STAT 654

PREDICTING AIR POLLUTANT CONCENTRATION WITH MULTISENSOR DATA

GROUP 6

Priyadharshini Ramesh Kumar - 534002478

Hasitha Varada - 734004713

<name> - <uin>

ABSTRACT

Pollution is a great problem for public health and environment protection. In this project, we propose to study the AirQuality Dataset, where we try to find how pollution is distributed during a day, when the peaks happen, and how pollutant concentrations are related to each other. The dataset: The data is collected from a gas multi-sensor device that is deployed in a heavily polluted area of an Italian city. The dataset is collected from March 10th 2004 to February 10th 2005 (one year). We are given the hourly averaged responses of gas multi-sensor arrays (12) exposed to different gaseous mixtures and an hourly averaged reference value which represents the Truth. The data contains various pieces of information like, a date and time attribute, the concentrations of various pollutants (CO, NMHC, C6H6, NOx, NO2, and the O3), and some meteorological attributes, like, temperature, relative humidity, absolute humidity. The target is to estimate the gas concentration using various Machine Learning models, with Random Forest achieving the lowest RMSE and highest R2 value.

TABLE OF CONTENTS

CONTRIBUTION

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 PROBLEM STATEMENT

One of the most dangerous threats to environmental quality, urban sustainability, and public health is still air pollution. However, due to strict regulations and improvements in emission control, pollutants such as carbon monoxide (CO) have continued to rise unchecked to dangerous levels, especially in densely populated urban areas. The dynamics of air quality are subtle, and managing and predicting it is a difficult task due to a confluence of factors such as industrial discharges, car emissions, and complicated meteorological conditions. Predictive analytics and precise real-time monitoring are essential for putting preventative measures in place to lower pollution levels. The only issue with accurate prediction is that environmental variables typically have a sporadic nature while air pollutants have a heterogeneous nature.

1.2 OBJECTIVE

Our research is motivated by two main goals that are essential to improving the way that air quality is currently understood and managed in urban environments. The thorough investigation of the connections among a variety of airborne contaminants is the first aspect of our goal. This entails a detailed analysis of sensor data to clarify the mutually beneficial and antagonistic relationships between different contaminants, such as carbon monoxide (CO), benzene, and nitrogen oxides (NOx). By figuring out these connections, we will be able to identify the origins of emissions and comprehend how they change in response to various environmental conditions. This kind of knowledge is essential to public health and urban planning because it opens the door to focused pollution management strategies.

The creation of an advanced predictive model using the multisensor data is the second aspect of our goal. This model, which was developed to capture the complex patterns seen in the sensor data, is more than just a forecasting tool. It is an example of the complicated interaction between statistical science and machine learning. Our objective is to develop a strong, predictive model that can precisely estimate the amounts of pollutants by utilizing cutting-edge machine learning techniques. The model's utility is found in its application; it is not an end in and of itself. We see a predictive system that can both predict growing pollution levels and offer useful information for preventative actions. This may involve alerting locals to imminent poor air quality.

CHAPTER 2
LITERATURE SURVEY

Author Name/ Journal/ Year	Title of the paper	Methodology	Pros/Cons
World Health Organization, 2022	Ambient (Outdoor) Air Pollution [1]	The fact sheet synthesizes data from global health studies and provides an overview of the health impacts associated with exposure to fine particulate matter, carbon monoxide (CO), ozone (O3), nitrogen dioxide (NO2), and sulfur dioxide (SO2).	Offers global and up-to-date insights into the health burden from air pollution. Lacks detailed technical data and analysis that might be found in full research papers.
Ishita Chanana, Aparajita Sharma, Pradeep Kumar, Lokender Kumar, Sourabh Kulshreshtha, Sanjay Kumar, Sanjay Kumar Singh Patel / Fire / 2023	Combustion and Stubble Burning: A Major Concern for the Environment and Human Health [2]	It provides an overview of the problems associated with uncontrolled combustion activities, the sources and dispersal of pollutants, as well as mitigation techniques for human health and the environment.	Discusses the broad range of pollutants from such activities, including greenhouse gases and particulate matter. The complexity of some discussed mitigation strategies may not be easily understood by laypersons without sufficient background knowledge.
Vicki MacMurdo (Anoka-Ramsey Community College) / LibreTexts/ 2021	16.4:Air Pollution [3]	The document provides a comprehensive educational overview of air pollution, detailing types of pollutants, their sources, and effects on health and the environment.	It discusses both natural and anthropogenic sources of air pollution, with an emphasis on understanding human-caused pollution which is actionable and can be mitigated. The paper may not address specific case studies or provide localized insights into air pollution issues.
Our Nation's Air - Trends Through / U.S/ Environmental Protection Agency (EPA) / 2020	Our Nation's Air: Trends Through 2020 [4]	It shows a significant reduction in the emissions of key pollutants by 78%, alongside a growing economy. The report notes declines in average concentrations of harmful air pollutants across the nation from 1990 to 2020, including reductions in carbon monoxide, lead, nitrogen dioxide, ozone, and particulate matter	Significant reduction in key air pollutants since 1970, with a 78% decrease in the emissions of criteria pollutants. Air quality is still a concern, especially in areas affected by wildfires and other natural events that can worsen pollution levels.

CHAPTER 3
DATA PREPROCESSING AND EDA

3.1 INTRODUCTION

Our dataset's integrity is provided by a thorough preprocessing step before we dive into predictive analytics. This vital step, which identifies and corrects inconsistencies, handles missing values, and smoothes out anomalies, serves as the cornerstone of our exploratory data analysis (EDA). By applying EDA, we not only get our dataset ready for the use of sophisticated machine learning algorithms, but we also start to reveal the fundamental structure of the data and the first insights it contains.

3.2 ABOUT THE DATASET

The dataset, which has 9,357 instances, provides a snapshot of the urban air quality over a calendar year from UCI repository. This represents the longest duration for which such sensor device response data are publicly accessible. The sensors are intended to gather data regarding the subsequent air pollutants:

Nitrogen dioxide (NO2), Total Nitrogen Oxides (NOx), Benzene, Non-methane Hydrocarbons (NMHCs), and Carbon Monoxide (CO). A dual dataset of sensor responses with confirmed concentration levels was produced by co-measuring these pollutants and using certified reference analyzers stationed at the same location to provide ground truth values. As we can see in the figure 3.2.1 our dataset includes a variety of characteristics, ranging from environmental factors like humidity and temperature to the quantity and concentration of particulate pollutants and how sensors react to those pollutants.

The primary variables are records of the date and time, hourly averaged concentrations from the reference analyzer of different pollutants, averaged hourly sensor readings from the chemical sensors intended to monitor particular gases, the absolute, relative, and temperature humidity values.

	Date	Time	CO(GT)	PT08.S1(CO)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
0	2004-03-10	18:00:00	2.6	1360.00	11.881723	1045.50	166.0	1056.25	113.0	1692.00	1267.50	13.60	48.875001	0.757754
1	2004-03-10	19:00:00	2.0	1292.25	9.397165	954.75	103.0	1173.75	92.0	1558.75	972.25	13.30	47.700000	0.725487
2	2004-03-10	20:00:00	2.2	1402.00	8.997817	939.25	131.0	1140.00	114.0	1554.50	1074.00	11.90	53.975000	0.750239
3	2004-03-10	21:00:00	2.2	1375.50	9.228796	948.25	172.0	1092.00	122.0	1583.75	1203.25	11.00	60.000000	0.786713
4	2004-03-10	22:00:00	1.6	1272.25	6.518224	835.50	131.0	1205.00	116.0	1490.00	1110.00	11.15	59.575001	0.788794

Fig 3.2.1 Air Quality Dataset

3.3 PREPROCESSING

Data Cleaning: Every instance that was found that had a "-200" was marked as missing. To strengthen the study, any column that had a substantial percentage of missing data such as NMHC(GT) was removed. In figure 3.3.1 there are mean and median values mentioned.

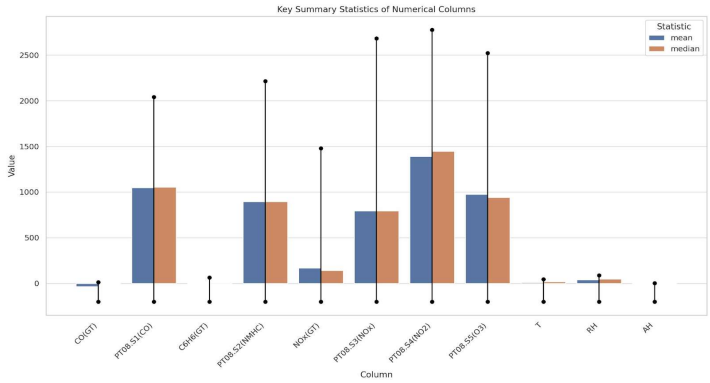

Fig 3.3.1 Summary Statistics of Numerical Columns

Data Transformation: Time parsing is the process of converting date and time data into a format that is appropriate for creating a time series. The columns with a tolerable amount of missing values shown in figure 3.3.2, relevant approaches for data imputation based on the kind of data were implemented. Recalculating additional features, like hourly variations in pollution levels, using feature engineering may help the model perform better.

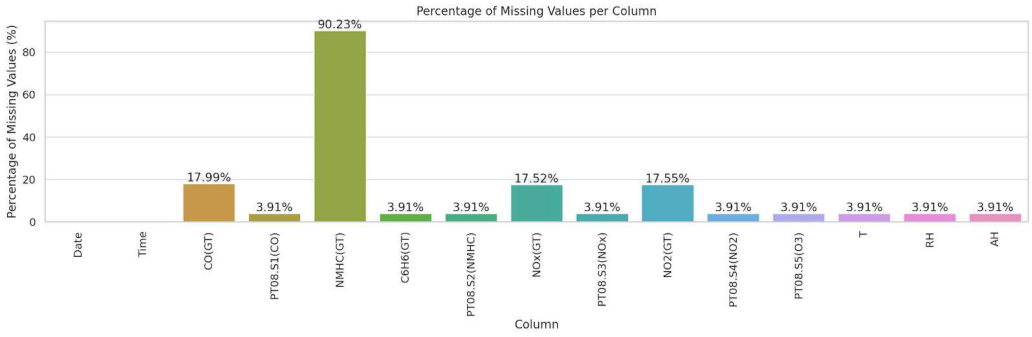

Fig 3.3.2 Percentage of Missing Values per column

3.4 OUTLIER DETECTION

Identification: To visualize the data's distribution and potentially spot outliers, a box plot was made for each pollutant and sensor response variable. Given in figure 3.4.1 sample boxplots , where in every box plot, an outlier is usually represented by a point outside the "whiskers" that extends to 1.5 times the interquartile range (IQR) from the quartiles.

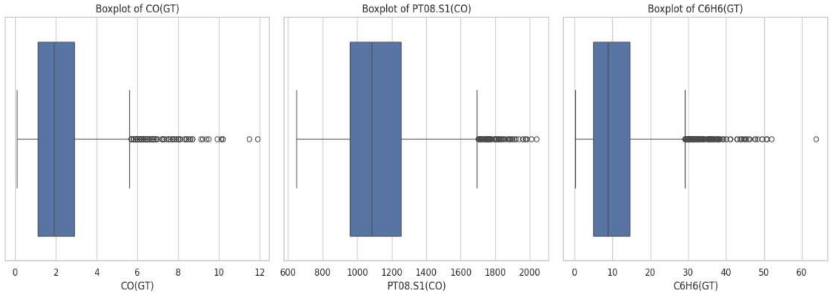

Fig 3.4.1 Boxplots of CO(GT), PT08.S1(CO), C6H6(GT)

Treatment: Using the IQR approach, lower and upper boundaries were computed to determine whether values can be considered outliers. In figure 3.4.2 we have mentioned the distributions of every column after removing outliers in order to preserve as much data as possible, also we only eliminated the most extreme deviations when identifying outliers, using a conservative 1.4 times the IQR. We aimed to preserve the actual representative in situ data of typical environmental circumstances by removing 846 outliers.

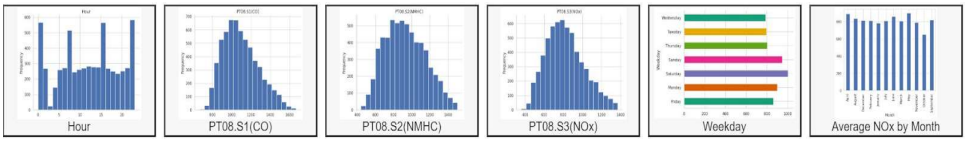

Fig 3.4.2 Distributions of every column

Removing Reference Analyzers: The information in the pertinent columns, which refer to "CO (GT), NOx (GT), C6H6 (GT), and NO2 (GT)," was not significant to us because we were more interested in the sensor responses than the concentrations of these pollutants.

Techniques: To properly delete the aforementioned columns from your working dataset, all you have to do is use the DataFrame's.drop() method to remove the columns. This dataset's EDA appears The dataset is now ready for EDA, which aims to comprehend sensor behaviors and influences on the environment, to make readings because these columns have been removed, leaving it with only the values of sensor responses, environmental parameters, and time variables.

3.5 EXPLORATORY DATA ANALYSIS (EDA)

Monthly Analysis: To investigate the variation in CO levels as recorded by the PT08.S1(CO) sensor, scatter plots were made for each month. In figure 3.5.1 I have provided information on seasonal effects on air quality by highlighting particular patterns, such as variations and concentrations of pollutants within various months.

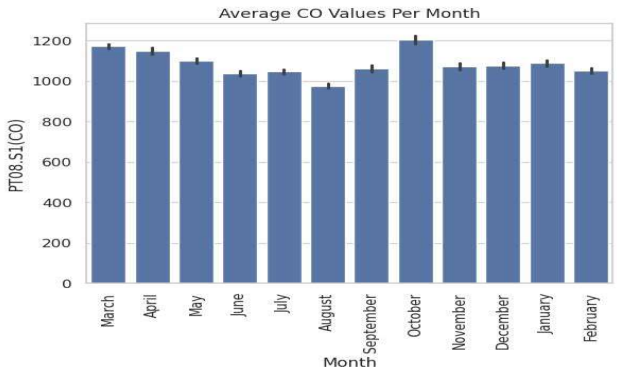

Fig 3.5.1 Average CO values per month

Weekday Analysis: The differences in CO levels were also shown by scatter plots, one for each day of the week. In figure 3.5.2 I have mentioned that the research revealed notable variations between weekdays and weekends, providing an insight into the weekly cycle of air pollution.

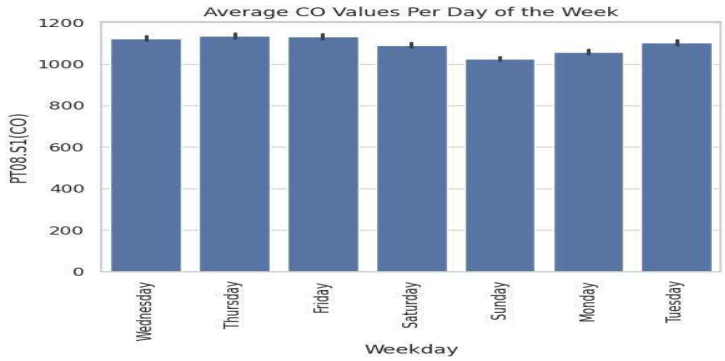

Fig 3.5.2 Average CO values per day of the week

Hourly Analysis: Diurnal trends in pollutant levels were revealed by scatter plots spanning a full day. In figure 3.5.3 I have mentioned the possible peaks during rush hours and troughs during periods of predicted decreased vehicular and industrial activity were displayed in the visualization.

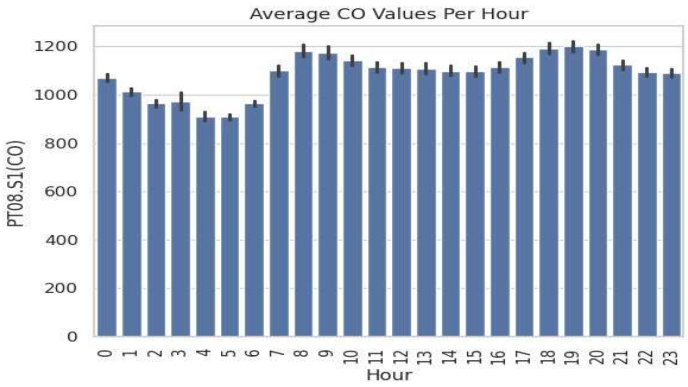

Fig 3.5.3 Average CO values per hour

Correlation Analysis: The sensor responses show significant positive relationships with one another. For instance, there is a significant positive association between PT08.S1(CO), PT08.S2(NMHC), and PT08.S5(O3). This essentially indicates that when one pollutant's concentration rises, so does the concentration of other pollutants. In figure 3.5.4 I have mentioned the correlations between variables. This may suggest that emissions from industry and traffic are frequent sources of these contaminants.

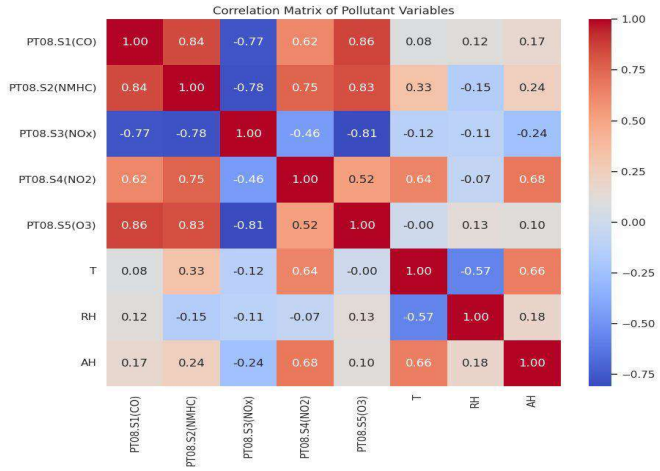

Fig 3.5.4 Correlation Matrix

Correlations with Environmental Factors: There is a moderate positive correlation between temperature (T) and PT08.S4(NO2), and a moderate negative correlation between temperature (T) and RH. In figure 3.5.5 we have also mentioned the pairplot to visualize more clearly and this may imply that humidity may decrease with temperature, which may be brought on by increased photochemical processes occurring in the warm atmosphere, and that NO2 content is likely to increase with temperature.

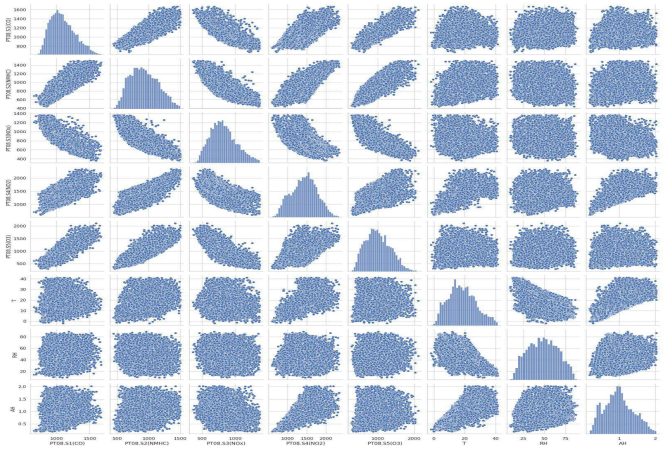

Fig 3.5.5 Pairplot of all variables

CHAPTER 4
METHODOLOGY

4.1 INTRODUCTION

After defining the attributes of the AirQuality Dataset in the preceding chapter, this part explores the process used to harness the potential for prediction of different machine learning models. Predicting CO concentrations using the remaining pollutants in the dataset is our major goal. To do this, we employ a typical machine learning process that includes training and evaluating the models. Twenty percent of the data is reserved for validation, while the remaining eighty percent is used to train the models.

4.2 LINEAR REGRESSION

One of the machine learning models used for CO prediction in this work is Multiple Linear Regression (MLR) using scikit-learn's linear_model.LinearRegression. MLR is a statistical method to make a linear relationship between a dependent variable and various independent variables. The dependent variable is the one you want to predict, and the independent variables are those that predict the behavior of the dependent variable. In our study, the dependent variable is CO concentration, and all other pollutant concentrations are independent variables. The fundamental equation of MLR can be formulated as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon$$

The projected CO concentration (Y) is expressed as a function of an intercept (β_0), independent variable coefficients (β_1 to β_n), and a random error term (ε) in the linear regression model. The model finds these values for the coefficients β that minimize the error term ε across all the data points. Basically, this provides a best-fit line through this multidimensional space of independent variables. This allows us to predict the CO concentration for new, unseen data based on the values of the other pollutants and the meteorological measurements.

4.2.1 Regularization

The overfitting happens when the model memorizes too tightly the training data. It will fail to generalize well on unseen data. To avoid overfitting and thus improve the generalization of the MLR model used for the prediction of CO concentration, we have to look into some regularization techniques. The two most widespread methods are the following:

1. Ridge Regression (L2 Regularization):

$$\text{Minimize: } J(\beta) = (1/N)\ \Sigma(yi - \hat{y}i)^2 + \alpha\ \Sigma(\beta j)^2$$

2. Lasso Regression (L1 Regularization):

$$\text{Minimize: } J(\beta) = (1/N)\ \Sigma(yi - \hat{y}i)^2 + \alpha\ \Sigma|\beta j|$$

Through the use of a penalty term based on the sum of either squared coefficients (L2) or absolute coefficients (L1) for regularization, Ridge and Lasso regression minimizes a cost function (J(β)) that combines squared errors between predicted and actual CO concentrations.

4.3 DECISION TREE REGRESSOR

Also, we used Decision Tree Regression using sklearn.tree.DecisionTreeRegressor, an alternative for linear regression, to run this model. The method creates a tree-like model where every internal node is a split over some feature and some particular threshold, and the procedure iteratively partitions the data over these splits till achieving leaf nodes that represent predicted CO concentration values. It benefits from nonlinear relationships between the variables. The Decision tree regression model benefits from nonlinear relationships between the variables. It can also give insight into the decision-making process because of the structure of the tree. Yet, it tends to be prone to overfitting when not tuned appropriately.

$$\hat{y} = \sum(i=1 \text{ to } N)\ ci * I(x \in Ri)$$

The constant values (ci) associated with each region (Ri) specified by the characteristics (x) are added up to predict the CO concentration ($\hat{y}$), and the indicator function $I(x \in Ri)$ determines whether a given data point belongs in that region.

4.4 ENSEMBLE LEARNING

Ensemble learning is a technique in machine learning, where the output of multiple models is pooled to give a better overall prediction. Imagine a group of experts with their own interpretation. When pooled together, you have a more robust, more exact prediction. Ensemble methods work on similar lines—training multiple models on the same data and mixing the results. This, obviously, results in improved accuracy, can handle complex relationships, and avoids being over-dependent on any one model that's flawed. You'd be quite familiar with two of the most common ensemble methods, Random Forest (a collection of individual decision trees) and Gradient Boosting, which builds models sequentially to improve on one another.

4.4.1 Random Forest

A group of decision trees called the Random Forest model, using sklearn.ensemble.RandomForestRegressor, cooperate to enhance prediction accuracy. It lessens the possibility of overfitting that is typical with a single decision tree by averaging the output of several trees. The model's dependability is further supported by the residual plot, which displays a random distribution of residuals along the horizontal axis. The model is a powerful tool for predicting air quality measures since it can explain a substantial percentage of the variance in the CO concentration data, as evidenced by its R-squared value of 0.87.

4.4.2 Gradient Boosting

Gradient Boosting is an ensemble method, using sklearn.ensemble.GradientBoostingRegressor, that constructs trees in a step-by-step fashion,

trying to fix the mistakes of the prior trees. The RMSE and R-squared values of this technique demonstrated promise in our investigation, indicating that it has successfully captured the underlying patterns of the data. Its result is nevertheless impressive and shows the promise of improving algorithms in environmental data modeling, where prediction accuracy is crucial, even though it didn't exceed the Random Forest.

4.5 SUPPORT VECTOR REGRESSOR

Support Vector Machines, which are typically employed for classification tasks, provide the foundation for the Support Vector Regressor (SVR), using sklearn.svm.SVR. This idea is extended to continuous outputs by SVR, which seeks to match the optimal line within an epsilon threshold error margin. Because it makes use of kernel functions, which convert the input data into a higher-dimensional space where defining a linear separator is simpler, this method excels at managing non-linear relationships. SVR has the ability to capture the complexities of the relationship that can exist between highly non-linear and complicated inputs (such as different contaminants and environmental conditions) and outputs (such as sensor readings) in the context of environmental data. This is accomplished by concentrating on the data points that are closest to the decision boundary—the most important data points, or support vectors. This makes SVR less susceptible to data noise or outliers, which are common in real-world sensor data.

$$\text{Minimize: } 1/2\ ||w||^2 + C * \Sigma(\max(0, 1 - y\_i * (w^T * x\_i + b)))$$

||w||^2 (weight vector norm squared) controls model complexity + C * Σ(max(0, 1 - yi * (w^T * xi + b))) (hinge loss sum) for SVR.

4.6 MLP

Multi-layer Perceptron (MLP) is an artificial neural network with multiple layers of nodes that utilize feedforward architecture for learning complex data relationships. It consists of layers of nodes, including an input layer for features, hidden layers for pattern learning, and an output layer for predictions. Each node applies an activation function to introduce nonlinearity. The network adjusts weights and biases through feedforward propagation and backpropagation to minimize prediction errors. MLPs are versatile and can be used for regression, classification, and unsupervised tasks. In this case, we use it to predict air pollutant concentrations based on other pollutants in the dataset,i.e. regression task.

Optimization: To improve the output of our MLP (Multi-layer Perceptron) model, we performed Hyperparameter Tuning, which is an experiment with different values for hyperparameters such as the number of hidden layers, the number of neurons in each layer, activation functions, learning rate, and batch size. Hyperparameter tuning can help us find the optimal configuration for our model.

Grid search is a technique commonly employed to systematically explore the hyperparameter space and identify the optimal configuration for the MLP model.

CHAPTER 5
RESULTS AND DISCUSSION

5.1 TITLE

5.2 TESTING

5.2.1 Performance Metrics

Table 5.1 Comparison of Models

Model	RMSE	R2
PSNR		
SSIM		
MSE		

5.2.2 Residual Analysis

5.3 DISCUSSION

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

6.2 FUTURE WORK

REFERENCES

[1] T.

[2] Fw

[3] Fe

[4] Fwe

[5] scikit-learn developers. (2019). sklearn.linear_model.LinearRegression — scikit-learn 0.22 documentation. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[6] sklearn.tree.DecisionTreeRegressor — scikit-learn 0.23.2 documentation. (n.d.). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

[7] scikit-learn. (2018). 3.2.4.3.2. sklearn.ensemble.RandomForestRegressor — scikit-learn 0.20.3 documentation. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[8] 3.2.4.3.6. sklearn.ensemble.GradientBoostingRegressor — scikit-learn 0.21.2 documentation. (2009). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

[9] sklearn.svm.SVR — scikit-learn 0.23.1 documentation. (n.d.). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[10]