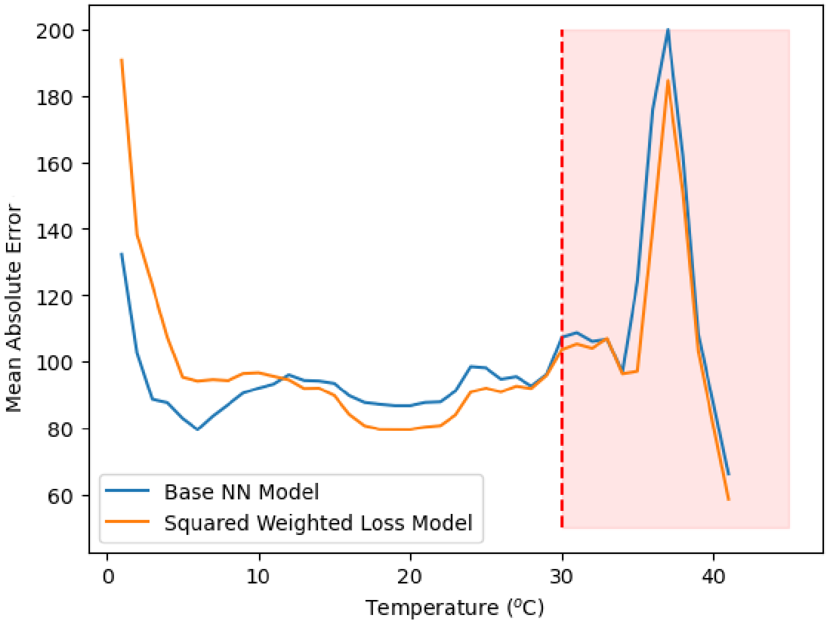


Figure 5.7: Half-hour forecasting error comparison between base NN model and the same model with a loss weighted by temperature squared

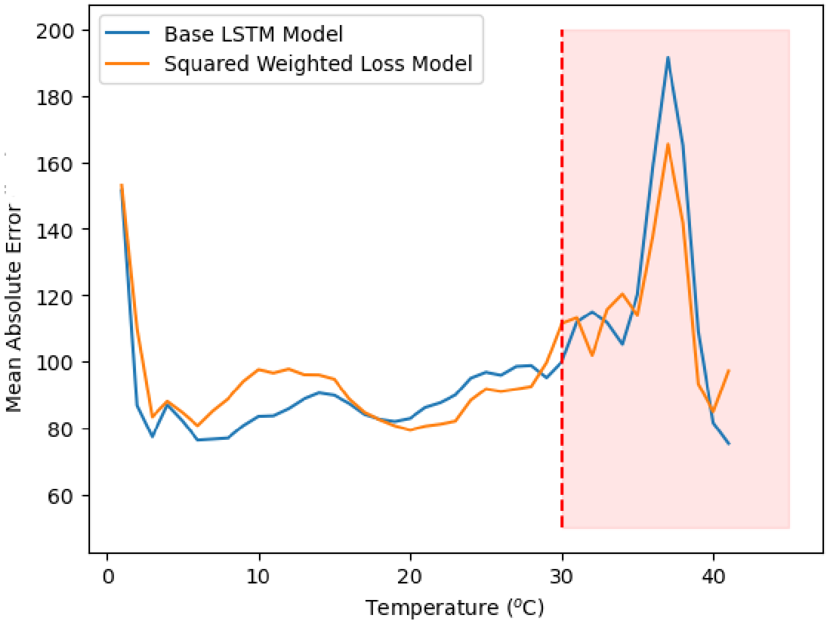


Figure 5.8: Half-hour forecasting error comparison between base LSTM model and the same model with a loss weighted by temperature squared

Comparing the metrics, it was found that the neural network weighted by temperature squared had the best performance under high-temperatures, with the lowest error of all the models for both half-hour and hour forecasts. It also had better overall performance, with lower forecasting errors except for cold temperatures below 10^{o}C. Metrics for this best model are shown in 5.4.

Table 5.4: Comparison of Specialised Model Performance for Test Data where Temperature >= 30^{o}C

Forecast	Metric	AEMO	Base Neural Network	Specialised Neural Network
Half hour	MAE	193.30	124.00	120.37
	MAPE	2.07	1.34	1.30
	RMSE	230.38	196.51	194.83
	R^2	0.9762	0.9827	0.9829
Hour	MAE	272.91	205.48	187.50
	MAPE	2.88	2.18	1.99
	RMSE	350.61	324.20	300.90
	R^2	0.9415	0.9499	0.9568

Another strategy that was trialled to improve performance under high-temperatures was to limit the training data to just high-temperature observations. The thought was that this might give better results for high-temperature conditions, but would generalise very poorly given it was not trained on any normal or cool temperature data. When testing this, the model was found to have very poor generalised performance as expected, but it also had much worse performance under high temperatures.

5.6 Final Model Details

The best performing model out of all the options tested was a 4-layer fully connected neural network with 50 nodes per hidden layer. The neural network was trained with the Adam optimiser using a learning rate of 0.001 and a L2 regularisation constant of 1e-05. A mean square error weighted by temperature squared was used as the loss function during training.

Comparing the model performance to the AEMO forecasts, there is a significant improvement in accuracy. One of the main issues with the AEMO forecasts for high temperatures is that they were biased and generally under predicted demand. Figure 5.9 demonstrates that the best model has almost no bias associated with it, with the points tightly bunched along the $y = x$ line. There is one outlier present in the best model for 30-minute forecasts that was not present in the AEMO model.

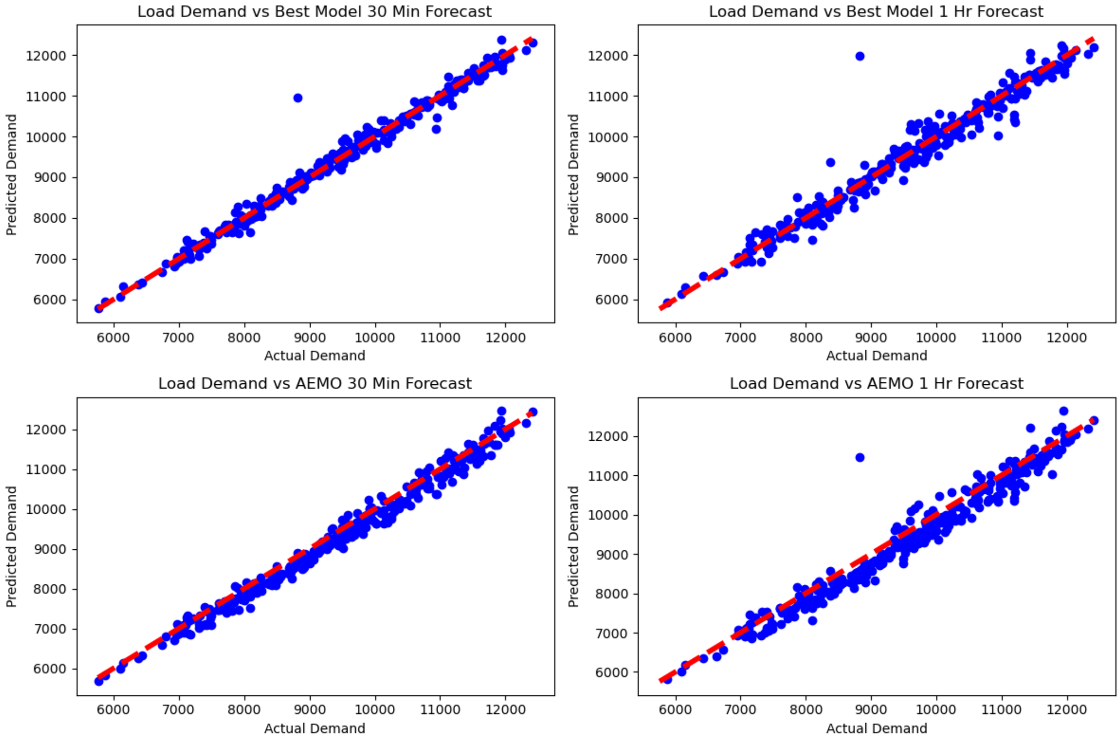


Figure 5.9: Comparing the predicted demand to the actual demand for the final model and for the AEMO model.

Comparing how the model errors vary as a function of temperature, Figures 5.10 and 5.11 show that the final model outperforms AEMO under all temperature conditions for 30-minute forecasts. The final model outperforms AEMO for most temperatures, with a noticeable improvement during hot temperatures. The only area where the final model is worse than AEMO is for one hour forecasts in very cold conditions. This is to be expected since our model is not designed for demand forecasting in these conditions.

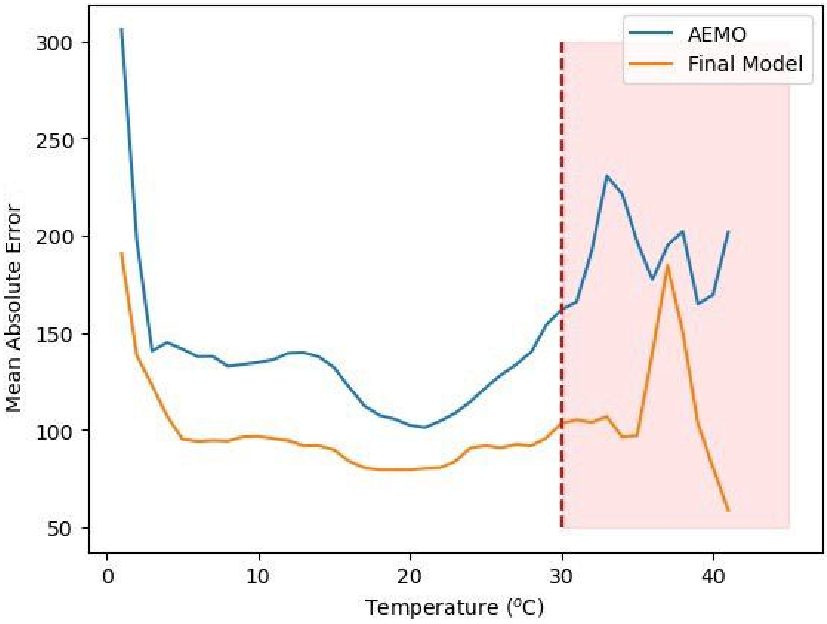


Figure 5.10: Forecasting errors for 30-minute forecasts for the selected model and for the AEMO model.

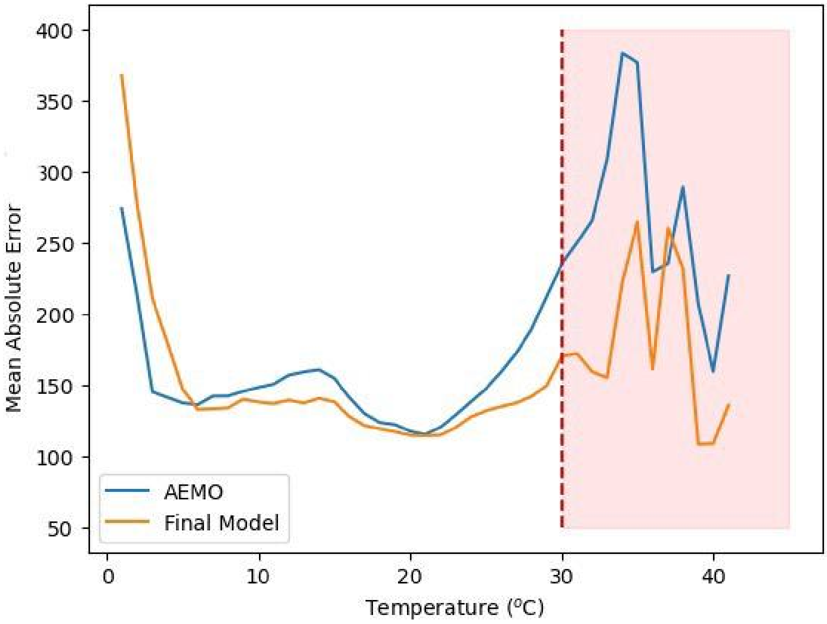


Figure 5.11: Forecasting errors for 1 hour forecasts for the selected model and for the AEMO model.

The model predictions across a sample week in the test data are shown in Figures 5.12 and 5.13. The sample week was chosen to be in summer (the first 7 days of January) so that temperatures would be relatively high. The sample predictions look very accurate for 30-minute forecasts. There is slightly lower accuracy for 1 hour forecasts, but it still appears to be in an acceptable range. The largest errors tend to appear at turning points in the demand. The model tends to slightly over-estimate the peak demand values each day.

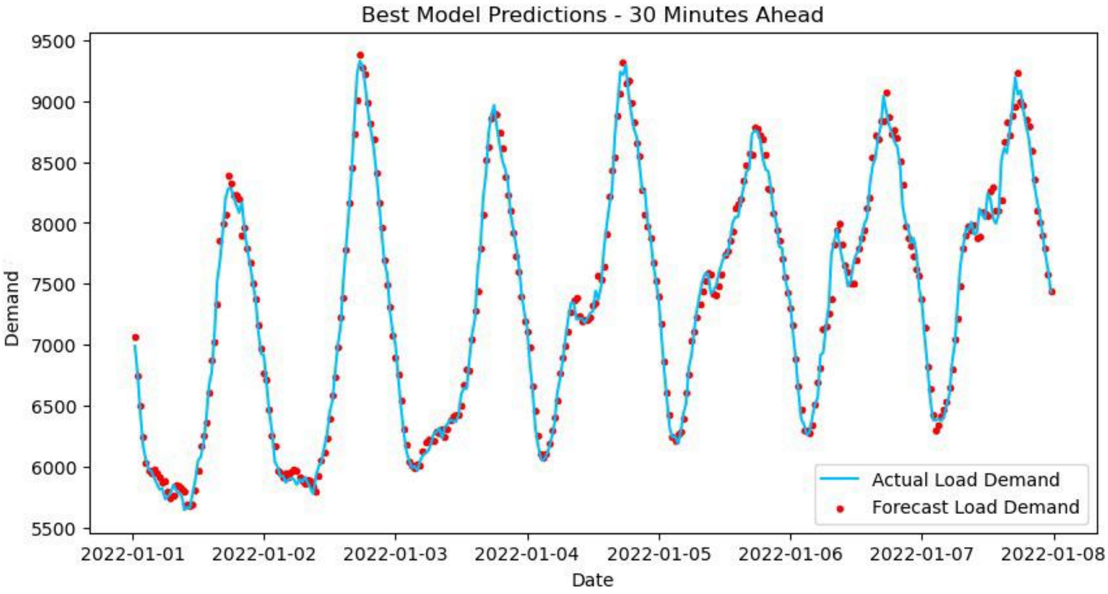


Figure 5.12: Sample week of 30-minute forecasts from the best model.

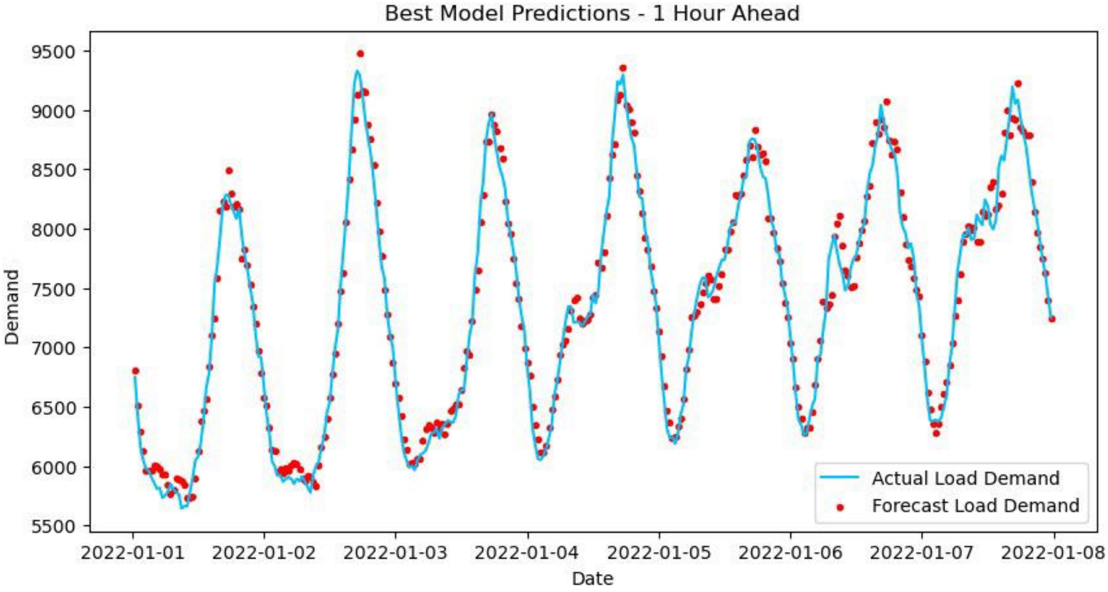


Figure 5.13: Sample week of 1 hour forecasts from the best model.

Chapter 6

Discussion

6.1 Model Performance Compared to AEMO Forecasts

Every tuned model that was created out-performed the AEMO forecasts comprehensively - both generally and for high-temperature observations. This result is unsurprising given the models were all more sophisticated than the AEMO model. The AEMO model is trained only using previous demand values, whereas these models are multivariate and consider additional information such as temperature and datetime properties. The AEMO model is also very simple, being a neural network with a single hidden layer containing 4 nodes. The machine learning methods employed in this report were deeper learning methods that were able to detect more complicated trends in the data.

6.2 Performance of Different Model Types

Comparing the three model types explored, the neural networks (ANN and LSTM) had fairly comparable performance to each other. It was expected that the LSTM model would perform better given it is more complex, can capture time related dependencies, and was so extensively featured when researching demand forecasting.

One challenge for the LSTM model is that some, but not all of the features were time related. The demand lags and temperature lags featured time dependant information, but the datetime features (e.g. hour, month) do not require lags - they are deterministic and known ahead of time. For the purpose of this project a simple LSTM model was used, which was fed all the features as a time series. One approach that could perhaps offer improvement may be a hybrid model approach, where LSTM networks are used to handle the time-dependancies between the lag variables and a traditional neural network is used to handle all the datetime features. Another challenge of the LSTM model was the training time. The increased training time meant that deeper networks and more epochs were not explored.

The XGBoost model performed slightly worse than the other models. Despite this, it was still a very useful model to build, with its feature importance plots providing crucial information about the value of each feature in the model. Throughout the project, this tool was used to help decide which features would be included in the final models.

6.3 Specialising the Model for Forecasting during High Temperatures

The weighted loss functions that were investigated to improve forecasting under high temperatures ended up having a small, but typically positive effect on model performance. It was a bit surprising that the linear weighting functions had such a small effect how the forecasting accuracy varied with temperature. It seems that there were so little hot temperatures in the >= 30°C range that a heavier weighting needed to be applied to considerably shift the model error distribution.

Another result that was somewhat surprising was that the weighted loss functions had a very small impact on general forecasting accuracy, and in some cases were able to improve it. It would have been expected that the traditional RMSE loss function would perform better for general performance (all temperatures). Since the general performance between RMSE and the weighted RMSE models was so similar, it is possible that the differences in errors are due to randomness and are not statistically significant.

The strategy of restricting the training data to only contain high temperature observations generalised poorly as expected, but also did more poorly on hot temperatures than simply training on all of the training data. This can mainly be attributed to the training sample size - with only 1.4% of observations having hot temperatures, there were very little observations in the training data when filtering. This approach also has the downside of class imbalance when it comes to datetime features. For example, filtering for hot temperatures would primarily present data in summer months. The model would not know how to make predictions in other months since there are little to no details provided in the training data.

6.4 Final Model Performance and Areas for Improvement

The performance of the final model was clearly a drastic improvement over the AEMO forecasting model. The final model performance was better generally, but was especially more accurate during high temperatures which was the goal of the model. One reason the final model was so much better than the AEMO forecasts was that it had close to zero bias in its predictions, whereas the AEMO forecasts underestimate demand during high temperatures on average.

Despite the large improvements offered over the AEMO forecasts, it is notable that the final specialised model still had slightly higher errors under high temperatures compared to the middling temperature range close to 20°C. While not explored during this project, it is possible that resampling techniques could be used to further minimise this difference. This could result in synthetically creating more high temperature data for the models to be trained on.

Chapter 7
Conclusion and Further Issues

The aim of this report was to develop a model that would accurately predict short-term demand forecasts during high temperatures. Through testing different machine learning models and training processes, several models were developed that were able to achieve higher accuracy in these conditions. The best performing model of the options tested was a 4 layer neural network that featured a mean square error loss function that was weighted by temperature squared. This model was significantly more accurate than AEMO's model during high temperatures, with a mean absolute percentage error of 1.30% compared to AEMO's 2.07% for 30-minute forecasts. This model also had strong generalised performance, meaning it could reliably be used for forecasting under most temperature conditions.

It is recommended that this best performing model be used for short-term demand forecasting in NSW moving forward. This model significantly outperforms the forecasts provided by AEMO and this gap is expected to grow even further given the predicted rising temperatures due to El Niño. The model will output a 30-minute forecast and a 60 minute forecast for demand. The model has been saved on GitHub under the name 'demand_forecast_model.keras'.

There are many further studies that are suggested that have not been covered in this report:

- Resampling techniques such as block bootstrapping could be explored as an alternative strategy to improving forecasts during high temperatures.
- Different forecasting horizons (e.g. 1 day, 1 week) could be explored based on the needs of suppliers and regulators.
- One significant feature not contained in the current model is the use of solar PV. With this expected to grow, it would be worthwhile exploring how this effects demand, and whether its inclusion in a model could improve forecasting accuracy.
- While this study focused on high-temperatures, many of the strategies employed could be used to produce more accurate forecasts during cold temperatures also. Demand is high during cold temperatures, so this could be useful for suppliers and regulators that are interested in peak demand values during winter.

References

Abbasi, R.A., Javaid, N., Ghuman, M.N.J., Khan, Z.A., Ur Rehman, S., Amanullah, 2019. Short Term Load Forecasting Using XGBoost, in: Barolli, L., Takizawa, M., Xhafa, F., Enokido, T. (Eds.), Web, Artificial Intelligence and Network Applications, Springer International Publishing, Cham. pp. 1120–1131. doi:10.1007/978-3-030-15035-8_108.

Adewuyi, S., Aina, S., Oluwaranti, A., 2020. A DEEP LEARNING MODEL FOR ELECTRICITY DEMAND FORECASTING BASED ON A TROPICAL DATA. Applied Computer Science 16, 5–17. doi:10.35784/acs-2020-01.

Ahmad, T., Zhang, H., Yan, B., 2020. A review on renewable energy and electricity requirement forecasting models for smart grid and buildings. SUSTAINABLE CITIES AND SOCIETY 55, 102052. URL: https://www.webofscience.com/wos/woscc/summary/d82d32a2-f340-4b13-b911-0b02e159d69c-a40758f3/times-cited-descending/1, doi:10.1016/j.scs.2020.102052.

Ahmed, T., Vu, D.H., Muttaqi, K.M., Agalgaonkar, A.P., 2018. Load forecasting under changing climatic conditions for the city of Sydney, Australia. Energy 142, 911–919. URL: https://www.sciencedirect.com/science/article/pii/S0360544217317759, doi:10.1016/j.energy.2017.10.070.

Aisyah, S., Simaremare, A., 2021. Correlation between Weather Variables and Electricity Demand. IOP Conference Series: Earth and Environmental Science 927, 012015. doi:10.1088/1755-1315/927/1/012015.

Al-Musaylh, M.S., Deo, R.C., Adamowski, J.F., Li, Y., 2018. Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia. Advanced Engineering Informatics 35, 1–16. URL: https://www.sciencedirect.com/science/article/pii/S1474034617301477, doi:10.1016/j.aei.2017.11.002.

Bashari, M., Rahimi-Kian, A., 2020. Forecasting Electric Load by Aggregating Meteorological and History-based Deep Learning Modules, in: 2020 IEEE Power & Energy Society General Meeting (PESGM), pp. 1–5. doi:10.1109/PESGM41954.2020.9282124.

Bedi, J., Toshniwal, D., 2018. Empirical Mode Decomposition Based Deep Learning for Electricity Demand Forecasting. IEEE Access 6, 49144–49156. doi:10.1109/ACCESS.2018.2867681.

Behera, D.K., Das, M., Swetanisha, S., Nayak, J., 2022. XGBoost regression model-based electricity tariff plan recommendation in smart grid environment. International Journal of Innovative Computing and Applications 13, 79–87. URL: https://www.proquest.com/docview/2673615626/2B4459B91E9F47B5PQ/6, doi:10.1504/IJICA.2022.123223.

Bianchi, F.M., Maiorino, E., Kampffmeyer, M.C., Rizzi, A., Jenssen, R., 2017. An Overview and Comparative Analysis of Recurrent Neural Networks for Short

Term Load Forecasting. URL: http://arxiv.org/abs/1705.04378, doi:10.1007/978-3-319-70338-1, arXiv:1705.04378.

Bouktif, S., Fiaz, A., Ouni, A., Serhani, M.A., 2018. Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches. Energies 11, 1636. URL: https://www.proquest.com/docview/2108519352/abstract/1E240A87773F4682PQ/1, doi:10.3390/en11071636.

Chen, X., Gupta, L., Tragoudas, S., 2022. Improving the Forecasting and Classification of Extreme Events in Imbalanced Time Series Through Block Resampling in the Joint Predictor-Forecast Space. IEEE Access 10, 121048–121079. doi:10.1109/ACCESS.2022.3219832.

Chung, J., Jang, B., 2022. Accurate prediction of electricity consumption using a hybrid CNN-LSTM model based on multivariable data. PLOS ONE 17, e0278071. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0278071, doi:10.1371/journal.pone.0278071.

Clements, A., Hurn, A., Li, Z., 2016. Forecasting day-ahead electricity load using a multiple equation time series approach. European Journal of Operational Research 251, 522–530. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221715011698, doi:10.1016/j.ejor.2015.12.030.

De Gooijer, J.G., Hyndman, R.J., 2006. 25 years of time series forecasting. International Journal of Forecasting 22, 443–473. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000021, doi:10.1016/j.ijforecast.2006.01.001.

Dehdashti, A.S., Tudor, J.R., Smith, M.C., 1982. Forecasting of Hourly Load by Pattern Recognition??? a Deterministic Approach. IEEE Power Engineering Review PER-2, 47–48. URL: http://ieeexplore.ieee.org/document/5519486/, doi:10.1109/MPER.1982.5519486.

Dong, D., Wen, F., Zhang, Y., Qiu, W., 2023. Application of XGboost in Electricity Consumption Prediction, in: 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI), pp. 1260–1264. URL: https://ieeexplore.ieee.org/document/10176934, doi:10.1109/ICETCI57876.2023.10176934.

Fan, C., Xiao, F., Wang, S., 2014. Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. Applied Energy 127, 1–10. URL: https://www.sciencedirect.com/science/article/pii/S0306261914003596, doi:10.1016/j.apenergy.2014.04.016.

Gokce, M.M., Duman, E., 2022. Performance Comparison of Simple Regression, Random Forest and XGBoost Algorithms for Forecasting Electricity Demand, in: 2022 3rd International Informatics and Software Engineering Conference (IISEC), IEEE, Ankara, Turkey. pp. 1–6. URL: https://ieeexplore.ieee.org/document/9998213/, doi:10.1109/IISEC56263.2022.9998213.

Hong, T., Fan, S., 2016. Probabilistic electric load forecasting: A tutorial review. International Journal of Forecasting 32, 914–938. URL: https://www.sciencedirect.com/science/article/pii/S0169207015001508, doi:10.1016/j.ijforecast.2015.11.011.

Hyndman, R., Koehler, A., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22, 679–688. doi:10.1016/j.ijforecast.2006.03.001.

Ibrar, M., Hassan, M.A., Shaukat, K., link will open in a new tab Link to external site, t., Alam, T.M., link will open in a new tab Link to external site, t., Khurshid, K.S., Hameed, I.A., link will open in a new tab Link to external site, t., Aljuaid, H., Luo, S., 2022. A Machine Learning-Based Model for Stability Prediction of Decentralized Power Grid Linked with Renewable Energy Resources. Wireless Communications & Mobile Computing (Online) 2022. URL: https://www.proquest.com/docview/2709592533/abstract/2B4459B91E9F47B5PQ/4, doi:10.1155/2022/2697303.

Ismail, A., Baysal, M., link will open in a new window Link to external site, t., 2023. Dynamic Pricing Based on Demand Response Using Actor–Critic Agent Reinforcement Learning. Energies 16, 5469. URL: https://www.proquest.com/docview/2843057304/abstract/4F9785526E5B427FPQ/1, doi:10.3390/en16145469.

Jiang, X., Jiang, M., Zhou, Q., 2023. Day-Ahead PV Power Forecasting Based on MSTL-TFT. URL: http://arxiv.org/abs/2301.05911, doi:10.48550/arXiv.2301.05911, arXiv:2301.05911.

Jovanović, S., Savić, S., Bojić, M., Djordjević, Z., Nikolić, D., 2015. The impact of the mean daily air temperature change on electricity consumption. Energy 88, 604–609. doi:10.1016/j.energy.2015.06.001.

Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y., Zhang, Y., 2019. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. IEEE Transactions on Smart Grid 10, 841–851. doi:10.1109/TSG.2017.2753802.

Lewinson, E., 2020. Choosing the correct error metric: MAPE vs. sMAPE. URL: https://towardsdatascience.com/choosing-the-correct-error-metric-mape-vs-smape-5328dec53fac.

Li, J., Han, L., Qiu, W., Jiang, Q., Ge, Y., Liu, S., Zhao, J., Hou, Y., Zhao, Z., Lin, Z., 2023. Optimal inventory strategy of electricity meters based on the combination of Holt-Winters and XGBoost, in: 2023 IEEE 6th International Electrical and Energy Conference (CIEEC), pp. 3987–3992. URL: https://ieeexplore.ieee.org/document/10167274, doi:10.1109/CIEEC58067.2023.10167274.

Liu, A., Ma, Y., Miller, W., Xia, B., Zedan, S., Bonney, B., 2022. Energy Analysis and Forecast of a Major Modern Hospital. BUILDINGS 12, 1116. URL: https://www.webofscience.com/wos/woscc/summary/ac95b782-5c37-4a51-8b7b-1309656466a8-a406f401/times-cited-descending/1, doi:10.3390/buildings12081116.

Manandhar, P., Rafiq, H., Rodriguez-Ubinas, E., 2023. Current status, challenges, and prospects of data-driven urban energy modeling: A review of machine learning methods. Energy Reports 9, 2757–2776. URL: https://www.sciencedirect.com/science/article/pii/S2352484723001026, doi:10.1016/j.egyr.2023.01.094.

McCulloch, J., Ignatieva, K., 2017. Forecasting High Frequency Intra-Day Electricity Demand Using Temperature. URL: https://papers.ssrn.com/abstract=2958829, doi:10.2139/ssrn.2958829.

Raza, M.Q., Khosravi, A., 2015. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renewable and Sustainable Energy Reviews 50, 1352–1372. URL: https://www.sciencedirect.com/science/article/pii/S1364032115003354, doi:10.1016/j.rser.2015.04.065.

Son, H., Kim, C., 2020. A Deep Learning Approach to Forecasting Monthly Demand for Residential–Sector Electricity. Sustainability 12, 3103. URL: https://www.mdpi.com/2071-1050/12/8/3103, doi:10.3390/su12083103.

Tan, M., Yuan, S., Li, S., Su, Y., Li, H., He, F., 2020. Ultra-Short-Term Industrial Power Demand Forecasting Using LSTM Based Hybrid Ensemble Learning. IEEE Transactions on Power Systems 35, 2937–2948. doi:10.1109/TPWRS.2019.2963109.

Taylor, J.W., de Menezes, L.M., McSharry, P.E., 2006. A comparison of univariate methods for forecasting electricity demand up to a day ahead. International Journal of Forecasting 22, 1–16. URL: https://www.sciencedirect.com/science/article/pii/S0169207005000907, doi:10.1016/j.ijforecast.2005.06.006.

Tran, N.T., Tran, T.T.G., Nguyen, T.A., Lam, M.B., 2023. A new grid search algorithm based on XGBoost model for load forecasting. Bulletin of Electrical Engineering and Informatics 12, 1857–1866. URL: https://beei.org/index.php/EEI/article/view/5016, doi:10.11591/eei.v12i4.5016.

Vu, D.H., Muttaqi, K.M., Agalgaonkar, A.P., 2014. Assessing the influence of climatic variables on electricity demand, in: 2014 IEEE PES General Meeting — Conference & Exposition, IEEE, National Harbor, MD, USA. pp. 1–5. URL: http://ieeexplore.ieee.org/document/6939377/, doi:10.1109/PESGM.2014.6939377.

Wu, K., Chai, Y., Zhang, X., Zhao, X., 2022. Research on Power Price Forecasting Based on PSO-XGBoost. Electronics 11, 3763. URL: https://www.proquest.com/docview/2739418901/abstract/2B4459B91E9F47B5PQ/9, doi:10.3390/electronics11223763.

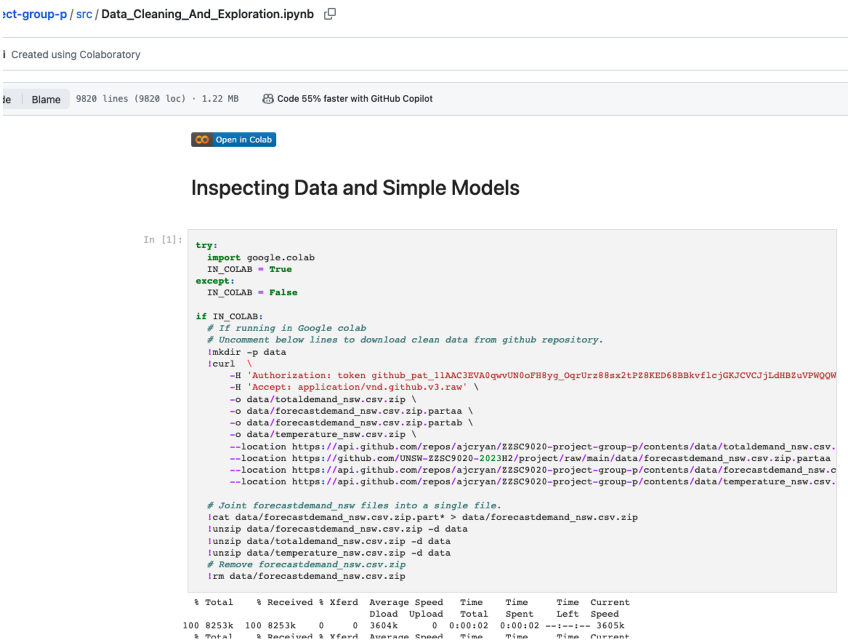
Zhang, R., Dong, Z.Y., Xu, Y., Meng, K., Wong, K.P., 2013. Short-term load forecasting of Australian National Electricity Market by an ensemble model of extreme learning machine. IET Generation, Transmission & Distribution 7, 391–397. URL: https://www.proquest.com/docview/1492896596/abstract/664C35588F644CB2PQ/1.

Appendix

Appendix A - Codes

7.0.1 Data Cleaning and Exploration

Link to code: https://github.com/ajcryan/ZZSC9020-project-group-p/blob/main/src/Data_Cleaning_And_Exploration.ipynb



ect-group-p / src / Data_Cleaning_And_Exploration.ipynb
i Created using Colaboratory
le
Blame
9820 lines (9820 loc) · 1.22 MB
Code 55% faster with GitHub Copilot
Open in Colab

Inspecting Data and Simple Models

In [1]:

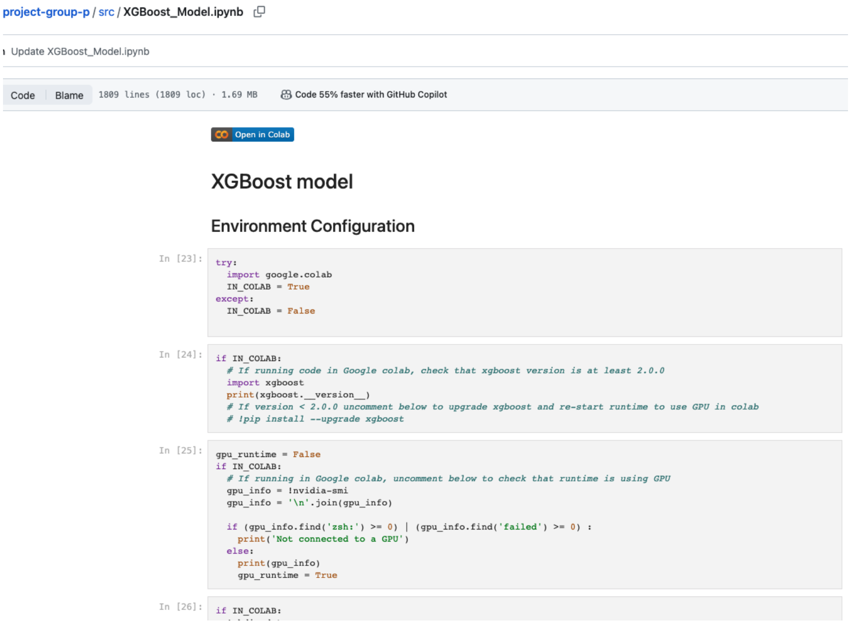
```
try:
  import google.colab
  IN_COLAB = True
except:
  IN_COLAB = False

if IN_COLAB:
  # If running in Google colab
  # Uncomment below lines to download clean data from github repository.
  !mkdir -p data
  !curl  \
      -H 'Authorization: token github_pat_11AAC3EVA0qwvUN0oFH8yg_OqrUrz88sx2tPZ8KED68BBkvflcjGKJCVCJjLdHBZuVPWQQW
      -H 'Accept: application/vnd.github.v3.raw' \
      -o data/totaldemand_nsw.csv.zip \
      -o data/forecastdemand_nsw.csv.zip.partaa \
      -o data/forecastdemand_nsw.csv.zip.partab \
      -o data/temperature_nsw.csv.zip \
      --location https://api.github.com/repos/ajcryan/ZZSC9020-project-group-p/contents/data/totaldemand_nsw.csv.
      --location https://github.com/UNSW-ZZSC9020-2023H2/project/raw/main/data/forecastdemand_nsw.csv.zip.partaa
      --location https://api.github.com/repos/ajcryan/ZZSC9020-project-group-p/contents/data/forecastdemand_nsw.c
      --location https://api.github.com/repos/ajcryan/ZZSC9020-project-group-p/contents/data/temperature_nsw.csv.

  # Joint forecastdemand_nsw files into a single file.
  !cat data/forecastdemand_nsw.csv.zip.part* > data/forecastdemand_nsw.csv.zip
  !unzip data/forecastdemand_nsw.csv.zip -d data
  !unzip data/totaldemand_nsw.csv.zip -d data
  !unzip data/temperature_nsw.csv.zip -d data
  # Remove forecastdemand_nsw.csv.zip
  !rm data/forecastdemand_nsw.csv.zip
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 8253k  100 8253k    0     0  3604k      0  0:00:02  0:00:02 --:--:-- 3605k
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
```

Figure 7.1: Data_Cleaning_And_Exploration.ipynb

7.0.2 XGBoost Model

Link to code: https://github.com/ajcryan/ZZSC9020-project-group-p/blob/main/src/XGBoost_Model.ipynb



project-group-p / src / XGBoost_Model.ipynb

Update XGBoost_Model.ipynb

1809 lines (1809 loc) · 1.69 MB

Code 55% faster with GitHub Copilot

XGBoost model

Environment Configuration

```
In [23]:
try:
  import google.colab
  IN_COLAB = True
except:
  IN_COLAB = False
```

```
In [24]:
if IN_COLAB:
  # If running code in Google colab, check that xgboost version is at least 2.0.0
  import xgboost
  print(xgboost.__version__)
  # If version < 2.0.0 uncomment below to upgrade xgboost and re-start runtime to use GPU in colab
  # !pip install --upgrade xgboost
```

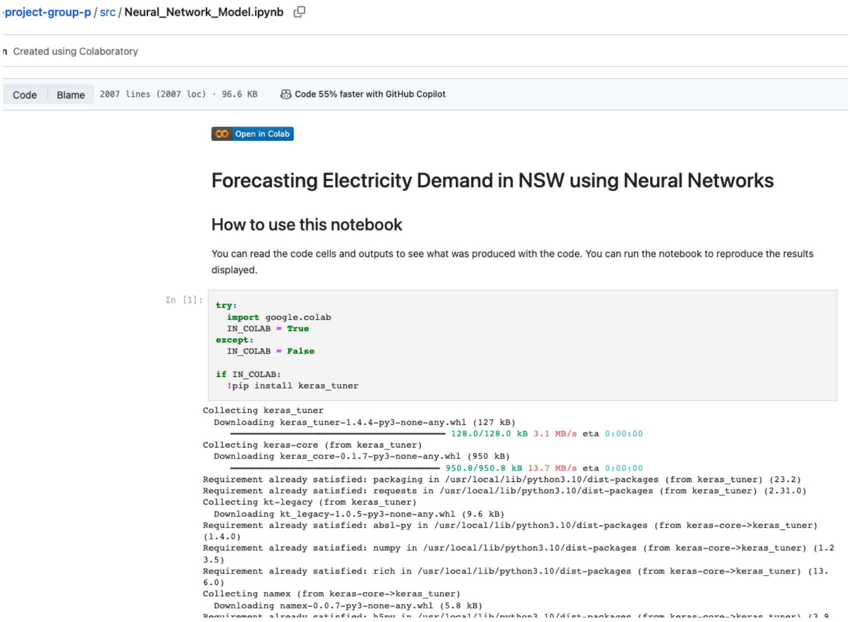
```
In [25]:
gpu_runtime = False
if IN_COLAB:
  # If running in Google colab, uncomment below to check that runtime is using GPU
  gpu_info = !nvidia-smi
  gpu_info = '\n'.join(gpu_info)

  if (gpu_info.find('zsh:') >= 0) | (gpu_info.find('failed') >= 0) :
    print('Not connected to a GPU')
  else:
    print(gpu_info)
    gpu_runtime = True
```

```
In [26]:
if IN_COLAB:
```

Figure 7.2: XGBoost_Model.ipynb

7.0.3 Neural Network Model

Link to code: https://github.com/ajcryan/ZZSC9020-project-group-p/blob/main/src/Neural_Network_Model.ipynb



project-group-p / src / Neural_Network_Model.ipynb

Created using Colaboratory

2007 lines (2007 loc) · 96.6 KB

Code 55% faster with GitHub Copilot

Forecasting Electricity Demand in NSW using Neural Networks

How to use this notebook

You can read the code cells and outputs to see what was produced with the code. You can run the notebook to reproduce the results displayed.

```
In [1]:
try:
  import google.colab
  IN_COLAB = True
except:
  IN_COLAB = False

if IN_COLAB:
  !pip install keras_tuner
```

```
Collecting keras_tuner
  Downloading keras_tuner-1.4.4-py3-none-any.whl (127 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 128.0/128.0 kB 3.1 MB/s eta 0:00:00
Collecting keras-core (from keras_tuner)
  Downloading keras_core-0.1.7-py3-none-any.whl (950 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 950.8/950.8 kB 13.7 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from keras_tuner) (23.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from keras_tuner) (2.31.0)
Collecting kt-legacy (from keras_tuner)
  Downloading kt_legacy-1.0.5-py3-none-any.whl (9.6 kB)
Requirement already satisfied: absl-py in /usr/local/lib/python3.10/dist-packages (from keras-core->keras_tuner)
(1.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from keras-core->keras_tuner) (1.2
3.5)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages (from keras-core->keras_tuner) (13.
6.0)
Collecting namex (from keras-core->keras_tuner)
  Downloading namex-0.0.7-py3-none-any.whl (5.8 kB)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-packages (from keras-core->keras_tuner) (3.9
```

Figure 7.3: Neural_Network_Model.ipynb

Appendix B - Tables

7.0.4 Data Summaries

1) File: totaldemand_nsw.csv
Contains total electricity demand values in NSW from 01/01/2010 to 01/08/2022. (File size: 43.915 MB)

Table 7.1: Total Demand Variables

Column Name	Data Type	Description
`DATETIME`	`datetime64[ns]`	Datetimes in 5-minute intervals from 01/01/2010 to 01/08/2022.
`REGIONID`	`object`	Contains the region in which demand was measured. 'NSW1' is the only region present in this data set.
`TOTALDEMAND`	`float64`	Total power demand measured in megawatts.

2) File: forecastdemand_nsw.csv
Contains AEMO's electricity demand forecasts in NSW from 01/01/2010 to 01/08/2022. (File size: 769.555 MB)

Table 7.2: Forecast Demand Variables

Column Name	Data Type	Description
`DATETIME`	`datetime64[ns]`	Datetimes in 30-minute intervals from 01/01/2010 to 01/08/2022. Each datetime is repeated 71 times based on forecasts made at different time horizons.
`REGIONID`	`object`	Contains the region in which demand was measured. 'NSW1' is the only region present in this data set.
`FORECASTDEMAND`	`float64`	Forecast power demand measured in megawatts.
`PREDISPATCHSEQNO`	`int64`	A unique identifier attached to each model forecast.
`PERIODID`	`int64`	Represents the number of periods (30-minute intervals) of the forecast horizon.
`LASTCHANGED`	`datetime64[ns]`	Datetime representing when the model made the forecast.

3) File: temperature_nsw.csv

Contains temperature data from Bankstown airport from 01/01/2010 to 01/08/2022. (File size: 8.195 MB)

Table 7.3: Temperature Variables

Column Name	**Data Type**	**Description**
`DATETIME`	`datetime64[ns]`	Datetimes in 30-minute intervals from 01/01/2010 to 01/08/2022.
`LOCATION`	`float64`	Contains the location at which temperature was measured. 94766 is the only location present in this data set.
`TEMPERATURE`	`float64`	Temperature measured in degrees Celsius.