

Optimal Medium Term Electricity Load Forecasting for New South Wales

Liam Walsh (z5429207),
Sreedevi Nandakumar (z3444815),
John D'Souza (z5389733),
Mina Mariano (z5337974).
22/04/2024

Table of Contents

1	Introduction	2
2	Literature Review.....	3
3	Material and Methods.....	6
4	Exploratory Data Analysis	12
5	Analysis and Results.....	19
6	Discussion	34
7	Conclusion and Further Work.....	36
	References	37
	Appendices	40



UNSW
SYDNEY

1 Introduction

In a 2019 economic report McKinsey [1] specifies two forces that are disrupting trends in global primary energy demand: rise of electrification, and growth of renewables in electricity generation. Electricity will become mankind's primary source of energy for the near future. The economic importance of electricity demand forecasting cannot be understated. As Oliver Nunn from Endgame Economics states, "If you can forecast [electricity] demand, you can make money." [10].

Soliman and Al-kandari [11] define electricity demand (load) forecasting as the science of predicting future load for a specific look-ahead period. Accurate load forecasting ensures that there is enough electric power supply to meet demand at any time [13]. This helps maintain the balance and stability of the power grid, which, in turn, brings about reliability, efficiency and cost savings. Through load forecasting, utilities can manage their resources better, and implement demand response programs, whereby customers are incentivized to reduce their demand during high-usage periods. Load forecasting also supports strategic planning decisions such as capacity expansion, maintenance scheduling and infrastructure development. In deregulated electricity markets, like the National Electricity Market (NEM) of Australia, load forecasting helps market participants make informed bidding strategies, mitigate risks, and manage energy contracts. There are several methods used in load forecasting, each of which analyses historical load data and other inputs to generate forecasts for different time horizons:

- Long term load forecasting (LTLF) is applied to predict loads from a year to as far as 50 years ahead, to facilitate system planning and optimisation. It helps utilities make decisions about where to invest in new power generation capacity and how to balance various sources of energy.
- Medium term load forecasting (MTLF) is exploited to predict monthly and yearly loads to carry out efficient operational planning and electricity trading. It considers seasonal variations in electricity consumption as well as planned outages.
- Short term load forecasting (STLF) is used to forecast loads from a few minutes up to a week ahead to minimize daily dispatching and distribution costs. This is crucial for managing the power grid in real-time. Accuracy of these predictions helps reduce wasted electricity generation and overloaded power lines.

Since STLF is an essential tool for daily market operations, it has been paid the most attention in academia and industry [12]. Jin et al [15] have stated that whereas STLF is useful for controlling and scheduling power generation for generators, accurate medium- and long-term peak load forecasting allows system operators to determine the generation capacity required to satisfy future demand, address various vulnerabilities in advance, and develop reliable and economic demand response programs. Failure to accurately forecast peak loads in the medium and long terms can lead to overinvestment in the construction of

power equipment or cause instability in supply and demand. This has motivated us to focus our efforts on MTLF in this project.

Industry and academia have applied many different statistical, computational, and econometric methods to forecast electricity demand. Univariate and multivariate linear regression models have been used. Recent forecasting techniques focus more on Machine Learning. This approach offers more flexibility and adaptability to changing patterns in complexities in data over time, helps uncover hidden patterns and relationships and handle high-dimensional datasets with substantial number of features, which traditional models may not manage effectively. Feed forward and recurrent neural networks have also been successfully applied.

In this project we used multiple models to forecast electricity demand for the state of New South Wales (NSW) in the medium term – a few months to two years. Using input factors, such as temperature, population growth, solar photovoltaic (solar PV) installation and economic factors such as gross state product (GSP), we answered the question, “Which predictors and modelling approach are best suited for Medium Term Load Forecasting (MTLF) of electricity?”

2 Literature Review

As stated in the Introduction, the rise of electrification and the growth of renewables in electricity generation are resulting in increasing trends in electricity demand, as well as increasing complexity in managing the supply-demand balance due to the intermittent nature of renewable energy resources. This has made MTLF and LTLF crucial tools to manage and plan the resources and conduct feasibility studies for building new power generation units [14].

Khuntia et al [17] have compiled a comprehensive review of MTLF and LTLF methods that are being employed today. They confirm that MTLF and LTLF are much less popular as research topics compared to STLF, stating that “dozens of papers on STLF are published each year for each paper on MTLF or LTLF.” They further confirm that MTLF has received less coverage in research than LTLF. This paucity of research, as well as the growing importance of MTLF as a tool for power system reliability management through applications in maintenance scheduling and economic operation in a deregulated environment, have motivated us to focus our efforts on MTLF as the use case for our modelling efforts.

MTLF’s applications include the allocation of available resources and development of infrastructure elements in the medium-term horizon, such as improving congestion management in transmission grids and consequently improving system efficiency and cost of energy for consumers. Transmission and distribution service providers can use MTLF to guide the improvement of their networks. Further, MTLF can help participants in the energy markets make informed trading decisions for the purchase of energy, development of medium-term generation, distribution, and transmission contracts [19]. An important learning for us from the Khuntia et al’s review is that typically, researchers have used one

of two approaches for MTLF. The first is termed a “conditional modelling approach” encompassing historical load, weather data and socio-economic indicators (like GDP). The second is termed an “autonomous approach” which is dependent only on historical load and weather data. The “autonomous approach” is well suited for stable economies and has proven to be better suited if the forecast horizon is less than or equal to one year. We shall explore both approaches in our project.

McGrath and Jonker [13] have summarised the typical approach used in load forecasting in the following way. The process begins with historical load data collection. Data from many factors that affect electricity use is included, such as weather data, time of day, calendar variables (seasons, holidays, weekdays versus weekends) and demographic factors (population density, economic activity). Once data is collected, a forecasting model is developed. They summarised the typical models used in the following way. *Regression models*: Linear regression models are often used for LTLF, relating the load demand to weather conditions and economic indicators. *Time series models*: Autoregressive Integrated Moving Average (ARIMA) are popular for all three look-ahead ranges, relying on past load data to predict future demand. *Artificial Intelligence models*: Neural networks and support vector machines (SVM) are increasingly used for their ability to model non-linear relationships. Deep learning models are also used to improve forecasting accuracy by automatically extracting relevant features from the dataset. The forecasting model is trained using a portion of the historical data and tested for validation. The accuracy of forecasts is typically evaluated using metrics like Mean Absolute Percentage Error (MAPE). Once the model is validated and fine-tuned it can be used to generate future load forecasts. These forecasts are then used for operational planning and decision-making activities. As new data becomes available, the models require updates and retraining.

Semekonawo and Kem, in August 2022, conducted an electricity demand forecasting study in West African countries [8]. They utilised multivariate linear regression and Autoregressive Integrated Moving Average (ARIMA) models to forecast electricity demand. They compared these two models based on Mean Absolute Deviation (MAD), Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). They included socio-economic factors like Gross Domestic Product (GDP) and population growth, and a technological factor, annual electricity demand, in the study. They concluded that the ARIMA model more accurately predicted electricity demand for most of the West African countries.

Chapagain et al. used artificial neural networks and deep learning methods to create a electricity demand forecast [3]. They utilised historical electricity demand and weather conditions to create their models. They studied two different scenarios, where for scenario 1, they utilised data available on weekdays and for scenario 2, they included all days of the week. Three different models were utilised on both scenarios: Feedforward Neural Network (FNN), Recurrent Neural Network based Gated Recurrent Network (RNN-GRU) and RNN with Long Short-Term Memory (RNN-LSTM). On evaluating forecasting accuracy, scenario 1 achieved lowest MAPE of 2.47% using the FNN method. The RNN-GRU model predicted scenario 2 more accurately compared to the other two models with the lowest Mean Absolute Percentage Error (MAPE) of 3.44%.

The application of Machine learning (ML) in electricity demand analysis in literature is divided into unsupervised learning methods, mostly in the domain of descriptive analytics and data pre-processing, and supervised learning, usually for predictive modelling. Much has been reported about the use of unsupervised methods such as clustering, to group together households and neighbourhoods with similar demand behaviour, and anomaly detection, for fault detection and dynamic electricity pricing [14]. The focus of our project, however, is predictive modelling, hence we have limited our ML models to supervised learning algorithms.

ML and artificial neural networks (ANNs) are often used to capture non-linear patterns in the data [6]. In our project, we have used a type of ANN called Feedforward Neural Network (FNN) to forecast electricity demand [5]. As per literature reviews, FNNs are prone to overfitting due to the chance of network being trapped in a local optimum rather than reaching a global optimum [4]. Complex pattern data can be understood using a branch of machine learning called deep learning [9]. A deep learning network has complicated architecture with multiple hidden layers compared to a simple neural network which has only one hidden layer. A deep learning network can also have structures like long short-term memory (LSTM). To capture long term dependencies and sequence dependent behaviour, RNN-LSTM models are well suited and so we have decided to use this model in the project [2].

In 2022, Long et al [18] demonstrated the use of Prophet, a time series forecasting model open-sourced by Facebook data scientists in 2017 [20], to improve the medium- and long-term load forecasting accuracy compared to an ARIMA as well as RNN-LSTM methods. Prophet can automatically handle missing values and outliers, decompose and predict the future trend of the time series and give confidence intervals. Its advantage lays in its ability to decompose the time series into trend, seasonal and holiday terms. The trend term fits the non-periodic variation of the time series; the seasonal term fits the periodic variation of the time series with annual and weekly cycles; while the holiday term fits the effect of the holiday time window on the time series.

Divinia et al [25] proposed a *stacking ensemble* learning model to predict short-term electricity demand. They have utilised a two layered stacking ensemble with three base learners whose predictions are fed to a top-level method which combines this prediction to produce final forecast. The three base learning algorithms utilised in the study are Random Forest, Artificial Neural Network, and decision tree based on Evolutionary computation. The top-level algorithm used is Gradient Boosting. They have found that the predictions obtained from the ensemble were superior to the results from other techniques.

Jin et al [15] demonstrated the importance of analysing the characteristics of the peak load and selecting appropriate features for accurately forecasting the load. They demonstrated that peak loads are correlated not only between years but also between the months of the same year. They also concluded that using monthly data as input data for the MTLF and LTLF models is more appropriate. We have been guided by this research and have chosen a monthly grain to form the basis for the dataset that we feed our models.

3 Material and Methods

3.1 Software

In this project our team collectively chose the mix of software tools to suit our skill sets.

Python was chosen as the programming language of choice for conducting exploratory data analysis, data cleaning, modelling, forecasting and data visualization. It has extensive libraries that are used for efficient data manipulation, time-series based analysis and forecasting and machine learning. For implementing machine learning algorithms, we used the [sklearn](#), [keras](#), [xgboost](#), [statsmodels](#), [pmdarima](#) and [prophet](#) libraries, which provided a diverse selection of algorithms.

For code documentation, narrative texts and enhanced data visualization, Jupyter Notebooks in Visual Studio Code were utilized. They facilitate the representation of Python code alongside documentation, easing collaboration among team members during model development, experimentation, insight sharing, and result reproduction.

We utilized various software tools for collaborative tasks. GitHub served as our platform for uploading datasets, models, and project notebooks. It allowed multiple team members to contribute while managing version control. Additionally, we utilized GitHub for organizing project management. Microsoft Teams was utilized for team meetings, messaging and storing and sharing documents via its integration with OneDrive. The project report was written in Microsoft Word and the project video presentation was carried out using Microsoft PowerPoint.

The Project Jupyter Notebook containing the codebase is stored at this GitHub location: <https://github.com/z5429207/ZZSC9020-Group-P/blob/main/src/Project.ipynb>.

3.2 Description of the Data

The datasets for our project consisted of three sets of data files provided by the project sponsor and three sets of independently sourced data files. In our project we used only those files that held data relevant to New South Wales. A summary of these data files is presented in Table 1.

Dataset	Format	No. of files	Grain (approx.)	Historical Depth (approx.)	Source
<i>NSW Total Demand</i>	Comma Separated Value (with header)	1	30-minute intervals	From 01/01/2010 to 17/03/2021	Provided by project sponsor
<i>NSW Temperature</i>	Comma Separated Value (with header)	1	30-minute intervals	From 01/01/2010 to 17/03/2021	Provided by project sponsor
<i>NSW Forecast</i>	Comma Separated	2	30-minute	From	Provided by

Dataset	Format	No. of files	Grain (approx.)	Historical Depth (approx.)	Source
<i>Demand</i>	Value (with header)	partial zipped	intervals	30/12/2009 to 17/03/2021	project sponsor
<i>Australian State Population</i>	Microsoft Excel multi-worksheet workbook	1	Quarterly per State	From June 1981 to June 2023	Australian Bureau of Statistics [26]
<i>Australian Renewable Energy Installations</i>	Microsoft Excel multi-worksheet workbook	20	Monthly per Post Code	From January 2003 to December 2022	Australian Clean Energy Regulator [27]
<i>Australian Gross State Product</i>	Microsoft Excel multi-worksheet workbook	1	Annual per State	From 1990 to 2023	Australian Bureau of Statistics [28]

Table 1 - Source Data File Summary

Our decision to focus on data for NSW stems from NSW's significant role in Australia's energy consumption landscape, rendering it a suitable representation of the variables crucial for our analysis. Moreover, this subset offers practicality in terms of computational resources and mitigates the risk of overfitting.

The energy demand data contains more than 196,513 rows of electricity demand in NSW in 30-minute intervals and amounts to 5.6MB. The variables include the date and time interval, total demand (MW) and region identifier. This data contains a combination of trend and seasonality elements. Data for the year 2021 is incomplete.

In this project we did not use the NSW Forecast Demand datasets that were provided.

The temperature dataset records the air temperature in New South Wales (NSW), measured from Bankstown. It comprises approximately 220,326 rows and amounts to a total size of 6.7MB. The variables encompass date and time intervals, location, and air temperature in °C. Although temperature readings are captured every half hour, the dataset does not exhibit a consistent time interval throughout. Data for the year 2021 is incomplete.

The Solar Photovoltaic (PV) dataset was sourced from the website of the Australian Government Clean Energy Regulator (CER). This dataset covers new installations, upgrades to existing systems, and stand-alone (off-grid) systems, excluding major installations. Notably, this data is not publicly accessible via the REC Registry and does not include systems awaiting registration or those that have failed, as stated on the CER website. These files were retrieved in March 2024. While the current year's dataset remains accessible on the website, historical data is currently unavailable. Each file is presented as an Excel sheet, containing data for one year along with data from the previous year. Data is segmented by postcode, with each file comprising approximately 2740 records. The information includes

the number of installations and the installed power capacity (KW) for each postcode, collected monthly. Only data from New South Wales (NSW) will be utilized from this file, for the purpose of this analysis.

The economic variables were sourced from a dataset containing information on gross state product (GSP) and gross domestic product (GDP) across various states and territories in Australia and overall Australia, accompanied by percentage changes over time. Spanning from June 1990 to June 2023, the dataset consists of 34 observations for most series, except for the percentage change series, which has 33 observations. The unit of measurement is in millions of dollars. Furthermore, the "DERIVED" data type implies that the values are derived or calculated based on other data sources. The data acquisition occurs each June and is collected annually. This dataset is selected because it offers a comprehensive overview of GSP and GDP across different Australian regions, facilitating the analysis of economic trends and performance over time. Additionally, the percentage change data provides further insights into the fluctuations in GSP growth or decline over the years.

3.3 Pre-processing and Data Cleaning Steps

The raw data for each dataset needed to be made available in an easy-to-use format to conduct profiling and data analysis. For this, we unpacked individual files from their original format (compressed, compressed-parts, Microsoft workbooks) into comma separated value (CSV) text files. The NSW Forecast Demand data exceeded the 100 megabytes file size limit of GitHub's free account, so we stored this data in multiple files, each containing a year's data. This dataset was, however, excluded from our project.

Each dataset was then profiled using the [ProfileReport](#) class of the python [ydata_profiler](#) library. This generated a comprehensive report of each dataset in a prebuilt template. It provided us with an efficient means to understand datatypes, value ranges, frequencies, correlations, and missing values.

After conducting Exploratory Data Analysis, we decided to process all source data into a monthly grain, making appropriate computations on the base measures.

3.3.1 NSW Total Demand

The NSW Total Demand dataset comprised several fields, including Datetime, RegionId, and TotalDemand. All entries in the dataset corresponded to the NSW region by examining the distinct RegionId values, which were all identified as "NSW1". For our analysis, we focused solely on the Datetime and TotalDemand columns. The values observed on the TotalDemand column are all positive integers and non-zero values which are valid. The minimum and maximum values are also proportionate to the population, so we retained all observations. Since data for the year 2021 was incomplete, we would exclude this data in our modelling. *Total Demand at the 30-minute interval grain was finally represented as the Average Daily Maximum Demand at the monthly grain.* Our MTLF use case was sensitive to 'peak' demand, so we decided to use this measure as our primary response variable.

3.3.2 NSW Temperature

The NSW Temperature dataset had three fields: Location, Datetime, and Temperature. To ensure the accuracy of our analysis, we examined the minimum and maximum temperature values within the dataset, confirming their validity. We retained data points with temperatures below 0 degrees Celsius as they were considered valid readings. Notably, the Location column primarily features the value "Bankstown," denoting the specific suburb where temperature recordings were taken. Given that Bankstown is assumed to represent the entire state of New South Wales, we excluded the Location column from further analysis. Since data for the year 2021 was incomplete, we excluded this data in our modelling. *Temperature was finally represented as Degree Days at the monthly grain.* Degree Days was computed as the absolute value of the difference between the average daily temperature and the threshold temperature of 18 degrees Celsius.

3.3.3 Population

The source data for population has records from the year 1981. For this report, data from the year 2010 was utilised since the demand and temperature data only starts from the year 2010. The population data is in 3-month intervals, therefore data from 1-Mar-2010 is utilised when merging the demand with population data. We used a linear interpolation calculation to present the quarterly population data in a monthly grain.

3.3.4 Solar Photovoltaic (PV) Installed Capacity

Extracting solar PV installation capacity from the Microsoft Excel workbooks included some challenges, such as non-standard worksheet naming across the multiple workbooks. These variations needed to be manually accounted for and handled as exceptions in the code. A python function was created to map post codes to each state. The sourced data contained the capacity installed per month at each post code. We needed to create a cumulative total over time to represent total installed capacity per month in a time series. Only data points within the 2010 to 2020 range were used for modelling.

3.3.5 Gross State Product (GSP)

Gross State Product was made available at an annual grain per fiscal year, from June 1990 to June 2023. Literature review confirmed that interpolating this annual figure into a monthly grain could involve complex processing to be accurate [24]. We felt that having an accurate interpolation was not the focus of our analysis, so we decided to use a simple averaging routine to apportion an equal amount of GSP to each month in the financial year. Only data points within the 2010 to 2020 range were used for modelling.

We explored the use of multiple measures of temperature at the monthly grain through exploratory data analysis, including Mean Temperature, Maximum Temperature, Minimum Temperature, Average Daily Minimum Temperature, Average Daily Maximum Temperature, but finally decided to use Degree Days as our temperature indicator. Similarly, we explored using Mean Demand, Minimum Demand, Maximum Demand, but finally decided to use Average Daily Maximum Demand as our demand indicator due to its ability to reflect peak demand behaviour over the course of the month.

We finally prepared our data set for modelling that included the following variables:

<i>Variable</i>	<i>Column name</i>	<i>Unit of measure</i>	<i>Data type</i>	<i>Formula</i>	<i>Rationale</i>
<i>Average Daily Maximum Demand</i>	demand_avg_daily_max	Mega Watts	Floating point	Monthly Average of Daily Maximum Demand	Inherits the characteristics of 'peak' demand that is important for MTLF
<i>Degree Days</i>	degree_days	Degree Celsius	Floating point	Absolute Value of difference between Daily Average Temperature and threshold Temperature of 18 degree Celsius. See A-3 – Glossary of Terms for definition.	Inherits the characteristics of Heating Degree Days and Cooling Degree Days used by AEMO [21] since both heating and cooling drives electricity use.
<i>Installed Solar PV Capacity</i>	pv_capacity	Kilo Watts	Floating point	Cumulative total of individual monthly installed capacity for NSW	We needed a predictor that represented total installed capacity.
<i>Population</i>	population	Count	Integer	Interpolated quarterly count to monthly grain	Used as a predictor.
<i>Gross State Product</i>	gsp	Millions of Dollars	Floating point	Interpolated annual figure to monthly grain by dividing by 12.	Used as a predictor.

Table 2 - Final Processed Dataset Summary

3.5 Assumptions

The temperature data provided in the NSW dataset pertains to the Bankstown suburb only. For the purposes of this analysis, we extrapolate this temperature to represent the entirety of the New South Wales state.

Use of a simple equal apportionment of annual GSP and linear interpolation for Population to each month in the year is acceptable.

3.6 Modelling Methods

Figure 1 depicts the modelling methods and rationale adopted by us in this project.

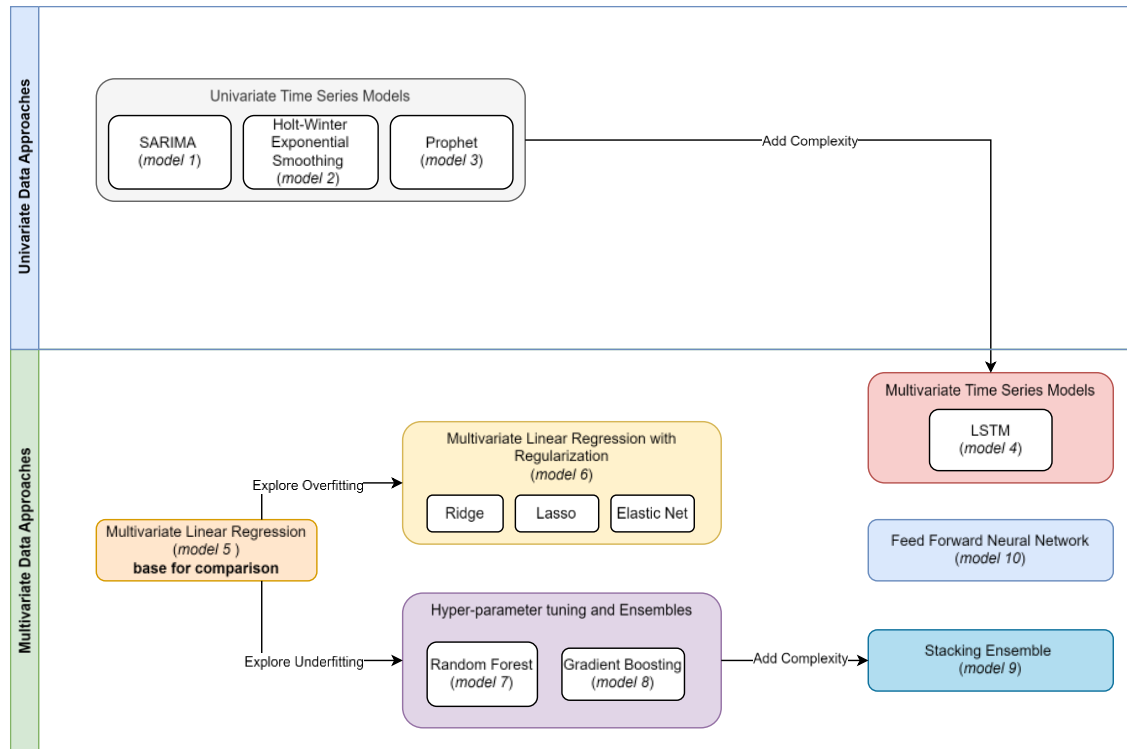


Figure 1- Modelling methods used in this project

We used a *Multivariate Linear Regression* model as the **base model for comparison** of other models.

We used 3 *Univariate Time Series* models to see if we could get comparable accuracy purely based on historical demand patterns. The models that were used were *Seasonal Auto-Regressive Integrated Moving Average (SARIMA)*, *Holt-Winter Exponential Smoothing* and *Prophet*. We then added complexity by using a *Long Short-Term Memory Recurrent Neural Network (LSTM)* Time Series approach. Literature review had highlighted the fact that ARIMA-based models were in use for electricity load forecasting for many years [8]. *Holt-Winter Exponential Smoothing* was deemed particularly successful for MTLF using data in the monthly grain [22]. Prophet was widely used for forecasting in industry, with some success shown in MTLF use-cases [18]. *LSTM* had successfully been used in STLF [2] and MTLF [15], so we were keen to explore its use.

We looked at the RMSE scores for Train and Test runs in our *Multivariate Linear Regression* model and validated our conclusion about absence of overfitting by exploring the results of adding *Regularization* using *Ridge*, *Lasso* and *Elastic Net Regression*.

Next, we observed the *Learning Curves* from our *Multivariate Linear Regression* model and explored the possibility of underfitting, by adding complexities in the form of *hyper-*

parameter tuning and *Ensembles* by running a *Random Forest Ensemble* model and *Gradient Boosting* model.

We then explored adding further complexity by using a *Stacking Ensemble* model to try to deliver better results.

Finally, guided by literature review, we used a *Feed Forward Neural Network* to see if we received comparable results.

Overall, we ran 10 models and compared the performance results of these to determine the optimal model for Medium Term Load Forecasting.

3.7 Model Accuracy Measures

The table below describes the performance measures used to assess our models:

<i>Measure</i>	<i>Abbreviation</i>	<i>Description</i>	<i>Rationale</i>
<i>Root Mean Squared Error</i>	RMSE	RMSE is the square root of the mean of the squared differences between the prediction and actual observation.	It tells us about the distribution of the residuals. A lower RMSE is indicative of a better fit for the data. It emphasizes larger errors over smaller ones, thus providing a more conservative estimate of model accuracy when large errors are particularly undesirable.
<i>Mean Absolute Percentage Error</i>	MAPE	MAPE is the mean of all absolute percentage errors between the predicted and actual values.	Lower values of MAPE indicate higher accuracy, while higher values indicate lower accuracy. It cannot be used when the actual values have instances of zero,

Table 3 - Model Accuracy Measures

4 Exploratory Data Analysis

This phase of the analysis helped us discover the relationships between the target variable and the features to better understand their impact on energy demand forecasting. These insights informed our understanding of the data. We used these insights to execute a feature selection.

4.1 Analysis of Independent Source Datasets

4.1.1. NSW Total Demand Dataset

We have used the pandas `describe` method to find the statistical interpretation of the demand dataset. The maximum total electricity demand in NSW between 2010 to 2021 is 14579.86 MW. The minimum total electricity demand in NSW during the same period is

5074.63 MW. 25% of the dataset have electricity demand less than 7150.07 MW. 50% of the dataset have electricity demand less than 8053.23 MW. 75% of the dataset have values less than 8958.55 MW. This suggests sporadic or infrequent instances of raised demand compared to the average and highlights the necessity for energy providers and grid operators to anticipate such occasional peaks in demand. The histogram plot depicted in Figure 2 explains the same. The mean electricity demand in the dataset is 8113.14 MW. The standard deviation was observed to be 1299.53 MW.

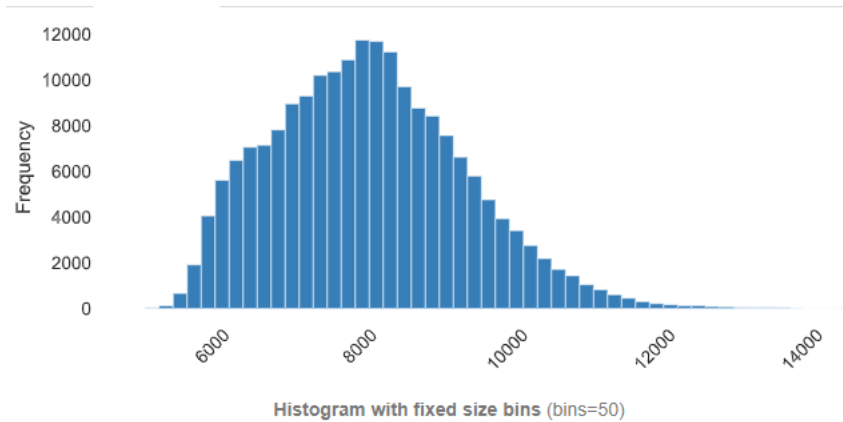


Figure 2 - Histogram of NSW Total Demand

4.1.2. NSW Temperature Dataset

We have used the pandas `describe` method to find the statistical interpretation of the temperature dataset. The maximum temperature recorded in Bankstown between 2010 to 2021 is 44.7°C. The minimum temperature recorded in Bankstown during the same period is -1.3°C. 25% of the dataset have values less than 13.4°C. 50% of the dataset have values less than 17.7°C. 75% of the dataset have values less than 21.3°C. The histogram plot depicted in Figure 3 explains the same. The mean temperature in the dataset was 17.42°C. The standard deviation was observed to be 5.84°C.

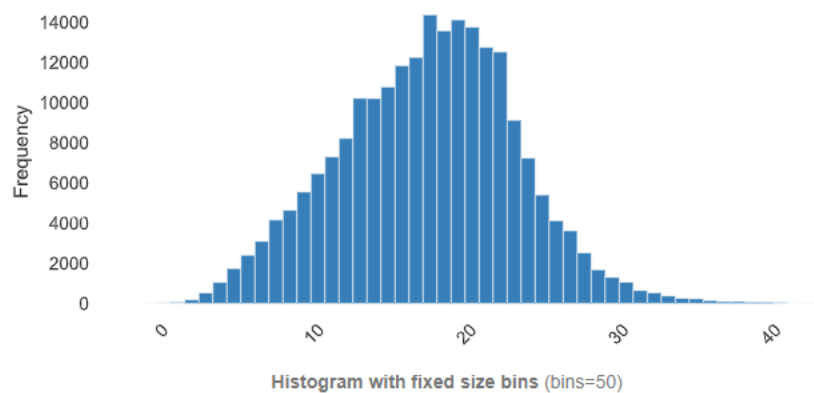


Figure 3 - Histogram plot of Temperature

4.1.3. Solar Photovoltaic (PV) Dataset

With 240 observations, the dataset displays substantial variability around the mean, indicating fluctuations in Solar PV installations. The distribution appears slightly more peaked than a normal distribution, hinting at occasional extreme values. Moreover, there is a significant concentration at zero frequency, indicating periods of no installations. The right-skewed distribution suggests increasing popularity in Solar PV installations, as ten-thousands of installations have been observed for the two-thirds of the sample. Additionally, the histogram exhibits a non-monotonic pattern, indicating that solar power installation does not follow a consistent trend or pattern throughout the observed period.

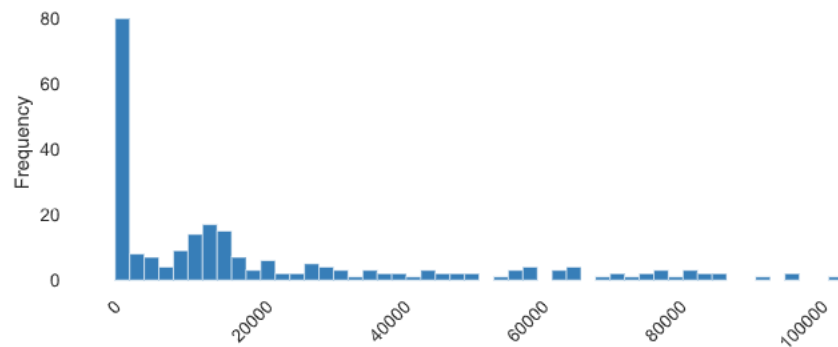


Figure 4 - Histogram of Solar PV Installations

Figure 5 shows that solar installations in NSW have increased from 2003 to 2022. After 2010, the solar installations have drastically increased. This increase is because the Australian solar power industry had installed more than 100,000 installations nationwide in 2020 compared to 80,000 installations between 2001 and 2009 [30].

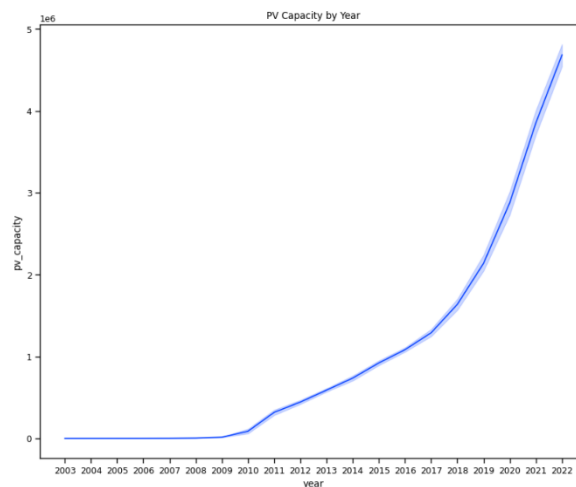


Figure 5 - PV Capacity by Year (2003-2022)

4.1.1. Population Dataset

Figure 6 shows the population growth as a monotonically increasing trend over the last 40 years.

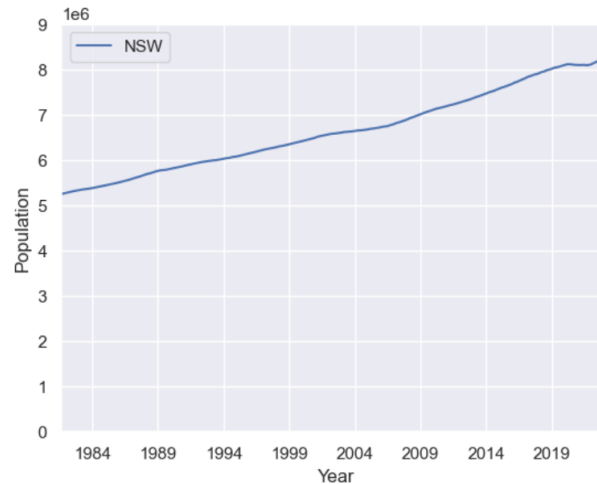


Figure 6 - Population Growth by Year (1984-2020)

4.1.1. Gross State Product (GSP) Dataset

Figure 7 shows Gross State Product for NSW also having a monotonically increasing trend over the last 30 years.

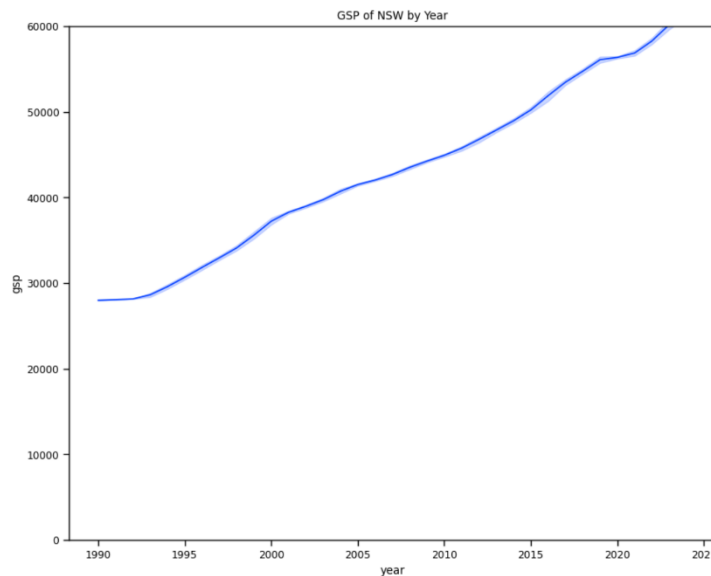


Figure 7 - Line Graph of GSP for NSW from 1990 -2022

4.2 Comparative Analysis of Features

The scatter plot in Figure 8 shows that the total electricity demand in NSW is observed to be lowest at approximately 18°C. This supports our degree days calculation which utilises

the threshold average temperature of 18°C. A higher demand for electricity is observed when the temperature increases or decreases from this threshold temperature of 18°C.

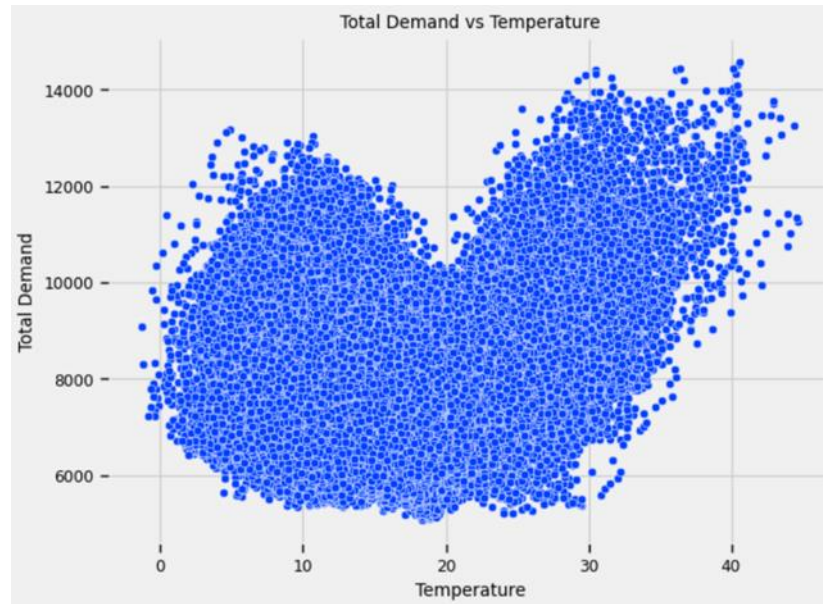


Figure 8 - Temperature vs Total Demand

Figure 9 shows the correlation between all variables used in this study. The exogenous variable, degree days is positively correlated to the independent variable, average daily maximum demand with a correlation of 0.787. Other exogenous variables, population and solar pv capacity, are negatively correlated to the average daily maximum demand with a correlation of 0.353 and 0.307 respectively. This correlation is also reflected in the scatter plots in Figure 10.

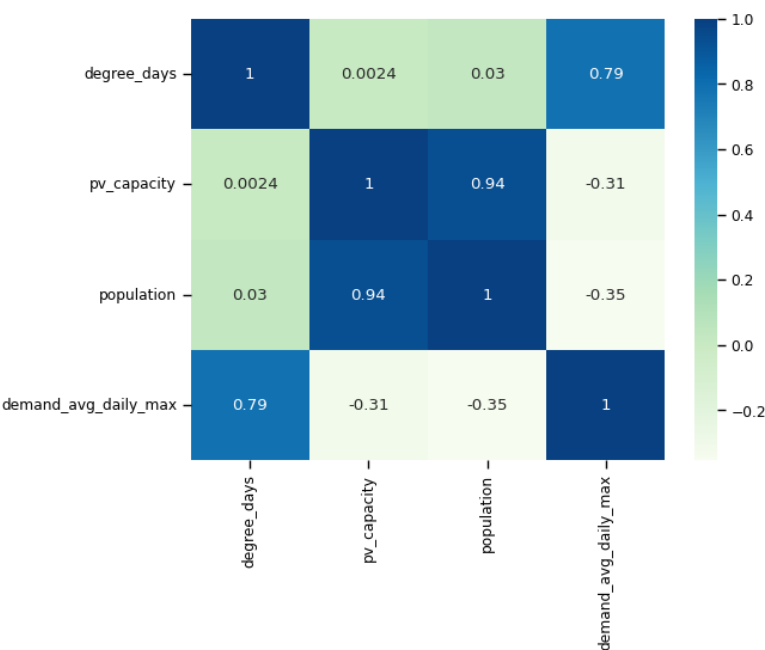


Figure 9 – Correlation Matrix Heat Map of Features

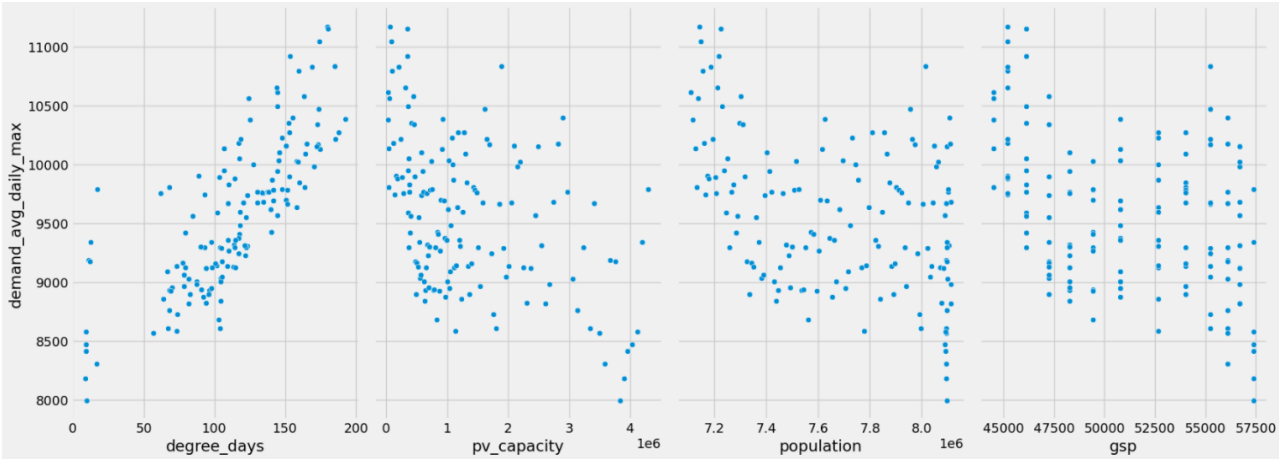


Figure 10 - Scatter Plot of Features

Figure 11 shows the trend of Average Daily Maximum Demand (on the left) compared to Degree Days (on the right). This shows that both variables display a similar trend in seasonality.

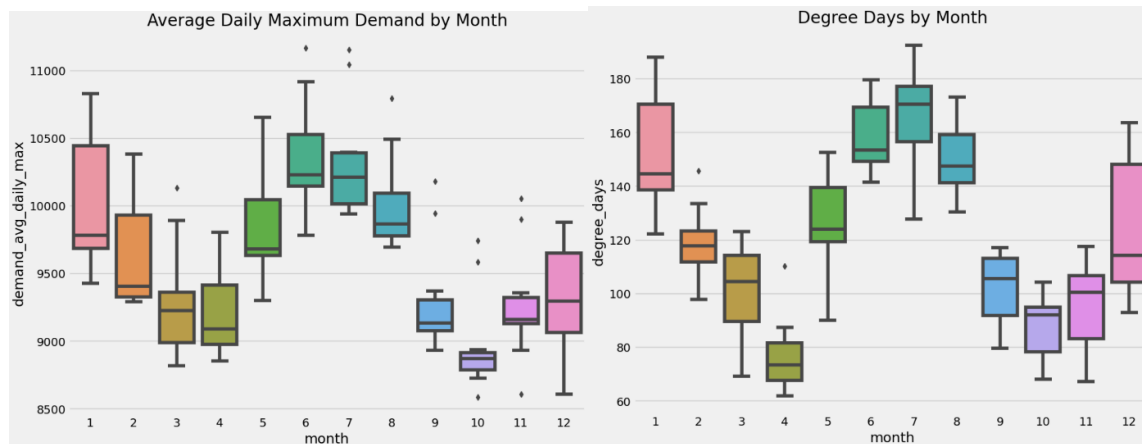


Figure 11 - Average Daily Maximum Demand & Degree Days by Month 2010-2020

Figure 12 shows the yearly seasonality of Average Daily Maximum Demand. It demonstrates a year on year decrease in demand. This could potentially be attributed to the uptake of distributed energy resources such as solar PV.

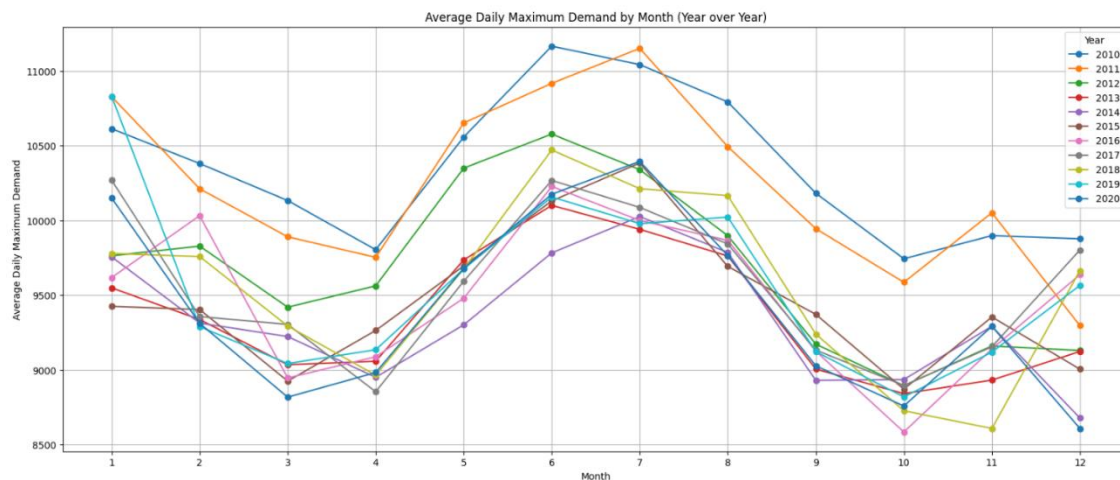


Figure 12- Average Daily Maximum Demand (Year on Year)

4.3 Feature Selection

Exploratory data analysis had revealed that basic temperature variants were not linearly related to the response variable, Average Daily Maximum Demand (`demand_avg_daily_max`). We, therefore, had decided to use Degree Days (`degree_days`) as our temperature measure.

To see the relative importance of predictors, we first looked at the correlation matrix for insights. This showed a good correlation of all predictors with the response variable, the strongest being with `degree_days`. Based on this insight, we did not drop any predictor.

Next, we looked at the results of *feature importance* provided by **XGBoost**. The results are displayed below:

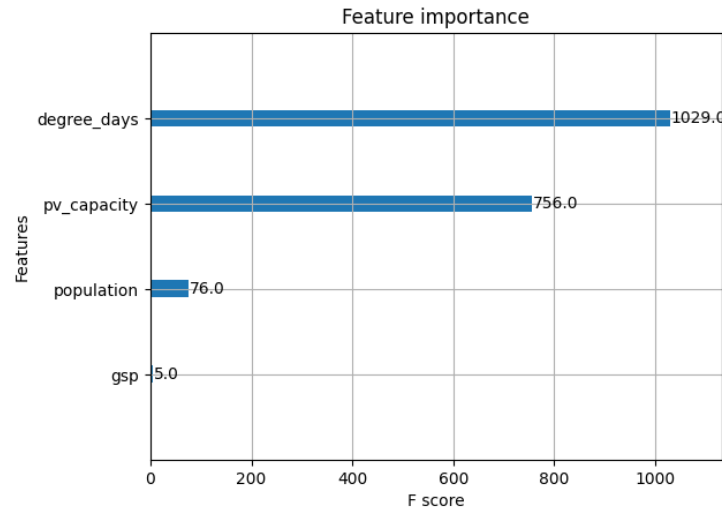


Figure 13- XGBoost Feature Importance (weight)

We also used the **scikit-learn** library's feature ranking with *Recursive Feature Elimination*. The results are displayed in the table below:

Variable	Selected	Rank
degree_days	True	1
pv_capacity	True	1
population	True	1
gsp	False	2

Table 4 - Result of scikit-learn Recursive Feature Elimination

Looking at the results, we decided to drop GSP as a predictor in our modelling. We attribute the low importance of GSP due to the simple method adopted by us to interpolate annual GSP into monthly GSP. This had produced equal values of GSP for each month of a year.

5 Analysis and Results

5.1 Time Series Methods for Forecasting

The following steps were performed to serve multiple time series models:

Step 1- Stationarity check of the time series: The *Augmented Dickey-Fuller (ADF)* test was used to determine the stationarity of the time series. *Akaike Information Criteria (AIC)* was used as the criterion for model selection.

Step-1 Results: The results of plot of time series trend, seasonality, and residuals for ADF test is shown below:

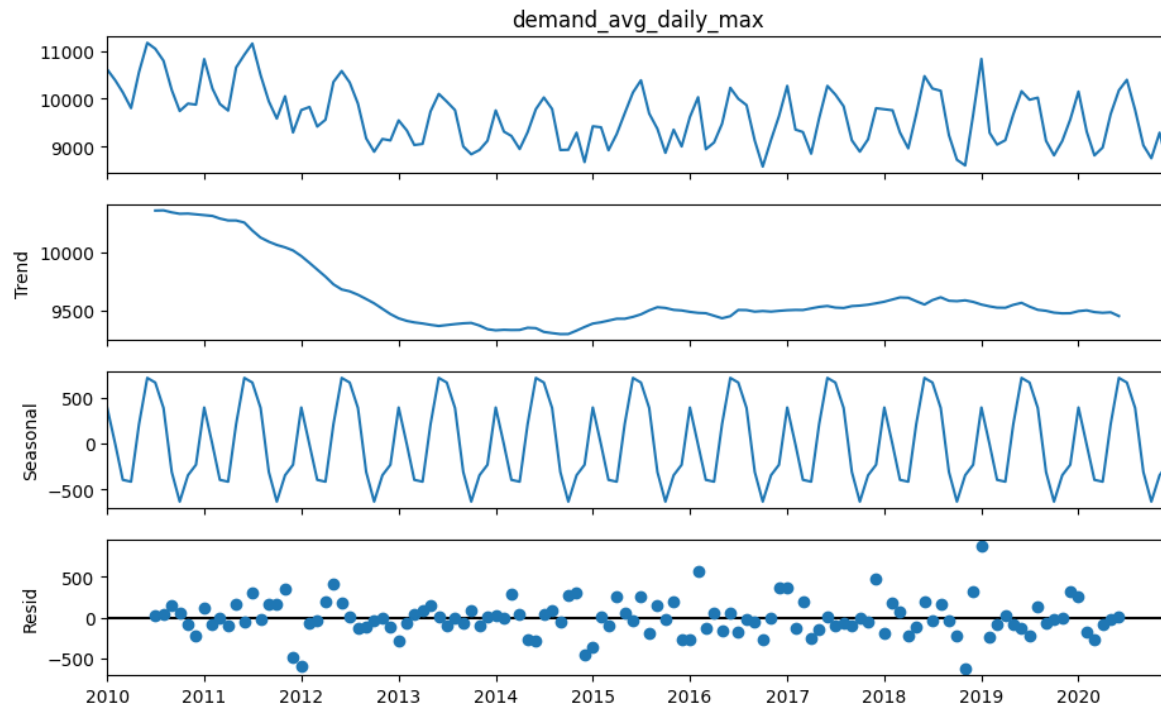


Figure 14 - ADF decomposition plot for Average Daily Maximum Demand 2010 – 2020

The ADF test produced a p-value of 0.088 and a Test Statistic of -2.623, while the 5% Critical Value was -2.886. Though the p-value is less than 0.05, the Test Statistic is greater than the 5% Critical Value, informing us that some differencing might be needed.

Step-2 - Data preparation for time series models: Due to incompleteness, monthly data for the year 2021 is excluded from the time series. The data was split into datasets for training and testing the model, keeping the last 24 months of monthly data as the test data to be used for forecasting.

Step-2 Results: There were 108 records used for training and 24 records for testing.

5.1.1 Seasonal Auto-Regressive Integrated Moving Average (SARIMA)

The following approach was used to apply SARIMA to the time series dataset:

Step-1 - Determination of optimal order parameters: Optimal seasonal and non-seasonal parameters were selected using the `auto_arima` (from `pmdarima`). We enabled the hyper-parameter, `seasonal=True`, to let `auto_arima` conduct the *Canova-Hansen* test to determine the optimal order of seasonal differencing, *D*, and then seek to identify optimal *P* and *Q* hyper-parameters. We set `start_p` and `start_q` to 0 (the default is

2) and left the `max_p` and `max_q` values to their default of 5. We asked `auto_arma` to use the ADF test to determine optimal order of differencing.

Step-1 Results: The results of the `auto_arma` selected optimal parameters were:

$p = 3, d = 1, q = 1, P = 1, D = 0, Q = 1$ for $m = 12$.

Step-2 - Fitting of SARIMA model using training dataset: We then fit the `SARIMA` model using the optimal parameters determined by `auto_arma`.

Step-2 Results: The results of the fit are shown in section A-2.1 [SARIMA model fit results](#).

Step-3 - Predicting forecast using fitted SARIMA model and test dataset: The `forecast` method of the model fit was applied to the test dataset to obtain a forecast for the years 2021 and 2022.

Step-3 Results: The results of forecasting Average Daily Maximum Demand for the 24 months from January 2019 to December 2020 are shown below.

SARIMA
Mean Absolute Percentage Error (MAPE) : 2.48%
Root Mean Square Error : 336.9956802350909

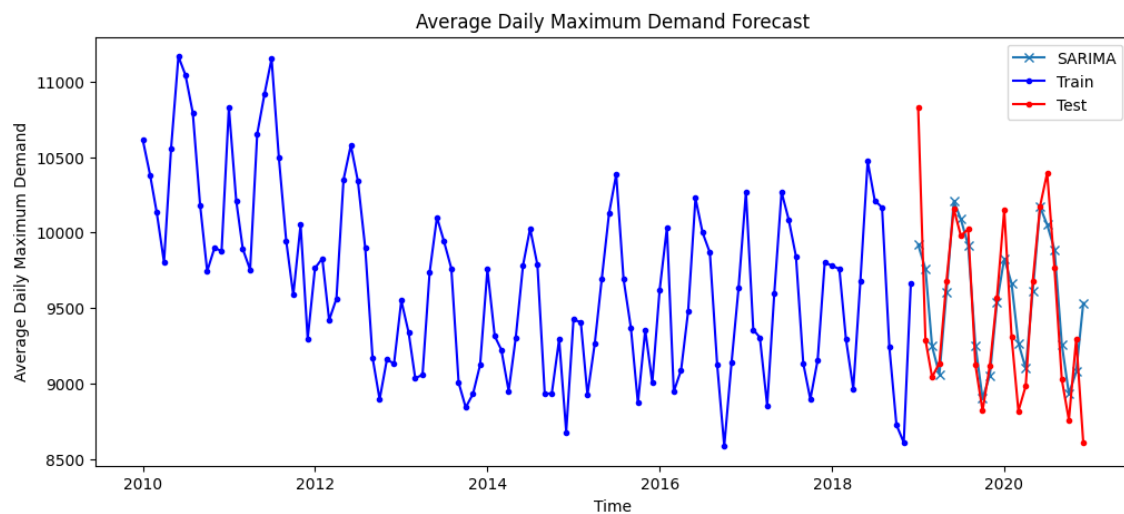


Figure 15 - SARIMA forecast results

Step-4 - Collecting model performance metrics: The *Mean Absolute Percentage Error (MAPE)* and *Root Mean Square Error (RMSE)* values for the model run were collected.

Step-4 Results: The model delivers a `MAPE` = 2.48% and `RMSE` = 336.996

5.1.2 Holt-Winter Exponential Smoothing

The following approach was used to apply Holt-Winter Exponential Smoothing to the time series dataset:

Step-1 - Fitting of Holt-Winter Exponential Smoothing model using training dataset:

We fit the [Exponential Smoothing](#) model using the following configuration parameters: `seasonal_periods = 12`, `trend = 'mul'`, `seasonal = 'mul'`. Choice of trend and seasonal types of 'multiplicity' were informed by the ADF test decomposition plots shown in Figure 14.

Step-1 Results: The results of the fit are shown in section [A-2.2 Holt-Winter model fit results](#).

Step-2 - Predicting forecast using fitted Holt-Winter model and test dataset: The forecast method on the [HoltWintersResults](#) object was applied to the test dataset to obtain a forecast for the years 2021 and 2022.

Step-2 Results: The results of forecasting Average Daily Maximum Demand for the 24 months from January 2019 to December 2020 are shown below.

Holt Winter
Mean Absolute Percentage Error (MAPE) : 1.93%
Root Mean Square Error : 306.2832932592546

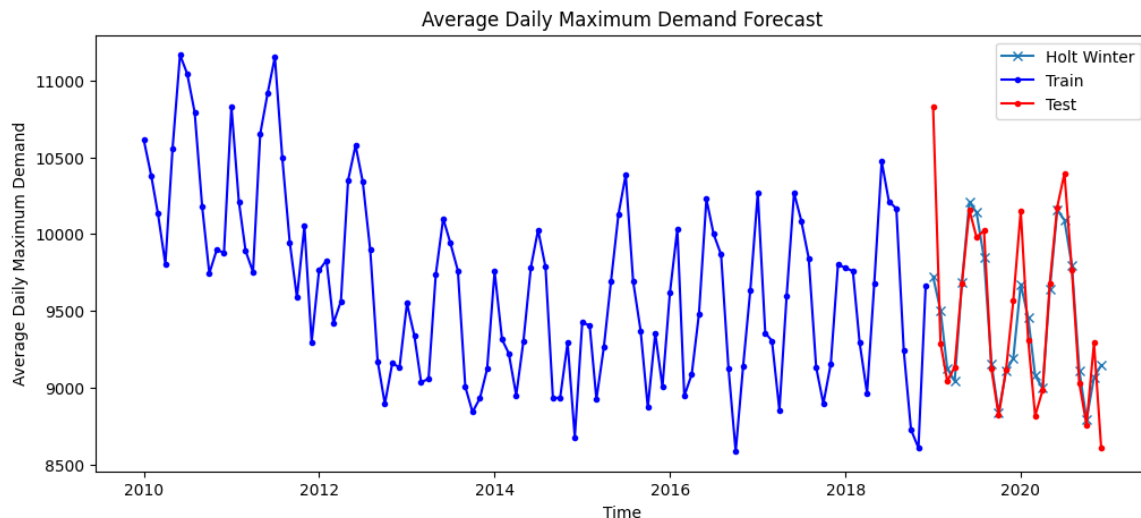


Figure 16– Holt-Winter Exponential Smoothing forecast results

Step-3 - Collecting model performance metrics: The Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) values for the model run were collected.

Step-3 Results: The model delivers a [MAPE = 1.93%](#) and [RMSE = 306.283](#).

5.1.3 Prophet

The following approach was used to apply Prophet to the time series dataset:

Step-1 - Data preparation for Prophet models: The prescriptive data structure and variable naming for Prophet was built.

Step-1 Results: The input train dataset with variable names 'ds' and 'y' and test dataset with variable name 'ds' was created.

Step-2 - Fitting of Prophet model using training dataset: We ran the Prophet model using the `Prophet` class from the python library `prophet`. The model was fitted using the following configuration parameters: `seasonality_mode='multiplicative'`, `daily_seasonality=False` and `weekly_seasonality=False`. All other input parameters were left to default values.

Step-2 Results: Prophet does not provide a 'summary' method for easily accessing the results of fit. The results were therefore observed in the context of predictions.

Step-3 Predicting forecast using fitted Prophet model and test dataset: The `predict` method was applied to model fit result object using the test dataset to obtain a forecast for the years 2021 and 2022.

Step-3 Results: The results of forecasting Average Daily Maximum Demand for the 24 months from January 2019 to December 2020 are shown below.

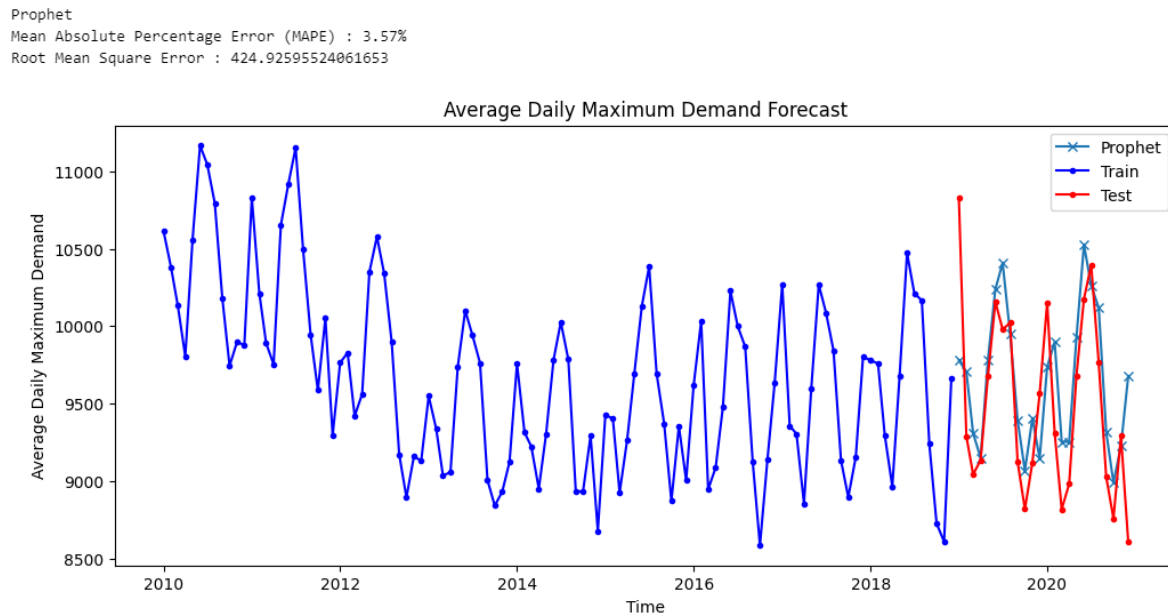


Figure 17 – Prophet forecast results

Step-4 - Collecting model performance metrics: The Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) values for the model run were collected.

Step-4 Results: The model delivers a `MAPE = 3.57%` and `RMSE = 424.926`.

5.1.4 Long Short-Term Memory (LSTM) Recurrent Neural Network

Strategy: The object that was used was the `keras.model` to create an LSTM model for the multivariate data. The optimizer used was 'Adam' and the loss calculation was `mean squared error`. The test data contains the last 24 months of data. This is to allow for sequential information to be trained. This was run over 50 epochs with a batch size of 32. The input variables were scaled using the minmax scaler between -1 and 1. This used the `sklearn.preprocessing.MinMaxScaler` routine.

Results: Below is the result of the model over the 50 epochs. The blue line is MSE loss from the training data and the orange for the test data.

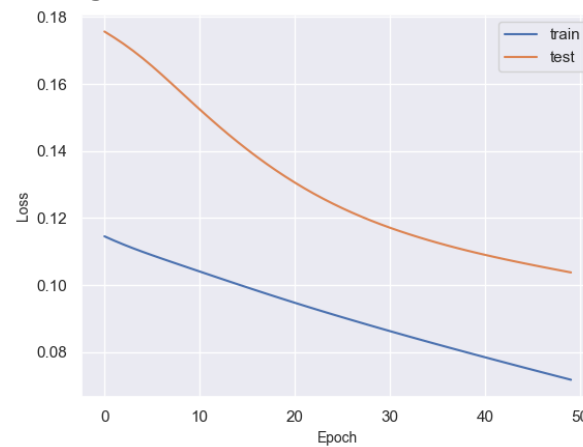


Figure 18 - MSE loss over Epochs in LSTM

The next figure shows the test data versus model prediction of demand (average daily max)

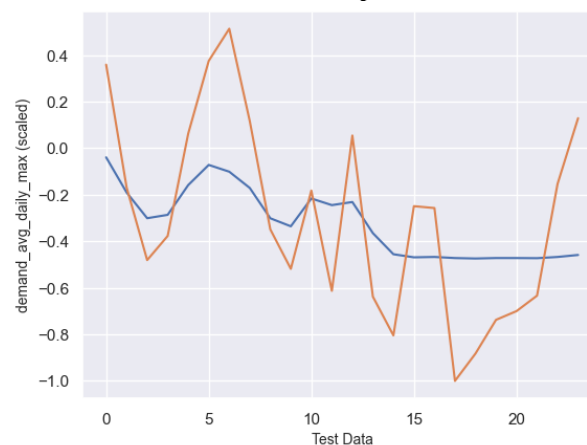


Figure 19- Actual versus Predicted plot for LSTM

For comparison with purposed models, the data is rescaled to produce the RMSE and MAPE figures.

Results: The model delivers a Test **MAPE = 24.3%** and **RMSE = 616.229**.

As you can see from the test data above figure above, the test predictions do not match the data very well. The prediction holds to the average of the demand over time and does not adapt to changes in changes based on the input variables. The RMSE and MAPE figures are remarkably high for this model.

5.2 Alternative Approaches to Time Series Methods for Forecasting

In addition to time series methods described earlier, we explored several alternative approaches for Forecasting Average Daily Maximum Demand.

Guided by feature selection described in section 4.3 [Feature Selection](#), the following variables were used in these processes. The dataset was at the monthly grain.

	Variable	Variable Type	Unit of Measure
1	Average Daily Maximum Demand	Response	Mega Watts (MW)
2	Degree Days	Predictor	Degree Celsius
3	Total Installed Solar PV Capacity	Predictor	Kilo Watts (KW)
4	Population	Predictor	Number

Table 5 - Variables used for Multivariate Forecasting

The following common approach was used to apply to multiple alternative models used:

Data preparation: Due to incompleteness, monthly data for the year 2021 is excluded from the dataset.

5.2.1 Linear Regression Model

The following approach was used to apply Linear Regression to the multivariate dataset:

Strategy: The following 4 exercises were carried out to investigate options for the best Linear Model. The `linear_model` class from the `sklearn` library was used for all exercises. The `train_test_split` function from sklearn's `model_selection` module was used for creating random sets of data for Training and Testing for each experiment. The `Normalizer` function from sklearn's `preprocessing` module was used for normalization of features.

1. Run 30 experiments using *all predictors without normalization*. Use a random split (80/20 Train/Test) for Train and Test sets of data per experiment. Calculate the Mean RMSE and MEAN MAPE across the 30 experiments as performance measures.
2. Same as exercise 1, above, except *with normalization* of predictors.
3. Run 30 experiments using *2 most highly correlated predictors without normalization*. Use a fresh random split (80/20 Train/Test) for Train and Test sets of data per experiment. Calculate the Mean RMSE and MEAN MAPE across the 30 experiments as performance measures.
4. Same as exercise 3, above, except *with normalization* of predictors.

Results: The performance measures for the 4 exercises are listed in the table below:

Exercise	No. of experiments	Normalize?	Predictors	Mean RMSE of Testing	Mean MAPE of Testing	Mean RMSE of Training	Mean MAPE of Training
1	30	No	degree_days, pv_capacity, population	289.249	2.47%	279.324	2.33%
2	30	Yes	degree_days, pv_capacity, population	270.267	2.22%	262.436	2.12%
3	30	No	degree_days, population	324.743	2.81%	317.708	2.75%
4	30	Yes	degree_days, population	612.336	4.70%	489.333	4.25%

Table 6 - Linear Regression Results

As illustrated in the table above, using all three predictors produces the best linear regression results. Normalization of predictors has improved this prediction.

The Learning Curve for the regression is shown below:

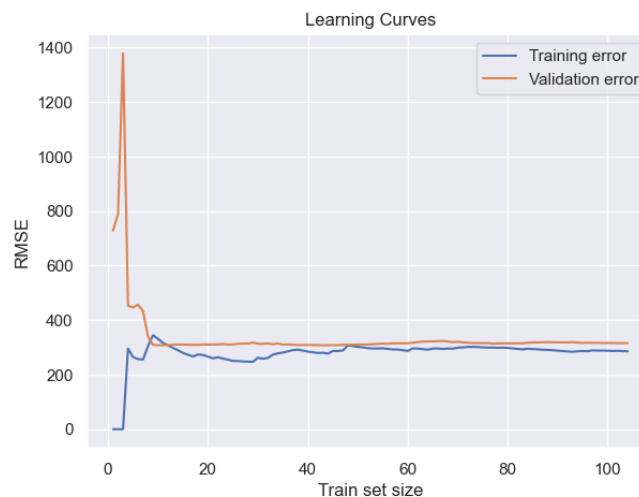


Figure 20 - Learning Curve for Linear Regression

Our exercises show Mean RMSE of Train and Test runs not having a significant difference: Mean RMSE of Train was 262.436 and of Test was 270.267. This leads us to conclude that overfitting is not a significant problem. We would like to explore the impact of Regularization to confirm this. Regularization treats overfitting.

5.2.2 Linear Regression Model with Regularization

The following approach was used to apply Regularization to the Linear Regression:

Strategy: *Ridge, Lasso and Elastic Net Regression* was carried out on the dataset. Hyper-parameter tuning using scikit-learn library's [GridSearchCV](#) module. The following sets of alpha hyper-parameters were used for each of the regularization models:

Alpha values for regularization: 0.001,0.005,0.01,0.05,0.1,0.5,0.99

Results: The performance results of regularization are shown in the table below:

Regularization method	Test RMSE	Test MAPE
Ridge	309.306	2.56 %
Lasso	307.374	2.57 %
Elastic Net	313.476	2.57 %

Table 7 – Regularization Results

The poorer performance using regularization confirms our suspicion that overfitting is not a significant problem.

5.2.3 Random Forest Model

The following approach was used to check the underfitting of data and to check if we can improve performance by utilising a more complex model.

Strategy: *Random Forest Regression* was carried out on the dataset. Hyper-parameter tuning using scikit-learn library's [GridSearchCV](#) module. The following sets of alpha hyper-parameters were used for each of the regularization models:

Minimum sample split: 2,4,6

Minimum samples leaf: 1,2,3

Results: The model delivers a Test [MAPE](#) = 2.02% and [RMSE](#) = 240.001.

The model performed better than the linear regression model.

5.2.4 Gradient Boosting Regression Model

The following approach was used to check the underfitting of data and to check if we can improve performance by utilising a more complex model.

Strategy: *Gradient Boosting Regression* was carried out on the dataset. Hyper-parameter tuning using scikit-learn library's [GridSearchCV](#) module. The following sets of alpha hyper-parameters were used for each of the regularization models:

Alpha values for Gradient Boosting Regression: 0.001,0.005,0.01,0.05,0.1,0.5,0.99

Results: : The model delivers a [MAPE](#) = 2.31% and [RMSE](#) = 277.030.

The model performed better than the linear regression model. Random Forest Regression model performed better than Gradient Boosting model.

5.2.5 Stacking Ensemble Model

Strategy: The base learners used in our study are KNearestNeighbour (knn) model, Decision Tree Regressor (cart) Model, Random Forest (rf) Regressor Model, Gradient Boosting(gb) Regressor Model, and Support Vector Regressor(svr) model. The predictions obtained from these base learner models are fed to the meta learner, Linear Regression Model. The overall strategy used for Stacking Ensemble (Figure 21) is depicted below:

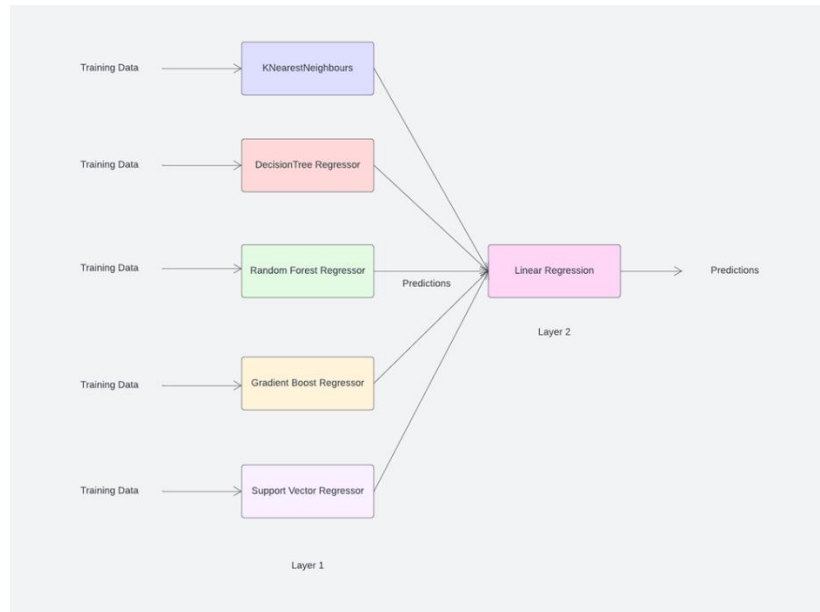


Figure 21 - Stacking Ensemble Strategy

Each base model was evaluated using cross validation. We have utilised [Repeated K-Fold](#) cross validator available on scikit-learn library's model selection module. Repeater K-fold cross validator repeats K-fold n times with distinct randomization in each repetition. *The parameter used for Repeated K-Fold cross validation are as follows:*

n_splits: [10](#)

n_repeats: [3](#)

random_state: [1](#)

Cross validation score for each model is estimated using [cross_val_score](#) available on scikit-learn library's model selection module. The parameters used for cross validation score estimation are as follows:

Estimator for each model: [KNeighborsRegressor](#), [DecisionTreeRegressor](#), [RandomForestRegressor](#), [GradientBoostingRegressor](#), [SVR](#)

Scoring: [neg_mean_absolute_error](#)

cv: [Repeated K-Fold](#)

n_jobs: [-1](#)

error_score: `raise`

The next step in ensemble learning is stacking the base estimators with a final regressor.

We have utilised a scikit-learn ensemble module called `Stacking Regressor` for stacking the output of each individual estimator and used Linear Regression model to compute the final prediction. The usage of each individual estimator's strength through a process of stacking enables utilising their output as input of the final estimator. The final estimator, Linear Regression model, is trained on cross validated predictions of the base estimators.

Results: The negative mean absolute error and its standard deviation obtained for each base estimator model and stacking are shown in the table below. Figure 22 shows boxplot of the scoring results for each estimator model and stacking ensemble.

Models	Scoring (negative mean absolute error)	Standard Deviation
KNeighbors Regressor	-245.989	58.313
Decision Tree Regressor	-252.108	56.727
Random Forest Regressor	-212.625	57.584
Gradient Boosting Regressor	-208.101	65.456
Support Vector Regressor	-524.214	80.028
Stacking	-213.696	62.949

Table 8 - Stacking Ensemble Model Results

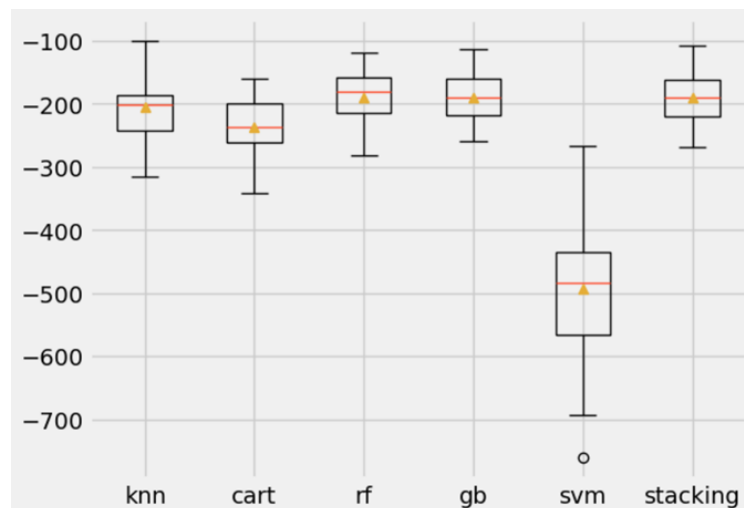


Figure 22 - Scoring Results for each Estimator Model and Stacking Ensemble

The residual plot (Figure 23) and a regression plot showing Predicted vs Actual (Figure 24) readings of the stacking ensemble are shown below.



Figure 23 – Residual Plot for the Stacking Ensemble

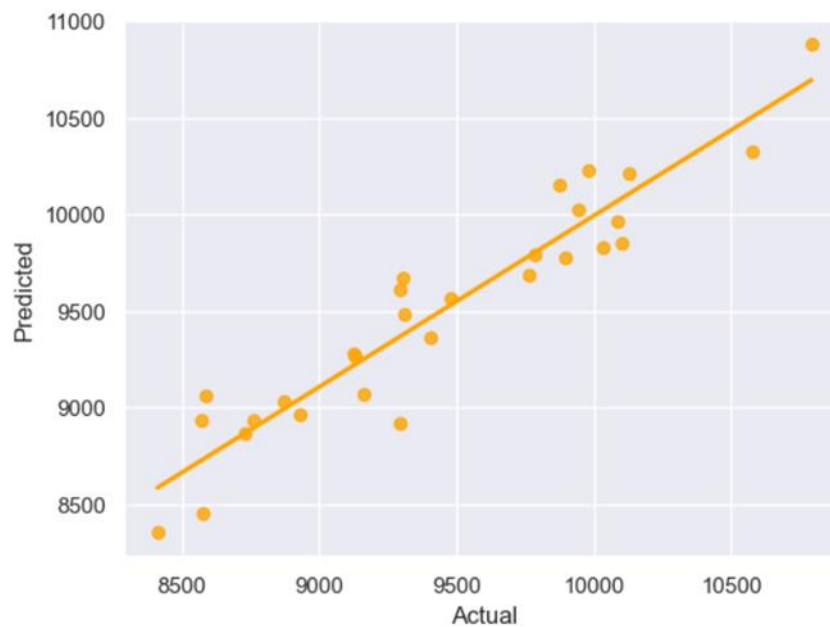


Figure 24 – Predicted vs Actual Readings for the Stacking Ensemble

Overall Results for Stacking Ensemble: The model delivers a Test $\text{MAPE} = 1.87\%$ and $\text{RMSE} = 210.08$.

5.2.6 Fast Forward Neural Network Model

The following approach was used to apply Neural network to the multivariate dataset:

The object was to use the `keras.model` to create a neural network of several layers to test the multivariate date.

The optimizer used was 'Adam' and the loss calculation was `mean squared error`. The train to test ratio was 80/20 using the `sklearn.model_selection.test_train_split` routine.

The model was run on various levels on hidden layers using different amounts of hidden nodes per layer.

Below are the results from the testing data:

Hidden Nodes	Layer Depth	MAPE	RMSE
40	3	0.027	333
60	3	0.028	338
80	3	0.026	299
40	4	0.027	302
60	4	0.029	337
80	4	0.028	320
40	5	0.023	275
60	5	0.027	325
80	5	0.024	288
40	6	0.03	377

Table 9 - Feed Forward Neural Network Test Results

From these results it was decided to use 5 hidden layers with 40 nodes per layer.

This was run over 50 epochs with a batch size of 32. The input variables were scaled using the minmax scaler between -1 and 1. This used the `sklearn.preprocessing.MinMaxScaler` routine.

Below is the result of this model over the 50 epochs. The blue line is MSE loss from the training data and the orange for the test data.

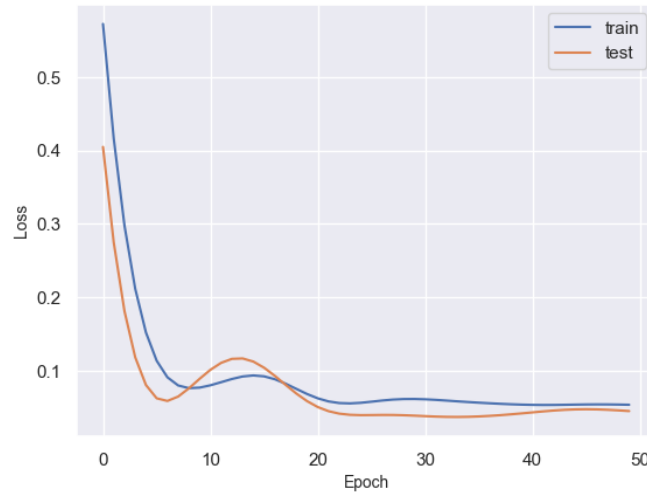


Figure 25 - MSE Loss over Epochs in FNN

The next figure shows the test data versus model prediction of demand (average daily max)

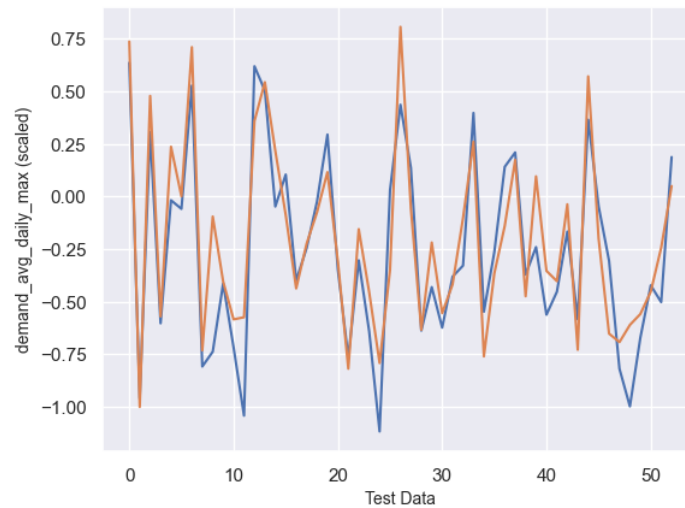


Figure 26 - FNN Prediction vs Actual Results

For comparison with purposed models, the data is rescaled to produce the RMSE and MAPE figures. The model delivers a Test **MAPE = 2.50%** and **RMSE = 284.563**.

5.7 Results Summary

The following table illustrates the results of prediction across all models explored in our project:

Model No.	Model Type	Test RMSE	Test MAPE	Characteristic of method
1	SARIMA	336.996	2.48 %	Univariate Time Series
2	Holt-Winter Exponential Smoothing	306.283	1.93 %	Univariate Time Series
3	Prophet	424.926	3.57 %	Univariate Time Series
4	Long Short-Term Memory Recurrent Neural Network	616.229	24.3 %	Time Series with Exogenous Variables using Recurrent Neural Network
5	Linear Regression	270.267	2.22 %	Multivariate Linear Regression
6	Linear Regression with Regularization	307.374	2.57 %	Multivariate Linear Regression
7	Random Forest Regression	240.001	2.02 %	Decision Tree Ensemble
8	Gradient Boosting Regression	277.030	2.31 %	Decision Tree with Gradient Boosting
9	Stacked Ensemble	210.078	1.87 %	Multi-model Ensemble
10	Feed Forward Neural Network	284.563	2.50 %	Simple Neural Network

Table 10 - Summary of Performance Measures for Modelling Average Daily Maximum Demand

6 Discussion

In this project we used a *Multivariate Linear Regression* model as the base benchmark against which to compare other modelling approaches. At a higher level, the *Linear* model delivered a reasonable result, with a MAPE of 2.22% and Mean Test RMSE of 270.267.

We were keen to explore *Univariate Time Series* approaches, based on our literature review and the inherent trends and seasonality in the demand data. The *Holt-Winter Exponential Smoothing* model delivered a good MAPE of 1.93% but poorer Test RMSE of 306.283. RMSE is susceptible to outliers, and it is likely that the outliers in the first and last month of the forecast range are responsible for this. This satisfactory performance of the *Holt-Winter* model for predicting monthly electricity forecasts is aligned with the research of Jiang et al [22]. The poorer performance of *SARIMA* (RMSE of 336.996 and MAPE of 2.48%), coupled with our results in multivariate approaches, suggests that the addition of exogenous variables (*SARIMAX*) might be an option to explore in the future. *Prophet* performed comparatively poorly (RMSE of 424.926 and MAPE of 3.57%). We used *Prophet* with default input parameters that are designed for simplicity and speed. Its performance could be improved by increasing complexity through adjusting seasonality settings and hyper-parameter tuning in the future.

LSTM results are an outlier amongst the models that we trained and used for predictions in our project, delivering the poorest performance scores of all. This is not in line with our literature review that showed that *LSTMs* have delivered promising results in electricity load forecasting. Our *LSTM* produced an RMSE of 616.229 and MAPE of 24.3%. We attribute this inferior performance to our lack of experience in tuning this model appropriately. There are methods for adding longer strings of X variables into the model each time (e.g. this month, last month and the month before) which were not explored and may be something to look at in the future.

It was evident from our literature review that weather, rooftop solar uptake and demographic variables like population and economic activity influence electricity demand. We have demonstrated that this is, indeed, true, as all multivariate approaches have outperformed univariate time series methods in our project, except for *LSTM*.

Our *Linear Regression* exercises showed Mean RMSE of Train and Test runs not having a major difference: Mean RMSE of Train was 262.436 and of Test was 270.267. This informed us that the impacts of overfitting were not significant. Nevertheless, we ran *Ridge*, *Lasso* and *Elastic Net regularization* on the *Linear Regression* to see if they improved results. *Ridge Regression* produced a RMSE of 309.306, *Lasso Regression* produced a RMSE of 307.374, and *Elastic Net* delivered a RMSE of 313.476. The degraded performance confirmed our analysis that overfitting was not significant in the *Linear* model and regularization was not required.

The *Learning Curve* for the *Linear Regression* (Figure 20) led us to explore the possibility of underfitting due to the RMSEs converging around the 260 MW mark. We therefore looked at compensating for potential underfitting by increasing complexity using *Decision Tree Ensemble* and *Gradient Boosting* methods with *GridSearch Cross Validation* for hyper-

parameter tuning. We felt that increasing complexity might increase the performance of models.

Gradient Boosting delivered an RMSE of 277.030 with a MAPE of 2.32%. This was comparable to the Linear Regression model (RMSE of 270.267). The *Random Forest (Decision Tree Ensemble)*, however, gave us much improved performance, with RMSE of 240.001 and MAPE of 2.02.

Guided by literature review, we decided to try a simple *Feed Forward Neural Network* (FNN) approach on the multivariate dataset. The performance of the FNN, though better than the time series modelling results, was poorer than the *Linear Regression* benchmark, delivering an RMSE of 284.563 and a MAPE of 2.50%. This is aligned to the insight provided to us by Oliver Nunn from Endgame Economics at the guest lecture in this course: he had said that electricity load forecasting is not a problem ideally solved by neural networks. It is also likely that in aggregating our dataset to a monthly grain we reduced the number of data points available to the neural network to learn from, thus degrading the accuracy of predictions. Aggregating data to the daily or weekly grain could be considered to explore better performance of the Neural Network in the future.

The promising results of the *Decision Tree Ensemble* approach led us to try adding additional complexities by using a strategy based on a *Stacking Ensemble* learning scheme, where predictions produced by multiple base learning methods are used by a higher-level method to produce final predictions. This decision was also guided by literature review wherein Divina et al have successfully used this approach for producing accurate predictions for electricity consumption forecasting [23]. The *Stacking Ensemble* approach has produced our overall best performance scores, with RMSE of 210.078 and MAPE of 1.87%.

We believe that the two-layered Stacking Ensemble approach yielded better results because it provides better generalizations compared to the ‘single strategy’ approaches. This approach can adapt to unseen cases, has a better capability of escaping from local optima and has superior search capabilities.

Here are a few things that we would have done had we more time to work on this project:

- Explore other approaches for LSTM modelling.
- Add exogenous variables to the SARIMA model (SARIMAX).
- Make better use of economic indicators: GSP is made available at an annual grain; we could explore using more complex or appropriate methods for interpolating GSP values to a monthly grain, such as the process described by the Department of Treasury and Finance Victoria [24].
- Use 2 independent temperature variables, Heating Degree Days (HDD) and Cooling Degree Days (CDD), as used by the Australian Energy Market Operator [21], rather than the combined measure, Degree Days (DD) used in our project so far.

7 Conclusion and Further Work

In this project we explored optimal methods for Medium Term Load Forecasting (MTLF) for electricity for the state of New South Wales, Australia. We were motivated by our literature review that indicated that out of the typical time horizons for electricity load forecasting (i.e., Short Term, Medium Term and Long Term), MTLF suffered from a significantly less amount of research [17]. MTLF is critical for grid operators and participants in the electricity market to ensure reliability of electricity supply in the 24-month forecast horizon.

In the MTLF context, intra-day or intra-week variations in demand are not critical to model. We therefore prepared our base dataset at the monthly grain. Grid reliability would be determined by the ability to predict ‘peak’ demand. We therefore used the Average Daily Maximum Demand (for each month) as our response variable. We simplified our model by collapsing the typical predictors for cooling (Cooling Degree Days) and heating (Heating Degree Days) based electricity consumption into a single predictor called Degree Days. In the MTLF context, however, the extent of influence of demographic and economic indicators is not as clear as in the LTLF context. We included Solar PV Installation Capacity and Population as predictors.

Our project results have demonstrated that the MTLF problem is best served by a two-tier *Stacking Ensemble* approach. We attribute this to its ability to adapt to unseen cases and escape from local optima, as well as its superior search capability.

Our recommendations for grid operators and participants in the electricity market from a MTLF perspective are as follows:

- MTLF is a multivariate data problem. Univariate time series approaches, though fairly accurate, do not perform as well as multivariate approaches.
- Use a Stacking Ensemble approach to get superior results.
- Use of temperature indicators, Solar PV capacity installation and population as predictors are recommended.

Future studies could improve on our approach in the following ways:

- Use Hybrid models. Like our Stacking Ensemble approach, there are other, more complex means of assembling multiple models together to optimize results. An example of this is the approach used by Jin et al [15].
- Use different models in the Stacking Ensemble approach.
- Use finer grained aggregations of demand data, such as weekly or daily data points to improve time series results as well as neural network performance.
- Expand the scope of the project to cover all regions of the NEM to provide comprehensive MTLF support to the market operator and participants that have cross-regional presence.

References

1. Sharma, N, Smeets, B, Tryggestad, C 2019, 'The decoupling of GDP and energy growth: A CEO guide', *McKinsey Quarterly*, accessed 18 March 2024, <<https://www.mckinsey.com/industries/electric-power-and-natural-gas/our-insights/the-decoupling-of-gdp-and-energy-growth-a-ceo-guide>>.
2. Bouktif, S, Fiaz, A, Ouni, A, Serhani, MA 2020, 'Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting', *MDPI Energies*, 13, 391, DOI:10.3390/en13020391.
3. Chapagain, K, Gurung, S, Kulthanavit, P, Kittipiyakul, S 2023, 'Short-Term Electricity Demand Forecasting Using Deep Neural Networks: An Analysis for Thai Data', *MDPI Applied System Innovation*, 6,100, pp. 1-29.
4. Hong, T, Fan, S 2016, 'Probabilistic electric load forecasting: A tutorial review', *International Journal of Forecasting*, vol. 32, no. 3, pp. 914-938, DOI: 10.1016/j.ijforecast.2015.11.011.
5. Harun, MHH, Othman, MM, Musirin, I 2010, 'Short term load forecasting (STLF) using artificial neural network based multiple lags and stationary time series', *2010 4th International Power Engineering and Optimization Conference (PEOCO)*, pp. 363-370, DOI: 10.1109/PEOCO.2010.5559238.
6. Hippert, HS, Pedreira, CE, Souza, RC 2001, 'Neural networks for short-term load forecasting: a review and evaluation', *IEEE Transactions on Power Systems*, vol. 16, pp. 44-55.
7. IBM Corporation 2021, *CRISP-DM Help Overview*, accessed 17 March 2024, <<https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview>>.
8. Semekonawo, KP, Kam, S 2022, 'Linear Regression and ARIMA Models for Electricity Demand Forecasting in West Africa', *Journal of Energy Research and Reviews*, vol. 12, no. 2, pp. 26-36, DOI: 10.9734/jenrr/2022/v12i230296.
9. Shi, H, Xu, M, Li, R 2018, 'Deep Learning for Household Load Forecasting—A Novel Pooling Deep RNN', *IEEE Transactions on Smart Grid*, vol. 9, pp. 5271-5280.
10. University of New South Wales 2024, 'Project Brief: Energy forecast', *ZZSC9020 Data Science Project, Hexamester 2, 2024*, accessed 16 March 2024, <<https://moodle.telt.unsw.edu.au/mod/page/view.php?id=6472271>>.
11. Soliman, SA, Al-kandari, AM 2010, *Electrical Load Forecasting - Modeling and Model Construction*. Elsevier, Burlington.

12. Kyriakides, E, Polycarpou, M 2007, 'Short term electric load forecasting: A tutorial', *Trends in Neural Computation*, Springer, Berlin, pp. 391–418.
13. McGrath, A, Jonker, A 2024, 'What is load forecasting?', *IBM Think 2024, Topics*, accessed 6 April 2024, <<https://www.ibm.com/topics/load-forecasting>>.
14. Shirzadi, N, Nizami, A, Khazen, M, Nik-Bakht, M 2021, 'Medium-Term Regional Electricity Load Forecasting through Machine Learning and Deep Learning', *MDPI Designs* 2021, vol. 5, no. 27, accessed 6 April 2024, <<https://doi.org/10.3390/designs5020027>>.
15. Jin, B, Zeng, G, Lu, Z, Peng, H, Luo, S, Yang, X, Zhu, H, Liu, M 2022, 'Hybrid LSTM–BPNN-to-BPNN Model Considering Multi-Source Information for Forecasting Medium- and Long-Term Electricity Peak Load', *MDPI Energies* 2022, accessed 6 April 2024, <<https://doi.org/10.3390/en15207584>>.
16. Shah, I, Iftikhar, H, Ali, S 2020, 'Modeling and Forecasting Medium-Term Electricity Consumption Using Component Estimation Technique', *MDPI Forecasting* 2020, accessed 7 April 2024, <https://mdpi-res.com/forecasting/forecasting-02-00009/article_deploy/forecasting-02-00009-v2.pdf?version=1590426570>.
17. Khuntia, SR, Rueda, JL, van der Meijden, M 2016, 'Forecasting the load of electrical power systems in mid- and long-term horizons: a review', *The Institution of Engineering and Technology Journals Generation, Transmission, Distribution*, vol. 10, no. 16, pp. 3971-3977.
18. Long, C, Yu, C, Li, T 2022, 'Prophet-Based Medium and Long-Term Electricity Load Forecasting Research', *Journal of Physics: Conference Series*, vol. 2356, 2022 International Conference on Electrical, Electronics and Information Engineering, DOI: 10.1088/1742-6596/2356/1/012002.
19. Oliveira, FS, Ruiz, C, Conejo, AJ 2013, 'Contract design and supply chain coordination in the electricity industry', *European Journal of Operational Research*, vol. 227, no. 3, pp. 527-537.
20. Taylor, S, Letham, B 2018, 'Forecasting at scale', *The American Statistician*, vol. 72, no. 1, pp. 37-45, DOI: 10.1080/00031305.2017.1380080.
21. Australian Energy Market Operator 2023, 'Forecasting Approach – Electricity Demand Forecasting Methodology', *NEM Electricity Statement of Opportunities Forecast 2023*, accessed 23 March 2024, <https://aemo.com.au/-/media/files/electricity/nem/planning_and_forecasting/nem_esoo/2023/forecasting-approach_electricity-demand-forecasting-methodology_final.pdf>
22. Jiang, W, Wu, X, Gong, Y, Yu, W, Zhong, X 2020, 'Holt–Winters smoothing enhanced by fruit fly optimization algorithm to forecast monthly electricity consumption',

- Energy*, Elsevier, vol. 193, no. 116779, accessed 18 April 2024, <<https://www.sciencedirect.com/science/article/pii/S0360544219324740>>
23. Divina, F, Gilson, A, Gomez-Vela, F, Garcia Torres, M, Torres, J 2018, 'Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting', *MDPI Energies* 2018, accessed 18 April 2024, <<https://www.mdpi.com/1996-1073/11/4/949>>.
24. Li, B, Gao, G 2021, 'Estimating Quarterly Gross State Product', *Victoria's Economic Bulletin*, vol. 5, Department of Treasury and Finance Victoria, accessed 15 March 2024, < <https://www.dtf.vic.gov.au/victorias-economic-bulletin-volume-5/estimating-quarterly-gross-state-product>>
25. Divina, F. et al., 2018. Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting. *Energies*, 11(949), pp. 3-31.
26. Australian Bureau of Statistics 2023, 'National, state and territory population', *Statistics, People, Population*, Latest Release, accessed 8 March 2024, <<https://www.abs.gov.au/statistics/people/population/national-state-and-territory-population/sep-2023/310104.xlsx>>
27. Australian Government Clean Energy Regulator 2024, 'Small-scale installation post code data', *Clean Energy Regulator Reports and Data*, accessed 8 March 2024, <<https://cleanenergyregulator.gov.au/markets/reports-and-data/small-scale-installation-postcode-data#postcode-data-files>>
28. Australian Bureau of Statistics 2023, 'Gross State Product, chain volume measures and current prices', *Australian National Accounts: State Accounts*, Latest Release, accessed 8 March 2024, < <https://www.abs.gov.au/statistics/economy/national-accounts/australian-national-accounts-state-accounts/latest-release>>
29. National Weather Service, 'What are Heating and Cooling Degree Days', *National Oceanic and Atmospheric Administration*, accessed 18 March 2024, <https://www.weather.gov/key/climate_heat_cool>
30. Martin, J., 2024. *2010 an extraordinary year for Aussie solar power*. [Online] Available at: <https://www.solarchoice.net.au/blog/solar-power-rebate-australia-extraordinary-year/>

Appendices

A-1 List of Tables

Table 1 - Source Data File Summary.....	7
Table 2 - Final Processed Dataset Summary	10
Table 3 - Model Accuracy Measures	12
Table 4 - Result of scikit-learn Recursive Feature Elimination.....	19
Table 5 - Variables used for Multivariate Forecasting.....	25
Table 6 - Linear Regression Results.....	26
Table 7 - Regularization Results.....	27
Table 8 - Stacking Ensemble Model Results	29
Table 9 - Feed Forward Neural Network Test Results	31
Table 10 - Summary of Performance Measures for Modelling Average Daily Maximum Demand.....	33

A-2 List of Figures

Figure 1- Modelling methods used in this project.....	11
Figure 2 - Histogram of NSW Total Demand	13
Figure 3 - Histogram plot of Temperature	13
Figure 4 - Histogram of Solar PV Installations.....	14
Figure 5 - PV Capacity by Year (2003-2022).....	14
Figure 6 - Population Growth by Year (1984-2020)	15
Figure 7 - Line Graph of GSP for NSW from 1990 -2022	15
Figure 8 - Temperature vs Total Demand	16
Figure 9 - Correlation Matrix Heat Map of Features.....	17
Figure 10 - Scatter Plot of Features	17
Figure 11 - Average Daily Maximum Demand & Degree Days by Month 2010-2020	18
Figure 12- Average Daily Maximum Demand (Year on Year)	18
Figure 13- XGBoost Feature Importance (weight)	19
Figure 14 - ADF decomposition plot for Average Daily Maximum Demand 2010 – 2020	20
Figure 15 - SARIMA forecast results.....	21
Figure 16- Holt-Winter Exponential Smoothing forecast results.....	22
Figure 17 - Prophet forecast results	23
Figure 18 - MSE loss over Epochs in LSTM	24
Figure 19- Actual versus Predicted plot for LSTM.....	24
Figure 20 - Learning Curve for Linear Regression.....	26
Figure 21 - Stacking Ensemble Strategy.....	28
Figure 22 - Scoring Results for each Estimator Model and Stacking Ensemble	29
Figure 23 - Residual Plot for the Stacking Ensemble	30
Figure 24 - Predicted vs Actual Readings for the Stacking Ensemble	30
Figure 25 - MSE Loss over Epochs in FNN	32
Figure 26 - FNN Prediction vs Actual Results	32

A-3 – Glossary of Terms

- **Consumption** – This measure refers to power used over a period, conventionally reported as megawatt hours (MWh) or gigawatt hours (GWh) depending on the magnitude of power consumed. It is reported on a “sent-out” basis unless otherwise stated (see below for definition) [21].
- **Degree Days** - This measure attempts to bring together the concepts of Heating Degree Days (HDD) and Cooling Degree Days (CDD) in the context of ascertaining the influence that temperature has on electricity demand. We have used a common threshold temperature of 18 Degrees Celsius to ascertain the net difference in temperature for cooling as well as heating as defined by the National Weather Service of the USA [29]. It assumes that when the temperature is 18 Degrees Celsius the population does not need heating or cooling to be comfortable.
- **Demand** – This measure has been defined as the amount of power consumed at any time. Maximum and minimum demand is measured in megawatts and averaged over a 30-minute period. It is reported on a “sent-out” basis unless otherwise stated (see below for definition) [21].
- **Distributed PV** – This is the term used for rooftop PV and PV Non-Scheduled Generators combined [21].
- **Rooftop PV** – rooftop PV is defined as a system comprising one or more PV panels, installed on a residential building or business premises (typically a rooftop) to convert sunlight into electricity. The capacity of these systems is less than 100 kilowatts (kW) [21].
- **“As generated” or “sent out” basis** – “sent out” refers to electricity supplied to the grid by scheduled, semi-scheduled, and significant non-scheduled generators (excluding their auxiliary loads, or electricity used by a generator). “As generated” refers to the same, but also adds auxiliary loads, or electricity used by a generator, to represent the gross electricity generation on site [21].

A-4 – Model Results

A-2.1 SARIMA model fit results

```

=====
                                SARIMAX Results
=====
Dep. Variable:                demand_avg_daily_max    No. Observations:                108
Model:                      SARIMAX(3, 1, 1)x(1, 0, 1, 12)  Log Likelihood                  -757.755
Date:                        Thu, 18 Apr 2024            AIC                           1529.509
Time:                        15:29:21                   BIC                           1548.219
Sample:                      01-01-2010                 HQIC                          1537.094
                                - 12-01-2018
Covariance Type:                opg

=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
ar.L1          0.1936      0.180        1.077      0.282      -0.159      0.546
ar.L2         -0.0518      0.108       -0.481      0.631      -0.263      0.159
ar.L3         -0.1820      0.147       -1.236      0.216      -0.471      0.107
ma.L1         -0.7441      0.170       -4.384      0.000      -1.077     -0.411
ar.S.L12        0.9470      0.026     36.935      0.000      0.897      0.997
ma.S.L12       -0.6153      0.092     -6.712      0.000     -0.795     -0.436
sigma2       6.773e+04   9585.787      7.066      0.000   4.89e+04   8.65e+04
=====
Ljung-Box (L1) (Q):                0.08    Jarque-Bera (JB):                0.23
Prob(Q):                          0.78    Prob(JB):                  0.89
Heteroskedasticity (H):            1.32    Skew:                      0.10
Prob(H) (two-sided):              0.41    Kurtosis:                  2.88
=====

```

A-2.2 Holt-Winter model fit results

```

=====
                        ExponentialSmoothing Model Results
=====
Dep. Variable:                demand_max    No. Observations:                108
Model:                      ExponentialSmoothing    SSE                    53286352.294
Optimized:                      True    AIC                    1447.778
Trend:                          Multiplicative    BIC                    1490.693
Seasonal:                      Multiplicative    AICC                   1455.464
Seasonal Periods:                12    Date:                  Fri, 19 Apr 2024
Box-Cox:                          False    Time:                  01:37:13
Box-Cox Coeff.:                  None

```

	coeff	code	optimized
smoothing_level	0.4292857	alpha	True
smoothing_trend	0.0225940	beta	True
smoothing_seasonal	0.0271769	gamma	True
initial_level	12717.154	l.0	True
initial_trend	1.0052585	b.0	True
initial_seasons.0	1.0822832	s.0	True
initial_seasons.1	0.9998136	s.1	True
initial_seasons.2	0.9999895	s.2	True
initial_seasons.3	0.9775199	s.3	True
initial_seasons.4	0.9814909	s.4	True
initial_seasons.5	0.9941850	s.5	True
initial_seasons.6	0.9946239	s.6	True
initial_seasons.7	1.0155072	s.7	True
initial_seasons.8	0.9723299	s.8	True
initial_seasons.9	0.9661314	s.9	True
initial_seasons.10	1.0013193	s.10	True
initial_seasons.11	1.0148062	s.11	True