Capstone Project By Team A

http:github.com/jlunsw/groupA-Bandits

A DATA SCIENCE APPROACH TO FORECAST ELECTRICITY CONSUMPTION IN AUSTRALIA

Amy Raggatt (z5294732), Ben Costello (z5324262), Brooke Gassner (z5207618), Jun Li (z5322278), Mohammed Ferkh (z5344380), Tomas Klein (z5294915)

April 16, 2022

Abstract

In 2019, Australia's energy consumption reached an astronomical 6,196 petajoules, highlighting the continual use of electricity. In the industry, demand lies at the centre of forecasting, with forecasting being a valuable practice, which has evolved over the last century and is critical in providing uninterrupted, stable electricity to Australian businesses and households. Since 2012, it has been the Australian Energy Market Operator (AEMO) that is solely responsible for providing electricity demand forecasts to the National Electricity Market. Moreover, short term load forecasting is considered a critical issue currently facing the energy sector, due to the privatisation of the industry over recent decades. Recent studies tend to explore a forecast horizon of over 24 hours leaving a gap for forecasting in the nearer future, such as the 5-hour ahead forecast explored in this paper. Here we report the development of deep Neural Network (NN) models and the application of feature engineering and a custom loss function to attain the highest possible level of accuracy. We show the performance and development of a baseline heuristic model, simple Feed Forward Neural Network, Convolutional Neural Network and Bi-directional Long Short Term Memory Network over a 10-step ahead forecasting period, with our best model providing a Root Mean Square Error (RMSE) of 204.19MW, only a margin away from AEMO'S RMSE of 188.59. Whilst the AEMO attain the highest level of accuracy, by the 5th timestep - 2.5hours in advance, the AEMO forecast error begins to flatten, contrasting a promising downward trend for the RMSE as our models increase in complexity. This indicates that with greater tuning and exploration, the models have the capacity to outperform AEMO's model. The outcome of an improved energy forecasting would be vastly beneficial to society, allowing supply to be managed more precisely, with lower costs and risks.

Contents

1 Introduction and Motivation

Short term load forecasting (STLF) for electricity demand has been an active area of study for the past 30 years [1]. STLF is important in reducing costs and improving reliability for power operators, since electricity storage limitations require electricity to be generated and consumed at the same time [2–7]. STLF is considered forecasting which has a prediction horizon of less than a week and is used in the day-to-day management and unit commitment of the distribution network [1, 8]. Factors that drive customers to make similar choices lead to an overall shift in demand [2]. The most used predictors of energy demand include daily temperature, type of day (weekend or weekday), holiday (yes or no), previous consumption, and time of day [2, 4, 5, 9, 10]. Moreover, Chapagain and Kittipiyakul investigated the effect of atmospheric variables such as temperature on forecasting accuracy for short term electricity demand [11]. The results of this showed the atmospheric variables as statistically significant during highly fluctuating and peak hours.

Artificial Neural Networks (ANNs) have generally proven to be the most accurate models for short term electricity demand forecasting [3, 4, 6]. They are considered to be appropriate for modelling non-linear data and deal well with the noisy, chaotic nature that is often associated with time series [8, 12–14]. However, they are notoriously slow to train, difficult to interpret and prone to overfitting [7, 10, 15, 16]. Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs) are all varieties of Neural Networks (NNs) that have proven to be accurate modelers for time series forecasting [14, 16, 17]. RNNs differ from the traditional Feed Forward Neural Networks (FNNs) in that the learning done at any given time will influence the learning in the next time step. That is, they have short-term memory [16]. This makes them ideal for recognising patterns in time series data, albeit, they have their limitations. RNNs do not recognize long-term dependencies due to vanishing or exploding gradients [18]. As a remedy, LSTMs were developed using "gates" that determine which long-term information is useful, and discard that which is not [16]. One study conducted, trained and tested an LSTM on a data set which contained electricity consumption data for different kinds of buildings in America [18]. The week-ahead forecast of this LSTM model out-performed the comparative models.

One-step ahead predictions involve predicting the value for the next timestep in a time series model [14]. Conversely, multi-step or multi-output predictions involve making predictions for each timestep up to a specified horizon. As the prediction horizon increases, producing accurate forecasts becomes more difficult. Taieb et al. explored a method involving multiple neurons in the output layer of a neural network. This multi-output strategy was highlighted as the approach which provided the best performance for multi-step time series forecasting [19]. Further, when forecasting wind speed in a multi-output model based on LSTM networks, optimal feature extraction and an error correction strategy, it was concluded that multi-output strategies resulted in the best performance [20].

There is much to be gained from designing, implementing and evaluating a set of ANNs for multi-output time series prediction. Currently, there is not a great deal of research comparing CNNs and RNNs for multi-output time series prediction [21]. Even less research has been completed on the use of an asymmetric custom-loss function with ANNs [22]. However, the ramifications for under-predicting demand are undoubtedly more serious than for over-predicting. While over-prediction is wasteful and can be costly if it occurs too frequently, under-prediction results in supply shortages and power outages, which have much larger economic consequences [23]. The Australian Energy Market Operator (AEMO) is responsible for electrical load and price forecasting in Australia [2]. A weighted loss function is not currently used in their models, but could prove to be a valuable addition to future forecasting models, which aim to minimize under-prediction.

In this paper, we used deep learning to predict energy demand in New South Wales (NSW). The motivation of this paper was to improve upon the AEMO forecast accuracy through the use of additional feature engineering, the design of deep NNs and by creating a custom loss function which penalises under-prediction. This evaluation-based study compares the performance of the deep learning models for multivariate multi-output short term time series forecasting. The forecasting horizon is 5 hours, consisting of 10 half-hourly time steps. The models were trained on 2 years of data (2018-2020) and tested on the most recent year of available data (2020-2021). Motivated by the literature, the models presented are a FNN, a CNN, and a Bidirectional Long-Short Term Memory (BD-LSTM).

The rest of the paper is organised as follows. Chapter 2 presents the materials used and the methods followed to develop the deep learning models. Chapter 3 provides an exploratory data analysis. Chapter 4 shows the analysis of the results. Chapter 5 provides a discussion and Chapter 6 concludes the paper with recommendations and future work.

2 Materials and Methods

2.1 Software

The exploratory data analysis and visualizations were carried out using Python and Tableau. Tableau allowed for the import of demand, temperature, time, and other feature data, from which the visualizations were constructed to effectively analyze trends in the data to assist our investigation.

The ANN models were built using Python in a Jupyter notebook. This allowed for a simple layout including text and code blocks to organize and explain code and methodologies. A table of contents was added to allow for smooth navigation between different sections. Google collab was utilized so that the final notebook was a collaborative environment. It also allowed for performance intensive predictions of the ANN models to be processed. In the case where predictions consumed too much time, they were processed locally instead, shortening the prediction time.

The key Python libraries accessed include Keras[1], Tensor Flow[2], Scikit Learn[3] as well as a range of other common data manipulation and visualization libraries. Pandas and Numpy were utilized to create the data frames and to pre-process and manipulate the feature data to fit the model and effectively produce the results required. Moreover, Tensor Flow and Keras were used to develop the variety of ANN models and optimization techniques to enhance predictions. Scikit Learn was used in the validation methodology to measure the different error types. All libraries and versions used for the models and visualizations are shown in Appendix 1. A GitHub repository was used to store all data, allowing for source and version control management and team collaboration.

2.2 Description of Data

The data provided to complete this project consisted of 12 CSV files. The data set contain data for Australian states, New South Wales, Victoria, Queensland, and South Australia. The data set was split into demand, forecasted demand and temperature, totalling 3 data sets for each state.

2.2.1 Total Demand

The 4 total demand data sets are structured with columns as in Table 1. The demand has been measured in 30-minute intervals with the 4 data sets covering the period from 1/1/2010 to 18/3/2021 as shown in Table 2. All states excluding SA commence at 12:00 am and SA commences at 12:30am, therefore containing one less row.

Table 1: Total Demand Data Structure

Name	**Description**
DATETIME	Date and time for the total demand reading
TOTALDEMAND	Demand measured in megawatts
REGIONID	Categorical variable that specifies the state - NSW1, QLD1, VIC1 or SA1

Table 2: Total Demand Data Period

State	**Start**	**End**	**Number of Rows**
NSW	1/1/2010 12:00:00 AM	18/3/2021 12:00:00 AM	196513
QLD	1/1/2010 12:00:00 AM	18/3/2021 12:00:00 AM	196513
SA	1/1/2010 12:30:00 AM	18/3/2021 12:00:00 AM	196512
VIC	1/1/2010 12:00:00 AM	18/3/2021 12:00:00 AM	196513

[1] https://keras.io
[2] https://www.tensorflow.org/
[3] https://scikit-learn.org/

2.2.2 Forecast Demand

The 4 forecast demand data sets are structured with columns as in Table 3. Again, the demand has been forecasted in 30-minute intervals, starting 30 minutes ahead, up until 4am the following day. The forecasts are updated half hourly, therefore each 30-minute time slot contains up to 79 forecasts with the final one being the most recent. The data for NSW covers 2010 to 2021, whilst the data sets for the other states only contain data from 1/1/2017 as shown in Table 4.

Table 3: Forecast Demand Data Structure

Name	**Description**
PREDISPATCHSQENO	Pre-dispatch identification number
REGIONID	Categorical variable that specifies the state - NSW1, QLD1, VIC1 or SA1
PERIODID	Number of 30 min steps before the forecast time arrives (PERIODID =2 means prediction is made 1 hour out)
FORECASTDEMAND	Demand measured in megawatts
LASTCHANGED	Date and time when the forecast was last changed (updated)
DATETIME	Date and time for the total demand forecast

Table 4: Forecast Demand Data Period

State	**Start**	**End**	**Number of Rows**
NSW	1/1/2010 12:00:00 AM	18/3/2021 12:00:00 AM	10906019
QLD	1/1/2017 12:00:00 AM	19/3/2021 4:00:00 AM	4095592
SA	1/1/2017 12:00:00 AM	19/3/2021 4:00:00 AM	4095592
VIC	1/1/2017 12:00:00 AM	19/3/2021 4:00:00 AM	4095592

2.2.3 Temperature

The 4 temperature files are structured with columns as in Table 5. The temperature has been measured in approximately 30-minute intervals. There are occasions where the increments differ, for example records at 22:30, 22:48 and 23:00. There are also some missing records at the exact 30-minute interval. Except for VIC, which commences on the 31st of May 2013, all other states commence at 12:00AM on the 1st of January 2010. All 4 states have the same end date of 18th of March 2021. Given the temperature measurements are not always at exactly 30-minute intervals, the number of rows differed for each data set as in Table 6.

Table 5: Temperature Data Structure

Name	**Description**
LOCATION	Categorical variable that specifies the location of the measurement (identifies the state)
DATETIME	Date and time for the total demand reading
TEMP	Temperature measurement in degree celsius

Table 6: Temperature Data Period

State	**Start**	**End**	**Number of Rows**
Bankstown	1/1/2010 12:00:00 AM	18/3/2021 12:00:00 AM	220326
Brisbane (Archerfield Airport)	1/1/2010 12:00:00 AM	18/3/2021 12:00:00 AM	208805
Adelaide (Kent Town)	1/1/2010 12:00:00 AM	18/3/2021 12:00:00 AM	208805
Melbourne (Olympic Park)	31/5/2013 3:00:00 PM	18/3/2021 12:00:00 AM	141681

2.3 Pre-Processing and Data Cleaning

2.3.1 Creating Data Frames

The original data sets for total demand, forecast demand and temperature across NSW, VIC, QLD and SA were provided in zip format. The zip files were saved to the GitHub in a data folder, with all members cloning the repository to their local drive or Google Collab. A Shell script file unzip_data.sh was written, which unzipped all the files into a data folder in the local repository. Given there were 12 CSV files, a Dataloader class was developed to enable convenient and easy loading of any state (NSW, VIC, QLS, SA) or any data class (forecast, demand, temperature) into Pandas data frames. This was used for both the exploratory data analysis and modelling sections.

2.3.2 Data Cleaning for Proposed Models

Both demand and temperature data frames contain a 'datetime' column, which was initially created as a string. As both demand and temperature are time series data, the datetime object was converted to datetime data type. As some of the temperature data was not provided in 30 min intervals, the Pandas 'resample' function was applied to ensure there was data for each half hour reading. For the data points with more than one value within the 30 min interval, a mean of those values was taken. The data from 1/1/2018 12:00am until 31/3/2020 13:00pm was selected for training data and testing was done for the period 31/3/20 13:30pm to 18/3/2021 12:00am. The demand and temperature data frames were then joined using 'datetime' as the index. After removing duplicates and missing values, the final data frame contained 56229 rows.

2.3.3 Data Cleaning for Error Calculation of the AEMO Forecast

Using the *Dataloader* class, a forecast demand data frame was developed for NSW only. Only forecasts for up to 10 periods (5-hour) ahead were retained and all others were removed to align the structure with the proposed model of a 5-hour forecast. All columns other than *datetime*, *periodid* and *forecastdemand* were removed. This data frame was joined to the NSW total demand data frame using the *datetime* index, totalling 3 columns: *totaldemand*, *forecastdemand* and *periodid*. This intermediate data frame was used to calculate various AEMO forecast metrics.

2.3.4 Feature Engineering

To improve model performance, additional feature variables were added in accordance with the learnings from the exploratory data analysis and past studies analysed in the literature review. The Holidays Python library was used to determine NSW public holidays by date. Summer and winter variables were allocated based on the month consistent with Australian seasons. A *FeatureEngine* class was developed with various methods to support the creating and manipulation of specific features.

Cyclical features such as hours, days and years were encoded as pairs of radial co-ordinates by applying the sin and cos functions to the underlying feature, effectively mapping them onto a circle, so that the beginning and end of the cycles were adjacent. This helped the models learn that Dec 31 and Jan 1 were adjacent, as were 11:59am and 12pm [24].

The method *add_features()* wrote the additional features in Table 7 into the final data frame that was used for modelling.

Table 7: Feature Engineering - Additional Variables

Features	Values	Description
HOUR_FACTOR	1 to 48	Every 30-minutes during 24 hrs
IS_WEEKEND	0 or 1	0 means workdays, 1 means weekends
IS_PUBLIC_HOLIDAY	0 or 1	1 means public holiday 0 otherwise
MONTH_FACTOR	1 to 12	Calendar month
IS_SUMMER	0 or 1	1 means summer 0 otherwise
IS_WINTER	0 or 1	1 means winter 0 otherwise
IS_LUNCHTIME	0 or 1	1 means during lunch time (12:00pm - 14:00pm) 0 otherwise
IS_WORKING_HOURS	0 or 1	1 means during working hours (8:00am - 18:00pm) 0 otherwise
DAY_OF_WEEK	1 to 7	Mon, Tue, Wed, Thu, Fri, Sat, Sun
DAY_SIN	-1 to 1	Use sin function to transform the total seconds of a timestamp into "time of day" signals.
DAY_COS	-1 to 1	Use cos function to transform the total seconds of a timestamp into "time of day" signals.
YEAR_SIN	-1 to 1	Use sin function to transform the total seconds of a timestamp into "time of year" signals.
YEAR_COS	-1 to 1	Use cos function to transform the total seconds of a timestamp into "time of year" signals.

2.3.5 Data Preparation for Modelling

The feature engineered data was restructured into a format more suitable for framing the problem as a supervised learning problem. The process of creating sliding windows is illustrated in Figure 1, with t representing the time step number. The first window is shown as a single row, W1. This first window combines the full set of features for 10 time steps (inputs) with the following 10 predicted demand values (labels). The window that follows (W2) will simply move along one time step (30-minutes). The value of 10 time steps was chosen to include the forecast horizon of 5 hours.

Figure 1: Multi-Output Sliding Window Diagram

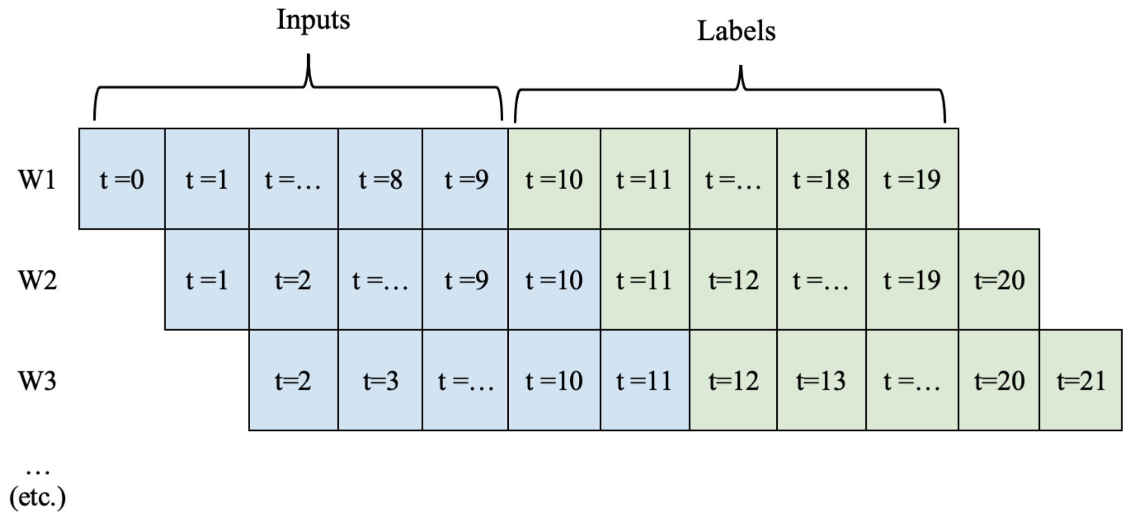
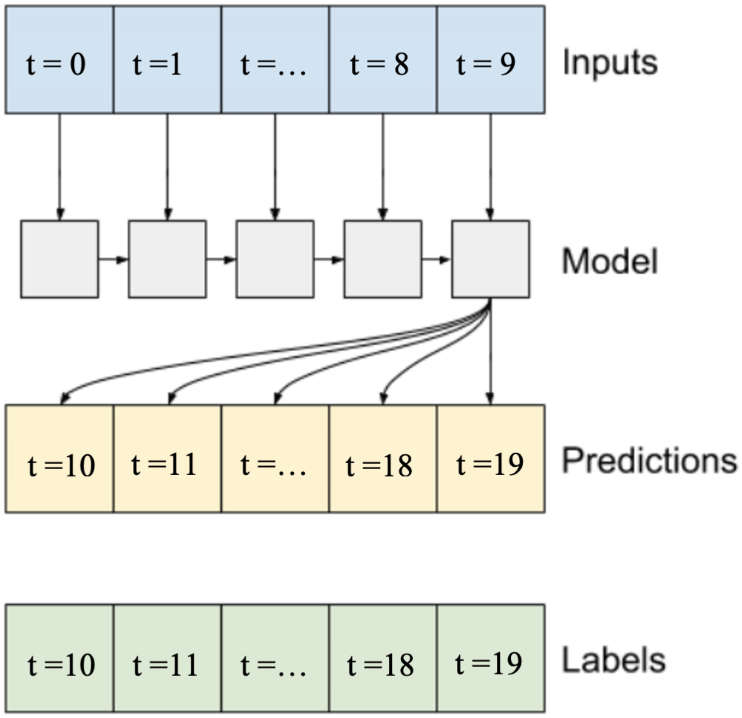


Figure 2 demonstrates the way in which windowing was used in the content of RNNs. An LSTM is a type of RNN which uses important information learned from previous predictions to improve future predictions. This is illustrated as figure 3B shows information in the grey "Model" boxes moving from one time step to the next. For example, information from time step 0 ($t = 0$) may be used to help improve the predictions for any time steps that follow it. At the last time step, the LSTM produces an output. This output consists of the predictions for the next 10 time steps.

Figure 2: Multi-Output Diagram



The data was then scaled to ensure all inputs were between zero and one. NNs are known to train more quickly and perform more accurately with inputs of the same scale and range, preferably between 0 and 1 [25]. The MinMaxScalar() from the Scikit-learn library was used for this purpose.

Finally, the data was split into consecutive training and test data sets using a split of 70:30. This was to ensure that the test set contained a full calendar year of data. The training set was further split into a train and validation set using a split of 90:10. Therefore, the training set took place first from 1/1/2018 12:00am till 31-3-2020 13:00 and the test set from 31/3/2020 13:30 till 18/3/2021 12:00 AM. Further, given the modelling used time series data, the sets were not shuffled, with the test data following the training data, so that future predictions were always made based on past data, simulating the real-world application.

2.4 Assumptions

To complete this project working with the data provided, some assumptions had to be made. First, the temperature data set only included one temperature reading for each state per time interval, therefore it was assumed that the temperature reading from this one station was sufficient and consistent in explaining the temperature across the whole state at any given time.

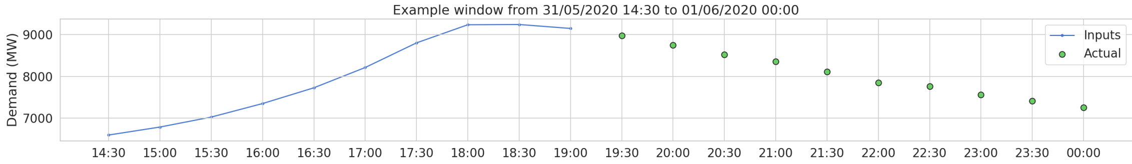
Further, there was some missing data for the half hourly intervals in the temperature data set. The temperature for these missing values was assumed to be the mean of all temperatures recorded within that half hour period.

2.5 Methodology

2.5.1 Overview of Models

A set of increasingly complex models were built and then tested to perform multi-step time series predictions. A series of 10, 30-minute demand and temperature observations, along with a set of engineered features, were used as predictors. At each time step, we used the previous 10 predictors to produce the next 10, 30-minute forecasts. Thus, our model is considered a multi-output model with a 5-hour prediction horizon. Figure 3 provides an example for a forecast made on the 08/11/2020 at 19:00. The blue line shows 10 demand values until present and the green dots are the true future values to be predicted. In this example, at 7pm, the model aimed to predict the next ten half hourly observations until midnight.

Figure 3: Example of 10 Future Observations



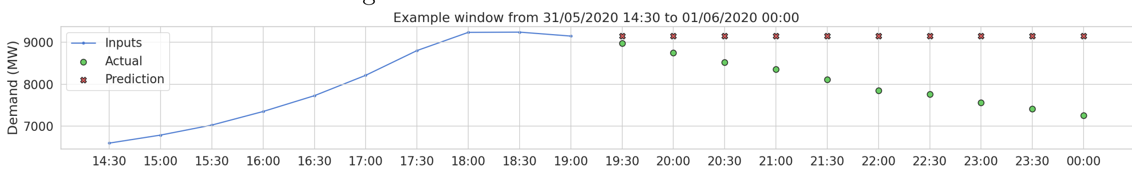
The primary measure of model complexity was the number of trainable parameters in each model and the primary measure of performance was the Root Mean Square Error (RMSE) between predicted and true values. We chose RMSE as it heavily weighs outliers, and outliers (very high or low demand values in this context) were considered particularly important to predict accurately [26].

The 4 models were:

- Baseline Heuristic Model
- Simple FNN
- CNN
- BD-LSTM

The naïve baseline model simply took the current demand observation and repeated it 10 times. This was more of a heuristic than a model and set a very simple baseline for the simplest possible prediction strategy. Figure 4 illustrates the heuristic model predictions.

Figure 4: Baseline Heuristic Predictions



The ANNs were constructed in very simple forms but in increasing order of model complexity. All had a single hidden layer of an arbitrarily chosen number of neurons as in Table 8.

Table 8: Configuration of Models

Model	**Hidden Layers**	**Trainable Parameters**	**Comments**
Heuristic	0	0	N/A
FNN	1	11,594	Hidden layer = 64 neurons
CNN	1	11, 882	Hidden layer = 256 neurons Convolutional Window=3 Training 40 epochs
BD-LSTM	1	566,282	Forward and backward layers 256 cells each. Drop Out Layer - 20% dropout Training 40 epochs
Best BD-LSTM	1	2,181,130	Forward and backward layers 512 cells each. Drop Out Layer - 10% dropout Training 60 epochs

2.5.2 Optimiser

ANN training uses a form of Gradient Descent for weight updates. Stochastic Gradient Descent (SGD) maintains a single learning rate across all weights, but the Adaptive Moment Optimizer (Adam) maintains a learning rate for each weight and updates (adapts) this as learning progresses. The Adam optimizer has been shown to improve learning over other optimizers for time series forecasting and was used for all the deep learning models throughout this paper [27].

2.5.3 Activation Function

Rectified Linear Units (ReLu) activation functions were applied for all hidden layer nodes. ReLu is less susceptible to vanishing gradients and has been shown to lead to better convergence performance than sigmoid activations [28].

2.5.4 Custom Asymmetric Loss Function

In a regression style problem, a simple distance metric is normally used to measure the size of the difference between predicted and true values. A standard measure is L2-norm or Mean Squared Error (MSE).

$$\text{Mean Squared Error (MSE)} = \frac{1}{N}[\sum_{i=1}^{N}(y_i - \hat{y_i})^2] \tag{1}$$

Where y is the observed data, $\hat{y}$ is the predicted data and N is the length of the observed data.

As aforementioned, one aspect of demand predictions considered was the asymmetry that under-predicting demand is substantially more detrimental than over-predicting. We wanted to use a loss function to address the asymmetry, and the ANN learning was nudged to favour demand estimates above the true values. Unfortunately, a standard loss function to support this was unable to be found, so a weighted L2-norm custom loss function was written. It was tested with a weight value of 2. This meant that any prediction below the true value would have a penalty of doubling the loss applied.

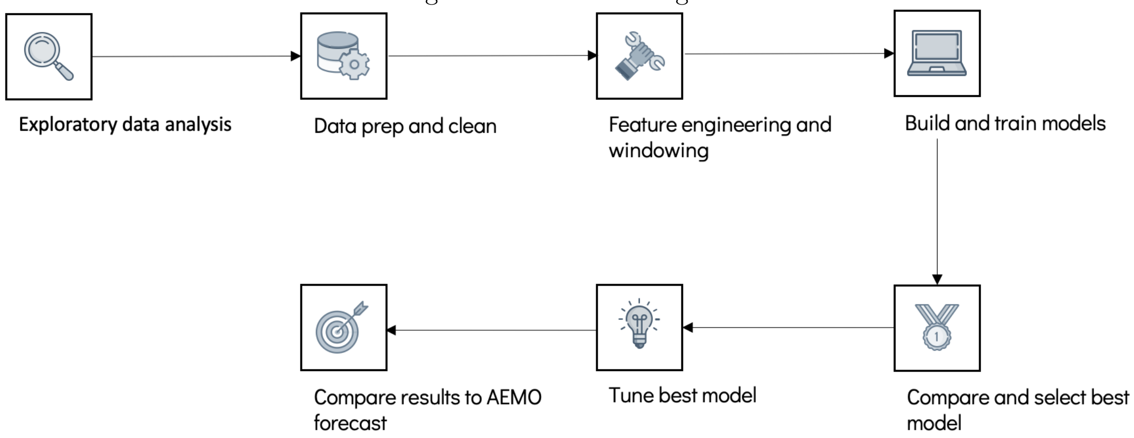
$$\text{Custom (MSE)} = \frac{1}{N}[\sum_{\hat{y_i}>y_i,i=1}^{N}(y_i - \hat{y_i})^2 + w\sum_{\hat{y_i}<y_i,i=1}^{N}(y_i - \hat{y_i})^2] \tag{2}$$

Where y is the observed data, $\hat{y}$ is the predicted data, N is the length of the observed data and w is a weighting.

2.5.5 Workflow Diagram

All processing was performed in a single Jupyter notebook. This included all data extraction, cleansing, feature engineering, data windowing and splitting of data into train, validation and test sets. The process is described in Figure 5.

Figure 5: Workflow Diagram



2.5.6 Experiment Setting

Each ANN model was trained 10 times and both step-wise and overall RMSEs were recorded in order to produce a 95% confidence interval of RMSE values for final comparison. comparison.

The RMSE values of interest were:

- Step-wise RMSE - an RMSE for each of the 10 time-step predictions
- Overall RMSE – an RMSE on all predictions

The RMSE of the 10th and last time step was of particular interest as the longer the prediction horizon, the more valuable the forecast. In particular, the further out the prediction, the more time energy producers have to respond to forecast demand changes. All RMSEs were compared to the corresponding AEMO forecast values, which were also calculated. The model that produced the best overall RMSE results was selected and then a limited exercise in hyperparameter tuning was performed in order to arrive at an optimal configuration of some of the key hyperparameters. We briefly tuned the number of hidden units, the dropout percentage and the training epochs. No attempt was made to optimize the remaining or heuristic models.

3 Exploratory Data Analysis

The exploratory data analysis (EDA) was conducted using Tableau Desktop and Seaborn and Matplotlib libraries in Python, utilizing Visual Studio Code. For Tableau analysis, a Tableau prep builder was utilized to create 3 hyper files, one for each category for each state using the union function – total demand, temperature and forecast demand. These files were used in Tableau EDA with Datetime used as a joining field, where data from different files were joined and plotted.

3.1 Total Demand – Time Series Analysis

Figure 6 below displays mean daily demand and mean daily temperature over the full period for the data provided for each state. The season attribute (spring, summer, autumn, winter) was created and overlaid on the chart to show the variability of both demand and temperature during different seasons in Australia. It is evident that demand increases during summer months as temperature rises and a similar pattern is evident in winter as temperatures drop.
To explore the demand in more detail and specifically in NSW, Figure 7 displays demand using a scatter plot with 30-minute demand increments from January 2018 on-wards. It can also be seen that

Figure 6: Demand and Temperature Time Series (2010 -2021) Mean Daily Data

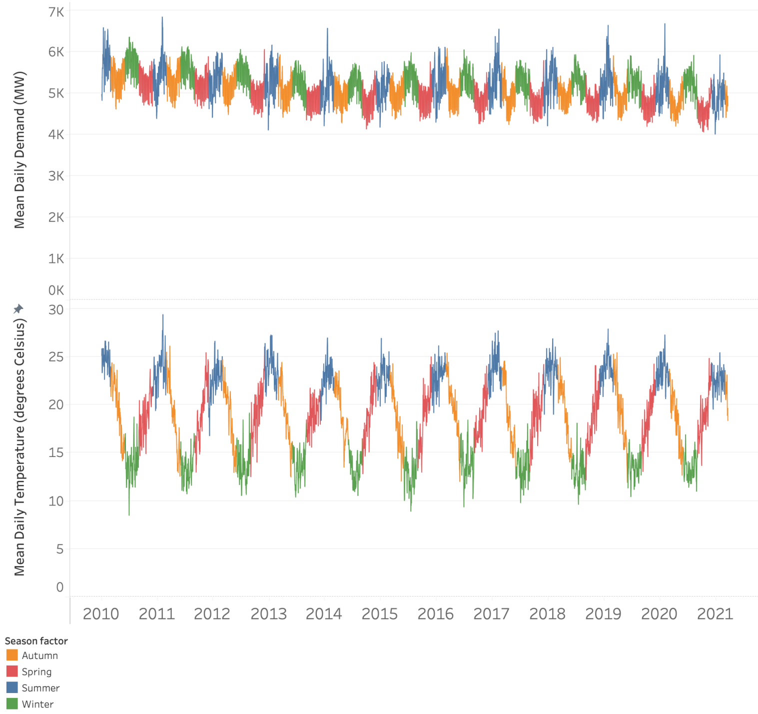
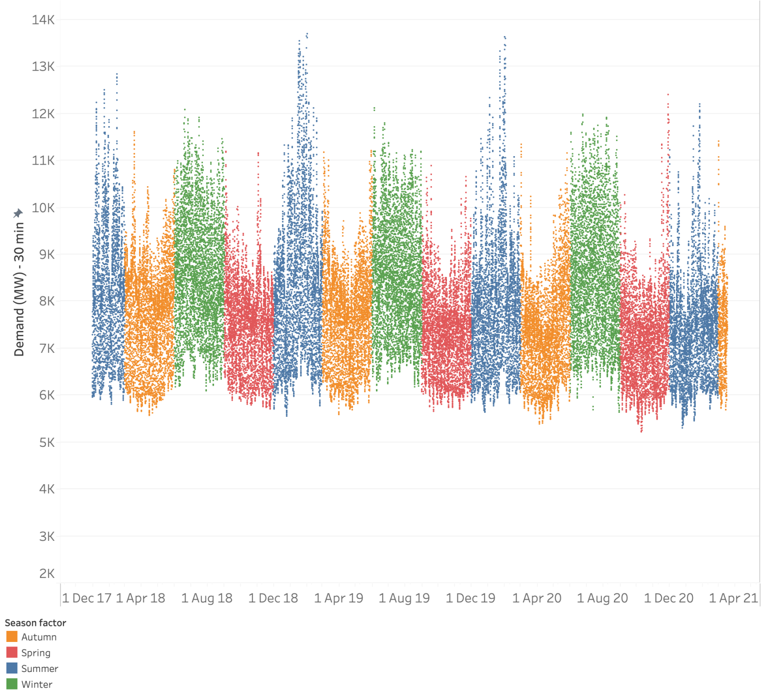


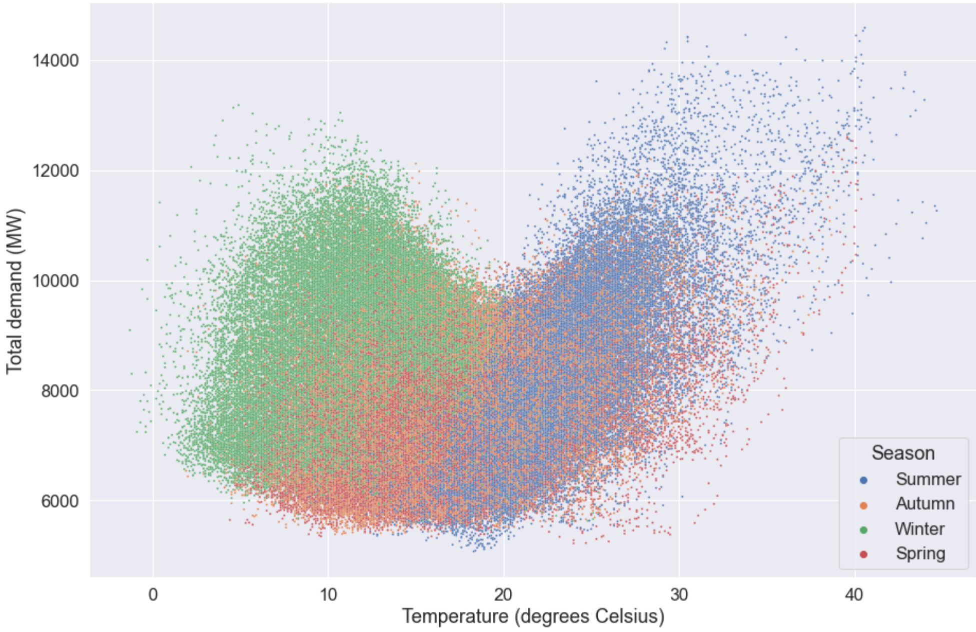
Figure 7: Demand in NSW Time Series (2018 -2021) 30-minute Intervals



there is much greater variability in the summer months with demand spikes.

To gain a better understanding of the NSW total demand time series, a seasonal trend decomposition using LOESS (STL) was applied for the period 2010 to 2020 [29]. The data was resampled to daily sum demand values and STL analysis using an additive model was implemented in Python as shown in Appendix 2. The process decomposed the time series into trend, seasonality and residuals. Whilst the trend was decreasing, the seasonality of total demand was evident in winter and summer, with winter's impact more pronounced. There are also some extreme unexplained residuals in 2010, 2016,2017, 2019 and 2020.
In order to better understand the relationship between the demand and temperature, Figure 8 shows a

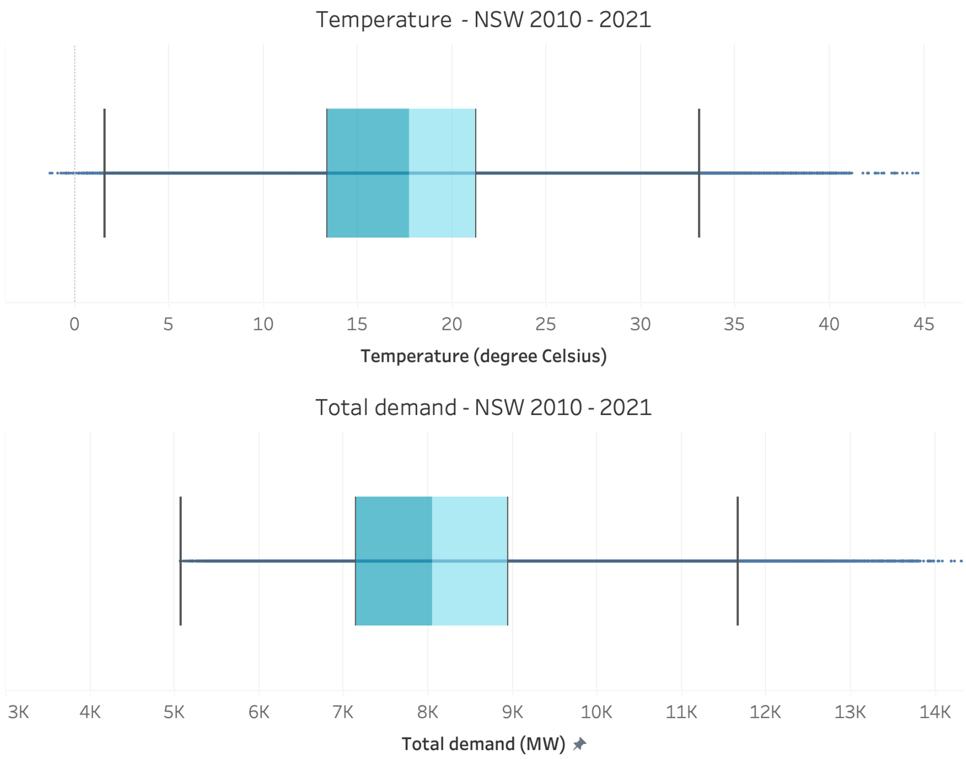
Figure 8: Demand and Temperature Relationship NSW (2010 -2021) 30-minute Intervals



scatter plot of demand and temperature in NSW from 2010 to 2021. The plot uses 30-minute interval data and again applies the season overlay. It can be observed that demand increases as temperatures rise in summer and as temperatures fall in winter, with it lower and less variable in spring and autumn. The electricity demand and temperature relationship can be seen as U-shaped, with demand increasing during low or high temperatures due the use of air conditioning and heating [30]. Both temperatures and total demand appear to have the most outliers in summer at high temperatures compared to any other seasons. In order to validate such hypotheses, box plots were used to display the same data set as shown in Figure 9.

Both distributions were positively skewed with many outliers on the right hand side for extreme temperature and extreme demand. For the temperature and demand data sets, the upper whisker is around 32.4 degree Celsius and 11,671 MW. Box plots for demand for each season can also be found in Appendix 3, showing that summer has more outliers than any other season, whilst the median value for winter is the highest in comparison to other seasons.

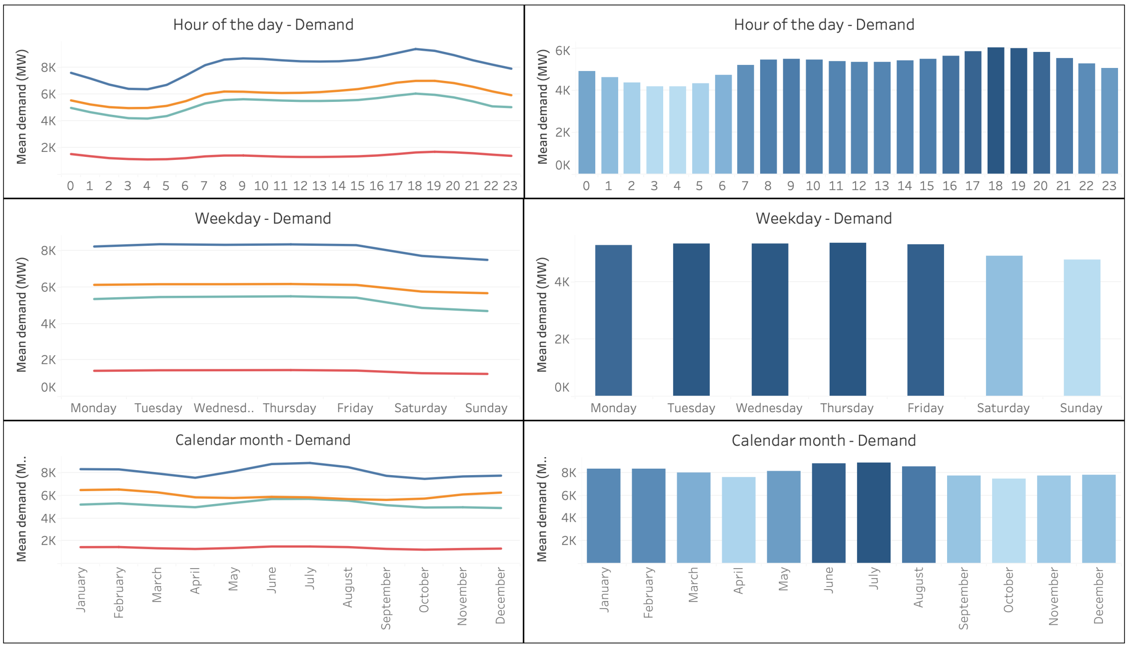
Figure 9: Demand and Temperature Boxplots NSW (2010 -2021)



3.2 Total Demand and Other Factors

Figure 10 examines total demand and how it changes in relation to the hour of the day, the day of the week and month of the year across different states. It can be concluded that demand peaks around 6-7pm for all states. A small dip during lunchtime can also be seen. In relation to the weekday, Monday to Friday appear very similar, with weekends having comparatively lower demand. Observing the months of the year, the winter months of June, July and August experience the highest mean demand, followed by the summer months of January, February and December.

Figure 10: Demand and Other Factors (Left –State Breakdown, Right - Mean Across All States



Further examination of hour of the day and day of the week for NSW as in Figure 11 shows increased demand between 5pm and 8pm Monday to Thursday and the lowest demand between 3am and 6am on Sunday.

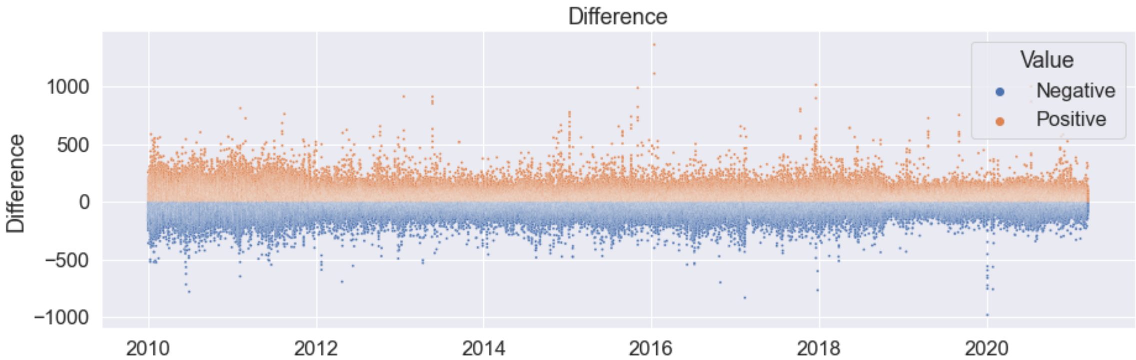
Figure 11: Mean Demand - Hour of the Day and Weekday NSW (2010-2021)

Hour ..	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
0	7,153	7,698	7,725	7,701	7,755	7,786	7,300
1	6,795	7,289	7,307	7,280	7,321	7,312	6,877
2	6,441	6,822	6,825	6,807	6,840	6,790	6,455
3	6,221	6,490	6,479	6,475	6,503	6,412	6,169
4	6,274	6,475	6,462	6,467	6,488	6,289	6,084
5	6,712	6,893	6,884	6,888	6,890	6,403	6,151
6	7,581	7,776	7,764	7,762	7,735	6,760	6,393
7	8,448	8,643	8,623	8,615	8,582	7,346	6,855
8	8,844	8,973	8,947	8,940	8,952	7,958	7,420
9	8,896	8,956	8,927	8,923	8,969	8,228	7,775
10	8,854	8,889	8,863	8,858	8,919	8,180	7,826
11	8,766	8,794	8,767	8,765	8,827	8,031	7,738
12	8,707	8,733	8,698	8,703	8,746	7,905	7,660
13	8,716	8,738	8,700	8,715	8,716	7,838	7,614
14	8,740	8,763	8,724	8,752	8,722	7,833	7,632
15	8,834	8,860	8,809	8,849	8,800	7,938	7,775
16	9,042	9,063	9,004	9,040	8,973	8,144	8,034
17	9,361	9,353	9,273	9,331	9,171	8,475	8,462
18	9,666	9,656	9,561	9,640	9,362	8,800	8,887
19	9,482	9,507	9,419	9,521	9,201	8,682	8,826
20	9,107	9,152	9,082	9,191	8,900	8,411	8,549
21	8,667	8,720	8,663	8,753	8,566	8,135	8,185
22	8,300	8,350	8,314	8,384	8,335	7,919	7,832
23	7,988	8,026	8,004	8,067	8,097	7,641	7,472

3.3 Forecast Demand

Figure 12 below shows the difference between the forecasted demand and actual demand time series for NSW for the period of 2010 to 2021. The difference chart shows that the forecasting between 2010 and 2012 is significantly worse than in the later years from 2012 onwards, with a further improvement in 2019. It can also be concluded that the difference between the forecasted and actual demand is relatively evenly distributed between positive and negative values. Given the importance of having sufficient demand, it is anticipated that overestimation would be preferred, with a larger proportion of positive values.

Figure 12: Difference; Forecasted Demand - Actual Demand NSW(2010-2021



4 Analysis of Results

The mean and 95% confidence interval for the RMSE are reported for the train and test data sets from 10 experimental runs in Table 9. The overall RMSE is reported for both test and train data sets and the RMSE for each step is reported only for the test data set. The AEMO and baseline Heuristic results are included for completeness, however, did not require experimental runs. The results in this table do not include models using the custom loss function, which is discussed below.

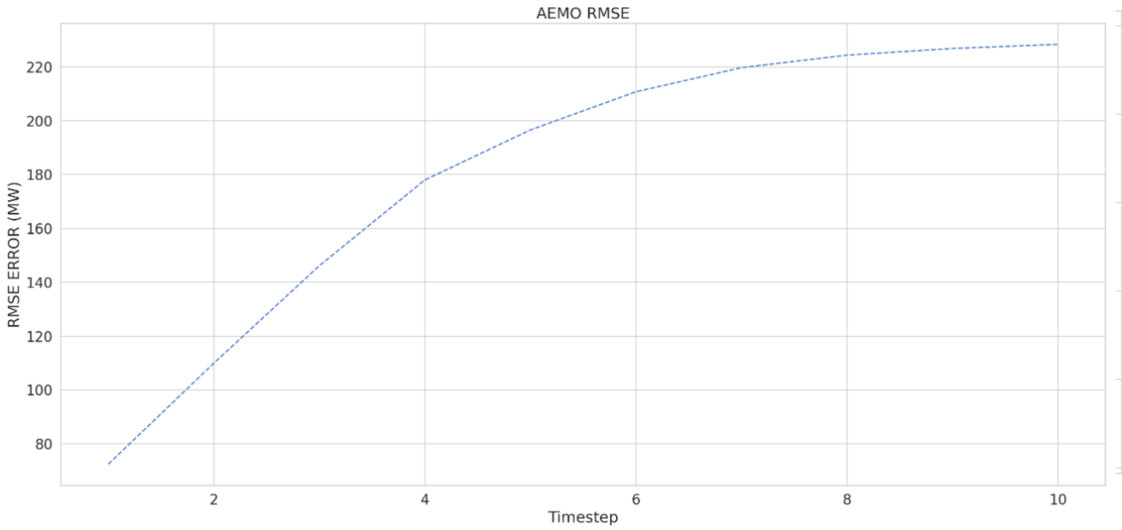
Table 9: Mean and 95% Confidence Intervals of the RMSE for Each Model

	AEMO Forecasts	**Heuristic**	**FNN**	**CNN**	**BD-LSTM**
Train Overall	166.00	756.38	331.9768 ± 22.843	260.2607 ± 8.0745	127.0663 ± 25.5302
Test Overall	188.59	775.45	388.1591 ± 23.6971	287.5244 ± 10.2192	204.1928 ± 6.1439
Step 1	72.41	223.31	391.257 ± 39.2317	244.6256 ± 25.8264	112.2425 ± 11.8291
Step 2	109.93	429.05	422.4578 ± 41.4051	278.1041 ± 22.4462	158.1823 ± 12.008
Step 3	146.13	620.66	450.0375 ± 37.1804	310.6815 ± 18.6815	202.1811 ± 11.2485
Step 4	178.00	794.60	476.7188 ± 34.0793	339.8267 ± 12.7224	244.1345 ± 9.5218
Step 5	195.58	949.26	496.1462 ± 33.0617	363.1166 ± 13.7736	282.669 ± 10.473
Step 6	210.76	1084.31	512.4452 ± 34.2359	392.502 ± 13.593	316.9006 ± 10.0881
Step 7	219.73	1199.60	527.5246 ± 28.26	417.1361 ± 15.1868	348.247 ± 8.9035
Step 8	224.36	1296.05	542.361 ± 24.8778	439.8835 ± 13.8077	374.0322 ± 8.7499
Step 9	226.84	1374.76	560.4495 ± 26.4075	463.5032 ± 11.1532	398.1423 ± 8.9552
Step 10	228.39	1437.84	593.2508 ± 28.6336	497.4868 ± 7.6987	421.0084 ± 10.4869

4.1 AEMO

The AEMO forecasts set a formidable benchmark. Understandably, like all the models, the AEMO model performance degrades as the forecast horizon increases as in Figure 13.

Figure 13: Step-wise RMSE Performance of AEMO Forecasts



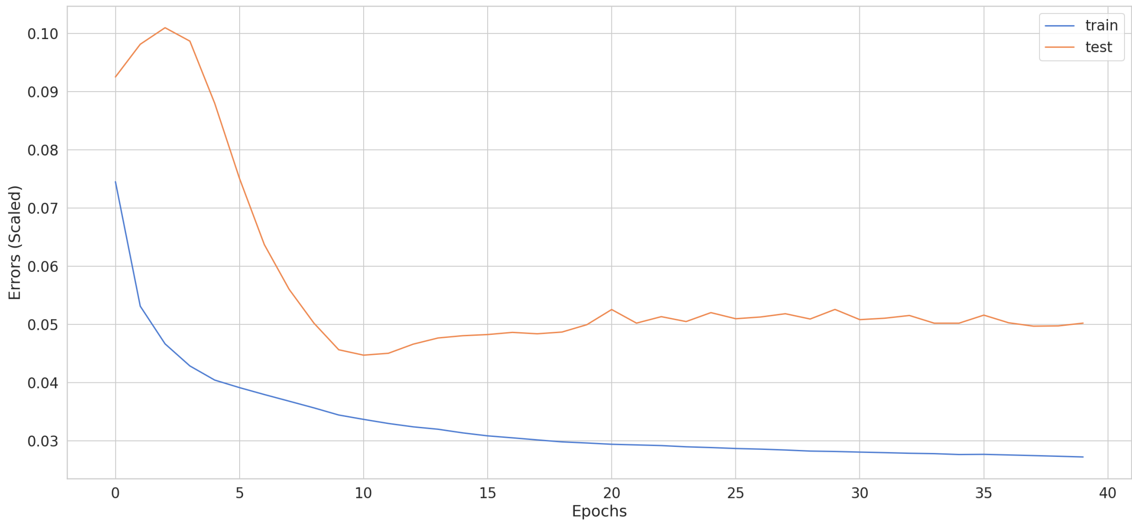
4.2 Baseline Heuristic

The baseline heuristic model results were unsurprisingly very poor and easily surpassed by all models.

4.3 FNN

The FNN also acted as a baseline for the ANN models and had the fewest trainable parameters. It was expected that since this was a linear model with no ability to remember trends from previous timesteps, this would be the poorest performer of the ANNs, with the experiments verifying this hypothesis. The loss curve from training this model as in Figure 14 shows that while training loss improves, the test loss decayed early and only slightly increased resulting in higher error as the epochs progressed. This suggested overfitting of the training data and non-linearity in demand prediction, which the linear structure of this simple ANN was incapable of learning.

Figure 14: Training Loss History FNN



4.4 CNN

It was expected that the CNN would improve performance as it had the ability to remember more information from previous time steps using a convolutional window. The training loss curve, Figure 15, was much more promising even though it appeared that the test loss had decayed and settled on quite a high value. Additionally, a plot of an example prediction window suggested reasonable performance as in Figure 16. It is clear from Table 9 that whilst this model was a significant improvement over the simple ANN, it was still not nearly as good as the AEMO benchmark.

Figure 15: Training Loss History CNN

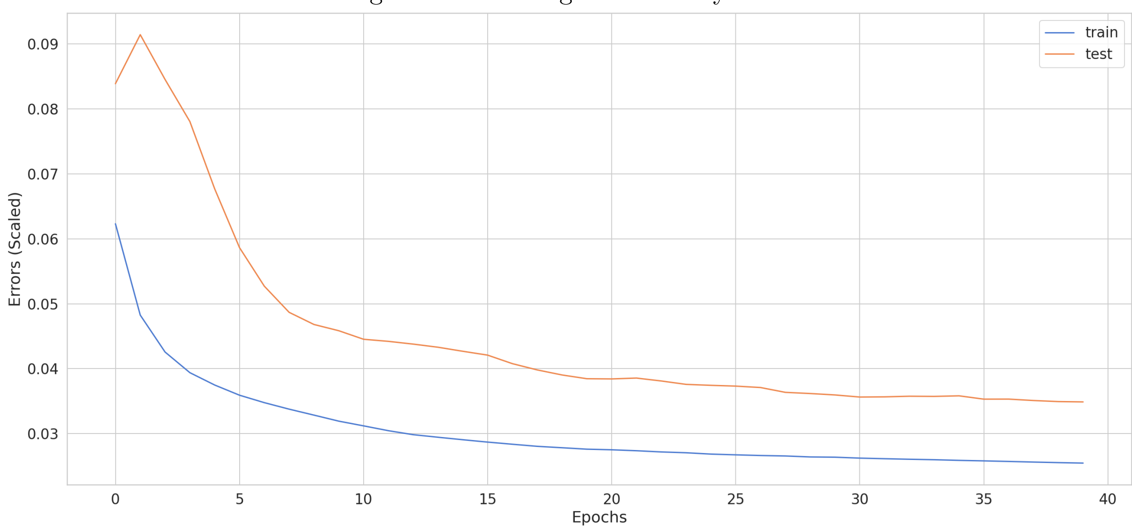
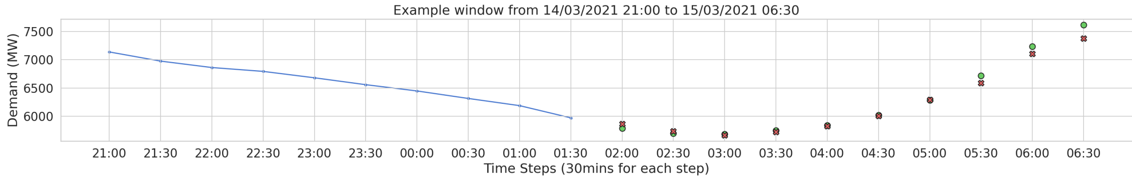


Figure 16: Sample Prediction Window for CNN Model



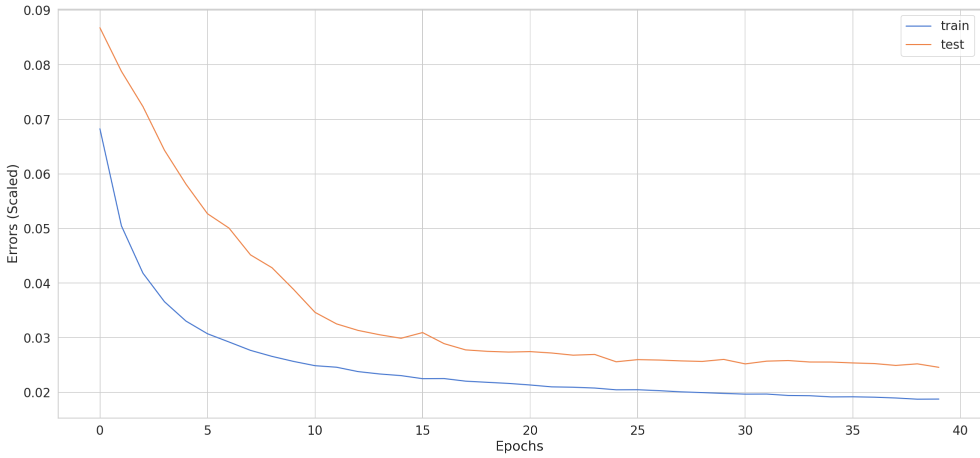
4.5 BD-LSTM

The final model was an LSTM, which maintains a long-term state that can learn to recognise important features (through its *Input* gate) and preserve these in the model [31]. The initial trained model consisted of a single bidirectional LSTM layer of 512 neurons (256 in each direction). This was not truly a single layer as the LSTM itself was constructed from a number of dense layers representing *Input, Forget* and *Output* gates.

A grid search across a limited set of hyperparameters was used to find a BD-LSTM model with the lowest overall RMSE. Due to time and processing constraints, this was a narrow search and from the available combinations, the best overall RMSE was obtained by:

- A single BD-LSTM with 1024 Nodes (512 in each direction)
- A drop out layer with 10% dropout
- A training period of 60 epochs

Figure 17: Training Loss Curve – BD-LSTM Model



The training history as in Figure 17 was much improved over the other models. The fact that the training and test loss decreased to a point of stability and that the gap between the two graphs is small, indicates a good fit. As a set of 3 sample prediction plots, Figure 18, shows predicted demand appeared very close to the true demand. Finally, plots of the mean and confidence interval of the test and training predictions as in Figure 19, shows that the 95% confidence interval for the test set was very narrow, so high confidence could be placed on predictions.

Figure 18: 3 Random Sample Prediction Windows for BD-LSTM Model

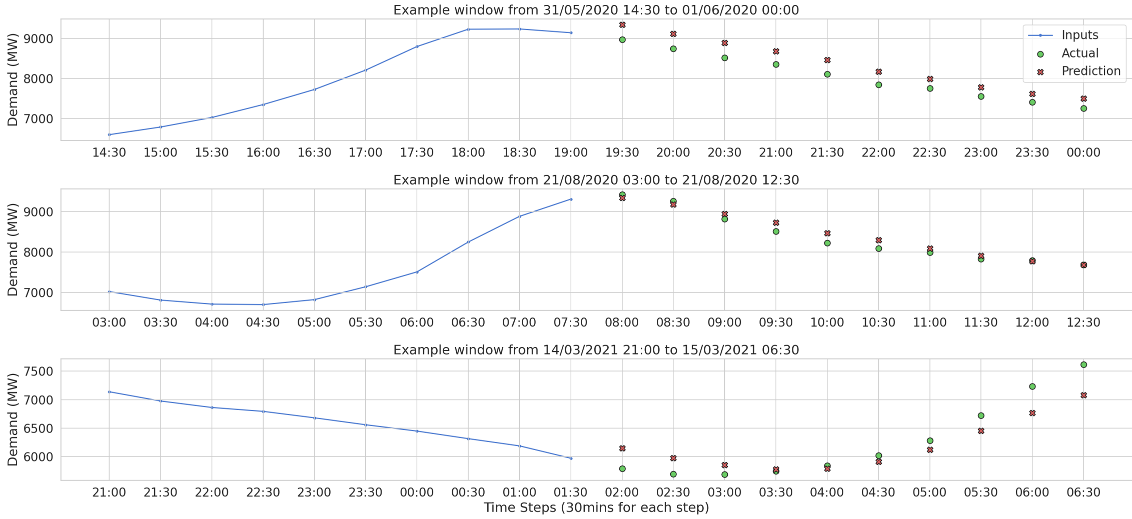
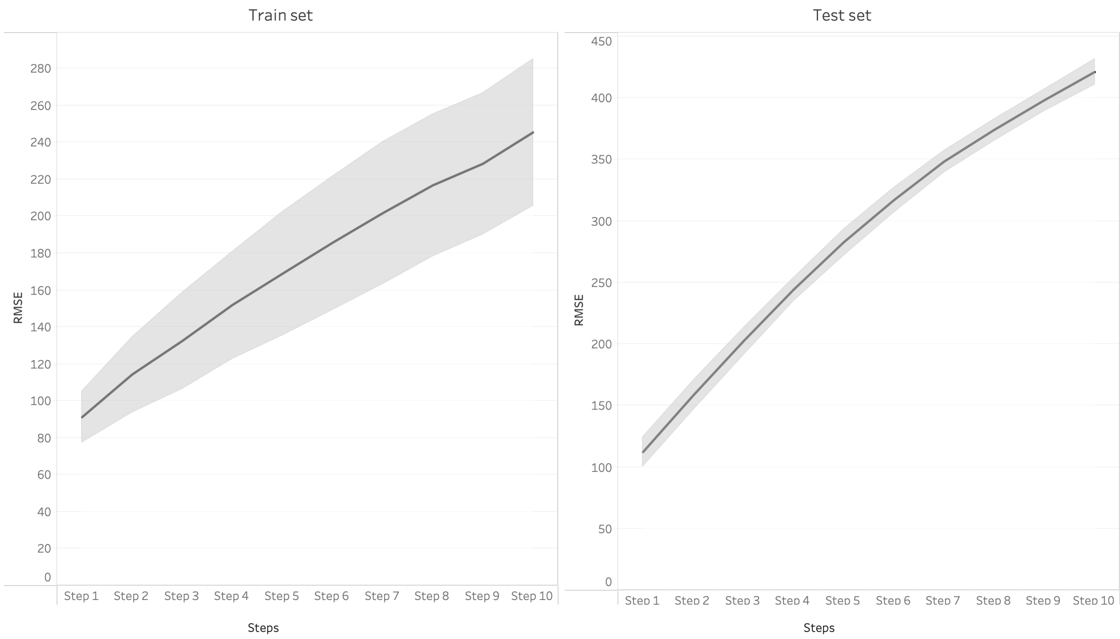


Figure 19: 95% Confidence Intervals for the Train and Test RMSE of the BD-LSTM Model



4.6 Custom Loss Function

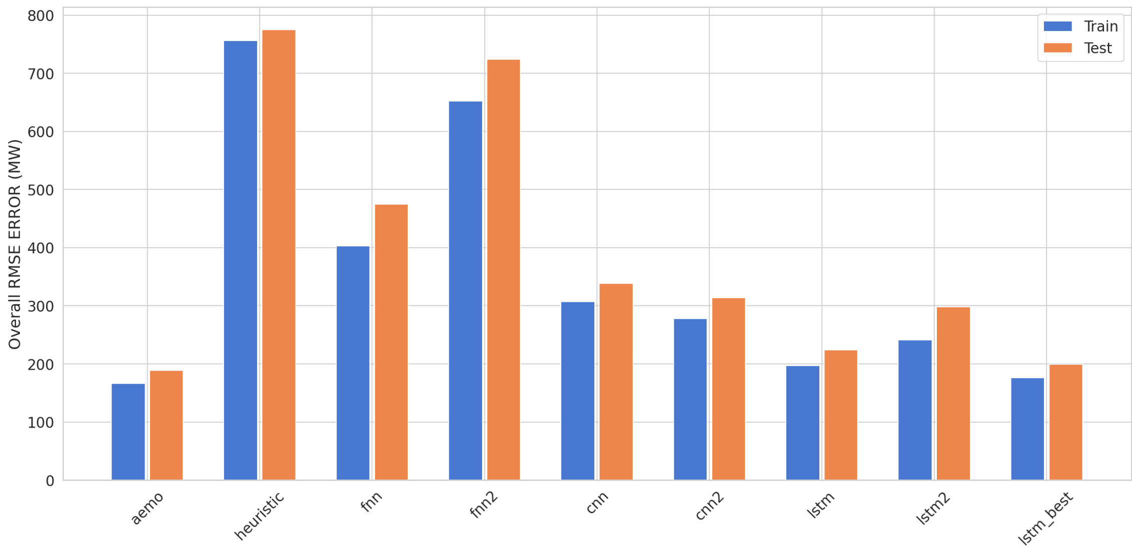
The goal of implementing a custom loss function was to influence model training towards meeting the business objective of reducing under-predictions. On this measure, the loss function succeeded. In Table 10 it is seen that the overall percentage of predictions under the observed true value is almost 50% for both the AEMO forecast and the best BD-LSTM model, with a standard MSE loss metric. Using a custom loss metric with a weighting of 2, the proportion of under predictions is reduced to less than a third.

Table 10: Percentage of Predictions Under Forecast at Each Step

	AEMO Forecasts	**BD-LSTM**	**BD-LSTM (With Custom Loss)**
Overall	46.70	47.54	32.31
Step 1	36.53	28.65	26.40
Step 2	38.94	33.40	26.49
Step 3	44.85	38.43	26.78
Step 4	47.67	46.32	28.78
Step 5	48.39	49.48	30.41
Step 6	49.17	53.23	32.51
Step 7	49.49	54.51	34.37
Step 8	50.27	57.41	37.00
Step 9	50.65	57.03	39.02
Step 10	51.03	56.90	41.34

Figure 20 shows the overall RMSE for all models, including models trained with the asymmetric custom loss function (denoted by a 2 after the model name). Here it is clear that the trade-off for under-prediction is made against general accuracy. For the simple FNN and LSTM models, the custom loss function resulted in a higher error rate. Interestingly, this was not the case for the CNN.

Figure 20: Model Performance Comparison - Overall RMSE

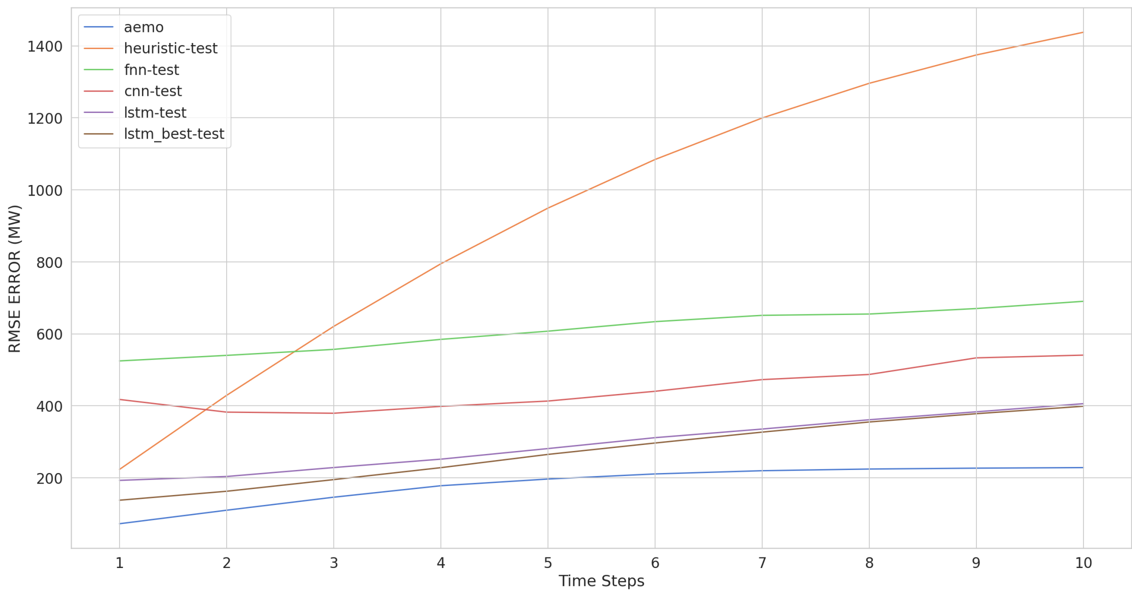


5 Discussion

The primary purpose of this study was to explore the ability of deep NN models to improve the standard of the AEMO forecasts. We aimed to achieve this through the investigation and evaluation of a range of ANN architectures and the addition of engineered time-variable features such as type of day (weekday, weekend or holiday) and season (summer, autumn, winter, spring). Figure 20 provides a ranking for our models based on the overall RMSE, where a lower RMSE results in a higher ranking model. While none of our models matched the accuracy of the AEMO forecasts, our most promising results were given by the BD-LSTM, whose overall test RMSE was only 15.6 MW above AEMOs model. This was followed by the CNN and the FNN, with test RMSEs of 287.52 and 388.16 MW respectively. Unsurprisingly, we observed that the heuristic model gave the worst RMSE. Table 5A displays the mean and 95% confidence intervals for each model. From this table, we can conclude that the BD-LSTM model had the lowest confidence interval ($\pm$ 6.1439 MW) and therefore provided the most precise and reliable forecasts. This was followed by the CNN ($\pm$10.2192 MW) and the FNN ($\pm$23.6971 MW). We also experimented with a custom loss function that was found to be successful in driving down under-predictions, but generally at the expense of overall accuracy measured by the RMSE (see Figure 20).

Furthermore, Figure 21 shows the comparative performance of each model across the prediction horizon. The AEMO forecasts shown in purple gave the lowest RMSE across all time steps and therefore provided the most accurate forecasts. Nevertheless, we see a promising downward trend for the RMSE as our ANN models become more complex, with the tuned BD-LSTM line closest to AEMO at each time step. The heuristic model shows the worst RMSE performance, followed by the FNN and the CNN. Moreover, by time step 5, a 2.5 hour ahead prediction, the AEMO forecast error begins to flatten, whilst the errors of the other CNN and BD-LSTM models tend to maintain a linear rate of increase. We note that it is natural for performance accuracy to deteriorate as the prediction horizon increases in multi-output time series problems [14].

Figure 21: Model Performance Comparison - Prediction Horizon Step-wise RMSE



Given that our very simple BD-LSTM model with limited tuning could produce results quite close to the AEMO forecasts, further experimentation with LSTM architectures may indeed lead to models that match or outperform the AEMO forecast. Promising results have been obtained forecasting short-term energy demand with slightly more sophisticated LSTM architectures such as multiple LSTM layers [18]. Recent work on Encoder-Decoder LSTMs [14] and Temporal Convolutional Networks [32, 33] suggest these architectures could improve accuracy and robustness of forecasts.

The custom loss function showed much promise in reducing the rate of under predicting demand. Whilst this involved a trade-off with forecast accuracy, this may be worthwhile if the impact of under predicting demand is more expensive than over predicting. We did not tune or experiment otherwise with variations of our loss function but do note that promising results have been obtained in other forecast settings with customizing loss functions to match business requirements, particularly with LSTMs [22].

Our models were trained on 2 years' worth of data (half-hourly predictions) in approximately 5 minutes on a desktop PC. This raises the prospect that a model could be fully retrained almost constantly to produce predictions every half hour. It may prove useful to include more outlier values from previous years to improve training for these events. This would also address the class imbalance problem. It may also be true that the data used by our models was limited in scope. We used a very simple set of temperatures, previous demand observations and simple weekend and public holiday data. It is plausible that other factors are likely to influence electricity demand, such as social and economic factors (e.g. COVID-19 shutdowns, sporting events, school holidays) and other trends in electricity consumption such as trends in take up of rooftop solar PV [34].

Time constraints on the project forced a limit to the scope of the experimental activities, counting several areas that would have been interesting to expand on given more time. We added features to our models with little experimental evaluation of whether they contributed to the model's performance, requiring further evaluation into warranting the importance of specific features. We had hoped to train and test our models on the data sets from South Australia, Victoria and Queensland but did not have sufficient time towards the conclusion of the project. We tuned hyperparameters only on what we found to be our strongest performing model and even then, our grid-search parameter space was very limited. Hyperparameter tuning all of our ANN models as well as increasing the parameter space may have yielded better results. Our models only considered the temperature readings from a single recording station in NSW's 800+ sq/km [35]. It is possible that this was an insufficient indicator of temperature across the entire state, causing an unjust measure of overall electricity demand. Additional temperature readings from different areas of the state would likely contribute to better model accuracy.

Finally, exploring additional data sources may prove worthwhile. Demand is undoubtedly affected by more factors than just temperature, previous demand and various short and long-term seasonality trends. AEMO uses features such as solar PV data and electric vehicle consumption, which we did not have access to [2]. We noted the long-term decline of demand in our EDA and posit that this was a result of increased household PV. It's quite likely that there is a correlation between demand and PV usage and take-up trends. The temperature data was useful in training the model; however, it was very basic. More meteorologic data such as humidity, rainfall, cloud cover and global warming may also contribute to improved model performance [11].

6 Conclusion

In this paper, we provided an in-depth evaluation of the use of deep NNs to predict short term energy demand in NSW. Our results show that a BD-LSTM network performed the best for multi-step ahead forecasting, with a prediction horizon of 5 hours ahead. It can be deterred that the FNN, CNN and LSTM models improved as they were progressively trained, with this being attributed to increased model complexity and improved memory capability. The created custom loss function showed promising results and may add value for operators who wish to reduce under-prediction of demand in future.

Moving forward, more in-depth hyperparameter tuning of BD-LSTM networks would be worthwhile to further improve the multi-output forecasts for time series models, as currently this is an area of much uncertainty. Further, other types of ANNs such as Temporal Convolutional Networks and Encode-Decoder LSTMs should be investigated in the future, as they have proven to be accurate time-series forecasters in the literature. It is also recommended that further work is undertaken to improve the asymmetric custom loss function. Comparison to asymmetric loss functions used in the hydrology industry may prove helpful and valuable. These deep learning methods might also be applied to other fields such as weather prediction and price forecasting.

References

[1] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.

[2] AEMO, "Electricity demand forecasting methodology," May 2021. [Online]. Available: https://aemo.com.au/en/consultations/current-and-closed-consultations/electricity-demand-forecasting-methodology

[3] M. S. Al-Musaylh, R. C. Deo, J. F. Adamowski, and Y. Li, "Short-term electricity demand forecasting using machine learning methods enriched with ground-based climate and ecmwf reanalysis atmospheric predictors in southeast queensland, australia," *Renewable and Sustainable Energy Reviews*, vol. 113, p. 109293, 2019.

[4] C. Rollins, *Forecasting Energy Demand & Peak Load Days with the Inclusion of Solar Energy Production.* Rochester Institute of Technology, 2020.

[5] K. Chapagain, S. Kittipiyakul, and P. Kulthanavit, "Short-term electricity demand forecasting: Impact analysis of temperature for thailand," *Energies*, vol. 13, no. 10, p. 2498, 2020.

[6] J. P. d. S. Catalão, S. J. P. S. Mariano, V. Mendes, and L. Ferreira, "Short-term electricity prices forecasting in a competitive market: A neural network approach," *electric power systems research*, vol. 77, no. 10, pp. 1297–1304, 2007.

[7] T. Hong *et al.*, "Energy forecasting: Past, present, and future," *Foresight: The International Journal of Applied Forecasting*, no. 32, pp. 43–48, 2014.

[8] I. Ghalehkhondabi, E. Ardjmand, G. R. Weckman, and W. A. Young, "An overview of energy demand forecasting methods published in 2005–2015," *Energy Systems*, vol. 8, no. 2, pp. 411–447, 2017.

[9] Y. Yang, Y. Chen, Y. Wang, C. Li, and L. Li, "Modelling a combined method based on anfis and neural network improved by de algorithm: A case study for short-term electricity demand forecasting," *Applied Soft Computing*, vol. 49, pp. 663–675, 2016.

[10] S. Kuldeep and G. Anitha, "Short term load forecasting methods, a comparative study," *International Journal Of Advance Research And Innovative Ideas In Education*, vol. 1, pp. 31–37, 2016.

[11] K. Chapagain and S. Kittipiyakul, "Performance analysis of short-term electricity demand with atmospheric variables," *Energies*, vol. 11, no. 4, p. 818, 2018.

[12] H. Saxena, O. Aponte, and K. T. McConky, "A hybrid machine learning model for forecasting a billing period's peak electric load days," *International Journal of Forecasting*, vol. 35, no. 4, pp. 1288–1303, 2019.

[13] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[14] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of deep learning models for multi-step ahead time series prediction," *IEEE Access*, vol. 9, pp. 83 105–83 123, 2021.

[15] N. J. Johannesen, M. Kolhe, and M. Goodwin, "Relative evaluation of regression tools for urban area electrical energy demand forecasting," *Journal of cleaner production*, vol. 218, pp. 555–564, 2019.

[16] A. M. N. Ribeiro, P. R. X. do Carmo, P. T. Endo, P. Rosati, and T. Lynn, "Short-and very short-term firm-level load forecasting for warehouses: A comparison of machine learning and deep learning models," *Energies*, vol. 15, no. 3, p. 750, 2022.

[17] Z. Chang, Y. Zhang, and W. Chen, "Effective adam-optimized lstm neural network for electricity price forecasting," in *2018 IEEE 9th international conference on software engineering and service science (ICSESS).* IEEE, 2018, pp. 245–248.

[18] X. Wang, F. Fang, X. Zhang, Y. Liu, L. Wei, and Y. Shi, "Lstm-based short-term load forecasting for building electricity consumption," in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE).* IEEE, 2019, pp. 1418–1423.

[19] S. B. Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition," *Expert systems with applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[20] J. Wang and Y. Li, "Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy," *Applied energy*, vol. 230, pp. 429–443, 2018.

[21] R. Chandra and S. Chand, "Evaluation of co-evolutionary neural network architectures for time series prediction with mobile application in finance," *Applied Soft Computing*, vol. 49, pp. 462–473, 2016.

[22] V. Eramo, F. G. Lavacca, T. Catena, and P. J. P. Salazar, "Application of a long short term memory neural predictor with asymmetric loss function for the resource allocation in nfv network architectures," *Computer Networks*, vol. 193, p. 108104, 2021.

[23] J. Steinbuks, J. de Wit, A. Kochnakyan, and V. Foster, "Forecasting electricity demand," Jan 2017. [Online]. Available: https://openknowledge.worldbank.org/handle/10986/26189

[24] D. Chakraborty and H. Elzarka, "Advanced machine learning techniques for building performance simulation: a comparative analysis," *Journal of Building Performance Simulation*, vol. 12, no. 2, pp. 193–207, 2019.

[25] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on nuclear science*, vol. 44, no. 3, pp. 1464–1468, 1997.

[26] N. Mehdiyev, D. Enke, P. Fettke, and P. Loos, "Evaluating forecasting methods by considering different accuracy measures," *Procedia Computer Science*, vol. 95, pp. 264–271, 2016.

[27] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[29] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[30] T. Ahmed, K. Muttaqi, and A. Agalgaonkar, "Climate change impacts on electricity demand in the state of new south wales, australia," *Applied Energy*, vol. 98, pp. 376–383, 2012.

[31] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019.

[32] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[33] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting," *Electronics*, vol. 8, no. 8, p. 876, 2019.

[34] M. Intelligence, "Australia solar power market trends, report, industry analysis: 2022 - 2027," 2022. [Online]. Available: https://www.mordorintelligence.com/industry-reports/australia-solar-power-market-industry

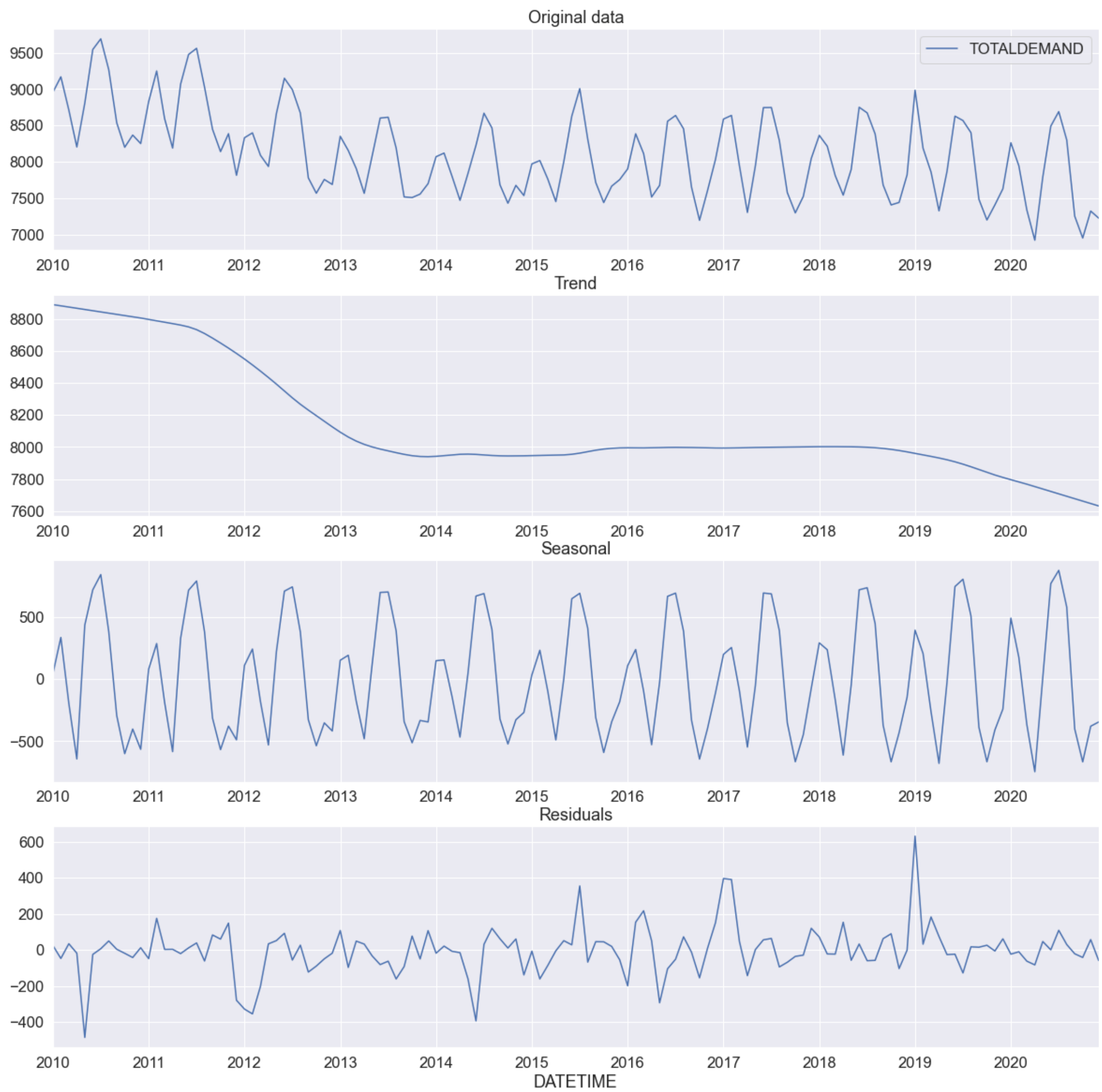
[35] "Area of australia - states and territories," 2001. [Online]. Available: https://www.ga.gov.au/scientific-topics/national-location-information/dimensions/area-of-australia-states-and-territories

Appendices

Appendix 1: Python Libraries and Versions

Name	**Version**
holidays	0.13
ipython	5.5.0
jupyter	1.0.0
keras	2.8.0
matplotlib	3.2.2
numpy	1.21.5
pandas	1.3.5
scikit learn	1.0.2
seaborn	0.11.2
tensorflow	2.8.0

Appendix 2: Demand Time Series Decomposition NSW (2010 -2021) Monthly



Appendix 3: Demand Box Plots - Seasons NSW (2010 -2021)