Appendix C - Group report front page

You will include the following information on the first page of your group report.

CMT307 Coursework 2 Group Project

Group number	G19
Project title	Energy Usage Prediction
Supervisor	Hanzhi Wang
	21126107 1840386 1823122 1823789 21088472 21014978 21123124 21099797

1. Introduction

Introduction

With global demand for energy increasing over the long term, and even more so in recent years (Tom Espiner, BBC news), along with the effects of climate change and the lack of clean renewable energy, there is a strong motivation for the reduction of energy usage worldwide. By predicting future energy usage readings, steps can be taken to allow humans to use energy more efficiently.

The Great Energy Predictor III from ASHRAE is a challenging task for any machine learning model, partly due to the very large nature of the datasets provided. By modelling the data and using variables such as the year the building was built, the wind speed, amongst others, the accuracy of the model can be enhanced to yield a more accurate reading. As such, it is crucially important to develop a robust model with exposure to various factors. By configuring the models to make precise predictions, we can achieve the following goals:

1. Predict the type and amount of energy used in a building based on the category of the building, the weather data, and the time of year.
2. Gain valuable insights into factors which effect the buildings energy demand. This information can then be used by building companies to improve buildings to be more efficient, and to use the conclusions made by the model to influence new building designs.
3. Alert building companies to abnormal amounts of energy usage, allowing them to identify issues for repair within the building.

2. Literature Review

The use of Machine Learning (ML) techniques in predicting future energy demands is a field that has been widely explored. Through various trials and studies, artificial neural networks (ANN) have been determined as one of the more effective techniques and are now readily used to produce accurate results (Seyedzadeh et al., 2019). After research, it has become clear that recurrent neural networks (RNN) are particularly efficient when using historic energy usage data as the input (Tun, Y.L et al., 2021). RNN's loop-like structure produces a time delay, which is especially effective when utilizing temperature data (Sun, Y et al., 2020.).

Whilst RNN has been widely used within this field, it is also commonly acknowledged that the basic model of RNN has its limitations and drawbacks. Since we are interested in long term energy prediction as well as short term, a naïve RNN tends to forget old information due to the commonly known vanishing gradient problem. To tackle this problem, we will implement LSTM-RNN model (Berriel, R.F et al., 2017), along with a basic RNN for comparison purposes.

The LSTM-RNN model was introduced by Hochreiter and Schmidhuber (1997). In the LSTM model, the summation units of the RNN model are replaced by memory units, providing the LSTM model with the capacity to store and recall information for longer (Heidari, A et al., 2020). The LSTM model has been successfully implemented to forecast energy demands and produced accurate results (Wang, J.Q et al., 2020), (Rahman, A et al., 2018).

There was some literature that has made us aware of some of the potential drawbacks of using this model. It was recognised that the LSTM model assumes knowledge of future weather conditions and does not consider any potential changes in weather (Rahman, A et al., 2018). Hence, should the weather differ significantly from our weather training data, there will be a loss in accuracy in the model. Secondly, there is a number of studies that noted difficulty in hyper-parameter tuning for this model. These difficulties include noted it took a large amount of trial and error to find the optimal parameters (Kim, T et al., 2019). It was noted that it took a combination of trial and error, grid search, random search and Bayesian optimization in order for the optimal parameters to be found (Ding, Z et al., 2021).

Thirdly, we decided to implement a decision tree model. There was a sufficient number of successful studies that give enough confidence in this model. However, from our readings we understood for the ease of use and the fact that decision trees are typically computationally inexpensive compared to other models, we may be giving up a small amount of performance (Amasyali, K et al., 2018). Based on our research we are under the impression that producing accurate results with LSTM-RNN and KNN, may be difficult and time-consuming thus it was right to potentially lose a small amount of accuracy, as this will allow to explore different techniques and produce further results for analysis and discussion. Nevertheless, there are still examples of decision tree models that produced accurate results. (Yu, Z et al., 2010) with decision tree model for energy demand prediction in buildings and their model providing 92% accuracy. Also, it was found that out of a neural network model, regression analysis, and decision trees, it was in fact the simpler decision tree model that produced the best results (Tso, G.K et al., 2007). It is worth noting that this study was carried out in 2007 and hence there have been developments in machine learning techniques since then.

3. Description of the task and dataset

The provided dataset comes from Kaggle's ASHRAE competition. It consists of data from over 1,000 buildings over a three-year timeframe within five CSV files. The building data file consists of 6 variables that provide information on each building's primary use, area, built year and floor count.. Next, there are training and test data files for weather readings which consist of 9 variables that provide information on air temperature, cloud coverage, dew temperature, precipitation depth, sea pressure, wind direction and wind speed. Lastly, there are two files, test and training both of which provide details on buildings meter readings. Overall, there are thirty-four columns that have four data types: decimal, integer, date and string.

To measure the quality of developed models the following evaluation metrics: Root Mean Squared Logarithmic Error (RMSLE).

The RMSLE is calculated as:

1) $$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(log(p_i + 1) - log(a_i + 1)\right)^2}$$

Where ϵ is the RMSLE value, n is the number of observations in the dataset, p_i is the predicted value and a_i is the target value. RMSLE is used because it is a common metric for regression problems due to it being robust to outliers, especially when compared to similar techniques such as Mean Squared Error.

Before processing, the the weather train, building data, and train files were all merged. The timestamp was broken down into hour, day of week, day of year, month and year to more accurately correlate time with meter reading.

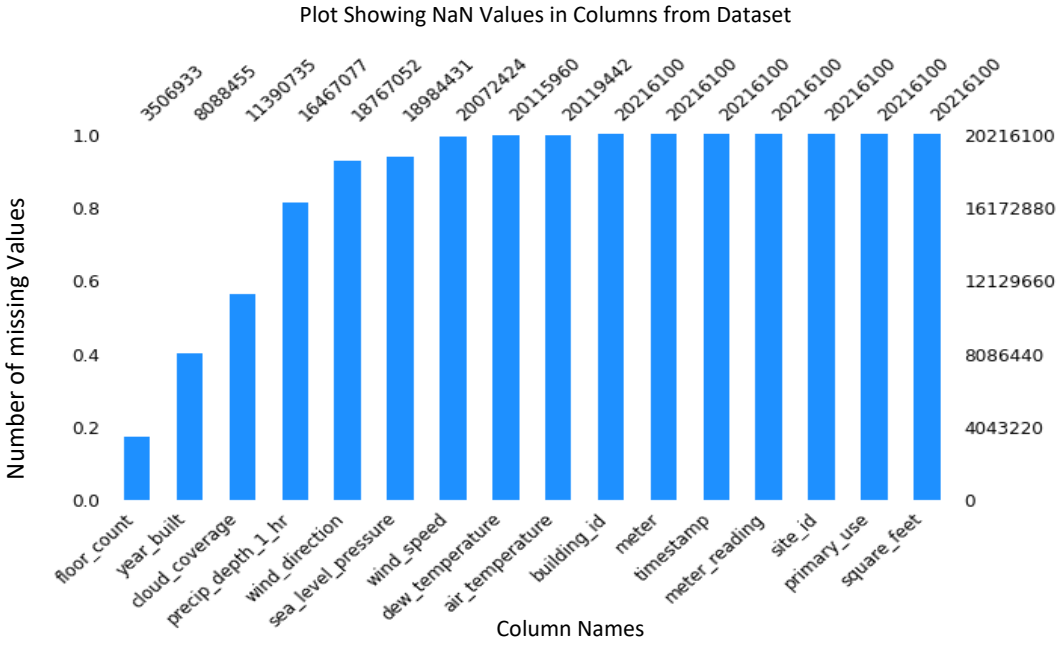


Figure 1: Graph showing the number of missing values within the categories of the dataset.

3.1 Exploratory Data Analysis

Once the data was combined, the target variable (meter reading) was explored. A log transformation of the variable was taken to adjust for high skewness and a density graph was plotted.

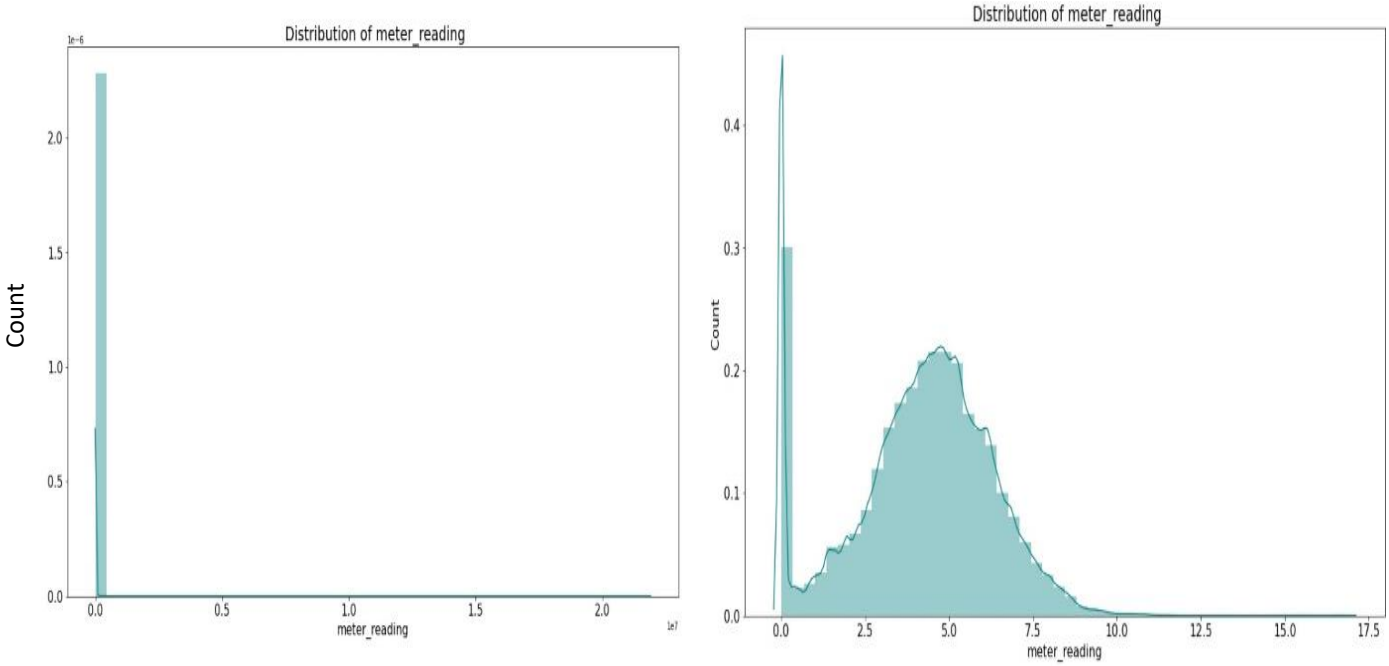


Figure 2a(left): Initial distribution of meter reading. Figure 2b (right): Logarithmic distribution of meter

Figure 2b shows a good variation in values along with a high number of 0-meter reading values. Then an exploration of any seasonality changes was made by plotting meter readings against time.

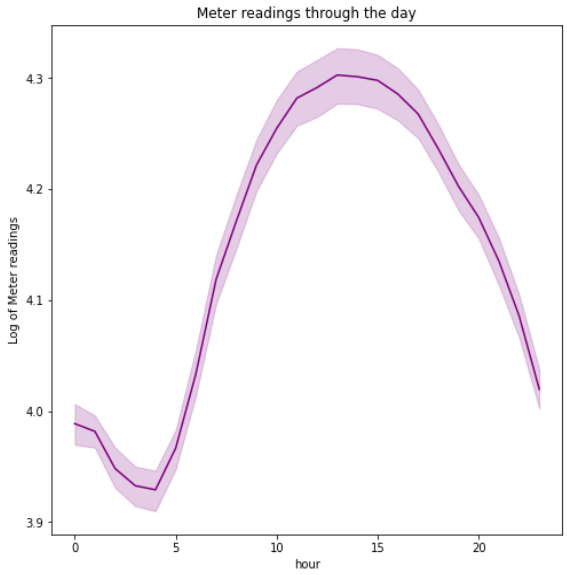


Figure 3: Average energy usage through a 24-hour period for all sites.

Figure 3 shows energy usage throughout the day. The energy usage is reasonably very low during the early morning hours and fairly high during the evening as it is the peak working time of all operating sites such as educational institutions and industrial areas.

Figure 4 shows the energy usage per annum. It is observed that energy consumption is low at the beginning of a year, rises sharply in the spring season and fluctuates during the summer season. June-September summer months report the highest level of energy consumption which may be due to the high AC usage.

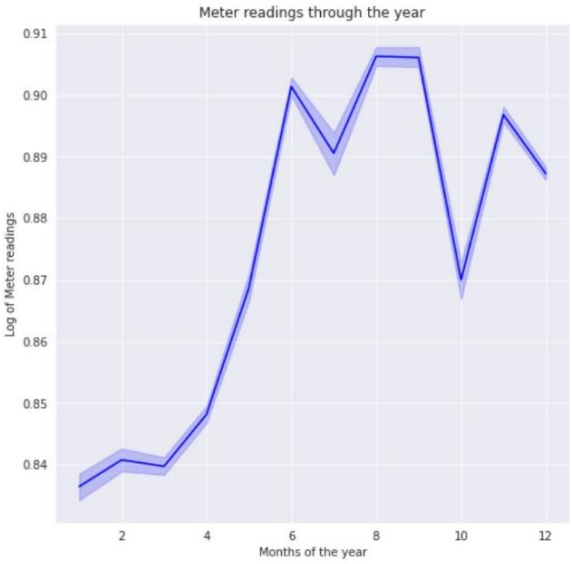


Figure 4: Average meter reading for all sites throughout months of the year.

Both Figure 5a and Figure 5b show the distribution of meters at the building and energy consumption from each of these meter types. Electricity meters are the most commonly used by chilled water and steam meter types. Also, steam and chilled water meter types consume the greatest energy followed by electricity type. It might be useful to replace hot water meter type with steam as it can greatly save energy.

Bar plot displaying the distribution of meters across the Dataset

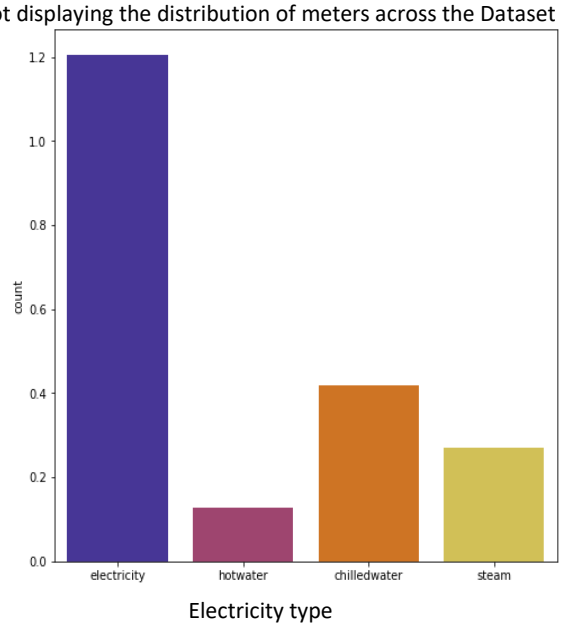


Figure 5a: Average meter by type of meter over all buildings.

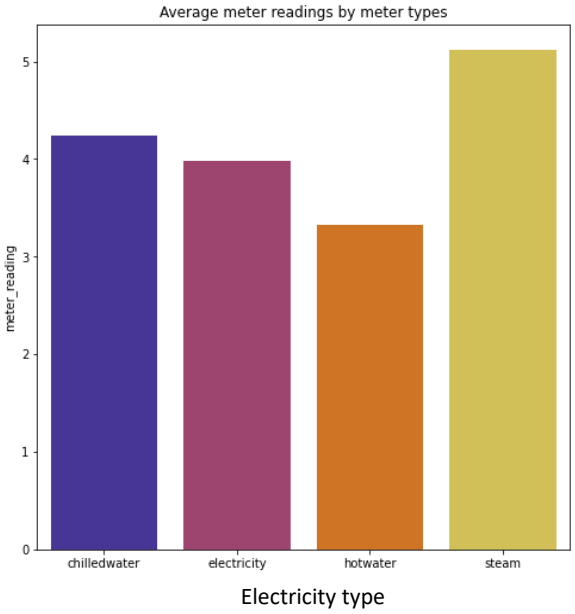


Figure 5b: Average meter reading by type of meter over all buildings.

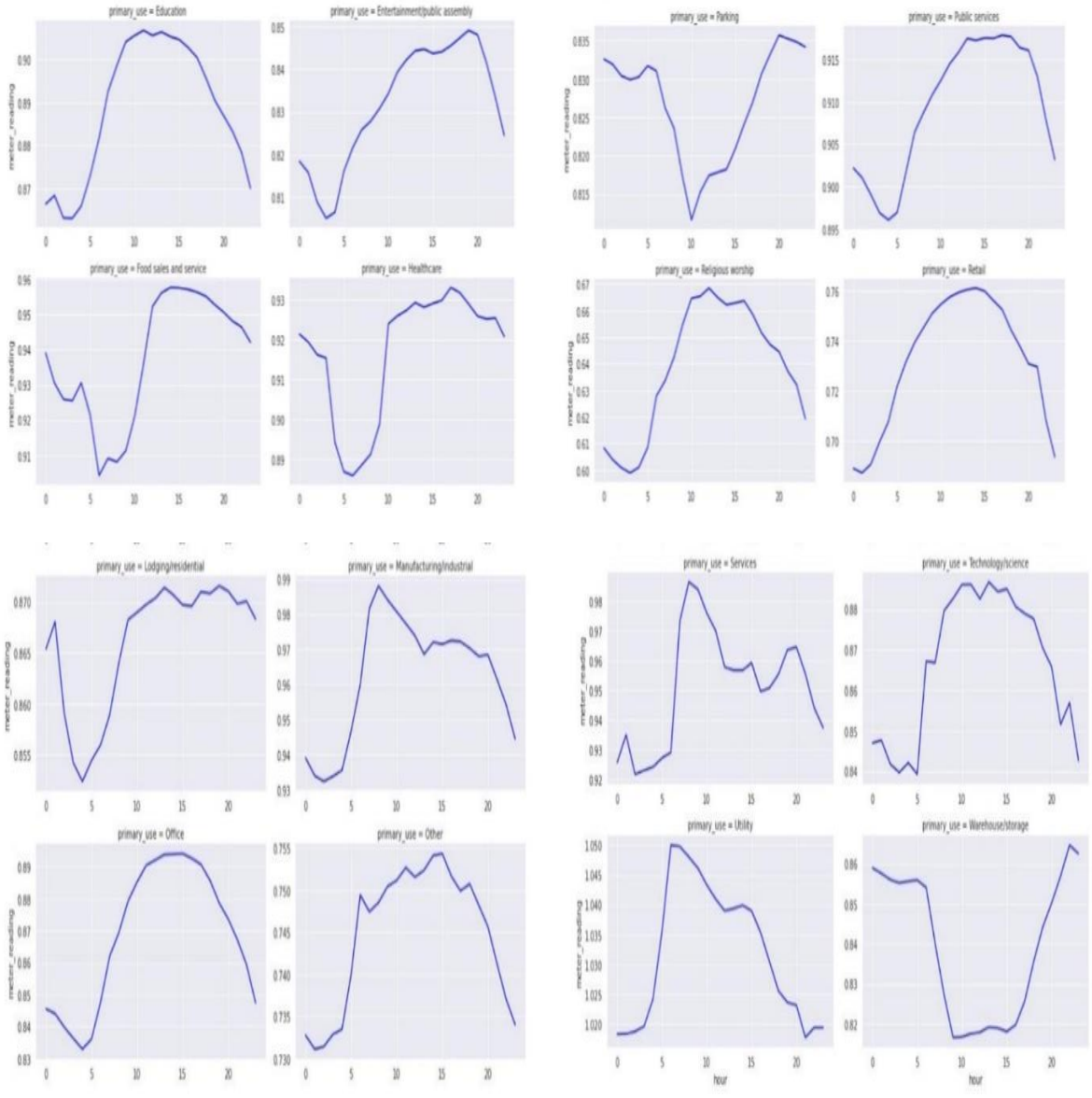


Figure 6: Average meter reading based on building type.

A meter reading distribution was explored based on primary usage in different areas. It is seen that educational institutions, offices and retail sites have the most energy consumption during the morning and evening times and the least consumption at the night.For entertainment and public assembly sites, there is low energy usage during night times and greater energy usage during evening times. For residential areas, there is a sharp decline after midnight and then meter reading keeps on increasing and reaches a high level and remains stable until midnightIt is directly related to the higher level of activities being performed throughout the day in a house that utilizes various appliances. Also, an overall analysis of the graphs shows that utility, industrial, healthcare and food sales sites report higher levels of energy consumption, whereas worship areas and retail sites consume a lower level of energy (figure 6).

3.1.1 Analysis of weather data

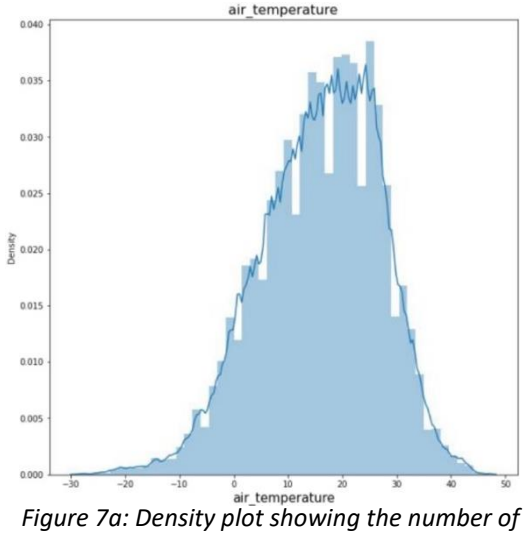


Figure 7a: Density plot showing the number of temperatures recorded at a certain temperature.

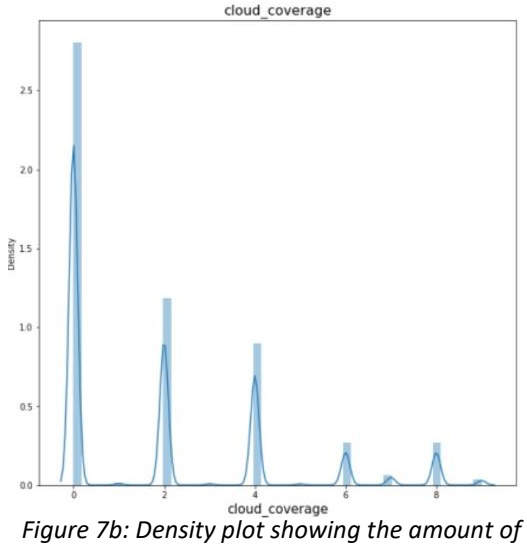


Figure 7b: Density plot showing the amount of cloud coverage recorded at a certain cloud coverage (Coverage in arbitrary units)

A density plot graph shows that air temperature variable follows a normal distribution (figure 7a & 7b). The mean value of air temperature is ~14.5 degrees Celsius. Cloud coverage is measured between a 0 to 9 scale where 0 means it is a clear sky and 9 means it is rainy. It is observed that most of the cloud coverage is zero. Sea level pressure follows a normal distribution with most values in the range of 1000-1025. Dew temperature has a skewed distribution with most values between 0-25 degrees (figure 8a & 8b)

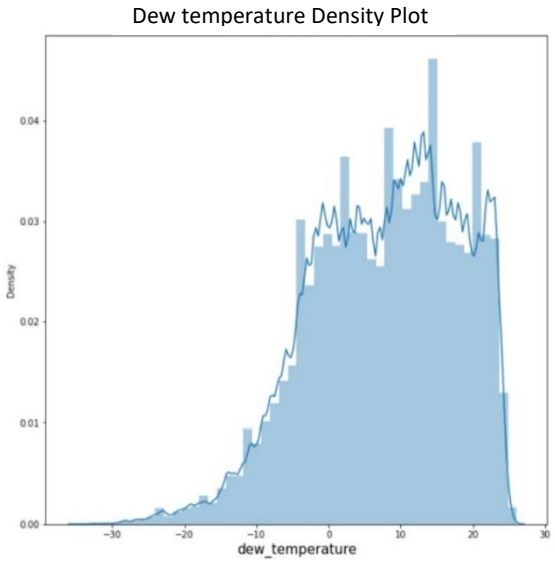


Figure 8a: Graph showing density of dew temperature.

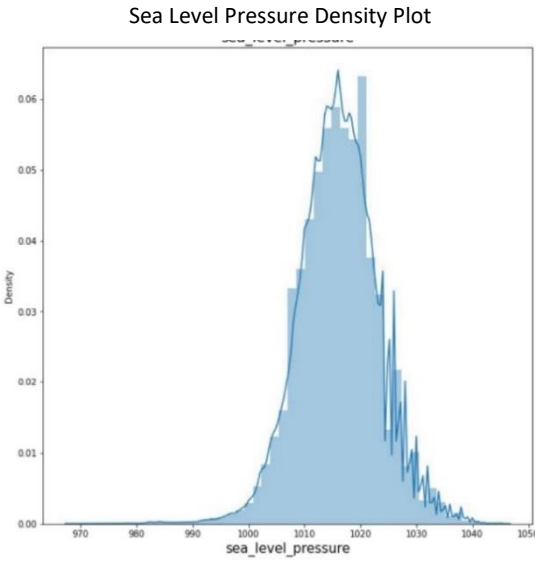
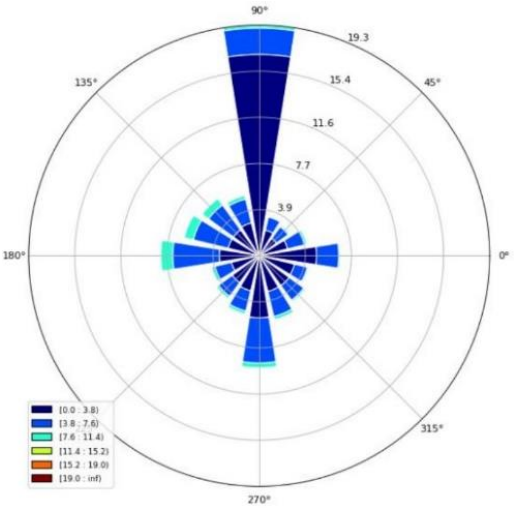


Figure 8b: Graph showing density of sea level pressure.



For a majority of the building sites, the wind mostly blows from the north direction with its speed between 0 to 3.8 m/s., followed by then south direction. Also, the Northeast direction has the minimum wind pressure.

Figure 9: Graph showing the wind speed and directions over all buildings.

3.2.2 Analysis of building data

Plot showing how the year each building was built

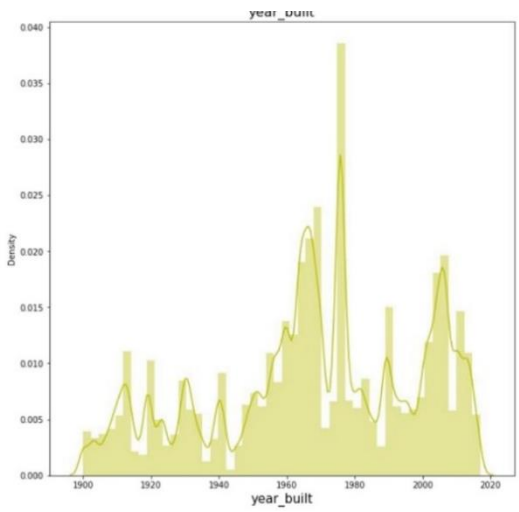


Figure 10a: Graph showing the number of buildings built in a particular year

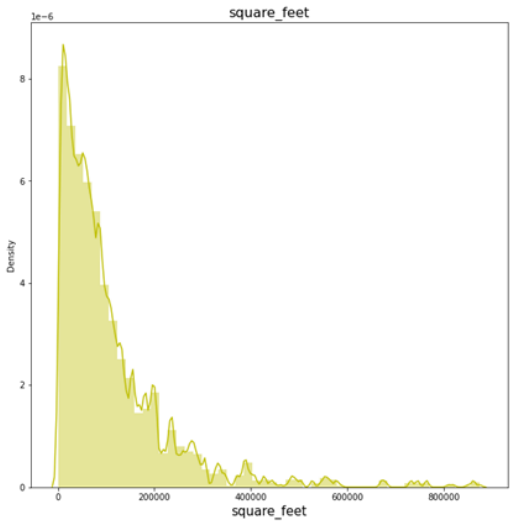


Figure 10b: Graph showing the number of occurrences of a particular square footage over all buildings.

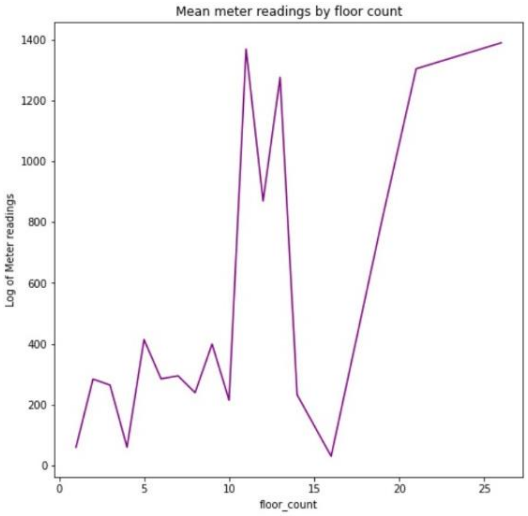


Figure 10c: The mean meter reading based on floor count of a building.

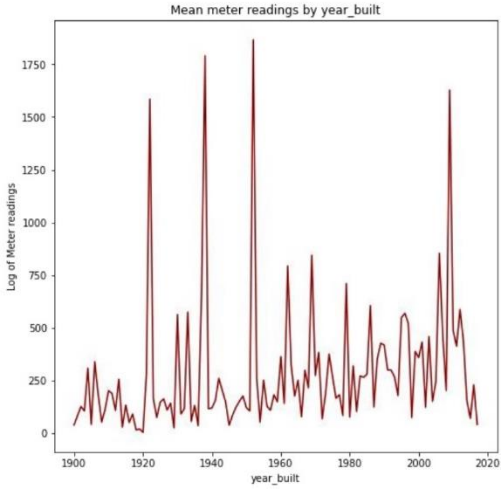


Fig.10d: Mean meters by year built

Figure 10 demonstrates the relationships (or lack thereof) between building year, square feet, floor count, and meter reading. There is little relationship between the year built and meter reading, however there is a clear correlation between both square feet and floor count with meter reading.

3.2.3 Correlation Matrix

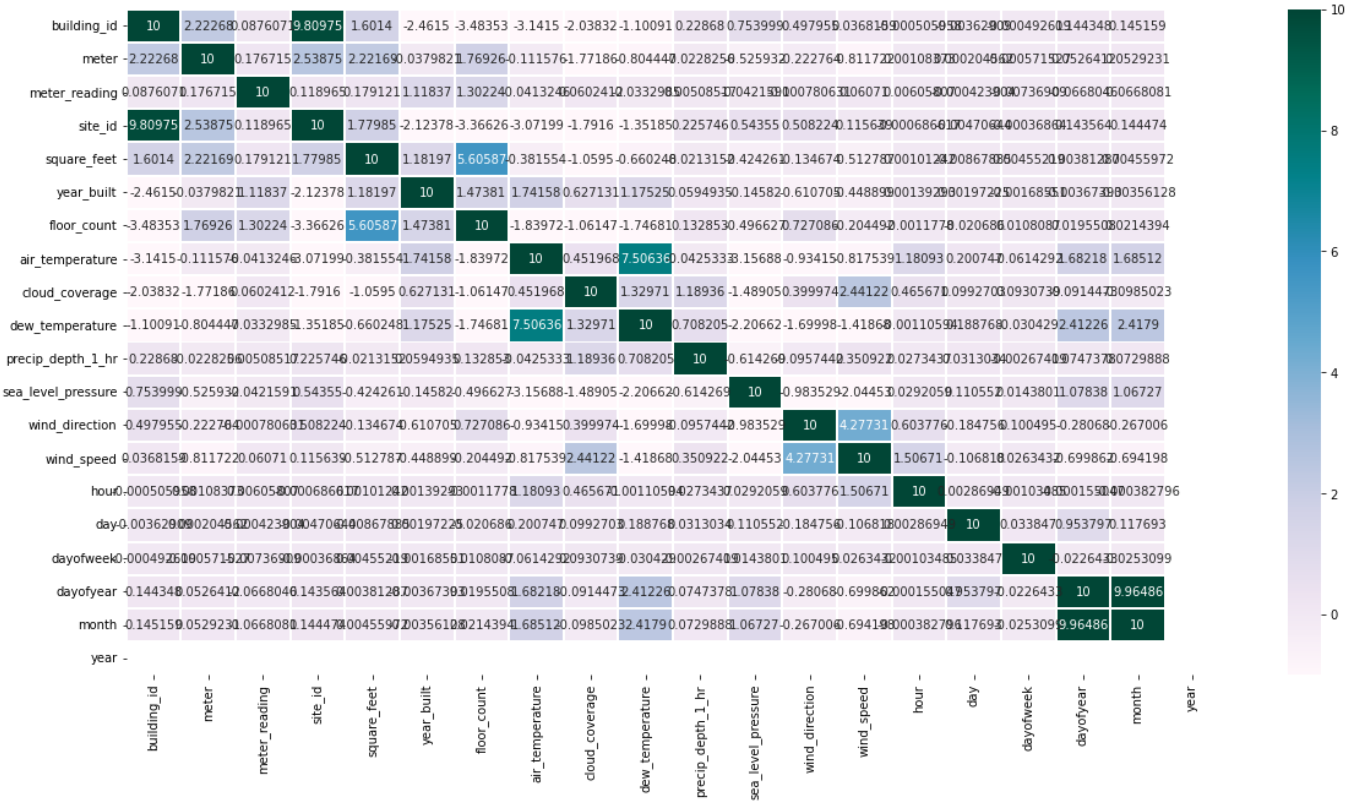


Figure 11: Correlation matrix demonstrating the correlation between variables for the purposes of feature selection.

It was plotted to understand relationships between feature variables. Square feet and year built have a positive corelation with the target variable. The greater the size of the building, the more energy it consumes. Air temperature is highly correlated with dew temperature and square feet is correlated with floor count. However, most of the features are less in correlation with meter reading.

4. Methodology

Lightgbm: LightGBM uses the leaf-wise tree growth algorithm, while many other popular tools use depth-wise tree growth. Compared with depth-wise growth, the leaf-wise algorithm can converge much faster however, this does have a tendency to cause over-fitting if not used with the appropriate parameters.

RNN: It is a form of machine learning that returns on itself to increase accuracy. For a typical neural network, firstly, an example from a dataset is loaded (in this case, the training data supplied). The network takes that example, and applies mathematical formulae to it using random variables, yielding a predicted result. Using the validation data, this prediction can be compared, and the difference between them will give an error. Returning the error back through the same path will adjust the variables, and this is all repeated until the variables are defined with minimal error. The recurrent neural network takes this a step further, by instead of taking in one example at a time and producing one result, it takes multiple neural networks which feed information to each other. This allows for low time complexity and makes it suitable for dealing with large datasets.

Decision Tree: It uses rules learnt during its training phase to make decisions. You begin with the root node, which splits off into two different regions. These regions also split into further two different regions and this process continues. It uses the rules supplied to decide which region to go to at a node, and this process is continued until all the rules are applied or until there are no data points left. The decision tree time complexity is of form O (n log(n) * d), where d is dimensionality in the data. A random forest uses multiple decision trees with an element of randomness for more accuracy, however, due to the multiple decision trees the computational memory cost is significantly higher.

4.1 Model Parameter Tunning:

To train the model and fit into the test data, which has around 20M and 41M rows respectively, the "groupNum_train" feature has been implemented, it divides the data into 60 groups, using 15 site_id's and 4 meter types, during the training and testing phase, a FOR loop has been implemented to iterate within 60 groups and get X_train, Y_train, X_valid, Y_valid, X_test and Y_test. These files are deleted after training into each iteration, to save memory and only will then only output the log. From here the regression errors and model has been saved, suffixing with the groupNum_train.

To split the data into train and validation, k-fold cross validation without shuffling was used, so that near-term data was not included in validation. In the dataset, multiple combination of folds was used to split the data, where it was observed that 3 folds works the best.

To pick the best iterations RMSE was implemented in the metric, and early stop was set to 20 iterations. After 20 consecutive rounds if the model's accuracy fails to improve then the LGBM stops the training process. Early stopping rounds reduce the overall training time.

5. Experimental Setting

As part of the pre-processing, one of the main factors was memory usage due to the large nature of the data. Converting the data to feather files and applying a function to reduce the memory usage (mainly by converting int64s to int8s, or float64s to float32s) allowed to run the programme more efficiently with very little loss of data. The exact method is provided in the 'Memory_Management.py' file.

5.1 Data Pre-processing & Feature Selection

Data pre-processing and feature selection with feature engineering procedures were followed to clean the data and feed it into the machine learning algorithms. It was observed by deep diving into the data that there is many combinations of data cleaning, and the models can be fed with the pre-processed data using similar model parameters to get higher accuracy.

5.1.1 Site Analysis

In the weather data, it was observed that at 16 sites, an hourly weather report was provided, but the site id has been encoded with integer values ranging from 0 to 15. Using an external weather report from the Kaggle dataset it was imperative that the sites can be divided into time zones. As shown from the initial data analysis, the energy utilization depends highly on the topographical season and the time of the day. The technique that was focused on the analysis is to match the temperature from the historic data with the weather data. It should be noted that the longitude and latitude with which the site id's matched with the historical data is not precise but was close enough to make a conclusion. It can be accessed from "site_analysis.py" file.

A spearman correlation plot was made between the temperature on particular dates and hours, between the Historic data and the weather data.

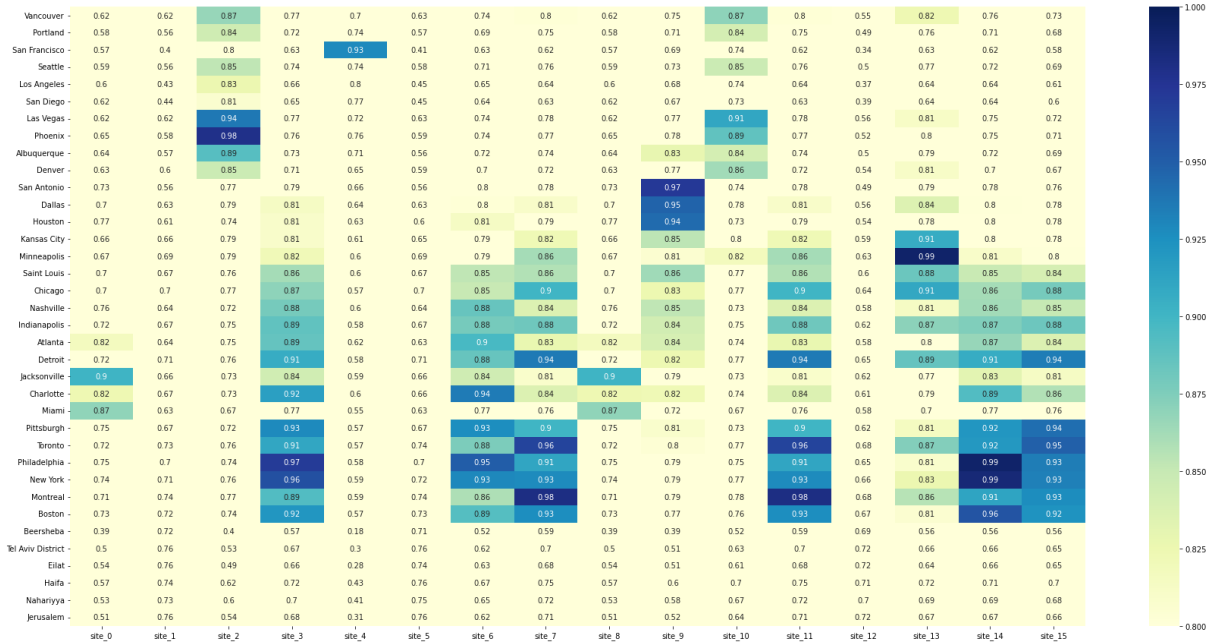


Figure 12: Spearman correlation coefficient between temperature and city on particular dates.

Based on this analysis, and using the hour offset (offset.hour()) method, the date was changed according to the local timezone. Also, a feature (Flag) "IsHoliday" was created as it was imperative that certain commercial buildings will consume less energy during holidays and weekends compared to non-holiday weekdays, for this particular task a method was created using Holidays library.

5.1.2 Feature Creation and Outlier Treatment

To combine the correlation between "floor_count" and "square_feet" that was shown in Fig.12, a new feature "floor_area" was created by dividing "square_feet" by "floor_count".

To train the model based on "site_id" and "meter" a new feature "groupNumTrain" was created, using this feature the dataset was divided into 60 batches and 60 models were made.

Each model was then fitted and predicted on the 60 batches of test data, this method helped by using less memory during both training and prediction of some models.

Since all the electricity meter reading is zero until May 20 2016 for site id = 0 and building id <= 104, this particular set of data was removed from training.

All the corresponding rows of building id which have zero meters reading from the start to a certain date were removed. Figure 13 shows the difference between before and after removal of zero meter reading, from building id 954.

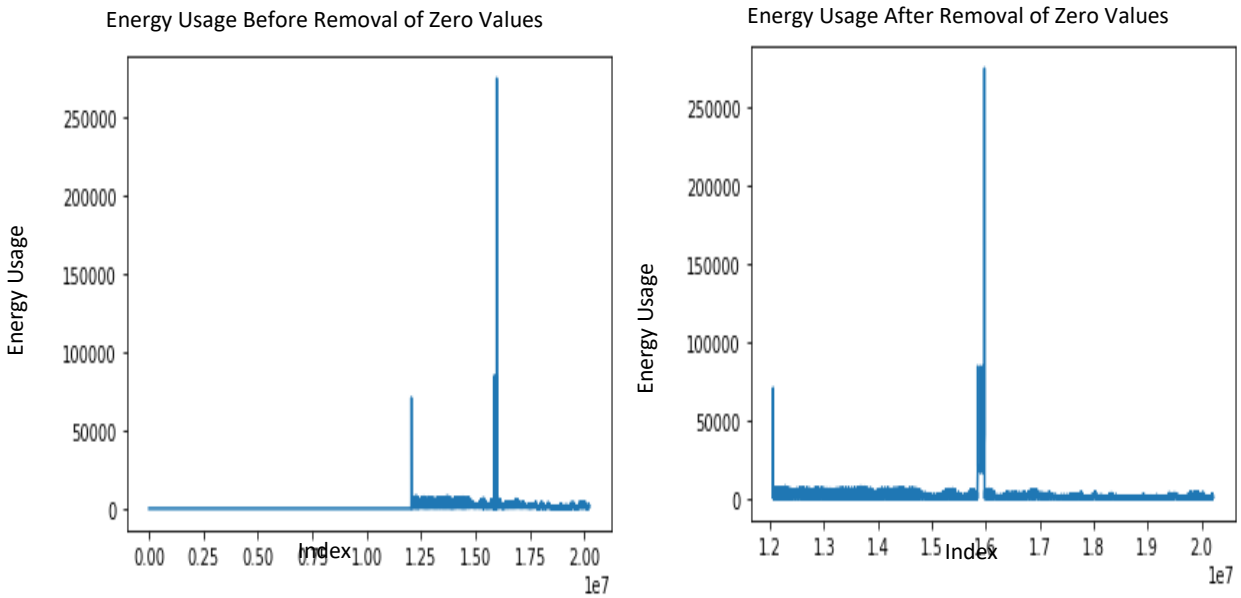


Figure 13: Left: Before removal of zero meter readings. Right: After removal of zero meter readings. Both are for building id 954.

5.1.3 Site-0 Correction

The meter readings for site 0 were not in kWh, as the rest of the data, but in kBTU. To convert it to kWh so it is comparable to the rest of the data, the site 0 readings were multiplied by 0.2931 (Dane, 2019).

5.1.4 Missing Value Imputation

The datasets provided for the project had a great deal of missing values, which led to various attempts at filling them. For the lightBGM and LSTM models, the pandas interpolate method was used to fill in the missing values with a forward fill implementation. For the RNN and Decision Tree models, columns with less than 50% missing data would be removed, whereas columns with more than 50% missing data had the remaining data points replaced with whatever the median value for that column was.

5.1.5 Adding Lag Features.

As we have time series kind of data, it has been imperative to add lag features in our data. A lag features is a variable which contains data from prior time steps. If we have time-series data, we can convert it into rows. Every row contains data about one observation and includes all previous occurrences of that observation. (mikulskibartosz, 2019)

Lag features has been added to air_temperature, cloud_coverage, dew_temperature, precip_depth_1_hr, sea_level_pressure, wind_direction, wind_speed.

5.1.6 Encoding and Counting

Two features bid_cnt and year_cnt has been introduced by encoding and counting categorical feature, building_id and year_built, this feature has proved to be useful and helped lower the RMSLE in LightGBM modelling.

In general, encoding is an important technique in machine learning as many algorithms cannot take in categorical feature, and techniques such as one-hot encoding increase the shape of the dataset, using is again memory inefficient while training large sets of data. (SagarDhandare, 2022)

New features, including bid_cnt and year_cnt were added to the data by encoding categorical features, for use in the lightGBM model. Furthermore, features like month was also added to the pre-processed data, however this was only useful in the recurrent neural network. By adding these new features, better scores of RMSLE were able to be obtained. However, care was taken not to apply encoders (such as one hot encoder) to the entirety of the dataset, as this can lead to poor memory usage when training large sets of data (Sagar Dhandare, 2022) due to the increase in shape of the dataset.

5.1.7 Smoothing Filter for Weather Data

It is imperative to smoothen the data as the weather data is time-sensitive, Savitzky-Golay filter was used to do so.

To smoothen the time-sensitive weather data, the Savitzky-Golay filter was used. Smoothing also reveals the trend and patterns within the data (Ridolfi, n.d.). Figure [INSERT NUMBER] demonstrates the relationships between air temperature and dew temperature

Smoothing also reveals the trend and patterns in the data. (Ridolfi, n.d.)

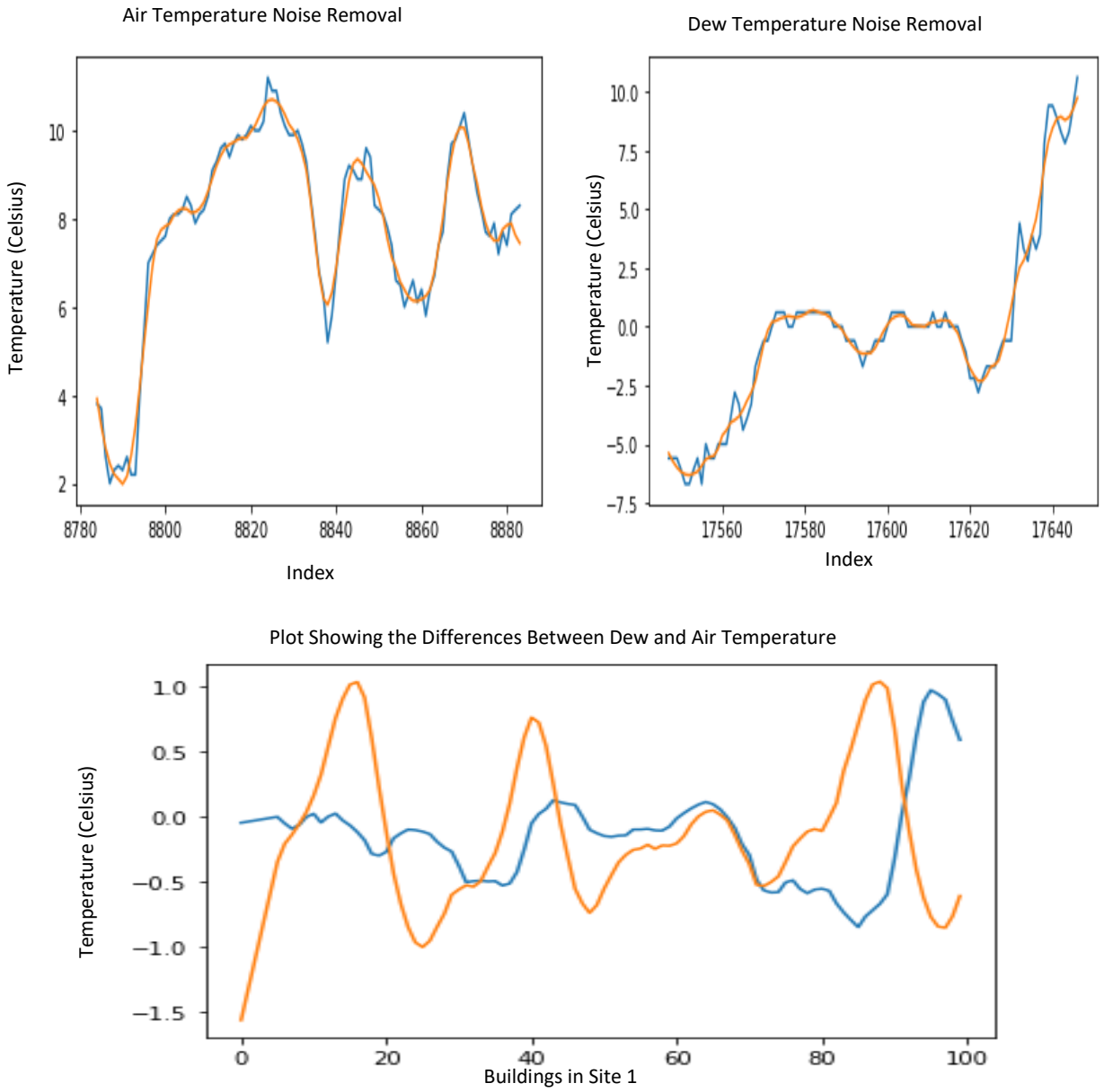


Figure 14: Top-left: Air temperature, smooth and unsmoothed. Top-right: Dew temperature, smoothed and unsmoothed. Bottom: Both smoothed air and dew temperature.

Figure 14 reveals trend and remove noise from air_temperature and dew_temperature and reveals the sine inverse proportionality between air_temperature and dew_temperature.

For the random forest model, the log transformation was not used as converting the raw train meter readings to integer values instead produced higher accuracy rates. However, to achieve accurate results it also required a large enough maximum depth which, in turn, increases the complexity, and therefore memory usage of the model exponentially.

With these adjustments, the hyperparameters that were found to have the greatest impact were the random state and max depth of the trees. To tune these hyperparameters, a small sample from the train data was looped and the highest accuracy values for these were found.

Figure 15: Graph showing the random state parameter test.

As expected, the random state showed a random distribution of accuracy values, and therefore the highest point the most accurate value that was tested was used. This hyperparameter is especially valuable when considering that the random state of the value has very littlr negative effects on memory or performance of the model.

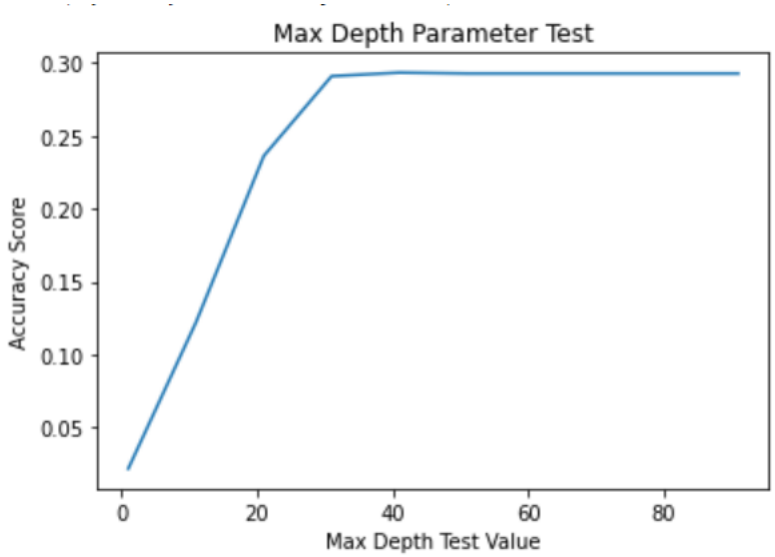


Figure 16: Graph showing max depth parameter test.

Figure 18 shows an inverse logarithmic shape, with it plateauing at roughly 40. Therefore, any value above this would cause the model to be unnecessarily complex and cause performance issues. However, the memory requirements of the max depth value of 40 was far too great for our

computational resources. The value of max depth 14 was the maximum possible with the computational power available and was therefore the compromise used.

6. Results

Table 1 demonstrates the results of the RMSLE calculated from the models used. The baseline average here is the RMSLE if the median from the meter readings within the training data is used as a prediction for all the meter readings in the test data. Given that this is just a simple average, we compare our models to this, and we are expecting them to have a lower RMSLE.

Model	**RMSLE (From Kaggle Submission)**
Decision tree	2.391
Lightbgm	1.060
RNN	2.069
LSTM	5.616*
Baseline Average	3.069

Table 1: Each of the models with their respective RMSLE score. (*This is due to the model being only binary due to computational limitations)

7. Analysis

In summary, most methods produce predictions with high levels of accuracy, with the worst being LSTM with an RMSLE of 5.616, and the best being Lightbgm with an RMSLE of 1.060. For some of the models, such as the decision tree model, there were certain performance and memory issues (Due to time complexity of O (m n^2), where m is the size of the data and n is the number of layers) that prevented them from producing nearly as accurately as they could and extra hyperparameter tuning could have been performed. Whilst the RNN model is not as accurate as Lightbgm, the short running time and low memory usage makes it an effective model to use if speed is a priority. As the two best models were the neural networks, it can be concluded they were the most effective method for this task.

In terms of errors, the greatest potential for errors within this project was the dataset itself. Some columns within the training dataset had to be removed entirely (Columns with greater than 50% missing data). Columns with less than 50% missing data had the missing data points replaced with the median of that column. Whilst it would be ideal if all the data was available, by keeping as many columns as possible, this led to optimal feature selection to enhance the accuracy of the neural network models.

Each model has weaknesses that could cause errors. LSTM and Lightbgm are prone to overfitting and RNN frequently has gradient vanishing and exploding problems. Decision trees are unstable if there is a small change in the training data, the optimal tree may have large changes.

7. Conclusion

During this project, we have successfully and unsuccessfully used numerous types of models with differing strengths and weaknesses. What is clear is through the overall accuracy and computing performance, is that this project is best suited to Neural Networks rather than classifications/ regression models as they provided a higher accuracy prediction, in a shorter time. With the strong results found, this data could be used to find which buildings are more efficient and can therefore be used in deciding future building designs as well as be used to discover anomalous results in the future which could be a sign of technical issues in buildings. Therefore, if this research were to be repeated or explored further, our suggestion would be to further explore Neural Networks, where a good starting spot would be a fully optimised LSTM. From there other deep-learning methods could be best suited and produce further great results. However, it has been shown by our results that some models such as Decision Tree can provide fairly accurate results if computational power and time were no issue.

References

Amasyali, K. and El-Gohary, N.M., 2018. A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews*, *81*, pp.1192-1205.

Berriel, R.F., Lopes, A.T., Rodrigues, A., Varejao, F.M. and Oliveira-Santos, T., 2017, May. Monthly energy consumption forecast: A deep learning approach. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 4283-4290). IEEE.

Deb, C., Zhang, F., Yang, J., Lee, S.E. and Shah, K.W., 2017. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, *74*, pp.902-924.

Ding, Z., Chen, W., Hu, T. and Xu, X., 2021. Evolutionary double attention-based long short-term memory model for building energy prediction: Case study of a green building. *Applied Energy*, *288*, p.116660.

Heidari, A. and Khovalyg, D., 2020. Short-term energy use prediction of solar-assisted water heating system: Application case of combined attention-based LSTM and time-series decomposition. *Solar Energy*, *207*, pp.626-639.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, *9*(8), pp.1735-1780.

Kim, T.Y. and Cho, S.B., 2019. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, *182*, pp.72-81.

Long, H., Zhang, Z. and Su, Y., 2014. Analysis of daily solar power prediction with data-driven approaches. *Applied Energy*, *126*, pp.29-37.

Olu-Ajayi, R., Alaka, H., Sulaimon, I., Sunmola, F. and Ajayi, S., 2022. Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. *Journal of Building Engineering*, *45*, p.103406.

Rahman, A., Srikumar, V. and Smith, A.D., 2018. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied energy*, *212*, pp.372-385.

Seyedzadeh, S., Rahimian, F.P., Rastogi, P. and Glesk, I., 2019. Tuning machine learning models for prediction of building energy loads. *Sustainable Cities and Society*, *47*, p.101484.

Sun, Y., Haghighat, F. and Fung, B.C., 2020. A review of the-state-of-the-art in data-driven approaches for building energy prediction. *Energy and Buildings*, *221*, p.110022.

Troncoso Lora, A., Riquelme Santos, J.M., Riquelme, J.C., Gómez Expósito, A. and Martínez Ramos, J.L., 2003, November. Time-series prediction: Application to the short-term electric energy demand. In *Conference on Technology Transfer* (pp. 577-586). Springer, Berlin, Heidelberg.

Tso, G.K. and Yau, K.K., 2007. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy*, *32*(9), pp.1761-1768.

Tun, Y.L., Thar, K., Thwal, C.M. and Hong, C.S., 2021, January. Federated learning based energy demand prediction with clustered aggregation. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 164-167). IEEE.

Wahid, F. and Kim, D., 2016. A prediction approach for demand analysis of energy consumption using k-nearest neighbor in residential buildings. *International Journal of Smart Home*, *10*(2), pp.97-108.

Wang, J.Q., Du, Y. and Wang, J., 2020. LSTM based long-term energy consumption prediction with periodicity. *Energy*, *197*, p.117197.

Yu, Z., Haghighat, F., Fung, B.C. and Yoshino, H., 2010. A decision tree method for building energy demand modeling. *Energy and Buildings*, *42*(10), pp.1637-1646.

Dane, S. (2019). *ASHRAE - Great Energy Predictor III*. [online] kaggle.com. Available at: https://www.kaggle.com/c/ashrae-energy-prediction/discussion/119261#latest-684102 [Accessed 24 Apr. 2022].

mikulskibartosz (2019). *Forecasting time series: using lag features*. [online] Bartosz Mikulski. Available at: https://www.mikulskibartosz.name/forecasting-time-series-using-lag-features/ [Accessed 24 Apr. 2022].

SagarDhandare (2022). *What Is Encoding? And Its Importance in Data Science!* [online] Medium. Available at: https://medium.datadriveninvestor.com/what-is-encoding-and-its-importance-in-data-science-6a2b0cce8e8e [Accessed 24 Apr. 2022].

Ridolfi, A. (n.d.). *Smoothing Your Data with the Savitzky-Golay Filter and Python – Finxter*. [online] https://blog.finxter.com/. Available at: https://blog.finxter.com/smoothing-your-data-with-the-savitzky-golay-filter-and-python/.

Reccurant Neural Networks. Available at: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

Jiang Su and Harry Zhang, A Fast Decision Tree Learning Algorithm, University of New Brunswick, NB, Canada

Tom Espiner, Energy bills could reach £3,000 as prices soar. Available at: https://www.bbc.co.uk/news/business-60600049