Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

SCHOOL OF ENGINEERING

INVESTIGATING THE PERFORMANCE OF DIFFERENT MACHINE LEARNING MODELS PREDICTING PARTICULATE MATTER IN DUBLIN

PARAIC O'REILLY

APRIL 17, 2023

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

B.A. (MOD.) COMPUTER ENGINEERING

Declaration

I hereby declare that this Final Year Project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed: Paraic O'Reilly Date:17/4/2023

Abstract

Air-pollution has been identified as the largest environmental risk to public health. Fine particulate matter with a diameter of less than 2.5 microns (PM2.5), is the air pollutant associated with the most respiratory diseases. Levels of PM2.5 have been linked to ambient weather conditions such as temperature, relative humidity, mean sea level pressure and wind speed. Studies have attempted to use machine learning to predict PM2.5 levels with varying levels of success, often using these meteorological variables as predictors. In Ireland, there is less PM2.5 data available than other many other countries. Which machine learning models are most effective for predicting PM2.5 levels will change based on the available data. In this paper, the available features in Ireland are analysed and the Random Forest, Deep Learning, Gradient Boosted Trees and Gated Recurrent Units machine learning models are implemented using meteorological data as features. The models are all trained based on stations in Dublin city. The ability of each model to predict PM2.5 is analysed using three common error metrics, the Mean Squared Error, the Mean Absolute Error and the Root Mean Squared error. The highest performing model XGBoost had a MAE = 3.2, a MSE = 18, and a RMSE = 4.2, though there was similar performance between the top performing models.The importance of each feature was tested through the inbuilt feature importance function of the XGBoost model. The robustness of the top models was tested by seeing how accurately they could predict PM2.5 in another city in Ireland based on the meteorological data for that city. The top performing model for predicting in Cork was XGBoost, with a MAE = 4.2, a MSE = 31, a RMSE = 5.5.

Acknowledgements

Thanks to my Mum and Dad for all the support and knowledge they have given me throughout college and life.

Thanks to Dr. Gregory O'Hare for supervising me through this process.

Contents

List of Figures

List of Tables

1 | Introduction - Chapter

Particulate matter (PM2.5) refers to minuscule aerodynamic particles which are less than two and a half microns in diameter. Elevated levels of PM2.5 are indicative of air pollution from sources of fossil fuel production, primarily coal (1). PM2.5 is associated with more adverse health affects then any other air pollutant according to the World Health Organisation's (WHO) Global Burden of Disease Project (2). WHO's global air quality guidelines recommend a daily value of less than 15 ug/m^3 and an annual value of less than 5ug/m^3 (3). Due to the risk that increasing levels of PM2.5 presents to public health, accurately predicting and forecasting PM2.5 values is of vital importance and is increasingly the focus of academic studies (4). In Dublin, the capital of Ireland, the annual PM2.5 level was 10 ug/m^3 from the years 2015 to 2017, which is 5 ug/m^3 over the recommended WHO level (5). Although low compared to the level average levels of Europe, over two hundred premature deaths occur in Dublin per year from prolonged exposure to ambient PM2.5 (6). PM2.5 level have been linked to ambient weather conditions such as temperature, relative humidity, mean sea level pressure and wind speed (7). This paper analyses the data available to predict PM2.5 levels and takes a comparative approach to the different models available for forecasting PM2.5. Traditionally, simple regression-based methods such as linear regression have been used for forecasting and predicting air pollutant values. However, new methods have been explored in the past few years that improve upon the accuracy of these traditional methods. Non-linear methods, such as Neural Nets, have been shown to have better performance (8) for prediction but require larger data sets and have been shown to struggle with over fitting and

converging to local minima. Extreme Gradient Boosting (XGBoost) has had more success in predicting air pollutants than traditional data mining methods like random forest regression (9). Recurrent Neural Nets have produced excellent forecasting results due to the consideration of time series data as a variable. (10). Due to the length of these recurrent networks, they are prone to vanishing or exploding gradients which can negatively affect performance. New architectures have attempted to deal with these issues, primarily the Long-Term Short Memory architecture and the Gated Recurrent Networks architecture. These models have had varying degrees of success, but they have been evaluated in countries with higher levels of PM2.5 where hourly data is available. In Dublin city, and throughout Ireland, only daily mean averages are available for ambient PM2.5 levels and the average ambient PM2.5 levels are much lower than countries such as China and the US. China implemented nation-wide monitoring for air pollutants in 2016, with hourly data recorded in each station, which increases the viability of machine learning models due to the vast increase in available data. (11). Despite the lower level ambient PM2.5 levels present in Dublin; it is still vital to forecast PM2.5 to meet the standards for public health set by WHO and Environmental Protection Agency (EPA). Dublin has begun monitoring and publishing more PM2.5 data with programs like the SMART Dublin scheme, yet there is still a lack of historical hourly PM2.5 data available. This project aims to find out which methods are most effective for predicting PM2.5 from ambient weather conditions based on the available PM2.5 meteorological data for Dublin. In this project, four models will be trained and compared to a basic Linear Regression:

- Random Forest Regression
- Fully Connected Neural Nets,
- Gradient Boosted Trees,
- Gated Recurrent Units.

Data analysis will be conducted on the PM2.5 and meteorological data to find

suitable features for prediction. Feature engineering will be employed to discard unnecessary features, perform transformations, and pre-process the data. Feature engineering is a crucial step in creating a successful model - “No algorithm alone, to my knowledge, can supplement the information gain given by correct feature engineering.” (12) Once the highest performing model is identified, the model will be tested on a station outside of the Dublin area to discover the robustness of the model and identify the different challenges in forecasting and prediction outside of the study area.

2 Relevant Materials

2.1 Target Variable

PM2.5 has been the primary focus in studies that model and forecast air pollutants. This is due to the higher links between PM2.5 and heart disease and respiratory illnesses. Outdoor and indoor, ambient PM2.5 exposure is among the top ten risk factors for diseases (13). Accurately predicting levels of PM2.5 in specific locations is essential for city planning and public health. For instance, studies show PM2.5 levels around maternity hospitals is consistently above recommended WHO levels (14). Being able to predict and pinpoint elevated levels of PM2.5 can reduce damage to the public health.

2.2 Study Area

Dublin is the capital and economic hub of Ireland. The greater Dublin area accounts for over 40% of the population of Ireland and is set to grow to 2.2 million by 2031 (15). Population density is related to increased ambient PM2.5 levels in cities in developed countries (16). Therefore, monitoring and predicting PM2.5 will be of growing importance to meet the WHO air quality guidelines. The EPA has historical data sets from three stations: Marino, Rathmines, and Phoenix Park. PM2.5 is measured in micro grams per cubic metre (ug/m^3). Each station provides a daily mean average of PM2.5 levels. The historical data sets are available from January 1st, 2009, to December 31st, 2019.

2.3 Meteorological variables

Hourly weather data was obtained from January 1st, 2009, to December 31st, 2019, from MET Eireann. Features were identified which could be converted to daily averages. The features selected were ambient air temperature (°C), mean hourly wind speed (kt), relative humidity (%) and mean level sea pressure (Pa). Categorical variables such as wind direction were discarded as they could not be converted properly to daily values. Other options such as wet bulb temperature contained too much similarity to air temperature and would increase model complexity without providing added information. Each station has a different range of variables available based on the recording devices present in the station. The Dublin airport station was chosen due to the extensive range of variables available in comparison to other stations and due to its proximity to the Marino PM2.5 station (within eight kilometres).

2.4 Feature Engineering

Feature engineering is a critical step in the process of developing machine learning models, especially in cases where the quality of the model depends on the input features. Feature engineering involves the process of selecting, extracting, and transforming raw data into meaningful features that can be used to train a machine learning model. The primary goal of feature engineering is to create a set of features that accurately captures the underlying patterns and relationships present in the data, while also being interpretable and computationally efficient. This process can involve a wide range of techniques including data pre-processing, feature selection, and feature transformation.

2.4.1 Data pre-processing

Data pre-processing involves cleaning and formatting raw data which can include tasks such as removing missing values, dealing with extreme outliers, normalizing data, and scaling data to a common range. Removing extreme outliers has been shown to improve performance in predicting air pollutants such as SO2(17).

2.4.2 Box Cox Transformation (BC)

The Box-Cox transformation is a statistical technique used to transform non-normal data into a normal distribution. As noted in a study on the benefits of feature engineering (18), the Box-Cox transformation is the most common method for the normalisation of features as it can improve the predictive power of features. The statistical technique is a power transformation, where the power is determined by a parameter lambda (λ). The optimal value of lambda is determined using maximum likelihood estimation, which involves finding the value of lambda that maximizes the likelihood of observing the data given the transformed distribution. It can help to stabilize the variance of the data, reduce the impact of outliers, and improve the performance of statistical models.

$$y(\lambda) = \begin{cases} \dfrac{(y^\lambda - 1)}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(y) & \text{if } \lambda = 0 \end{cases} \tag{1}$$

2.4.3 MinMax Scaler (MMS)

This class is implemented using the Scikit-Learn library. This class scales the input features in the range 0 to 1. This is achieved by subtracting the minimum value of the feature and dividing by the range of the feature. The class provides two methods, fit, and transform.

Fit: calculates the minimum and maximum of the input data.

Transform: scales the data in a specified range.

2.5 Modelling Techniques

This project investigates the viability of five main machine learning approaches, linear regression, random forest, neural nets, gradient boosted trees, and gated recurrent units.

2.5.1 Linear Regression (LR)

Linear regression is a statistical approach to modelling the relationship between a dependent variable and one or more independent variables. Linear regression works by finding a line of best fit through a set of data points. This line represents the relationship between the independent variable(s) and the dependent variable. The line is determined by minimizing the sum of the squared differences between the predicted values and the actual values of the dependent variable. The equation of the line is given by:

$$y = mx + b \tag{2}$$

2.5.2 Decision Trees

Decision Trees construct a tree structure of nodes and branches that represent decisions and outcomes. Each node in the tree corresponds to a decision based on a specific feature or attribute of the data, and each branch represents the possible outcomes of that decision. The goal of a decision tree is to find the most optimal path through the tree that leads to the desired outcome based on the available data.

2.5.3 Random Forest (RF)

Random Forest Model is an ensemble machine learning algorithm that is based on decision tree classifiers. It works by building multiple decision trees and combining their outputs to make a final prediction (19). The decision trees are constructed by

randomly selecting a subset of features from the training data, and then selecting the best split among them based on criterion, such as the Gini impurity or information gain. This process is repeated until the tree is fully grown, i.e., all the data points are assigned to a leaf node. To make a prediction, each decision tree is traversed and the output of all the trees is aggregated to make a final prediction. It is robust to outliers and missing values as it uses only a subset of features and data points to build each tree. It can manage high dimensional data, as it can automatically select the key features and ignore the irrelevant ones. Lastly, it is computationally efficient, as it can parallelize the training process and handle large datasets.

2.5.4 Neural Nets (NN)

Neural Nets are composed of interconnected processing nodes or neurons that work together to learn complex patterns from data. A neuron, which is modelled after the structure of a biological neuron, is the fundamental building component of a neural network. A neuron receives inputs that are multiplied by a collection of weights and then summed with a bias term. This result is then fed into an activation function, which decides the neuron's output. This output is then passed on to the network's other neurons, who repeat the process until a final output is created. A neural network's weights and bias terms are initially set at random, and the network is trained using a method known as backpropagation. Backpropagation involves presenting the network with a collection of training data and comparing the network's output. (20) Neural Nets are more robust to non-normalised and skewed data as skewed data does not overtly affect the performance of the model due to the complexity present within the hidden layer ((21))

2.5.5 Gradient Boost Trees (XGBoost)

XGBoost combines ensemble decision trees with the gradient boost framework. It works by constructing a series of decision trees, each of which is constructed to correct the errors of the previous tree. The algorithm begins with a single tree that is

used to generate preliminary predictions. The first tree's residuals are then used to construct the second tree, and the procedure is repeated until a predetermined number of trees are constructed or a convergence criterion is met. The cost function used by the algorithm is improved during training. The cost function is made up of two parts: a regularization term that penalizes complex models and a loss function that gauge's model deviation. XGBoost is highest performing machine learning algorithm for many data scientists on competition sites such as Kaggle, and requires less resources than other algorithms like Neural Nets ((22))

2.5.6 Gated Recurrent Units (GRU)

Recurrent Neural Networks (RNNs) are widely used in machine learning for dealing with consecutive data such as time series data. Traditional RNNs, on the other hand, suffer from a vanishing gradient issue during the training phase, which has an impact on the model's performance. Gated Recurrent Units (GRUs) are a type of RNN suggested to address this problem. They were first proposed as an improvement over traditional RNNs in a 2014 study (23).

The GRU is made up of a reset gate and an update gate that control the movement of information in the network. The reset gate decides how much of the prior state to forget, and the update gate decides how much of the new state to keep. Gated Recurrent units was chosen for its effectiveness in dealing with linear time series data and its success with smaller datasets compared to other recurrent network architectures.

2.6 Model training

When training the various models for the project, several techniques were used to ensure the most accurate results possible were obtained. Cross validation was used to maximise the small datasets and grid search was used to find optimal hyper paramaters.

2.6.1 Cross Validation

Cross-validation is a statistical method used in machine learning to evaluate the performance of a model. It involves partitioning a dataset into several subsets or folds where each fold is used to train the model and the remaining fold(s) are used to test the model's performance. The process of cross-validation helps to address the problem of overfitting which occurs when a model is too complex and fits the training data too closely, resulting in poor generalization to new data. By using cross-validation we can estimate the model's performance on unseen data and select the optimal hyperparameters that maximize the model's generalization ability. In a typical cross-validation procedure, the dataset is randomly split into k equal-sized folds. The model is then trained on k-1 folds and tested on the remaining fold. This process is repeated k times, each time using a different fold for testing and the remaining folds for training. The results of each fold are then averaged to obtain an overall performance metric, such as accuracy or mean squared error.

2.6.2 Grid search

Grid search is a powerful tool in machine learning that helps to find the best hyperparameters for a given model. Grid search works by systematically testing different combinations of hyperparameters, and evaluating the performance of the model on a validation set for each combination. The user specifies a set of hyperparameters to be tested, and grid search generates all possible combinations of these hyperparameters to create a "grid". Then, the model is trained and evaluated for each combination of hyperparameters, and the combination that produces the best performance on the validation set is selected as the final set of hyperparameters. In Python, grid search is implemented in the scikit-learn library using the GridSearchCV function. This function takes as input a machine learning model, a dictionary of hyperparameters and their corresponding values, and a cross-validation strategy. The function then performs grid search to find the best

hyperparameters, and returns the best set of hyperparameters along with the corresponding model performance metrics.

2.7 Performance Metrics

Mean Squared Error, Root Mean Squared Error and Mean Absolute Error are the metrics chosen to evaluate model performance. Root Mean Squared error and Mean Absolute Error are the most common error metrics when measuring air pollutants in academic studies, as they provide the error in the units of the target variable.

2.7.1 Mean Squared Error (MSE)

The average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss. MSE is the most common metric for regression problems as it penalises larger errors more than smaller errors.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

2.7.2 Mean absolute Error (MAE)

Mean absoluter error measures absolute difference of the errors between paired observations that express the same phenomenon. It is less sensitive to outliers then metrics such as MSE. It portrays the average magnitude of errors without considering the direction. It is expressed as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

2.7.3 Root Mean Squared Error (RMSE)

RMSE is a statistical measure commonly used in evaluating the accuracy of a predictive model. It is a measure of the average difference between the predicted

values and the actual values in a dataset, expressed in the same units as the original data. The calculation of RMSE involves taking the square root of the mean of the squared differences between the predicted values and the actual values. This formula allows for the amplification of larger errors while still incorporating smaller errors into the calculation. The use of RMSE in scientific research is important for assessing the performance of models and for comparing the accuracy of different models. A lower RMSE indicates a better fit of the model to the data, and the RMSE can also be used to estimate the range of error in the predictions made by the model.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

2.7.4 Coefficient of Determination (R2 value)

R2 value is an evaluation of the scatter points around the fitted line which measures how well the data fits. It can be prone to overfitting as a metric and does necessarily indicate an accurate network. This occurs when the model maps to spurious patterns in the data. The R2 value is only included for the linear regression model as it is not valid for the other non-linear models. It is included to give context to what a good fit of the data looks like. This can help when analysing the other model's predictions graphically.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

By using these four metrics, we can have a comprehensive evaluation of the model's performance. MSE, MAE and RMSE provide information on the magnitude and direction of the errors, while the R2 value measures how well the model fits the data. This allows us to identify the strengths and weaknesses of the model and make informed decisions on how to improve it.

3 Methods

3.1 Data Analysis

Comprehensive data analysis was conducted on the target variable and features to help identify which modelling methods will be most effective and to identify if feature transformations will be useful. The data set contains 3652 values including null/missing values. The data set contains 3302 values without missing values. Missing values were only present for the target variable. Missing rows were dropped for all models except for the recurrent neural net. Imputation of missing values did not help with accuracy for any models. Recurrent neural nets cannot have missing values in the linear time series therefore linear interpolation was used to impute missing values. The table below displays the basic descriptive statistics for the PM2.5 data set. A distribution plot of the target variable was created using the seaborn library in python. Distribution plots identify variance in continuous variables. This helps identify the range and distribution of the target.

The distribution plot is shown below in Figure 3.1.

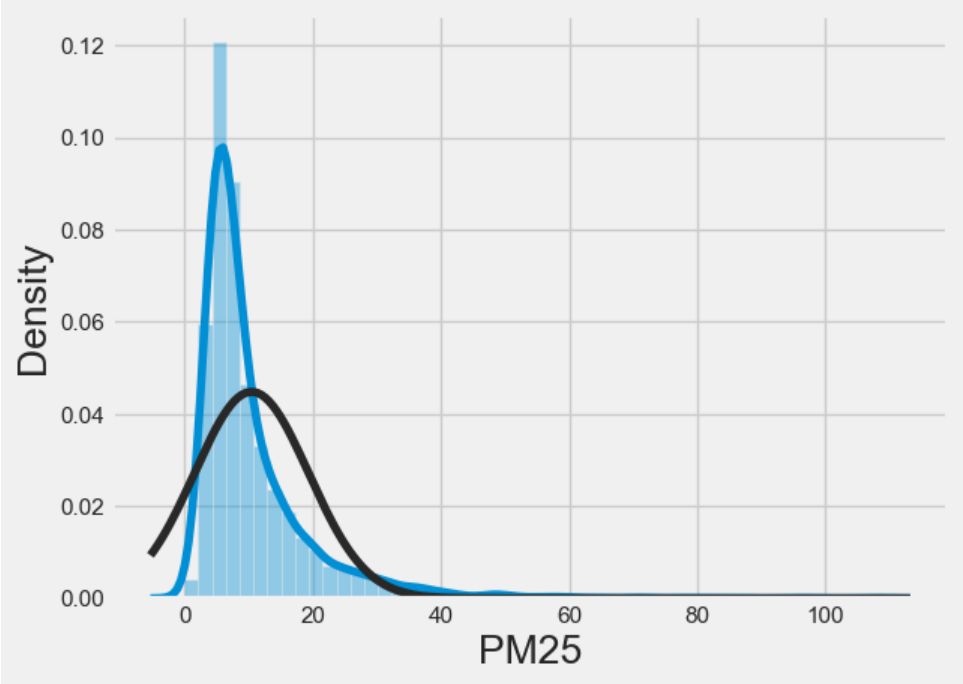

Figure 3.1: Distribution plot of the PM2.5 data

From this, the target variable is shown to deviate from the normal distribution; it is shown to be leptokurtic and shows considerable positive skewness.

The graph indicates the presence of extreme outliers which could affect model performance. To gain further insight a box-plot was created to identify how far the outliers deviated from the standard deviation.

The box-plot is show below in Figure 3.2.

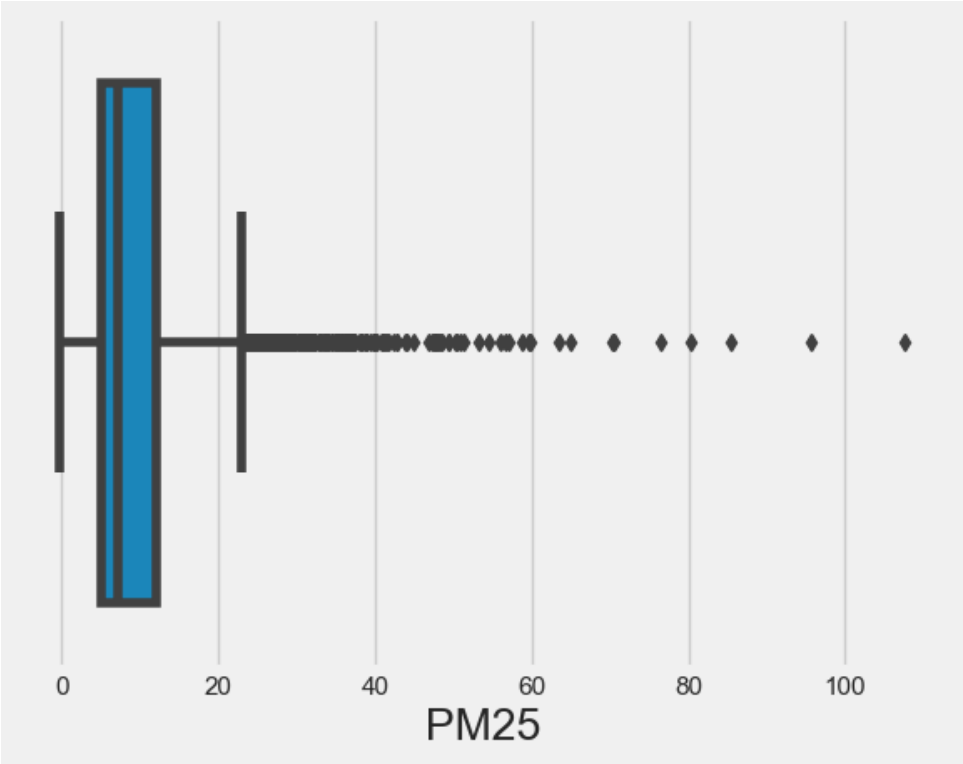

Figure 3.2: Box plot of the PM2.5 data

The box plot shows multiple values greater than two standard deviations. These extreme values will have to be removed as they are not indicative of the data set and will only hinder model performance. Next, the relationships between the target variables and four features were considered. First, to provide a visual reference a scatter plot plotting each feature versus the target was created using the Pandas module.

The scatter plots are displayed below in Figure 3.3.

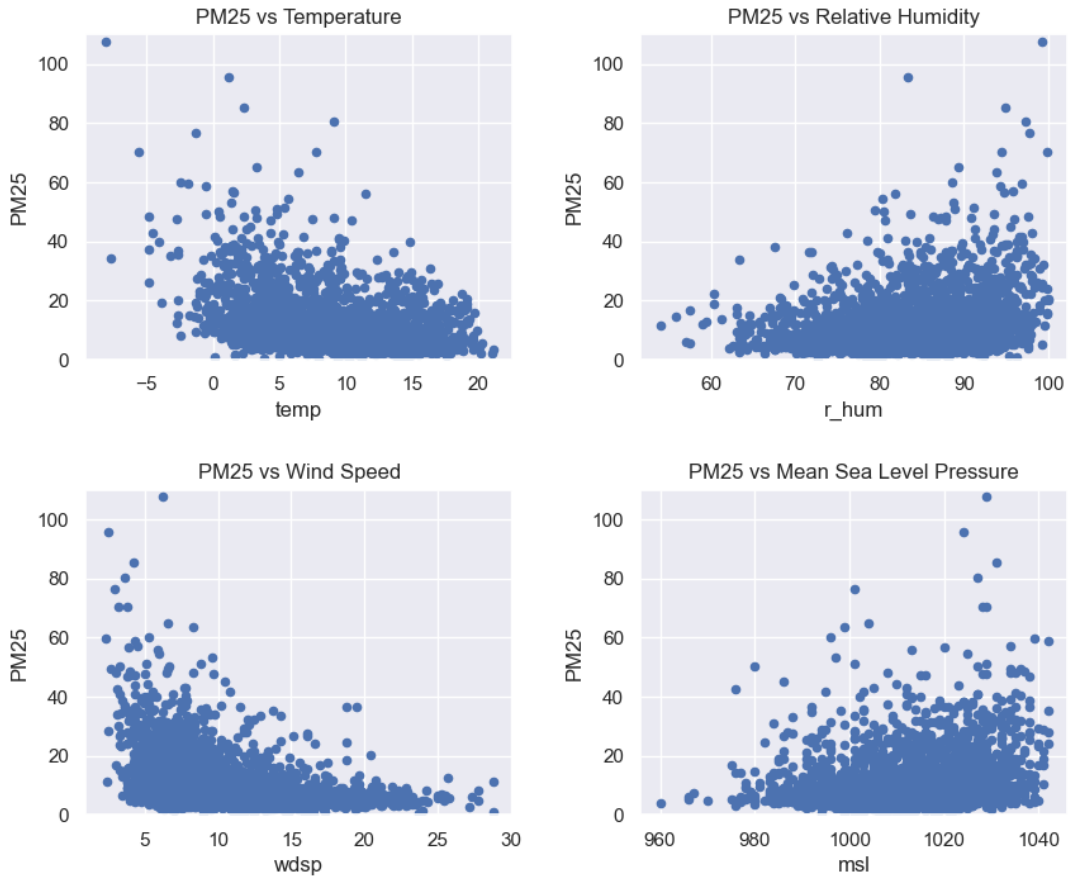

Figure 3.3: Scatter plots of PM2.5 versus each feature

A negative exponential relationship was identified between wind speed and PM2.5. The other variables demonstrated positive linear relationships with PM2.5. To provide more accurate information a correlation heat map was created with Seaborn. This displays the Pearson correlation coefficient between each variable.

The correlation heat map is shown below in Figure 3.4.

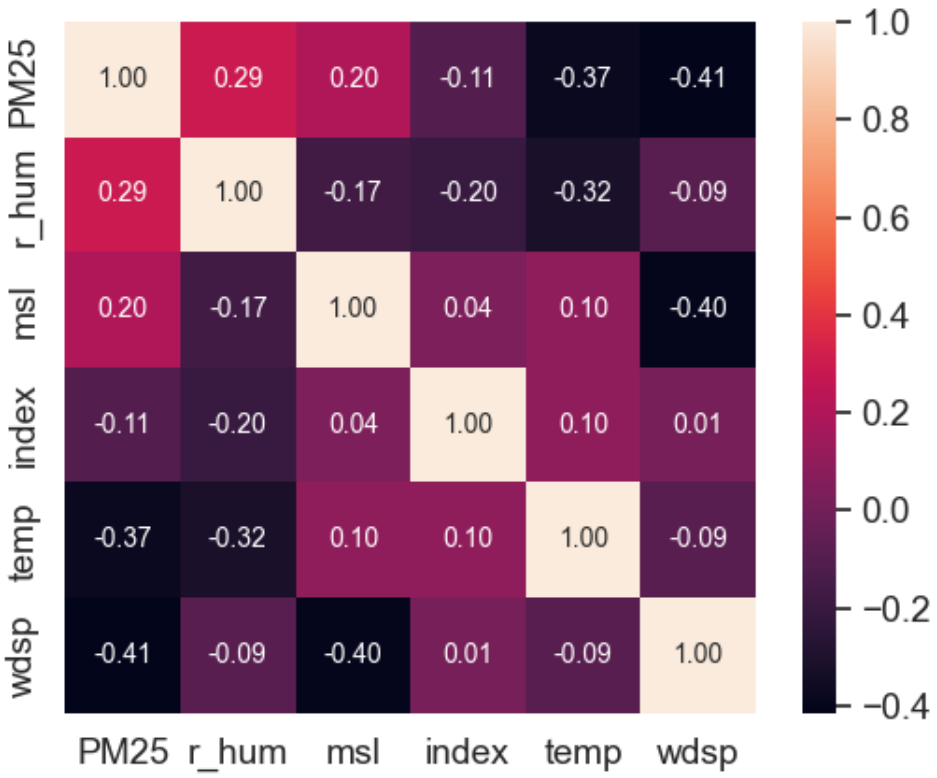

Figure 3.4: Correlation matrix displaying the Pearson correlation coefficient between all variables.

The correlation matrix demonstrates that wind speed and temperature are negatively correlated with PM2.5 with correlations of -0.41 and -0.38 respectively. Mean level sea pressure and relative humidity are weakly positively correlated with correlation values of 0.2 and 0.29, respectively.

None of the features are correlated strongly enough to infer multicollinearity and so were safe to be included. From the values provided we identify wind speed and temperature as strong indicators and are likely to be more effective as features.

3.2 Feature Engineering

Analysis of the distribution plots of each variable revealed that the target variable, wind speed and mean level sea pressure features do not follow the normal line. Univariate feature engineering is a key process for improving the performance of features that have non-normal distributions These transformations can only be performed on positive non-zero values. Therefore, the Box-Cox transformation was only performed on the positive values of the features. The wind speed, relative humidity and mean level sea pressure features were normalised.

The original wind speed and transformed wind speed are displayed below in Figure 3.5 and Figure 3.6.

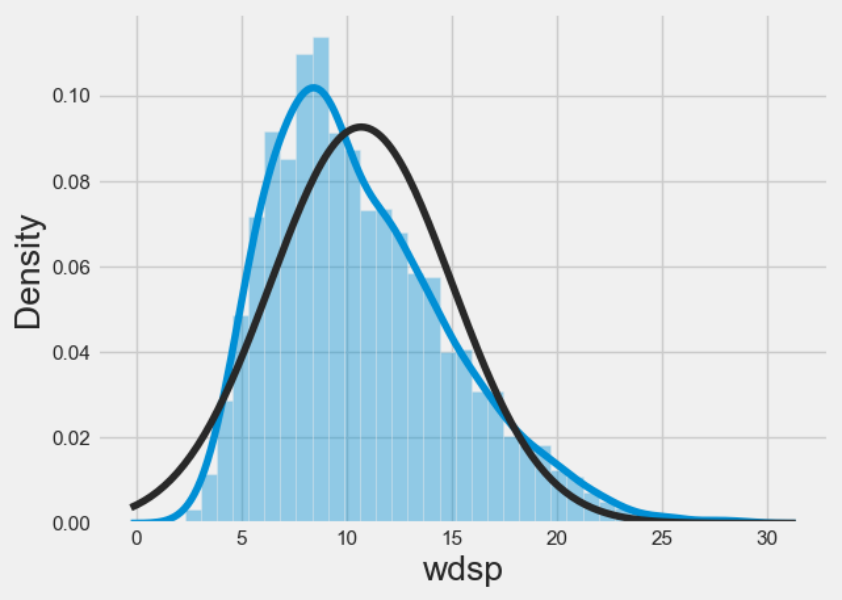

Figure 3.5: Distribution plot of the wind speed feature

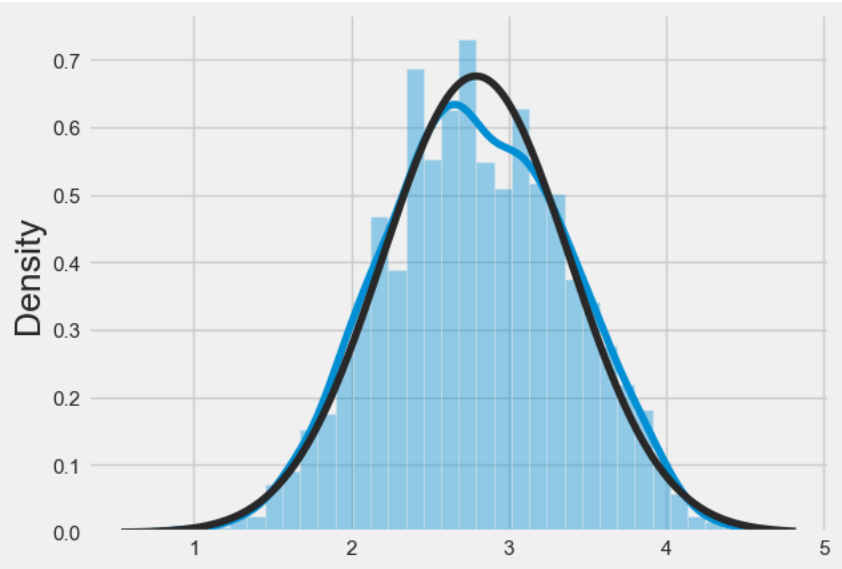

Figure 3.6: Distribution plot of the normalised wind speed feature

As each feature variable has a different data range, they need to be scaled to all be between 0 and 1. This ensures that none of the models are unfairly biased towards a feature due to it having a larger value. Scaling was conducted using the Min Max scaler class from the Scikit-Learn library in python.

The original and scaled features for mean level sea pressure are shown below in Figure 3.7 and 3.8.

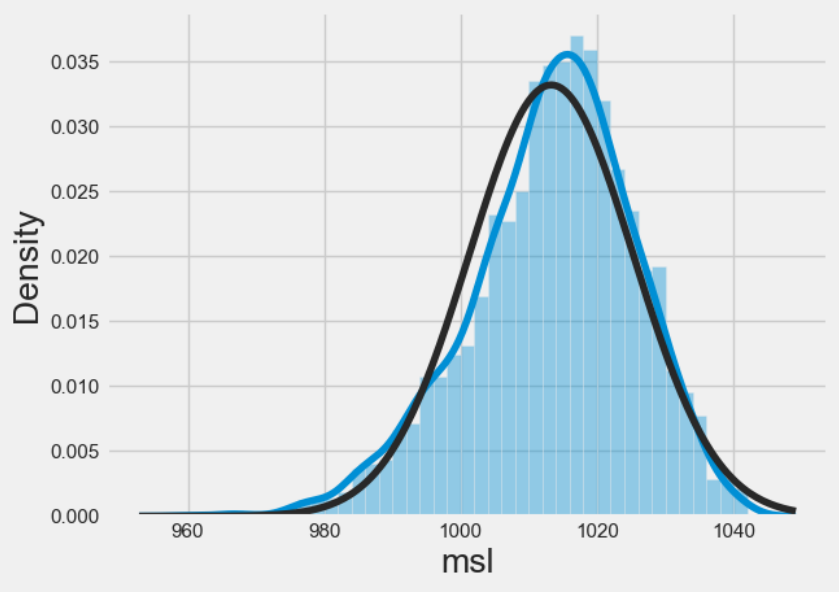

Figure 3.7: The original feature for the mean level sea pressure feature.

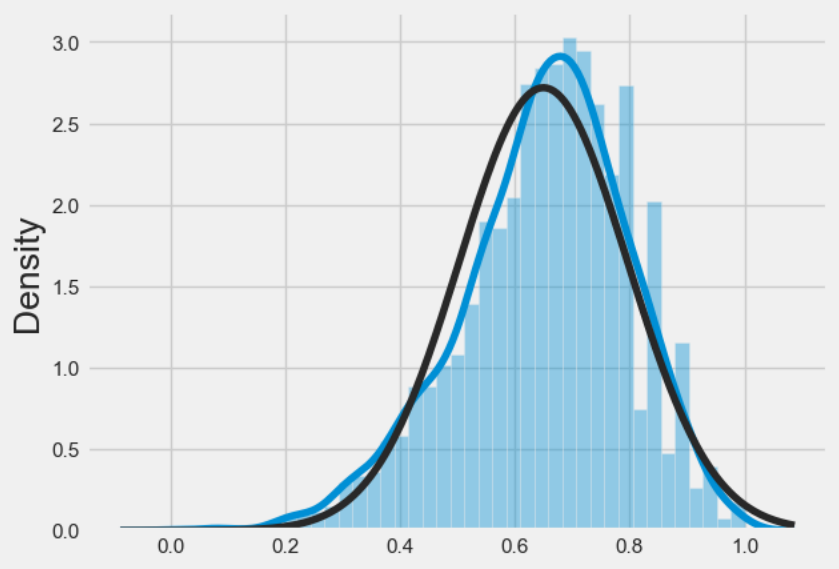

Figure 3.8: The scaled mean level sea pressure feature.

3.3 Modelling Process

The aforementioned models were used to predict the daily average based on the daily meteorological data. There were four basic stages:

(i) data cleaning and preparation

(ii) fitting the data to the proposed model

(iii) Making predictions with the fitted model

(iv) Evaluating the predictions of the model.

3.3.1 Data cleaning and preparation

The data sets of daily PM2.5 data contained missing values. For each model, how the missing data was handled was experimented with to identify which method resulted in improved results. Once missing values were handled, the feature transformations, scaling and normalisation were applied. Each model was fitted with data which had not been transformed and transformed data to identify which resulted in improved results. The data was then broken into training and validation sets. An 80,20 split was used for all models. One year of unseen data was kept aside for a test set to test the robustness of the models.

3.3.2 Fitting the models

Linear Regression:

A simple multi-linear regression was used as the base model experiment. Each feature was scaled and normalised. The model took in four features and mapped to the target variable. There was no additional optimizations made.

Random Forest:

Imputing missing data had no improvements with results and rows with missing values were dropped. To achieve the best performance with a Random Forest model, the hyper parameters must be fine tuned. A grid search was performed to find the optimal hyper parameters. 5-fold cross validation was used to compensate for the small data set when fitting the model and searching for hyper parameters. Cross Validation increased the size of the small data set, as it negates the need for a validation set, conserving the limited data for the training of the model. Multiple grid searches were used to fine optimal hyper parameters. For each parameter, three possible values were chosen to cover a large range. Once the optimal value was found, the range would be narrowed around that value to see if there was a closer optimal value.

Table 3.1 below displays the final parameters chosen by the grid search.

Table 3.1: Optimal hyper-parameters found through grid search

Parameters	**Values**
Maximum Depth	10
Number of Estimators	200
Maximum Features	3
Minimum Samples Leaf	2
Minimum Samples Split	1

XGBoost:

Imputing missing data had no improvements with results and rows with missing values were dropped. To achieve the best performance with a XGBoost model, the hyper parameters must be fine tuned. A grid search was performed to find the optimal hyper parameters. 10-fold cross validation was used to compensate for the small data set when fitting the model and searching for hyper parameters.

Table 3.2 below displays the final parameters chosen by the grid search.

Table 3.2: Parameters for XGBoost chosen by grid search

Parameters	**Values**
Maximum Depth	3
Number of Estimators	1000
Learning Rate	0.01
colsample$_bytree$	0.0.7
Minimum Child Weight	1

Neural Network:

Missing values were dropped. First, a basic architecture was attempted with no feature transformations, removal of outliers and no optimization of hyper parameters. This allowed the improvements of other adjustments to be tracked. The number of epochs was set at 500, but an early stopping function was applied which would stop training and revert to the previous best weights if no improvement was detected after a certain range. The architecture was slowly scaled up unit over fitting occurred. Over fitting was then managed using two layers of dropout. An eight layer deep neural net was used with the ADAM optimizer for the final architecture.The loss chosen for the model was the Mean Squared Error. L1 regularisation was used to help combat over fitting. Two levels of dropout were also used to combat the small size of the data set. The final batch size chose was 32. Four batch sizes were tried for each model configuration identify which gave the most accurate results.The final parameters are displayed below in Table 3.3.

Table 3.3: Final hyper parameters for the Neural Net model

Parameters	**Values**
Learning Rate	0.001
Batch Size	32
Epochs	48

The model loss is displayed in Figure 3.9. The graph shows spikes near the end of training indicative of slight over fitting however additional levels of dropout or batch normalisation did not result in improved results and this slight over fitting could likely only be solved through the acquisition of more training data.

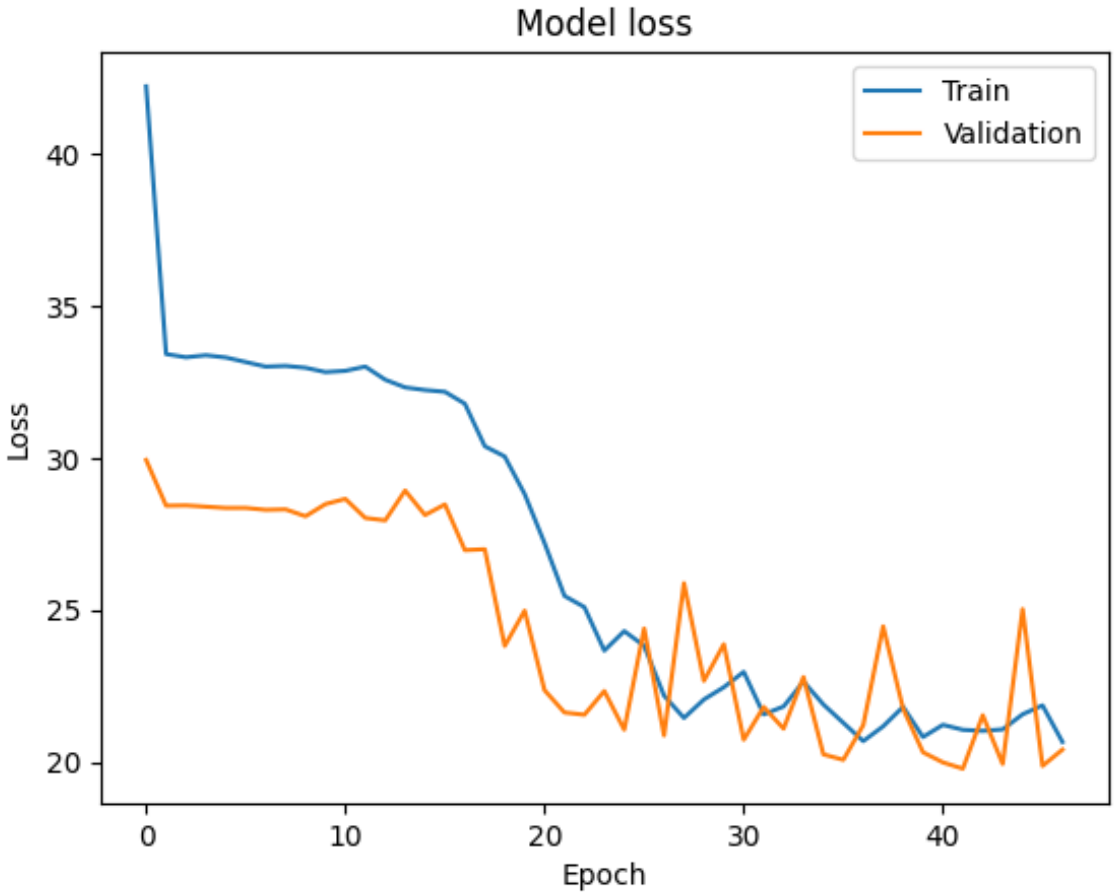

Figure 3.9: Training loss and validation loss for the Neural Net model

GRU:

To make use of the GRU architecture, the PM2.5 data had to be converted to time series data. The time series is shown below in Figure 3.10.

There are considerable in the time series. To use a recurrent neural net, the missing values in the time series were imputed using linear interpolation. The time series was then split into training and validation splits with an 80/20 split. The completed time series is show below in Figure 3.11.

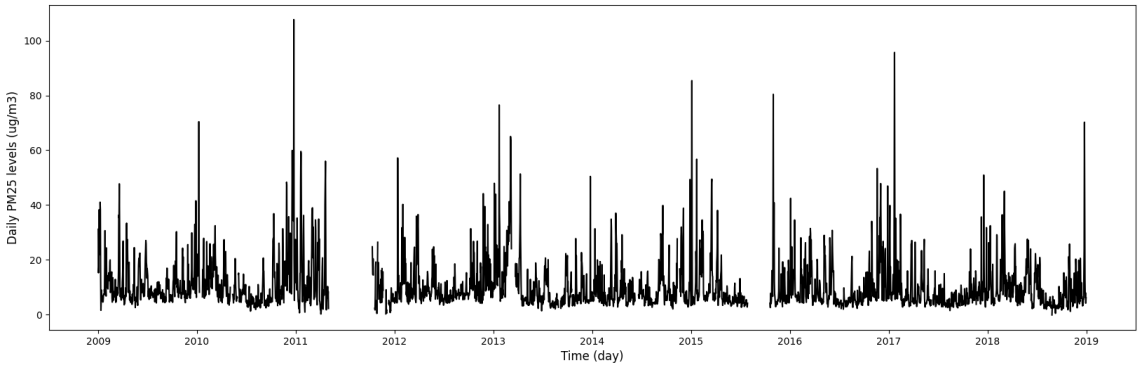

Figure 3.10: PM2.5 converted to a time series graph

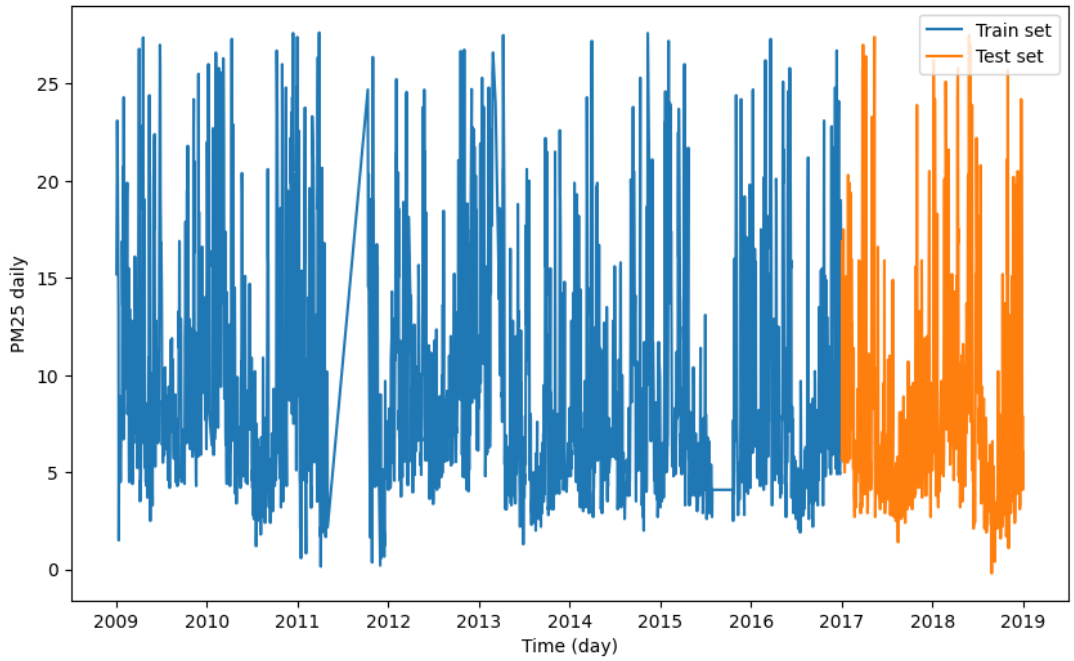

Figure 3.11: Time series graph with gaps filled in through linear interpolation

Grid Search could not be used to optimize parameters due the extended training time required by the model. Instead, a smaller range of parameters were tested. The training loss is displayed below in Figure 3.12

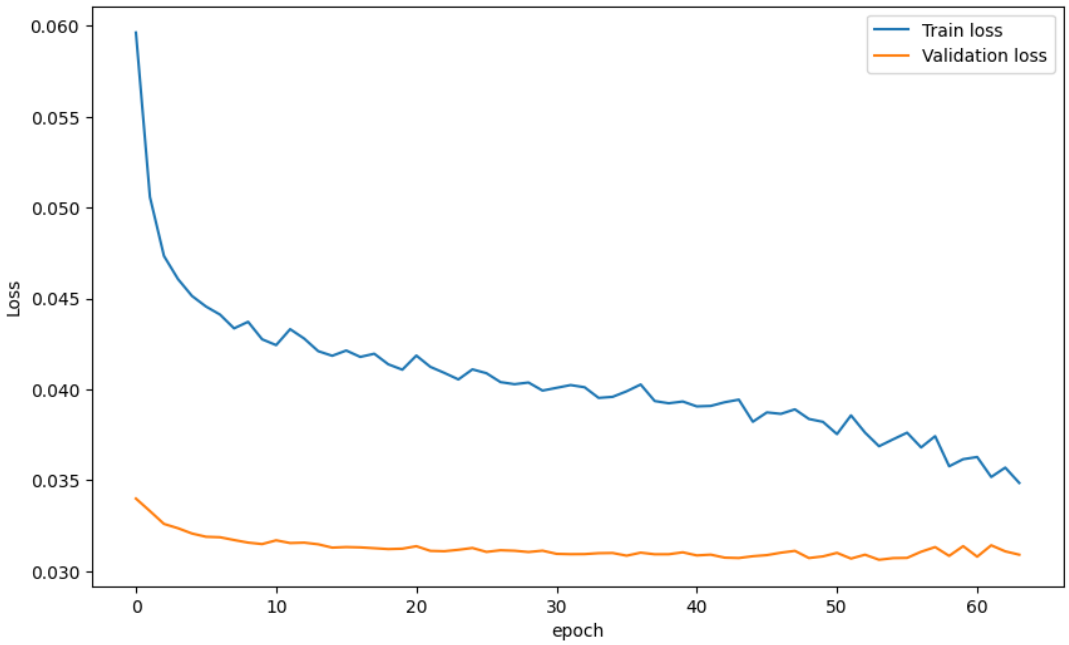

Figure 3.12: Training and validation loss for the GRU model

3.3.3 Making Predictions

Predictions were generated using the model from unseen data not used in the training process. The test set was one tenth of the training data and represented a full year after the training data time. These predictions were than graphed against the true values to provide a visual representation of model performance.

3.3.4 Evaluating Model Performance

The RMSE, MSE and MAE were calculated for each model either using the average 5-fold cross validation score or by taking the results for those metrics on the test set.

4 Results

4.0.1 Model Results

Results for each of the metrics is displayed below in the tables. The values displayed in each table represent either the average of cross validation, or the values for the test set, depending on the criteria of the respective model. The predictions for the final models were graphed versus the true values of the data set to give a visual of the accuracy of predictions. Lastly, feature importance was graphed for the Random Forest and XGBoost models to identify which weather variables were most critical for model performance.

Linear Regression Results:

The metrics for the linear regression are displayed below in Table 4.1.

Table 4.1: Metrics for the Linear Regression Model

Parameters	**Values**
Mean Squared Error	52.
Mean Absolute Error	1000
R2 Score	0.3
Root Mean Squared Error	7.2

The model displays a prominent level of errors. This was expected as the base model does not have the capabilities to map data that is this complex. There were also few attempts outside of basic feature engineering to optimize the base model. This provides an example of predictions that is better than random, while also showing the difficulty in mapping complex data.

The predictions versus true values are displayed in Figure 4.1.

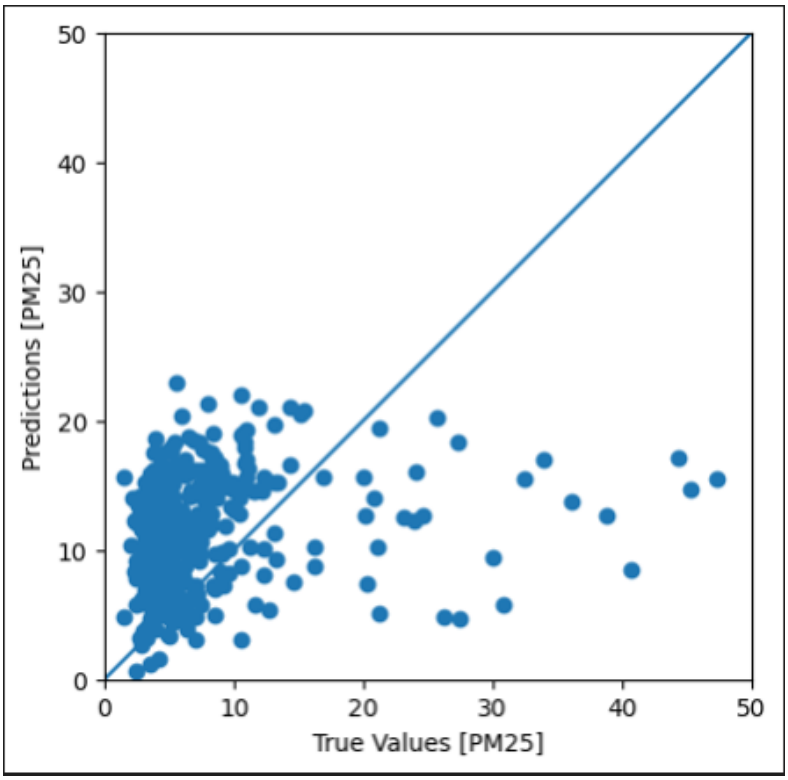

Figure 4.1: Predictions from the Linear Regression model graphed against the actual PM2.5 values

Random Forest results:

The metrics for the optimized random forest model are displayed below in Table 4.2.

Table 4.2: Metrics for the optimized Random Forest Model

Parameters	**Values**
Mean Squared Error	24.7.
Mean Absolute Error	3.62
Root Mean Squared Error	4.96

Before optimization, the random forest results were like the base model. After fine tuning the hyper parameters however the MSE decreased by 47% to 24.7. The RMSE is 2 lower, representing a prediction 2(ug/m^3) closer for the average prediction.

The predictions versus true values are displayed in Figure 4.2.

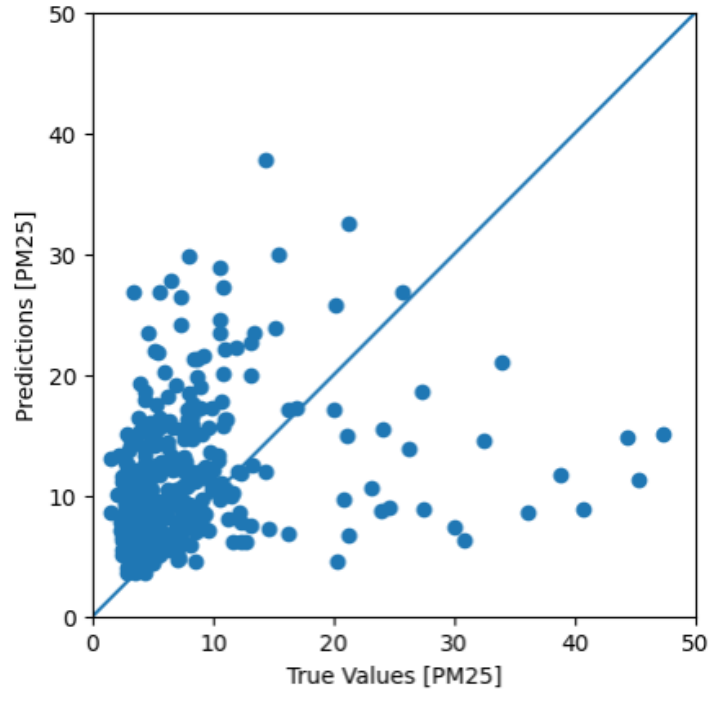

Figure 4.2: Predictions from the optimized Random Forest model graphed against the actual PM2.5 values

The model does not fit well to the data. It tends to over predict and choose extreme values. This is due to the small data set and outliers.

Neural Net results:

The metrics for the optimized Neural Net are displayed below in Table 4.3.

Table 4.3: Metrics for the optimized Neural Net model

Parameters	**Values**
Mean Squared Error	25.
Mean Absolute Error	4.2
Root Mean Squared Error	5

The model performed better on the validation set where it's MSE dropped to 19, however it did not carry over fully to the unseen test data. The true values versus predictions made by the Neural Net are displayed in Figure 4.3.

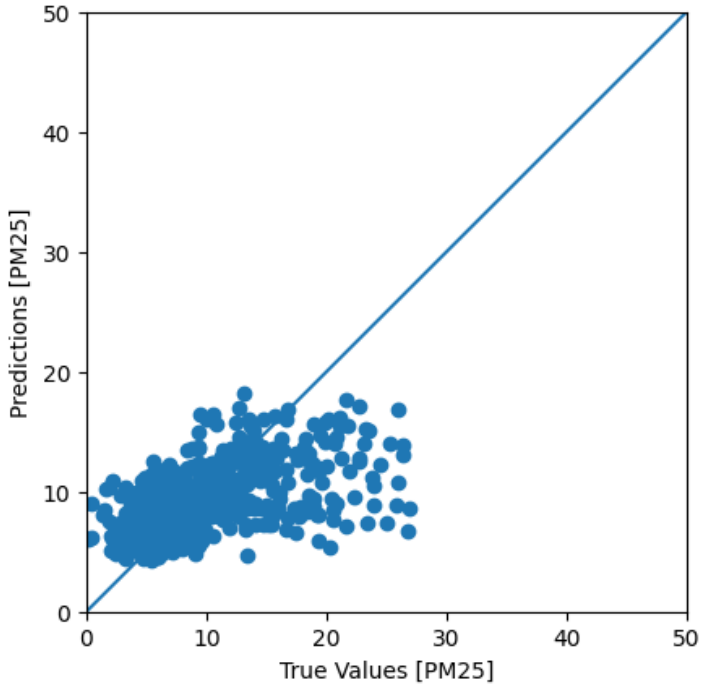

Figure 4.3: Predictions from the optimized Neural Net model graphed against the actual PM2.5 values

XGBoost results:

The metrics for the XGBoost model are displayed below in Table 4.4.

Table 4.4: Metrics for the optimized XGBoost model

Parameters	**Values**
Mean Squared Error	18.
Mean Absolute Error	3.2
Root Mean Squared Error	4.2

The true values versus predictions made by the XGBoost are displayed in Figure 4.4.

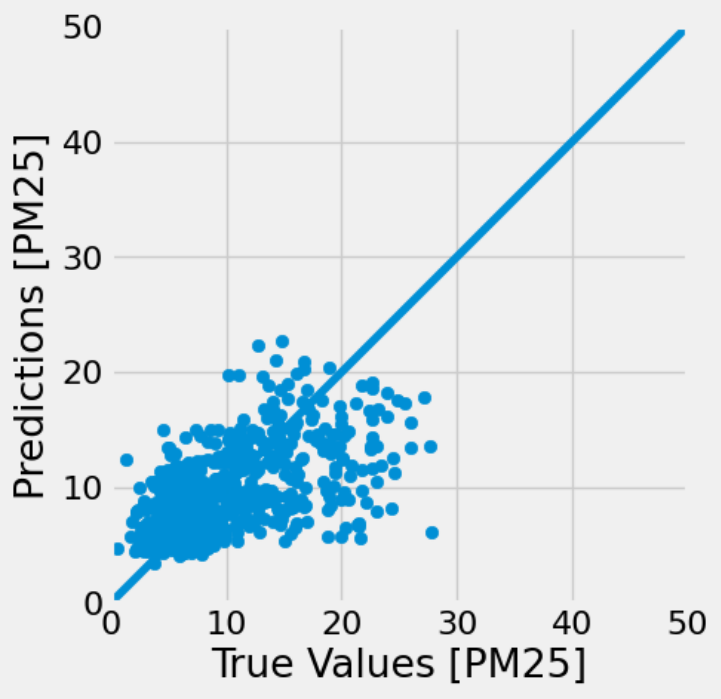

Figure 4.4: Predictions from the optimized XGBoost model graphed against the actual PM2.5 values

GRU results:

The metrics for the GRU model are displayed below in Table 4.5.

Table 4.5: Metrics for the optimized GRU model

Parameters	**Values**
Mean Squared Error	25
Mean Absolute Error	3.7
Root Mean Squared Error	5

The GRU performed similarly to the Neural Net model, with an the same value for RMSE and MSE. These models are likely limited by the same factors such as limited data.

The predictions versus true values are displayed in Figure 4.5.

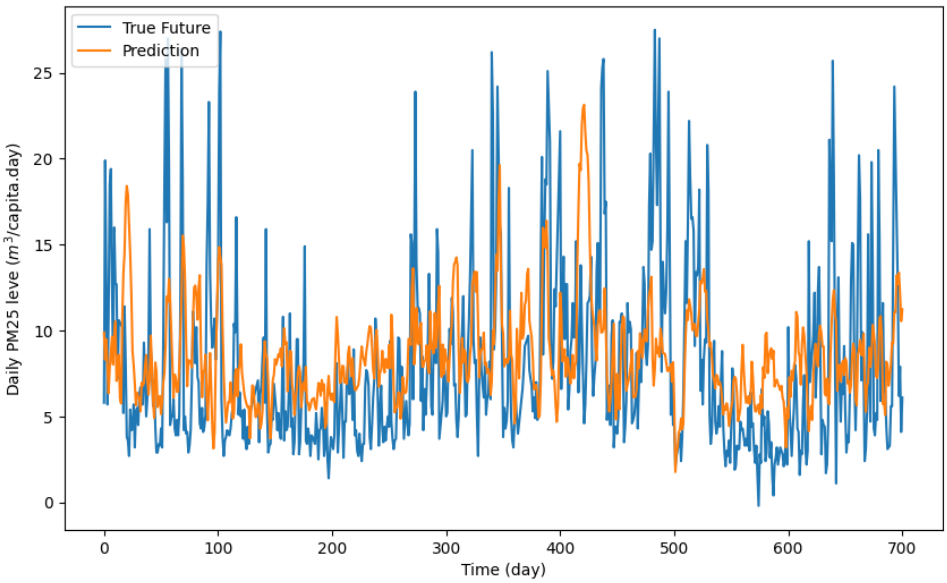

Figure 4.5: Predictions from the optimized GRU model graphed against the actual PM2.5 values

4.0.2 Feature Importance

Unnecessary features do not contribute to model performance. This is because they add to the overall complexity and hinder the model's performance. A preliminary analysis was conducted using the correlation matrix, however, to ensure the elimination of unnecessary features and to gauge the predictive power of each feature the built-in feature importance feature was used for both the Random Forest Model and the XGBoost model.

The results are displayed below in Figure 4.6.

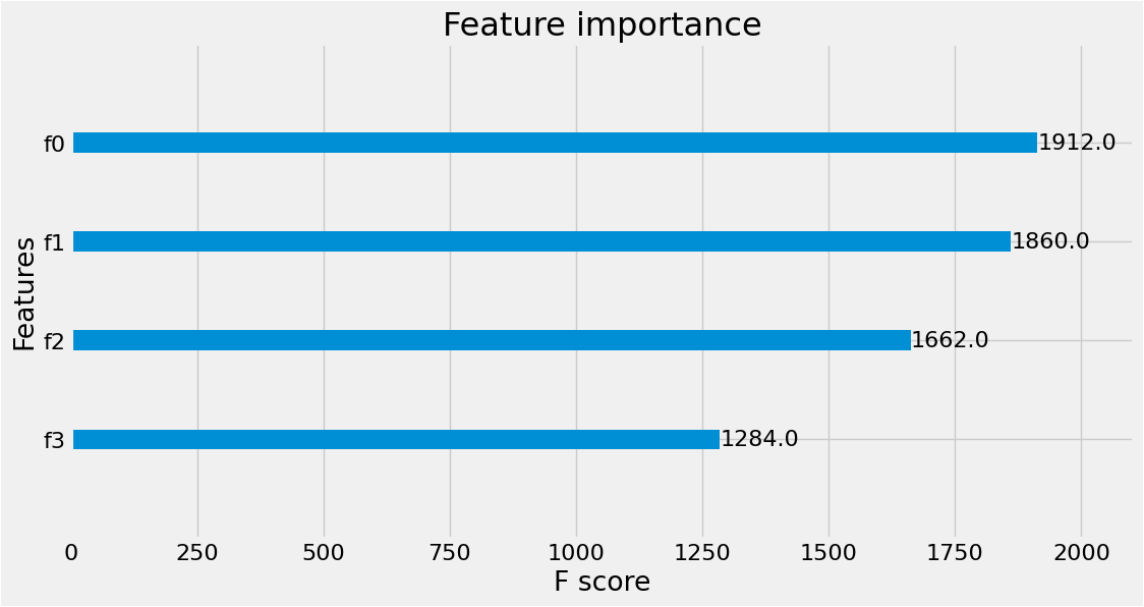

Figure 4.6: A bar chart of the F-score of each feature

F0 = wind speed, f1=temperature, f2=relative humidity, f3=mean level sea pressure. Relative humidity and wind speed are identified as the most accurate predictors of PM2.5 among the chosen features. With an F-Score of 1900 and 1894 respectively, they out predict temperature and mean level see pressure by 250 and 600. Mean level sea pressure is weaker than the other features, with an f score two-thirds of relative humidity. F Score: The f-regression score is a statistical measure that calculates the significance of each feature in a regression model. It works by measuring the variation in the target variable that is explained by each feature, while controlling for the effect of the other features. Specifically, the f-regression score calculates the ratio

of the variance in the target variable that is explained by the model, to the unexplained variance. This ratio is then scaled by the degrees of freedom to obtain an f-statistic. The resulting f-regression score for each feature indicates its importance in explaining the variation in the target variable. Features with higher f-regression scores are considered more important, as they explain more of the variation in the target variable than features with lower scores.

4.0.3 Model Performance outside target area

The performance of the top performing model was tested for a PM2.5 station in Old Station Road Cork.

The results for the XGBoost model in Cork are displayed below in Table 9.

Table 4.6: Metrics for the XGB models predictions for Cork

Parameters	**Values**
Mean Squared Error	18.
Mean Absolute Error	3.2
Root Mean Squared Error	4.2

The true value versus the predictions for Cork are displayed below in Figure 4.7.

The forecasting for Cork is displayed in Figure 4.8 below.

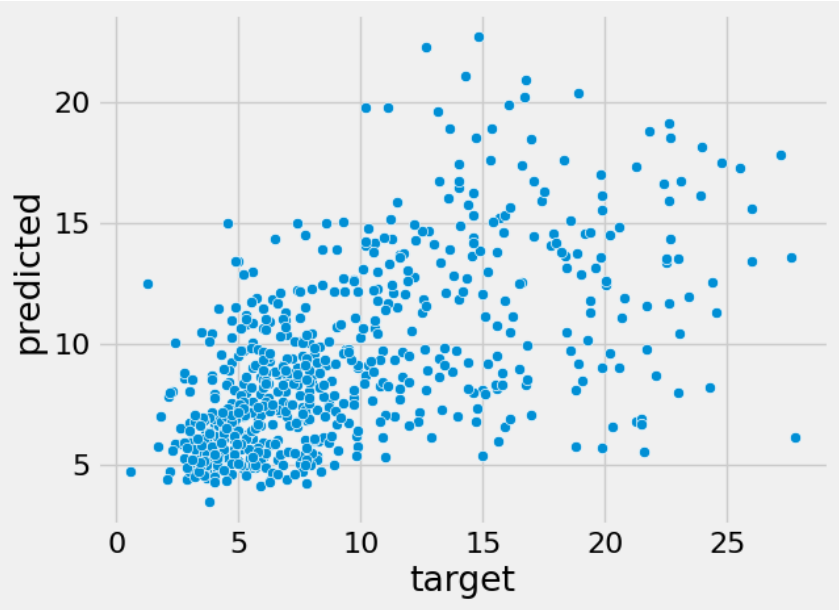

Figure 4.7: Predictions for Cork form the XGBoost model

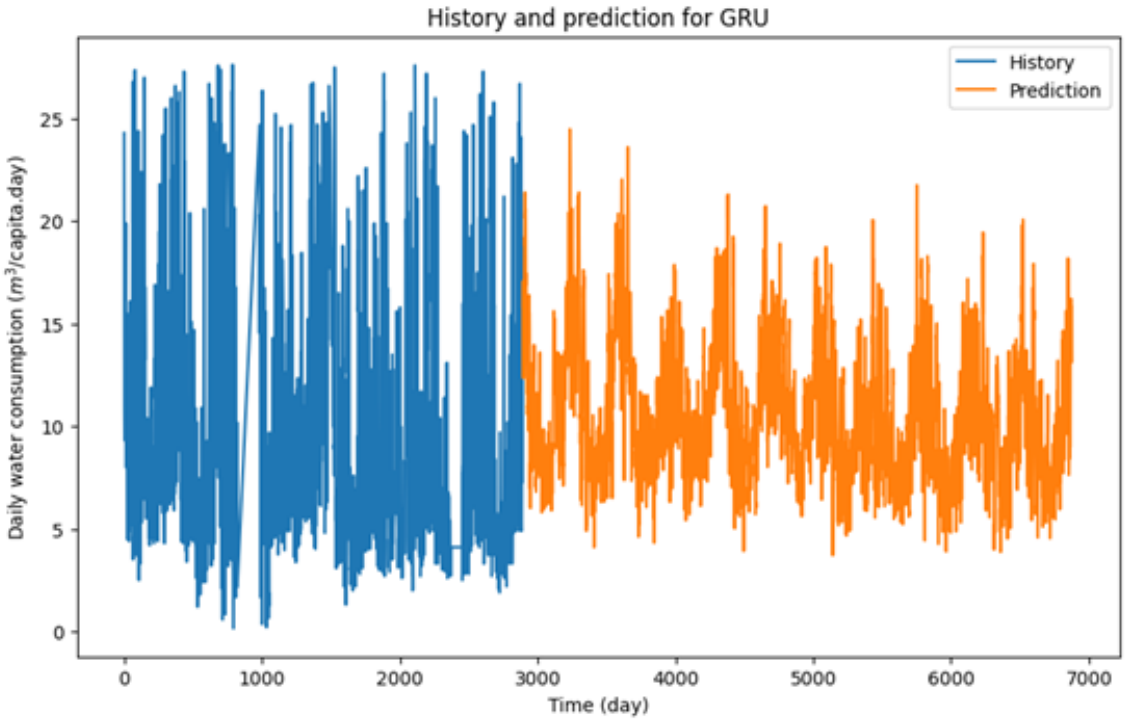

Figure 4.8: PM2.5 forecasting for Cork from the GRU model

5 Discussion

5.1 Top Model Performance

The top performing model for the data set was, XGBoost reducing the error of the base model by 65%. XGBoost outperforming the other metrics, is consistent with the literature, as it had outperformed other models with small data sets in other studies (24). XGBoost can perform well with small or large data sets and fits complex data well. However, despite XGBoost having the lowest error, all the models performed similarly once optimized. Although the MSE was 7 less than the other models on average, the RMSE for the Neural Net and the GRU model were 5, which was only 0.8 more than the XGBoost model. As the RMSE presents the error in the units of the target variable, this means the top model's predictions are only 0.8 (ug/m^3) closer to the correct value then the other models.

5.2 Performance of the model outside the study area

One of the limitations of machine learning algorithms is that they are a "black box". It is impossible to find out exactly how the model came to its predictions. To test the robustness of the model and to gain and understanding how much the specific characteristics of the city are affected its decisions, the model was tested on a data set in Cork City. The meteorological data was provided from a Met Eireann Station in Cork. The XGBoost model still decreased of the base model by 21, despite the model being developed for a different city. Although the RMSE increased by 1.3

ug/m^3,when making predictions in Cork, it was still at a level where it could make predictions within a useful range. This indicates that meteorological variables within a city radius could be enough to reliably forecast PM2.5 if enough data and a powerful enough model were acquired.

5.3 Applications of Model Applications of Model

Although the models still had levels of RMSE between 5.5 ug/m^3 and 4.2 ug/m^3, the models are still accurate enough to detect levels of PM2.5 that are harmful to public health. This can be seen in the forecast data in Figure 4.5 and Figure 4.8. The model has captured the trend of the data and can consistently predict when PM2.5 levels will rise above the daily recommended threshold of 15 ug/m^3. Even if it cannot accurately capture the extreme peaks, it can still provide warnings to take precautionary methods such as wearing a mask or staying inside when higher than normal are detected. The forecasting ability of the model also allows entities to plan around low levels. Open air events can be scheduled on days with lower work of exposure to air pollutants. However, if organisations such as the EPA, wished to use these models to create policies based on reducing PM2.5, or to predict PM2.5 levels in specific high priority urban areas such as hospitals, more powerful models trained on larger data sets, with smaller margins of error would have to be created. There is evidence of these types of models in China where they make use of geospatial data and satellite data to give more precise locations for predicted PM2.5 levels(25).

5.4 Limitations

The data set, which consisted only of daily values, values was the most limiting factor of the data set. The size of the data set limited the effectiveness of the Neural Net model, as Neural Net model's performance scales with data size. Hourly data would have provided a larger range of PM2.5, which could have allowed the models to map more complex patterns within the data. Neural Nets and Recurrent Neural

Nets require powerful graphics processing units (GPUs) to train and make predictions. This limited the size of the Neural Net and GRU models used for this project. The GRU model for instance, would have taken a full day of training to optimize with grid search. The computer used to run this would often crash during that process which limited the effectiveness of the GRU model. The stations considered in this were limited in terms of location. A wider spread of stations across the Greater Dublin area, or Ireland, would have opened the possibility of using Convolutional Neural Nets which could consider the relative proximity of each PM2.5 station and each meteorological data and their outputs to generate predictions. (26) Another aspect that limited the results of the project was the three-month time in which the study was carried out. Due to this time restriction, it was not feasible to collect live PM2.5 data, so the study was reliant on historical data sets. In a longer study, PM2.5 stations that have APIs could have been used to gather data. These APIs provide hourly data, which given the time restrictions of the project, would not have gathered more data than the historical data used. However, a longer study could have gathered hourly data for 2 years which would create a data set five times larger than the one used in this project.

6 | Conclusion

The meteorological data available was able to accurately predict PM2.5 values within the city to within 4(ug/m^3) using the XGBoost model. Model performance did not differ greatly between the top performing models, Neural Nets, Gated Recurrent Units and XGBoost, as there was a difference of less than 1 RMSE. Although XGBoost was the highest performing model, the consideration of time series data by the GRU model allowed for superior forecasting of PM2.5 levels. More advanced models could be considered in future; however, this is dependent of the monitoring data available. Hourly data would allow for more categorical data such as wind direction, while increasing the size of the available data set. More monitoring stations could allow models to include distance and longitudinal coordinates as features and opens other modelling options such as convolutional neural networks.

Bibliography

[1] E. E. McDuffie, R. V. Martin, J. V. Spadaro, R. Burnett, S. J. Smith, P. O'Rourke, M. S. Hammer, A. van Donkelaar, L. Bindle, V. Shah, L. Jaeglé, G. Luo, F. Yu, J. A. Adeniran, J. Lin, and M. Brauer. Source sector and fuel contributions to ambient pm2.5 and attributable mortality across multiple spatial scales. *Nature Communications*, 12:3594, 2021. doi: 10.1038/s41467-021-23841-w.

[2] World Health Organization. Global burden of disease 2019: Disease and injury incidence, prevalence, and years lived with disability for 354 diseases and injuries, 1990–2019: A systematic analysis for the global burden of disease study 2019. *Lancet*, 396(10258):1204–1222, 2020.

[3] World Health Organization. Who global air quality guidelines. particulate matter (pm2.5 and pm10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. Geneva: World Health Organization; Licence: CCBY-NC-SA3.0IGO., 2021.

[4] S. Yang, J. Sui, T. Liu, W. Wu, S. Xu, L. Yin, Y. Pu, X. Zhang, Y. Zhang, B. Shen, and G. Liang. Trends on pm2.5 research, 1997–2016: a bibliometric study. *Environmental Science and Pollution Research*, 25(13):12284–12298, 2018.

[5] C. Aves and M. Williams. Urban air quality modelling of dublin: Final report. Prepared for The Environmental Protection Agency, Ireland. Retrieved from FM1206_CERC_EPADublin_R5_08Jul19., 2019. Report No. FM1206/R5/19.

[6] P. Goodman, B. Jahanshahi, D. McVicar, and N. Rowland. Estimating local

all-cause and circulatory mortality burdens associated with fine particulate matter pollution in northern ireland and the republic of ireland. Report commissioned by The Irish Heart Foundation on behalf of The Irish Heart Foundation and The British Heart Foundation, January 2023.

[7] B. Cheng, Y. Ma, F. Feng, Y. Zhang, J. Shen, H. Wang, Y. Guo, and Y. Cheng. Influence of weather and air pollution on concentration change of pm2.5 using a generalized additive model and gradient boosting machine. *Atmospheric Pollution Research*, 12(9):210225, 2021. doi: 10.1016/j.apr.2021.210225.

[8] I. G. McKendry. Evaluation of artificial neural networks for fine particulate pollution (pm10 and pm2.5) forecasting. *Atmospheric Environment*, 45(6): 1096–1101, 2011. doi: 10.1016/j.atmosenv.2010.11.026.

[9] B. Pan. Application of xgboost algorithm in hourly pm2.5 concentration prediction. *IOP Conference Series: Earth and Environmental Science*, 113:012127, 2018. doi: 10.1088/1755-1315/113/1/012127.

[10] M. Niu, Y. Zhang, and Z. Ren. Deep learning-based pm2.5 long time-series prediction by fusing multisource data—a case study of beijing. *Atmosphere*, 14 (2):340, 2023. doi: 10.3390/atmos14020340.

[11] Y. Yuan, S. Liu, R. Castro, and X. Pan. Pm2.5 monitoring and mitigation in the cities of china. *Environmental Science & Technology*, 46(7):3627–3628, 2012. doi: 10.1021/es300984j.

[12] T. Verdonck, B. Baesens, M. Óskarsdóttir, and S. vanden Broucke. Special issue on feature engineering editorial. *Machine Learning*, 2021.

[13] GA Roth, D Abate, KH Abate, SM Abay, C Abbafati, N Abbasi, H Abbastabar, F Abd-Allah, J Abdela, A Abdelalim, et al. Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: A systematic analysis for the global burden of disease study 2017. *Lancet*, 392:1736–1788, 2018. doi: 10.1016/S0140-6736(18)32203-7.

[14] SA Lee, K Flynn, G Delaunay, MM Kennelly, and MJ Turner. Air pollution levels outside the capital's maternity hospitals. *Ir Med J*, 115(8):650, 2022.

[15] The Pensions Commission. Population and labour force projections technical sub-committee – working paper 1. July 2021.

[16] M.-J. Kim, Y.-S. Chang, and S.-M. Kim. Impact of income, density, and population size on pm2.5 pollutions: A scaling analysis of 254 large cities in six developed countries. *International Journal of Environmental Research and Public Health*, 18(17):9019, 2021. doi: 10.3390/ijerph18179019.

[17] M. Castelli, F. M. Clemente, A. Popovič, S. Silva, and L. Vanneschi. A machine learning approach to predict air quality in california. *Journal of Environmental Informatics*, 36(2):117–128, 2020. doi: 10.3808/jei.201900437.

[18] T. Verdonck, B. Baesens, M. Óskarsdóttir, et al. Special issue on feature engineering editorial. *Machine Learning*, 2021. doi: 10.1007/s10994-021-06042-2.

[19] W. Yuchi, E. Gombojav, B. Boldbaatar, J. Galsuren, S. Enkhmaa, B. Beejin, G. Naidan, C. Ochir, B. Legtseg, T. Byambaa, P. Barn, S.B. Henderson, C.R. Janes, B.P. Lanphear, L.C. McCandless, T.K. Takaro, S.A. Venners, G.M. Webster, and R.W. Allen. Evaluation of random forest regression and multiple linear regression for predicting indoor fine particulate matter concentrations in a highly polluted city. *Environmental Pollution*, 245:746–753, 2019. doi: 10.1016/j.envpol.2018.11.034.

[20] Raúl Rojas. The backpropagation algorithm. In *Neural Networks*, pages 433–494. Springer, 1996. doi: 10.1007/978-3-540-60505-8_10.

[21] A. Larasati, M. Hajji, and A. Dwiastuti. The relationship between data skewness and accuracy of artificial neural network predictive model. *IOP Conference Series: Materials Science and Engineering*, 523:012070, 2019. doi: 10.1088/1757-899X/523/1/012070.

[22] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016. doi: 10.1145/2939672.2939785.

[23] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modelling. `https://doi.org/10.48550/arXiv.1412.3555`, 2014.

[24] Boyan Nenchev, Qingfeng Tao, Zhili Dong, Chinnapat Panwisawas, Hongying Li, Bin Tao, and Han Dong. Evaluating data-driven algorithms for predicting mechanical properties with small datasets: A case study on gear steel hardenability. *International Journal of Minerals, Metallurgy and Materials*, 29(7): 836–847, 2022. doi: 10.1007/s12613-022-2357-6.

[25] Yanpeng Qi, Qingquan Li, Hamidreza Karimian, and Deshun Liu. A hybrid model for spatiotemporal forecasting of pm2.5 based on graph convolutional neural network and long short-term memory. *Science of the Total Environment*, 660:941–951, 2019. doi: 10.1016/j.scitotenv.2019.01.333.

[26] Youjin Park, Boyeong Kwon, Jaeseok Heo, Xiaoming Hu, Yu Liu, and Taesup Moon. Estimating pm2.5 concentration of the conterminous united states via interpretable convolutional neural networks. *Journal of Cleaner Production*, 241: 118361, 2019. doi: 10.1016/j.jclepro.2019.118361.