ASSIGNMENT TWO

PAPER NAME: Data Mining and Machine Learning

PAPER CODE: COMP809

TOTAL MARKS: 100

Students' Name: Vinnie Limbrick and Yesha Kaniyawala

Students' IDs: 0831133 and 23213969

- Due date: 09 Jun 2024 midnight NZ time.
- Late penalty: maximum late submission time is 24 hours after the due date. In this case, a **5% late penalty** will be applied.
- Submit the actual code (no screenshot) separately with appropriate comments for each task.

Note: This assignment should be complemented by a group of two students and both students MUST contribute in each part.

Submission: a soft copy needs to be submitted through the canvas assessment link.

INSTRUCTIONS:

- **The following actions may be deemed to constitute a breach of the General Academic Regulations Part 7: Academic Discipline,**
  - Communicating with or collaborating with another person regarding the Assignment
  - Copying from any other student work for your Assignment

    - Copying from any third-party websites unless it is an open book Assignment
    - Uses any other unfair means
- **Please email DCT.EXAM@AUT.AC.NZ if you have any technical issues with your submission on Canvas immediately**
- **Attach your code for all the datasets in**.

PM2.5 Prediction Using MLP Regression and LSTM models

Vinnie Limbrick
0831133

Yesha Jayeshkumar Kaniyawala
23213969

Table of Contents

***Abstract*—The Goal of this research is to predict PM2.5 concentrations from environmental data using machine learning algorithms, namely Multi-Layer Perceptron Regression (MLP) and Long Short Term Memory (LTSM) algorithms. PM2.5 is an important environmental metric associated with health outcomes and implications. This paper used data collected through the Auckland Council Environmental data Portal, which was then preprocessed to ensure data integrity and inform feature selection for creating the machine learning models. The model demonstrated some effectiveness at predicting PM2.5 from environmental metrics however further research is needed to create more effective models.**

Keywords—component, formatting, style, styling, insert (*key words****)***

I. Introduction

Air quality is an important environmental issue, with particulate matter such as PM10 and PM2.5 being closely linked to a wide range of health problems, from minor issues to serious diseases such as cancer and heart disease. Accurate and timely monitoring of PM2.5 concentrations is essential for public health and environmental policy-making. Traditionally monitoring busy of PM2.5 levels relies on physical sampling and statistical models, which often lack the ability to handle large datasets to predict future trends effectively.

Predictions of Particulate Matter

II. Data Preprocessing

This report details using data obtained from the Auckland Council Environmental Data Portal to build the model. The initial data pulled directly from the website was for the dates from the beginning of January 2019 to the end of December 2023, covering a 5 year period. The original dataset included four pollution variables, NO, NO_2, NO_x, and SO_2. It also included wind direction, wind speed, air temperature, relative humidity and the air quality index. The downloaded database contains hourly records where the pollutant variables are in $\mu g/m^3$, wind speed units are ms^{-1}, wind direction are in degrees (bearing with 0° being direct north), air temperature are in degrees Celsius, and relative humidity are in percentage of air saturation. The data was downloaded into a .csv file to be read into a pandas dataframe using the Python package Pandas.

The top rows of the dataframe were first viewed to give an idea of what the data looked like. The top four rows were eliminated as they were not relevant to the creation of the model, however the rows gave details related to the data such as feature names and with each feature's units of measurements. The columns were named according to the top rows.

The data showed 50,117 total entries, with the time data being present for all entries. Other features contained a variety of missing data, as illustrated in Figure 1.

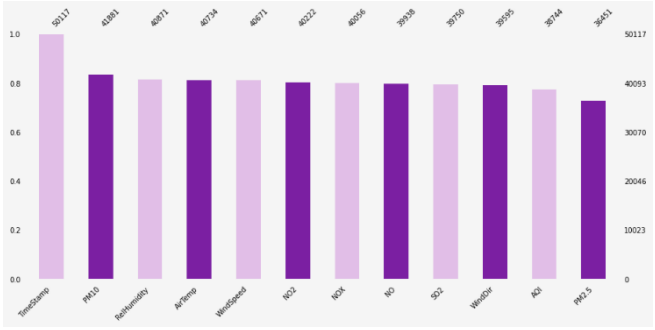


Figure 1: Missing data in last 5 years Auckland Council Environmental dataset

Before dropping the na (missing) values from the dataframe, we wanted to first eliminate other values which would not be included as to minimize the potential loss of relevant datapoints in attempting to clean the data set. The top 10 rows of the dataframe, indicate that that the NO_x variable is an aggregate of NO and NO_2, so the NO_x variable was dropped as it would not provide any information not provided by NO and NO_2, and possibly introduce multicollinearity into the model.

AQI was dropped as it was an index number determined by the other pollutants and the PM2.5 and PM10. These variables are what determine the AQI and not AQI being a feature that can determine the concentration.

The timestamp on the data points was at hourly intervals however there were some rows where the time was not on the hour and a record had been created either one minute to or one minute after the hour. On these timestamps, there was no matching data where the timestamp did not end in an even '00' were dropped from the dataframe. The dataframe was also checked for any duplicate values of which zero were found.

III. Data Exploration and Feature selection

The data was also visually explored to observe for patterns and distributions. PM10 and PM2.5 had both been included in this stage of preprocessing in order to establish if one variable was going to show stronger relationships with the features.

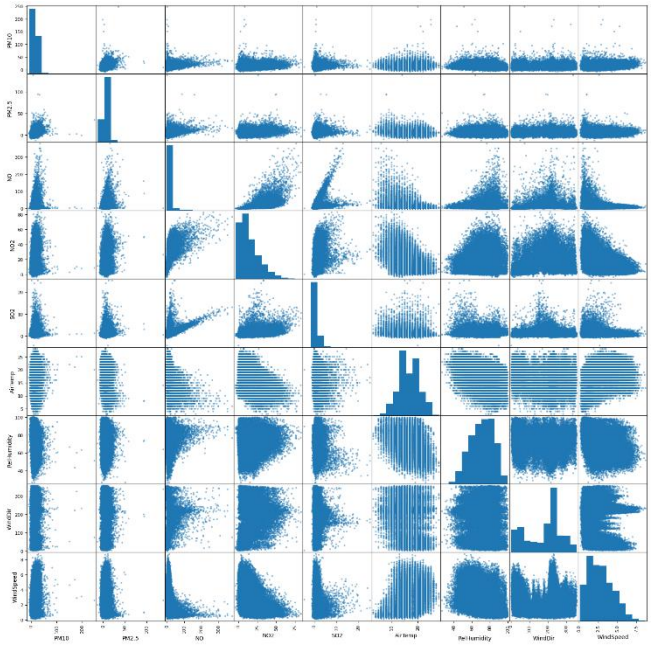


Figure 2 – Scatter Plot Showing Relationships

The targets PM10 and PM2.5 sightly positive linear relationships with the pollutants in the dataset. The other meteorological variables such as wind and temperature had less visually obvious patterns, instead looking to be somewhat flat in their distribution.

Figure ds

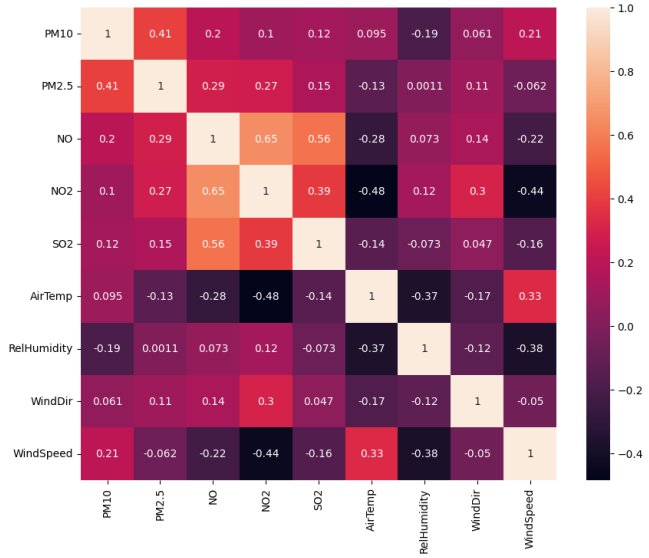


Figure 3 - Heatmap for Pearson Correlation

We used the seaborn package to generate the heatmap to view the correlation of the features. PM2.5 had larger Pearson correlations with the three pollutant variables, however PM10 had the larger correlations with windspeed and relative humidity (although this was a negative relationship for PM10). This is one of the primary reasons for selecting PM2.5 over PM10 as the target variable in the model. The meteorological variables tend to have weaker relations with the potential targets by Pearson's correlation,

Wind direction is an angular variable representing direction. The angular nature of the variable can present some issues when creating models, for example the mean directionality between 350° and 10° is 0, not 180. However, the wind direction is still an important consideration to be visualized as it could provide insight air quality given local industry or motorway locations relative to the measuring station.

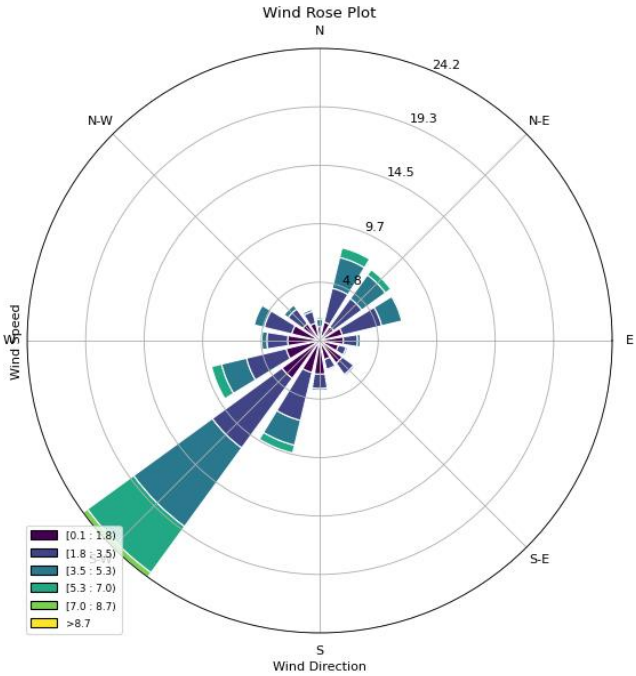


Figure 4 – Wind Rose Plot using Wind direction feature

As per figure 4, the wind direction is mostly clustered around wind in the south west direction, and somewhat the opposite to a much lesser extent in the north east. While this could provide important insight to the model it also brings new challenges to the model. Wind direction was therefore excluded from the data.

Using a violin-boxen plot the feature distribution and skewness were observed through a plot to show univariate feature distribution in figure 5. Using pandas method to show descriptive statistics in combination with the plot assisted with understanding the data. The features showed a normal distribution although the presence of outliers on the upper end was clear, as well as the presence of negative values for the pollutant data points. Air temperature, relative humidity and wind speed had some outlier presence however the ranges were not extreme and within realistic values for what Auckland experiences and so it was decided to include these values rather than removing them.

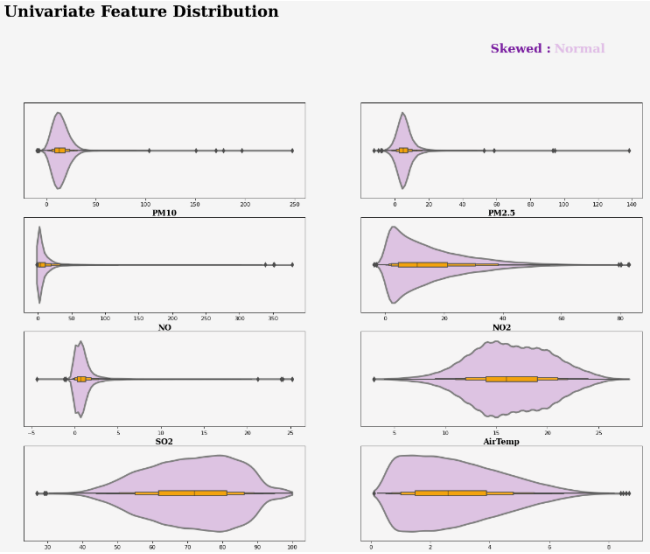


Figure 5 – Univariate Feature Distribution

Next the outliers and negative values of pollution were addressed. First the values which were negative concentrations for NO, NO_2, and SO_2 were changed to Nan values so they can be later discarded, as the negative value indicates a potential issue with the measuring equipment such as miscalibration and cannot be 'trusted'. The inter-quantile method was used to handle the upper outliers of the data. When handling the upper outliers, it was considered the effect of removing too much of the training data or too little on the models. The final model used the more conservative method of using 0.05 and 0.95 for the quantiles, however it was experimented with 0.25 and 0.75 also.

The variance inflation factor (VIF) of the remaining features was checked in order to determine the presence of multicollinearity.

	Feature	VIF
0	NO	3.373263
1	NO2	5.390534
2	SO2	2.786331
3	AirTemp	14.256221
4	RelHumidity	13.587869
5	WindSpeed	4.935844

Figure 6 – Variance Inflation Factor of the features

The was evidence of multicollinearity with air temperature and relative humidity. This is likely explained by higher temperatures in Auckland being associated with higher humidity, particularly through warmer time periods. Air temperature was dropped due to the higher VIF.

The fastai library function add_datepart was used to create time series data which would be more usable in the training data. This created data used in the model to represent temporal features such as year, day, day of week. The method also creates Boolean features for beginning of month however all the Boolean features were excluded from the data as it was not believed the specific day of month being the beginning or end would have any affect as opposed to weekly cycles which may affect pollution patterns with traffic and potentially local industries.

To conclude, the features chosen were NO, NO2, SO2, Relative Humidity and windspeed. These features were shown to have relative independence of each other through small enough VIF. The pollutants chosen all have positive linear relationships with the chosen target variable. NO_x was the only pollutant excluded as it was an aggregate of the other two nitrogen based pollutants. The relationship between the target and relative humidity and wind speed did not follow a strong linear relationship however, there is the possibility that given the temporal nature of the data it may be an important part of the training model.

IV. Experimental Methods

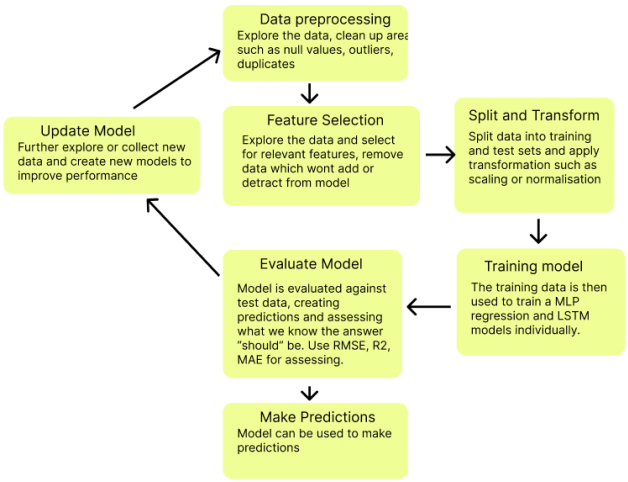


Figure 7 – Workflow Diagram

70% of the data was to be used for the training set and 30% was to be used as the testing set. There were three methods which were considered for splitting the data, and each set was used and compared to one another.

The first method was sci-kit learn's train_test_split method. This method is not optimal for time series data however as it randomizes the sets which means the time series nature of the data doesn't translate into the model well.

The second method was to manually split the data, which would mean the 'top' 30% of the data set becomes training data and retains the temporal nature of the dataset.

The third method was to use the FastAI library's method time_series_split with 5 folds.

For the MLP model the features had the sci-kit learn StandardScaler applied to them for feature scaling, as MLP models tend to be sensitive to the scale of input features. This

prevents features with large values such as relative humidity from dominating the training process in the MLP model. Other reasons for feature scaling include improved accuracy and speed of convergence.

V. MULTILAYER PERCEPTRON

A. MLP 1)

A Multi-Layer Perceptron (MLP) is an artificial neural network with numerous layers of connected neurons. It is often used for machine learning tasks including data categorization (classification) and numerical prediction (regression). The architecture of an MLP is very adjustable, allowing it to be adapted to various machine learning applications by altering the number of layers, neurons in each layer, and activation functions. Here's a general description of how an MLP operates and is structured:

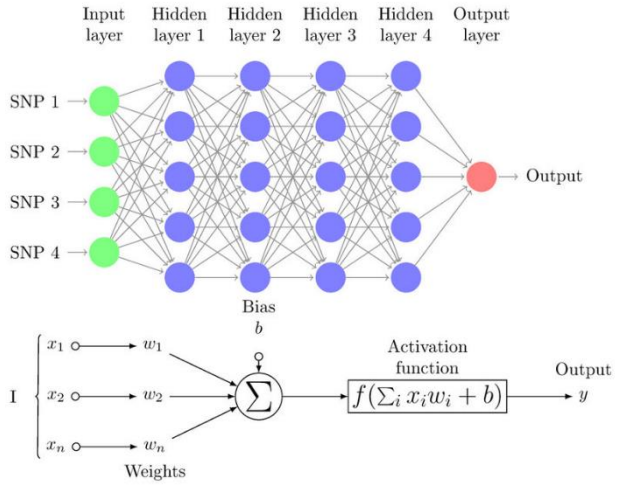


Figure 8 – MLP structure

Input Layer: The first data features are sent on to the input layer. Each neuron in this layer corresponds to a characteristic from the input dataset. The number of neurons equals the number of characteristics. Neurons in this case merely carry input data to the next layer without performing any calculation.

Hidden layer: Hidden layers apply nonlinear adjustments to incoming data, allowing the network to learn complicated patterns. Each neuron in a buried layer receives inputs from the neurons in the previous layer. The problem's complexity determines the number of hidden layers and neurons in each layer. Nonlinearity is introduced via activation functions (for example, ReLU and sigmoid).

Output Layer: The output layer is responsible for producing the network's final predictions or outputs. The number of neurons in this layer varies with the task at hand. For binary classification, typically one neuron determines the likelihood of one class. For multi-class classification, the number of neurons corresponds to the number of classes. In regression, typically one neuron predicts the continuous value.

Weights and biases: Each neuronal connection is assigned a weight that indicates its relevance. Weights are modified during the training process to reduce inaccuracy. Neurons (except for the input layer) feature a bias component that captures the input space offset from zero.

Activation Function: Activation functions introduce nonlinearity, which is necessary for the network to learn complicated mappings. ReLU is a popular activation function for hidden layers, while sigmoid or SoftMax is used for binary or multi-class classification and linear for regression.

An MLP is well-suited for modelling the nonlinear or complex relationships between various environmental factors (like NO, NO_2, and SO_2 levels, temperature, humidity) and PM2.5 concentrations. By training on historical environmental data, the MLP learns to predict PM2.5 levels based on current or future environmental conditions. The feedforward nature of MLP means steps must be taken to ensure the temporal nature of the data is maintained, through including lag variables of the target for the previous hour and two hours respectively so the historical data is passed through the hidden layers. This captures the temporal dynamics of the data allowing seasonal trends to be detected across the days, weeks, and years.

An issue with MLP to consider is the risk of overfitting where the model conforms too strictly to the training data and loses effectiveness at predictions due to learning noise and fluctuations. A model which contains too many neurons may fall into this trap. Poor feature selection where features are irrelevant but included can also contribute to this issue, where the model may start associating random noise with outcomes.

B. MLP 2) Regression Model – Single Layer

The next step in creating models was to use sci-kit learn's MLPRegressor neural network to create the first MLP model. The first model contained a single hidden layer with 25 neurons.

The model with a single layer and 25 neurons, with the rest of the parameters being default produced:

```
Mean Absolute Error: 2.5553399534776804
Root Mean Square Error:
3.495156631097296
R-squared: 0.26104767832865383
```

C. MLP 2) Regression – Best learning rate

Figure 10 shows the MLP cycling through various learning rates to find the rate that creates the best model all else being the same. The best identified learning rate across a variety of selected rates was 0.2. When a rate of 0.01 was run through the model it consistently produced the higher RMSE and lower R^2 values.

D. MLP 3 – Two layers, 25 neurons

The next step was to calculate the optimal number of neurons per layer across two different layers with a max number of neurons as 25. The optimal number of configurations was found to be 3 neurons in the first layer with 22 neurons in the second layer, which gave the results of RMSE is 3.418. Despite this indication, when the learning

rate was adjusted to 0.01, which as per the previous exercise was determined to be a worse performing learning rate, the overall model performed better, with an RMSE of 3.313, and an R^2 of 0.34.

E. MLP 4) Variation in performance

Throughout the experimental phase, it was observed that the performance of MLP models varied significantly with different architectural setups. The testing was performed to identify the optimal structure for predicting PM2.5 from environmental data.

Neuron distribution across layers had a significant effect on the model performance. The best results found were a split of 5 and 20 on layer one and two respectively, however a variety of configurations had different results. As evidence by changing parameters, the learning rate also has an effect on the model's performance and the learning rate from one architectural set up does not necessarily translate to what's optimal for another. Some configurations seem inexplicably poor performing despite 'surrounding' configurations performing better, such as the 9-16 configuration.

A network with a single heavy layer may fail to capture the depth of interactions as effectively as a multi-layered approach where each layer can build upon the features processed by the previous one. The optimal 5-20 configuration likely provided a balance between capturing detailed interactions at the first layer and allowing deeper abstractions and combinations in the second layer, which enhanced model performance.

```
Configuration: 24-1 -> RMSE: 3.8257336528603787 -> MAE: 2.80253148198601 -> R2: 0.11465488652158573
Configuration: 23-2 -> RMSE: 4.16596999916379 -> MAE: 3.1631276722036565 -> R2: -0.04982136859908004
Configuration: 22-3 -> RMSE: 3.4513859951348365 -> MAE: 2.4034299200326186 -> R2: 0.27943992123084416
Configuration: 21-4 -> RMSE: 3.735969720335848 6 -> MAE: 2.7744531370379875 -> R2: 0.1557135340739192
Configuration: 20-5 -> RMSE: 4.04830948242882 -> MAE: 2.9232228501189668 -> R2: 0.008641935836234294
Configuration: 19-6 -> RMSE: 4.175730108302559 -> MAE: 3.0110214159331528 -> R2: -0.05474621146963 11
Configuration: 18-7 -> RMSE: 3.869516102849429 -> MAE: 2.9188662296263757 -> R2: 0.09427480521765641
Configuration: 17-8 -> RMSE: 3.8181207281769916 -> MAE: 2.834046085888486 -> R2: 0.11817492224004011
Configuration: 16-9 -> RMSE: 3.6399728161515 2 -> MAE: 2.640975721262196 -> R2: 0.1985733603048332 52
Configuration: 15-10 -> RMSE: 4.227324917180759 -> MAE: 3.08978531554019 -> R2: -0.08097186670638219
Configuration: 14-11 -> RMSE: 3.61229017154907 87 -> MAE: 2.5227446950672814 -> R2: 0.21068857132417 44
Configuration: 13-12 -> RMSE: 3.926492535668433 -> MAE: 2.970741930941723 -> R2: 0.0674058549612514
Configuration: 12-13 -> RMSE: 3.9246719008360 16 -> MAE: 2.8015653162965575 -> R2: 0.06827050437829296
Configuration: 11-14 -> RMSE: 3.7991903409049295 -> MAE: 2.8286348393176124 -> R2: 0.1268974894037871
Configuration: 10-15 -> RMSE: 3.502232792897577 -> MAE: 2.487740614556896 -> R2: 0.2580525396802033 6
Configuration: 9-16 -> RMSE: 4.085473018472201 -> MAE: 3.05585017993777 -> R2: -0.009642969526811651
Configuration: 8-17 -> RMSE: 3.432892607658416 -> MAE: 2.4475779893457674 -> R2: 0.2871411147913744
Configuration: 7-18 -> RMSE: 3.6834795382290584 -> MAE: 2.7707969641252026 -> R2: 0.1792712337402524 4
Configuration: 6-19 -> RMSE: 3.5017826403272 -> MAE: 2.496133722075855 -> R2: 0.2582432569236037
Configuration: 5-20 -> RMSE: 3.312963401035184 -> MAE: 2.3384968712402365 -> R2: 0.3360789956973575 5
Configuration: 4-21 -> RMSE: 3.526488526289605 -> MAE: 2.5808923132682096 -> R2: 0.2477398044693594
Configuration: 3-22 -> RMSE: 3.3357442147203793 -> MAE: 2.35210650940097 -> R2: 0.32691701021806685
Configuration: 2-23 -> RMSE: 3.325905092070304 -> MAE: 2.3596211028394154 -> R2: 0.3308818101080335
Configuration: 1-24 -> RMSE: 3.3508637549256957 -> MAE: 2.3403878739883446 -> R2: 0.3208015720706302
Best Configuration: (5, 20) with RMSE: 3.312963401035184 with r2: 0.3360789956973575 5
```

Figure 9 - Single Layer MLP Learning Rates

From the experimental setups and the results obtained, it is evident that the architecture of an MLP significantly impacts its predictive performance. The best-performing model configuration involved an initial layer which had fewer neurons, most likely focusing on capturing primary relationships and passing refined features onto a larger second layer for deeper processing. This setup not only minimized the RMSE but also provided the highest R^2 value, indicating a better fit to the data variability compared to other tested configurations.

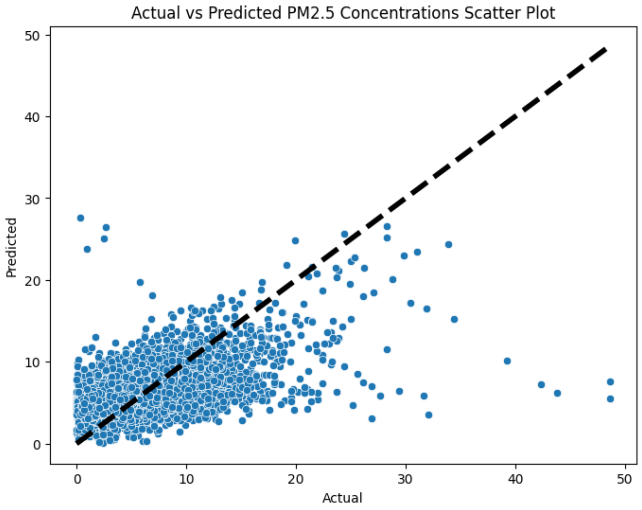


Figure 10 MLP Actual vs Predicted Scatter Plot

VI. Long Short Term Memory

A. LSTM 1) Describe LSTM Architecture

Long Short-Term Memory (LSTM) networks are a sort of recurrent neural network (RNN) that can handle sequential input while overcoming normal RNN constraints, specifically the vanishing gradient problem. LSTMs can learn long-term dependencies, making them ideal for tasks such as time series prediction, natural language processing, and speech recognition. An LSTM cell's architecture consists of numerous distinct components:

Input: At each time step, the LSTM accepts an input vector that represents the current observation or token in the series.

Cell State (Ct):

The cell state determines LSTM's ability to remember long-term information. It moves across the cell with few modifications, allowing information to be stored for lengthy periods of time.

Hidden State (ht):

The hidden state contains information that affects the output at the current time step. It is also transferred to the following time step, along with the cell state.

Gates:

LSTMs use gates to regulate the flow of information. There are three types of gates:

Forget Gate (ft): Determines the information to discard from the cell state. It takes the previous hidden state (ht−1ht−1) and the current input and returns a 0 to 1 value for each number in the cell state.

$$ft = \sigma(Wf \cdot [ht-1, xt] + bf)$$

Input Gate(it)**:** Determines which new information to incorporate into the cell state. It consists of two components:

the input gate layer and a vector of new candidate values that could be added to the state.

$$it = \sigma(Wi \cdot [ht-1, xt] + bi)$$

$$C\tilde{}t = \tanh(WC \cdot [ht-1, xt] + bC)$$

Output Gate(ot)**:** Determines what the next concealed state will be. It determines which aspects of the cell state should be output.

$$ot = \sigma(Wo \cdot [ht-1, xt] + bo)$$

How does LSTM differ from MLP?

Neural networks of the LSTM (Long Short-Term Memory) and MLP (Multilayer Perceptron) varieties have diverse structures and functions. The main applications of LSTMs are time series analysis and sequential data analysis. They have memory cells in them that can store data for extended periods of time, allowing the network to recall earlier inputs. LSTMs are made up of cells with input, forget, and output gates that regulate the flow of information, allowing them to manage dependencies over a range of time steps. For applications such as language modeling, speech recognition, and time series prediction, this makes LSTMs appropriate. However, because they require the management of sequential data and long-term dependencies, they are more difficult to train.

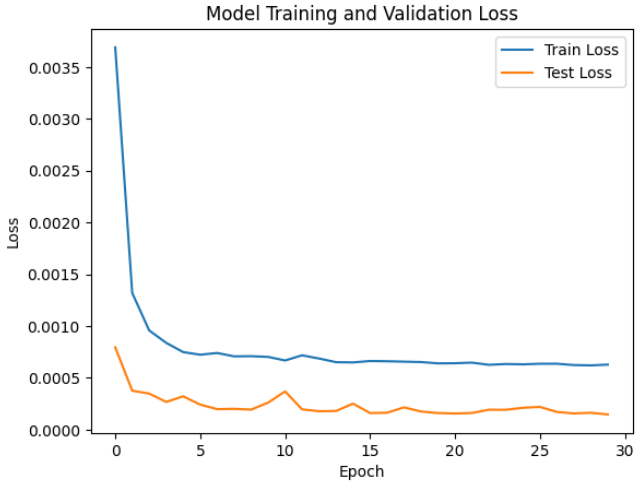
However, MLPs are employed for jobs involving general-purpose pattern recognition and categorization. An MLP is made up of an input layer, an output layer, and one or more hidden layers. Every layer in an MLP is fully connected to every other layer. MLPs, in contrast to LSTMs, rely on feedforward connections and activation functions and have a straightforward architecture devoid of memory cells. They handle inputs on their own, without taking dependencies or order into account. Because of this, MLPs are appropriate for applications like tabular data classification and recognition of images. Due to their ability to handle independent input points and lack of requirement to address temporal dependencies, MLPs are typically easier to train than LSTMs. The number of neurons in each LSTM gate (forget, input, output) determines the capacity of the model to learn and retain information. More neurons can capture more complex patterns but also increase the risk of overfitting and require more computational power.

In conclusion, MLPs are simpler neural networks utilized for applications where input data points are not dependent on each other, whereas LSTMs are specialized to handle sequential data and sustaining long-term dependencies. To control information flow over time, LSTMs use memory cells and gates, but MLPs have a more straightforward design with completely connected layers.

B. LSTM 2)

The lag variables are no longer necessary for the LSTM model as the there is a back-propagation which was absent from the MLP model. LSTM is a Recurrent Neural Network which makes it a suitable choice for temporal data as it can capture patterns over time and make use of historical data. The lag variables were therefore dropped prior to normalization of the data and fitting to the LSTM model.

Figure 11 LSTM Epoch



From figure 11 we can see the stablises with a low value after 10. From the numbers produced by running the model over 30 epochs the best epoch would be 30, where the cost function is smallest. The graph also indicates that is the case.

C. LSTM 3

VII. Model Comparison

A. Model Comparison 1)

The map comparing actual and anticipated PM2.5 concentrations reveals some critical insights about your model's performance. First, the model detects the overall increased trend in PM2.5 levels, demonstrating its ability to discern broad patterns in the data. However, the accuracy of the projections differs among time periods. Initially, the projected values greatly underestimate the actual concentrations, especially at Time 2. This gap narrows at the halfway, particularly around Time 4, when the model's predictions closely match the actual values, implying improved performance in this period.

Toward the end of the period, at Time 6, the forecasts diverge again, albeit less substantially than at the start. This variable precision, particularly where the model suffers with quick fluctuations in PM2.5 levels (as seen between Times 3 and 5), may indicate problems with the model's capacity to react to more volatile environmental changes.

These observations point to either overfitting to certain data properties or underfitting, as the model does not generalize effectively throughout the full period. Incorporating more detailed temporal features or altering the model's complexity may help improve it. Furthermore, investigating more robust feature engineering may assist improve accuracy and reliability. Understanding these dynamics is critical for fine-tuning the model, which might be further investigated by studying more plots with larger datasets or under different climatic conditions.

B. Model Comparison 2)

The RMSE for the LSTM model is 0.0155, whereas with the MLP Regression model the RMSE is 3.13. Given the discrepancy and the very small number of the LSTM model there may be overfitting occurring. The R^2 value for the LSTM is 0.99. A model which fits the data this well is almost certainly overfitted, as opposed to the MLP model which has an R^2 of 0.34 and may be an ineffective model do to low performance instead.

As lower RMSE and R^2 is closer to 1, it would be concluded that the LSTM model performs better. This is in line with what would be expected given LSTM's suitability to working with temporal data and it's architectural configuration with recurrent neural networks. However, given the results are 'too good to be true' there is likely overfitting, and more work needs to be done into improving the model.