IMPROVED AIR QUALITY PREDICTION USING MOBILE AND FIXED IOT SENSING

Rajender Kumar[1] and Kritika Sharma[2]

[1]Assistant Professor and [2]B. Tech Student Electronics and Communication Engineering Department, NIT Kurukshetra, India

ABSTRACT

Air pollution is a critical global concern, causing significant harm to human health and resulting in many respiratory conditions. Consequently, the necessity for innovative methods and systems to accurately predict air pollution has emerged, with the aim of mitigating health risks. With the advancement of machine learning and deep learning techniques, there has been a growing interest in leveraging these approaches to develop accurate models for air quality prediction. Machine learning algorithms, such as linear regression, random forests, gradient boosting, and neural networks, have been extensively applied to air quality prediction tasks. These algorithms analyze historical air quality data along with various environmental factors to capture complex patterns and relationships. Paper investigates different ML algorithms viz, Random forest, light GBM, ANN, Decision tree, linear regression and XGBoost. XGBoost is the recommended model for its consistent accuracy and efficiency. It achieved the lowest error rates (MAE: 19.02, RMSE: 36.81) and the highest \(R^2 \) score (0.81), proving to be the most reliable choice for this regression task. However, Light GBM remains a close second and could be considered in scenarios where computational efficiency is a priority. The research on air quality prediction using machine learning and deep learning techniques holds great promise for improving understanding of air pollution dynamics and developing effective strategies for pollution control and public health management. Further advancements in these fields will contribute to the development of smarter and more sustainable cities with cleaner and healthier environments.*

Keywords*-Air Quality, ANN, Machine Learning, Deep Learning, RMSE, MAE, XGBoost.*

1. INTRODUCTION

The rapid urbanization and industrialization taking place worldwide have led to a severe crisis of air pollution in numerous countries. This pollution not only poses a substantial risk to public health but also significantly disrupts people's daily lives. In cities like Mumbai and Delhi, residents frequently need to wear masks before venturing outside due to the alarming levels of air pollution. Moreover, the fluctuating air quality throughout the day limits outdoor activities. The presence of various air pollutants leads to the occurrence of air pollution. Nitrogen dioxide, among these pollutants, plays a prominent role as one of the primary gases responsible for contributing to this environmental concern. Air particulate matter, categorized separately from other air pollutants, encompasses PM2.5 (particulate matter), which poses particular concerns for individuals. PM2.5 refers to fine particles in the air that have a diameter smaller than 2.5 µm. Numerous urban areas have taken proactive measures by establishing their own stations dedicated to monitoring air quality. These stations provide real-time and frequent updates on the current state of air quality throughout the day.

With the increasing awareness of air pollution, it has become increasingly vital to assess air quality in close proximity to individuals. This data enables people to make informed decisions about outdoor activities and plan the most favorable routes to their destinations. Traditionally, the conventional method for monitoring atmospheric conditions across a broad geographical region involves the establishment of fixed-location monitoring stations.While implementing a fixed sensor-based monitoring system may not be excessively complicated, it does face several challenges. Firstly, it requires a significant investment to construct and deploy monitoring units that can effectively cover a wide area. Furthermore, the accuracy of such a system diminishes as the distance from the monitoring stations increases, rendering it less reliable for remote locations. Furthermore, in proximity to road networks, even slight distances can result in notable disparities in air quality data attributable to emissions from vehicles. As a result, there is a growing demand for alternative methods that can gather air quality information in a more cost-effective and flexible manner, while also providing detailed air quality predictions.

In our study, we introduce a hybrid approach that combines the deployment of multiple static sensors with IoT mobile sensors to achieve effective air quality monitoring. By leveraging both static and mobile sensors, we aim to obtain a comprehensive understanding of air quality conditions. Static sensors offer continuous data streams, providing a holistic view of the overall air quality. In contrast, mobile sensors contribute more precise and accurate data regarding specific areas, thereby reducing potential errors associated with static sensors.

In this paper, our focus is twofold. Firstly, we construct a prediction model that utilizes the collected data from both static and mobile sensors. This model enables us to generate rapid and up-to-date information about air quality in the vicinity of individuals. Secondly, we aim to deliver this information promptly to people, ensuring that they have access to real-time air quality updates. By integrating both static and mobile sensor data, our hybrid approach enhances the accuracy and timeliness of air quality information provided to individuals.

1) The model introduced a component that enables the extraction and loading of pollution data from fixed monitoring stations. This facilitates comprehensive integration and analysis of data, enhancing our understanding of the subject matter.

2) The model utilized artificial neural networks (ANN) to predict pollution levels near roads based on the available data. ANNs are recognized for their proficiency in predicting fuzzy data and effectively modeling dynamic systems. This prediction capability greatly enhances our comprehension of the dynamics of air quality in close proximity to roadways.

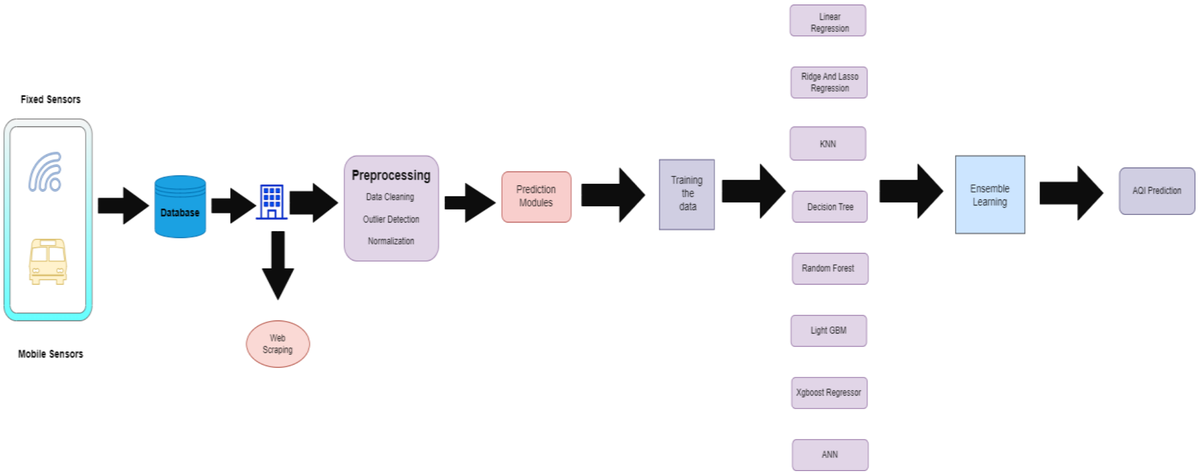
3) The proposed approach in this study entails the integration of both stationary and mobile IoT sensors to effectively forecast the data.

4) It utilized some machine learning algorithms also to predict pollution levels.

2. RELATED WORK

Various monitoring approaches have been proposed and employed to measure air quality. Zheng et al. [1] leveraged both public and private web services, alongside a compilation of public websites, to deliver up-to-the-minute meteorological information, weather predictions, and the data for the purpose of prediction. Alvarado et al. [2] employed small unmanned aerial vehicles to monitor PM10 dust particles and calculate emission rates. With the progress of technologies, IoT has demonstrated their efficacy in gathering real-time data on weather conditions, pollution levels, and traffic information, facilitating the analysis of air quality [3].

Apart from fixed sensors, air quality data collection has also been carried out using public transportation infrastructure such as buses [4]. One project [5] employed crowdsourcing, involving the participation of community members in data collection and the creation of an online system for monitoring air quality. These methods can be resource-intensive and time-taking. In our study, we aim to investigate the integration of stationary and mobile sensors to improve the accuracy of predictions, an area that has received limited research attention.



To meet the growing demand for real-time air quality information and enable citizens to respond promptly to pollution, various sensor networks have been developed to share real air quality data [6]. Garzon et al. [7] introduced a vigilant system that consistently detects regions where specific substance concentrations surpass predetermined thresholds and promptly notifies users upon entry. Maag et al. [8] presented a versatile, multi-contaminant surveillance platform utilizing budget-friendly wearable sensors. In contrast to the aforementioned approaches, our solution can effectively offer comparable capabilities to end-users with either a reduced sensor count or diminished computational requirements.

Regarding air quality prediction, regression models are commonly employed. Zhao [9] proposed a multivariate linear regression model for short-term PM2.5 prediction, incorporating other gaseous pollutants such as SO2, NO2, CO, and O3. Although these technologies offer flexibility in modeling non-linear relationships, they often lack insight into the underlying mechanisms and have not consistently outperformed classical regression models in various scenarios [10]. Additionally, research has focused on modeling and simulating pollutants for prediction purposes [11]. Considering the limited dataset available for our project, we opted for conventional regression models as baseline methods due to their computational efficiency while still yielding favorable results.[12] in this approach Dan Zhang and Simon S. Woo presented real time localized air quality prediction using mobile and fixed iot sensing.

3. DATASET

To construct a resilient prediction model, it is imperative to preprocess the obtained data as it often contains noise, missing values, and other inconsistencies. To achieve this, we employ the following techniques for data pre-processing:

A. Data Cleaning

This involves tackling challenges like missing data, anomalies, and discrepancies in the dataset. To address missing values, various techniques can be employed, such as imputation, mean substitution, or interpolation. Outliers can be identified and either removed or adjusted based on their impact on the overall data.

B. Data Normalization/Standardization

Diverse features within the dataset may possess dissimilar scales. To rectify this, normalizing or standardizing the data is employed to bring all features to a comparable scale, thus facilitating precise model training. Normalization techniques encompass Min-Max scaling or Z-score standardization.

$$x^* = \frac{x - min}{max - min}$$

Here, max represents the highest value observed in the entire dataset, while min denotes the lowest value. x* represents the normalized data value after the normalization process.

C. Feature Selection

Not all features in the dataset may contribute significantly to the prediction task. Feature selection techniques help identify the most relevant features, reducing complexity and improving model performance. Techniques such as correlation analysis, backward/forward feature selection, or regularization methods like Lasso can be employed.

D. Splitting Data into Training and Testing

Sets

The collected data must be partitioned into training and testing sets. The training set is utilized for model training, whereas the testing set serves the purpose of assessing the performance of the trained model.

Our split is typically 70-30, but it may vary based on the dataset size and characteristics. By employing these pre-processing techniques, we can enhance the quality and reliability of the data, ensuring that the prediction model is more robust and capable of providing accurate insights and forecasts

METHODOLOGY

The dataset used in this study was collected from two types of sensing devices. One set of devices was designed for stationary locations, while the other set was specifically designed for mobile vehicles. A total of six IoT sensor devices were utilized, with three deployed at fixed sites and the remaining three arranged on cars. The devices utilized in this study incorporate the following sensors:

1. Temperature and Humidity sensor

Temperature sensors measure the ambient temperature of the surrounding area. They can provide accurate readings in various temperature ranges and are commonly used in weather monitoring, HVAC systems, and industrial processes. Temperature sensors utilize different technologies such as thermocouples, resistance temperature detectors (RTDs), and thermistors to detect changes in temperature.

Humidity sensors, on the other hand, measure the amount of moisture or water vapor present in the air. They provide information about the relative humidity, which is expressed as a percentage. Thermal conductivity-based sensors are commonly used for measuring humidity.

These sensors provides measurements of minimum, maximum, and average temperature and pressure. These parameters serve as the features utilized in our analysis.

a) Deployment at static location

b) Deployment at a mobile vehicle

2. Micro Dust Sensor

The PM2.5 value has been measured using a micro dust sensor. This specialized sensor utilizes advanced technology, such as optical or laser-based methods, to accurately detect and quantify fine particulate matter with a diameter of 2.5 micrometers or smaller in the air. It enables precise monitoring of PM2.5 levels, which is essential for assessing air quality and understanding potential health risks associated with exposure to these fine particles.

3. GPS sensor

GPS sensors are incorporated into mobile IoT devices to determine the location and capture the Air Quality Index (AQI) value in conjunction with other features obtained from temperature and humidity sensors. This integration enables precise tracking of AQI levels, along with comprehensive environmental data collected through temperature and humidity measurement.

The dataset formed by data scraping from the Tutiempo weather site contains valuable weather-related information collected from the website's pages. Data scraping refers to the process of extracting data from websites by automatically navigating through web pages and extracting the desired content.

By scraping the Tutiempo weather site, a dataset can be formed with various features such as temperature, humidity, wind speed, precipitation, atmospheric pressure, and other meteorological parameters. It is important to note that when using data scraped from websites, proper attribution and compliance with any legal or ethical considerations are essential. Additionally, data quality and consistency should be ensured, as errors, inconsistencies in the scraped data could impact the reliability and accuracy of subsequent analyses or models.

Overall, the dataset formed by data scraping from the Tutiempo weather site provides a valuable resource for studying past weather conditions, understanding climate patterns, and conducting weather-related research and analysis of days with tornado, Number of days with hail. In addition to these readings, there are also corresponding measurements for PM2.5 (particulate matter 2.5) and the Air Quality Index (AQI).The output feature in this case is the Air Quality Index (AQI).

4. REGRESSORS

A. LINEAR REGRESSION

Linear regression is a fundamental and extensively employed statistical method for modeling the association between variables and Its objective is to determine the optimal line of best fit, which minimizes the disparity between the observed data points and the predicted values. Linear regression provides valuable insights into the strength, direction, and significance of the relationships between variables.

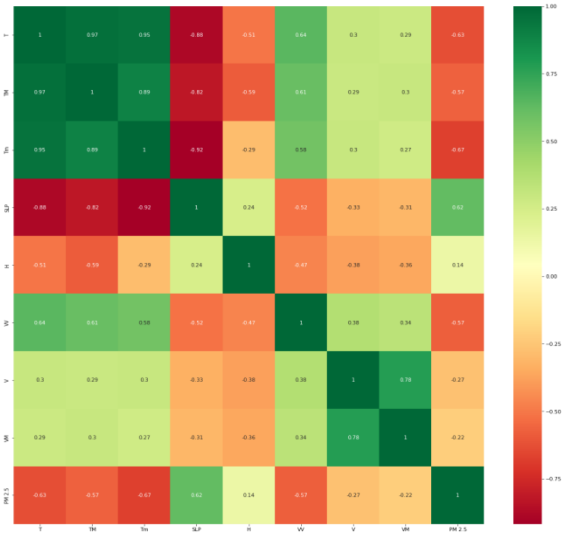


Figure: Heatmap

It allows us to estimate the impact of independent variables on the dependent variable, make predictions, and understand the overall trend in the data.

$$y = a + bx$$

$where$

$$b = \frac{n\Sigma xy - (\Sigma x)(\Sigma y)}{n\Sigma x^2 - (\Sigma x)^2}$$

$$a = \frac{\Sigma y - b(\Sigma x)}{n}$$

In simple linear regression, the variables x and y are used to establish the regression line. The slope of the line is denoted by b, while the y-intercept is represented by a. The cost function in simple linear regression captures the discrepancy between the predicted and actual values, aiming to minimize the overall error.

$$\sum_{i=1}^{M} (Actual(i) - Predicted(i))^2 =$$

$$\sum_{i=1}^{M} (Actual(i) - \sum_{j=0}^{p} w_j \times x_{ij})^2$$

B. RIDGE AND LASSO REGRESSION

Ridge and Lasso regression are two regularization techniques that extend the traditional linear regression model. They are widely used to address the issue of overfitting and improve the performance of linear regression in situations where there are a large number of features or multicollinearity.

Ridge regression introduces a regularization term to the linear regression objective function, which penalizes the model for large coefficients. This penalty term, controlled by a tuning parameter (lambda or alpha), shrinks the coefficients towards zero, reducing their magnitudes and effectively reducing the complexity of the model. Ridge regression helps to mitigate overfitting and can improve the generalization ability of the model.

$$\sum_{i=1}^{M} (Actual(i) - Predicted(i))^2 =$$

$$\sum_{i=1}^{M} (Actual(i) - \sum_{j=0}^{p} w_j \times x_{ij})^2 + \lambda \sum_{j=0}^{p} w_j{}^2$$

Lasso regression, on the other hand, not only adds a regularization term but also uses a different penalty term called the L1 norm. This penalty term promotes sparsity by forcing some coefficients to become exactly zero. This property of Lasso regression allows it to perform feature selection by automatically identifying and excluding irrelevant or redundant features from the model.

$$\sum_{i=1}^{M} (Actual(i) - Predicted(i))^2 =$$

$$\sum_{i=1}^{M} (Actual(i) - \sum_{j=0}^{p} w_j \times x_{ij})^2 + \lambda \sum_{j=0}^{p} |w_j|$$

Both Ridge and Lasso regression provide a balance between model simplicity and predictive accuracy. The selection between the two methods relies on the particular demands and specifications of the given problem. Ridge regression is suitable when all features are expected to contribute to the prediction, while Lasso regression is preferred when there is a need for feature selection and a desire to have a more interpretable model.Overall, Ridge and Lasso regression are valuable tools in the realm of regression analysis, offering ways to improve model performance, handle multicollinearity, and select relevant features, ultimately leading to more robust and interpretable models.

C. DECISION TREE

The decision tree regressor is a powerful and intuitive algorithm used for regression tasks. The algorithm operates by recursively dividing the data into smaller subsets using feature conditions, resulting in a hierarchical structure resembling a tree. Each internal node in this decision tree corresponds to a feature along with its associated condition, dictating the process of splitting the data. The leaves of the tree represent the predicted values for the target variable. During the training process, the algorithm determines the optimal splitting points based on criteria such as minimizing the mean squared error or maximizing the variance reduction. Decision tree regressors are versatile and interpretable models that can effectively capture complex relationships in the data. By understanding the feature conditions and tree structure, valuable insights can be gained, making them valuable tools in regression analysis.

D. RANDOM FOREST REGRESSOR

The Random Forest Regressor is a potent ensemble learning technique that integrates decision trees and bootstrap aggregating (bagging) principles to construct a robust and precise regression model. The algorithm generates multiple decision trees by randomly selecting subsets of the training data and features. Each tree is trained on distinct subsets of the data, resulting in diverse sets of predictions. During the prediction phase, the final output is obtained by averaging the predictions from all individual trees, thereby improving the reliability and accuracy of the forecast.

This algorithm offers several advantages. First, it can handle large and complex datasets with high-dimensional feature spaces. Second, it is robust against overfitting due to the randomness in selecting subsets and features. Thirdly, the algorithm adeptly manages missing values and outliers, showcasing its robustness in handling data imperfections. Fourthly, it offers valuable insights into feature importance, enabling feature selection and facilitating model interpretation.

The Random Forest Regressor exhibits remarkable versatility and enjoys extensive utilization across diverse domains, including finance, healthcare, and environmental sciences. It excels in capturing complex relationships and nonlinear interactions between features, making it suitable for both small and large-scale regression problems.In summary, the Random Forest Regressor is a powerful ensemble learning algorithm that leverages the strengths of decision trees and aggregation techniques. It provides accurate predictions, handles complex datasets, and offers interpretability, making it a popular choice for regression tasks.

$$\underline{m}\,(x) = \frac{1}{F}\sum_{f=1}^{n} \quad m(x;\Theta f)$$

In the given expression, m(x; θk) represents a set of tree predictors, indexed by f = 1, . . .F. The variable θf denotes a random vector, defining the characteristics of the fth tree in the Random Forest. The input variable x signifies the observed input data, which are assumed to be independently drawn from the joint distribution (x, y).

E. LIGHT GBM

Light GBM (Light Gradient Boosting Machine) is a fast and efficient gradient boosting framework for supervised machine learning. It uses gradient-based learning algorithms, leaf-wise tree growth, and exclusive feature bundling to achieve high performance. Light GBM is designed to handle large-scale datasets and offers customizable parameters for optimal model tuning. It supports parallel and GPU learning, making it suitable for computationally intensive tasks. With its ability to handle categorical features and its speed, accuracy, and scalability, Light GBM is a popular choice in various domains for tasks such as classification, regression, and ranking.

$$f(x) = \sum_{n=1}^{N} \quad \mu_n \quad k(X; b_n)$$

The function k(X; b_n) represents the basis function, which is typically selected as a simplified representation of x with parameters b = {b1, b2, ...}, while μ_m denotes the expansion coefficients with n = 1, 2, ..., N. In our model, we employ regression trees as the fundamental function.

F. XGBOOST REGRESSOR

The XGBoost Regressor represents a cutting-edge machine learning algorithm, renowned for its outstanding performance in diverse applications. It is known for its exceptional performance and efficiency in handling complex datasets. XGBoost Regressor combines the strengths of gradient boosting and regularization techniques to improve model accuracy and reduce overfitting. It sequentially adds weak learners (decision trees) to the ensemble, each correcting the errors made by the previous learners. This iterative process allows the model to learn complex relationships and make accurate predictions. One of the key advantages of XGBoost Regressor is its ability to handle various types of data, including numerical and categorical features. It automatically handles missing values by using a technique called "sparse aware" split finding. It also offers built-in support for handling imbalanced datasets through customizable weights and sampling strategies. Furthermore, XGBoost Regressor includes regularization terms that control the complexity of the model, preventing it from becoming overly complex and reducing the risk of overfitting. It provides hyperparameter tuning options, enabling fine-tuning of the model for optimal performance. XGBoost Regressor is a highly efficient and effective algorithm for regression tasks. Its gradient boosting framework, regularization techniques, and support for diverse data types make it a top choice for accurate predictions and handling complex datasets.

G. KNN

K-Nearest Neighbors (KNN) is a straightforward yet robust algorithm employed for classification and regression tasks. It functions on the principle of resemblance, wherein it forecasts the label or value of a novel data point by taking into account the labels or values of its closest neighbors in the training dataset. KNN calculates the distances between the new point and all other points in the training dataset, selecting the K closest neighbors. For classification, it assigns the majority class label among those neighbors, while for regression, it calculates the average value.

One of the key advantages of KNN is its simplicity and non-parametric nature, as it doesn't assume any specific data distribution. It can handle complex decision boundaries and remains effective even with noisy or sparse data. However, the performance of KNN can be influenced by the choice of K and the distance metric used for similarity measurement. Additionally, KNN can be computationally expensive for large datasets due to the need to calculate distances for each new data point.

K-Nearest Neighbors (KNN) is an intuitive and versatile algorithm that utilizes similarity to make predictions. While it has certain limitations, it remains widely used in scenarios where interpretability and flexibility are crucial.

ENSEMBLE TECHNIQUES

Ensemble techniques in machine learning encompass the amalgamation of predictions from multiple individual models to produce predictions that are more accurate and robust. These methods leverage the collective intelligence of diverse models to enhance predictive performance and improve generalization across various scenarios. The key idea behind ensembles is that the collective intelligence of multiple models can often outperform a single model. Common ensemble methods include bagging, boosting, and stacking. Bagging creates diverse models by training them on random subsets of the data and averaging their predictions. Boosting iteratively builds models, focusing on instances that were previously mispredicted. Stacking is a technique in machine learning that aggregates the predictions of multiple models and employs them as inputs to a meta-model. Ensemble techniques help reduce bias, improve generalization, and enhance predictive performance in various machine learning tasks.

DEEP LEARNING

ANN

Artificial Neural Networks (ANNs) belong to a category of machine learning models that draw inspiration from the architecture and behavior of

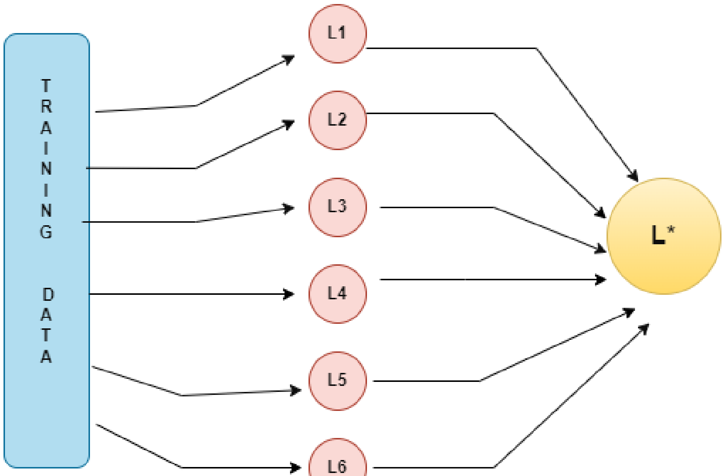
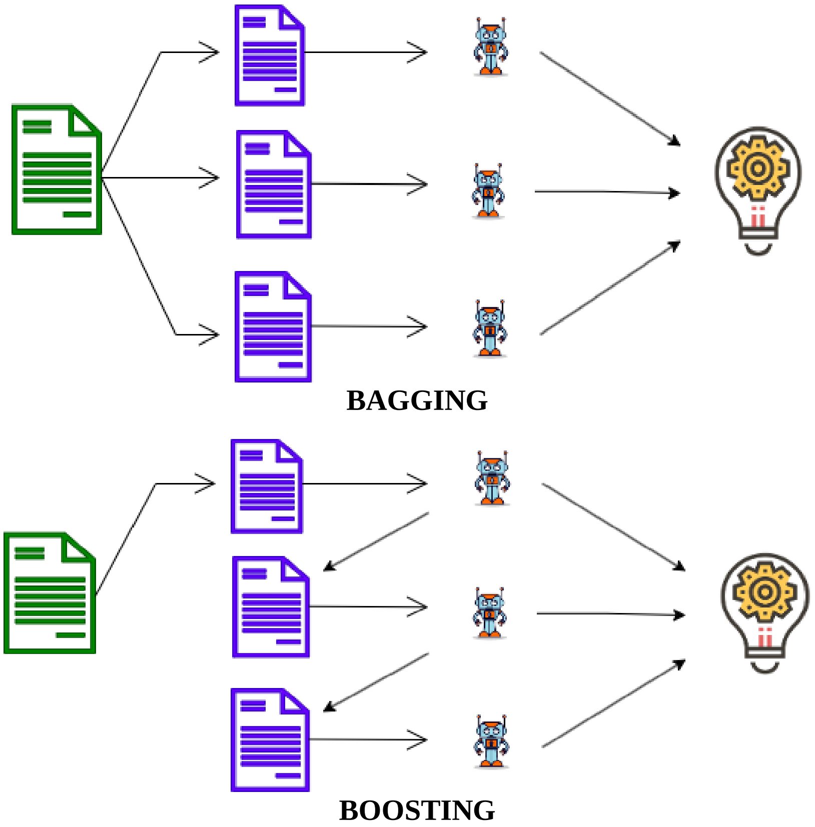


Figure: Ensemble Learning Model

biological neural networks. ANNs are composed of interconnected units known as neurons, arranged in a hierarchical manner through layers. Frequently, these structures comprise an input layer to receive data, one or more intermediary layers known as hidden layers, and a final output layer. Each neuron in the network receives input signals, computes the weighted sum of its inputs, applies a non-linear activation function to this sum, and then transmits the resulting output to the subsequent layer. The neural network's connections, symbolized by weights, undergo learning in the training phase, during which the network adapts its parameters to minimize the discrepancy between predicted and actual outputs.

ANNs are highly flexible and capable of modeling complex patterns and relationships in data. They excel in tasks such as classification, regression, and pattern recognition. Deep neural networks, which contain multiple hidden layers, have significantly improved the performance of ANNs and enabled breakthroughs in areas such as computer vision and natural language processing. Training ANNs often involves backpropagation, a process where the error is propagated backward through the network to adjust the weights. Various optimization algorithms, such as gradient descent, are employed to iteratively update the weights and improve the network's performance. Despite their power, ANNs can be computationally intensive and require large amounts of data for training. They are also prone to overfitting if not properly regularized and may lack interpretability due to their black-box nature.



ANNs have found applications in numerous domains, including image and speech recognition, recommender systems, financial forecasting, and healthcare diagnostics. They continue to advance the field of machine learning and artificial intelligence, contributing to groundbreaking research and practical applications. In summary, Artificial Neural Networks (ANNs) are robust computational models that draw inspiration from the intricate workings of biological neural networks. They are capable of learning complex patterns and have made significant contributions across various fields, with deep neural networks revolutionizing the capabilities of ANNs.

5. PERFORMANCE METRIC

Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are prevalent evaluation metrics used to assess the efficacy of regression models.

1. Mean Squared Error (MSE):

MSE (Mean Squared Error) assesses the average of the squared deviations between the predicted and actual values. It computes the average of the squared residuals, representing the differences between predicted and actual values. MSE gives higher weight to larger errors due to the squaring operation. A smaller MSE implies improved model performance, with a value of 0 indicating a flawless fit. Nonetheless, MSE is susceptible to the impact of outliers and can be influenced by the scale of the target variable.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \quad (Actual\,(i) - Predicted\,(i))$$

2. Coefficient of Determination (R²):

R² (R-squared) is a statistical metric that quantifies the goodness of fit of a regression model. It signifies the fraction of the variability in the dependent variable (y) that can be accounted for by the independent variable(s) included in the model. The R² value ranges from 0 to 1, with 1 denoting an ideal correspondence between the model and the data.

$$R^2 \; = \; 1 - \frac{\sum_{i=1}^{n} \quad (Actual(i) - Predicted(i))^2}{\sum_{i=1}^{n} \quad (Actual(i) - ActualAverage)^2}$$

3. Root Mean Squared Error (RMSE):

RMSE (Root Mean Squared Error) is derived by taking the square root of the MSE (Mean Squared Error) and is frequently favored over MSE due to its alignment with the unit of the target variable. It offers a gauge of the average magnitude of the residuals, which are the differences between predicted and actual values. RMSE serves as a valuable indicator for interpreting the model's efficacy concerning the target variable. Similar to MSE, a smaller RMSE signifies superior model performance, with an RMSE of 0 indicating a perfect fit.

$$RMSE = \sqrt{\sum_{i=1}^{n} \quad \frac{(Predicted(i) - Actual\,(i))^{\;\;2}}{n}}$$

4. Mean Absolute Error (MAE):

MAE (Mean Absolute Error) quantifies the average absolute deviation between the predicted and actual values. It assesses the average of the absolute residuals, which represent the absolute disparities between the predicted and actual values. In contrast to MSE (Mean Squared Error), MAE displays reduced sensitivity to outliers and offers a more interpretable metric for evaluating error. A lower MAE indicates better model performance, with 0 representing a perfect fit.

When comparing models or evaluating the performance of a regression model, it is common to consider these metrics to assess how well the model predicts the target variable. It is important to choose the most appropriate metric based on the specific context and requirements of the problem at hand.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \quad |Actual(i) - Predicted(i)|$$

6. RESULTS

When dealing with small datasets, classical models often yield satisfactory results. These models include linear regression, lasso, and ridge regression, which are employed to tackle the challenge of overfitting by regularizing the model parameters. Decision trees and random forests are subsequently introduced to enhance accuracy through their ability to capture complex relationships in the data. However, to further optimize performance, more advanced techniques are employed.

Gradient boosting, a popular ensemble method, is applied to iteratively improve the model's predictive ability by combining multiple weak learners into a strong predictor. Its implementation involves training a sequence of models, each one focused on minimizing the errors made by its predecessors. Light GBM, another boosting algorithm, is then introduced, which leverages a histogram-based approach to achieve faster training times and better scalability.

Finally, the XGBoost regressor, an advanced implementation of gradient boosting, is utilized. XGBoost combines the strengths of gradient boosting with additional enhancements, such as parallelization and regularization techniques, to achieve even higher accuracy and performance. Among all the classical models discussed, the XGBoost regressor consistently delivers the most optimal results.

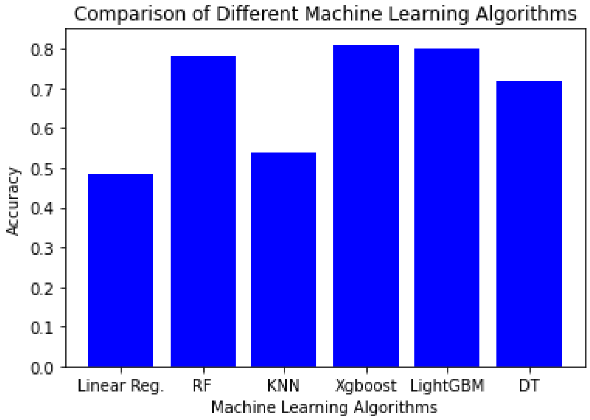
In summary, when working with small datasets, a progression of classical models is often employed, starting from linear regression and gradually incorporating more sophisticated techniques like decision trees and random forests. Eventually, advanced ensemble methods like gradient boosting, light GBM, and XGBoost regressor are utilized to achieve the best possible predictive performance.

As the dataset size increases, deep learning methods tend to exhibit improved performance, as measured by the R^2 value. This improvement can be attributed to the ability of deep learning models to effectively capture complex patterns and relationships within large amounts of data.

The deeper architectures and intricate network connections allow them to learn intricate representations and make more accurate predictions. However, it is worth noting that while deep learning models thrive with larger datasets, some classical models may experience a decline in performance. This phenomenon occurs due to the fundamental differences in the underlying principles of classical models and deep learning methods.

Classical models, such as linear regression, lasso, ridge regression, decision trees, and random forests, are typically designed to handle smaller datasets and rely on assumptions that may not hold in the context of large and high-dimensional data. The increased dataset size can lead to issues such as overfitting, limited capacity to capture intricate relationships, and reduced generalization ability.

Consequently, these models may struggle to adapt to the complexity and richness of information present in larger datasets.



On the other hand, deep learning models, with their highly flexible architectures and trainable parameters, can effectively leverage the increased dataset size to improve their performance. By employing multiple layers of interconnected neurons, these models can learn hierarchical representations of the data, automatically extracting relevant features and capturing intricate patterns that may be present within the expanded dataset.

When the dataset size increases, deep learning methods tend to excel due to their capacity to handle the complexity of large and high-dimensional data. Conversely, classical models may encounter difficulties, as their underlying assumptions and limitations become more apparent in the face of larger datasets. This underscores the need to choose the appropriate modeling approach based on the dataset characteristics and desired performance goals.

Regressors	MAE	MSE	RMSE	R^2
Linear Regression	44.83	3687	60.72	0.485
Ridge and Lasso	44.50	3627	60.23	0.50
KNN	34.97	2970	54.49	0.54
Decision Tree	22.09	1977	44.46	0.72
Random Forest	24.22	1551	39.39	0.78
Light GBM	20.06	1426	37.60	0.80
Xgboost	19.02	1355	36.81	0.81
Light GBM + Xgboost	23.65	1675	38.787	0.759
Xgboost+ Random Forest	19.90	1420	37.45	0.804
ANN	38.53	3314	57.57	0.52

The initial training begins with the Linear Regression, which is chosen due to its ability to provide relative feature importance. By using the Linear Regression,we can gain insights into the significance of different features in the dataset. After this, many other algorithms have been used such as, XGBoost algorithm, KNN and Artificial Neural Network (ANN) are applied for predicting AQI (Air Quality Index).

These models are chosen due to their effectiveness in capturing complex patterns and relationships in the data. The hyperparameters for the models were carefully selected using the grid search approach. In the case of the deep Artificial Neural Network (ANN), we conducted experiments to determine the optimal architecture. Based on our findings, a configuration with three hidden layers, consisting of 256 nodes each, was deemed the most suitable. The Rectified Linear Unit (ReLU) activation function was used for all hidden layers, while the output layer employed a linear activation function since the prediction was expected to be a numeric value. During training, we utilized the Adam optimization algorithm for model optimization. A training batch size of 100 epochs, indicating the number of times the entire training dataset was iterated, were chosen to strike a balance between computational efficiency and model performance. These choices were made through careful experimentation and consideration of the dataset characteristics to ensure effective training and accurate predictions. To leverage the strengths of multiple algorithms, we selected the three best-performing models, namely Random Forest (RF), LightGBM and XGBoost. These models were chosen based on their individual performance and ability to capture different aspects of the data. We employed two ensemble learning techniques, namely voting ensemble and stacking ensemble, to combine the predictions of these models. In the voting ensemble, the predictions from each individual model were averaged to produce the final prediction. This approach benefits from the collective wisdom of the models and can improve overall prediction accuracy. In the stacking ensemble, the predictions from the individual models were used as inputs to a final estimator, which was trained using cross-validation. This final estimator learned to make predictions based on the patterns and relationships identified by the base models, leading to potentially enhanced performance. The obtained test scores are summarized in Table for RF, LightGBM and XGboost have similar performance while ANN is slightly worse.Furthermore, ensemble learning models demonstrated improved performance compared to some individual machine learning models. Upon reviewing the existing literature, it is evident that the results obtained in this work surpass those reported in previous studies that assessed similar evaluation metrics.In contrast to the reference paper, our study incorporated additional algorithms such as XGBoost and ensemble methods, which outperformed the gradient boosting and random forest algorithms mentioned in the reference. Moreover, while the reference paper utilized SVM, we found that it performed poorly on our dataset. Consequently, we explored alternative algorithms including linear regression, ridge regression, lasso regression, and K-nearest neighbors (KNN), all of which demonstrated superior performance compared to SVM. In our experimentation, we observed that as the size of the dataset increased, the performance of our machine learning models deteriorated. However, we found that artificial neural networks (ANN) exhibited robust performance even with larger datasets. Despite the decline in performance of other machine learning models, the ANN consistently demonstrated good performance and maintained its effectiveness in handling larger datasets.

7. RESULT AND DISCUSSION

Modeling and simulation of ML algorithms discussed in previous sections are implemented and results are compared in this section. From the analysis, *XGBoost* proved to be the most effective regressor, outperforming others across key metrics. Its performance highlights its superiority in handling complex datasets. Here's a comparison of models with notable observations:

1. *XGBoost vs. Linear Regression*:

- XGBoost performed significantly better, reducing errors by approximately *57%* in MAE and improving \(R^2 \) by *67%*, making it a far more accurate choice for predictions.

2. *XGBoost vs. Random Forest*:

- XGBoost demonstrated a *21% lower MAE* and a *7% improvement in RMSE*, indicating its stronger predictive power and lower error margin.

3. *XGBoost vs. LightGBM*:

- While LightGBM was a strong competitor, XGBoost was *5% better in MAE* and achieved a slightly higher \(R^2 \) score, showcasing its edge in overall accuracy.

4. *Random Forest vs. Decision Tree*:

- Random Forest was *10% better in MAE* and achieved a *5% higher \(R^2 \)* score, proving the advantage of ensemble learning over standalone trees.

5. *KNN vs. Linear Regression*:

- KNN reduced errors by *22% in MAE* and improved \(R^2 \) by *11%*, showing its ability to capture non-linear relationships better than Linear Regression.

6. *XGBoost + Random Forest vs. Standalone Models*:

- This ensemble was *18% better than Random Forest in MAE* but slightly less effective than XGBoost alone, suggesting that hybrid models can help but might not always outperform the best standalone approach.

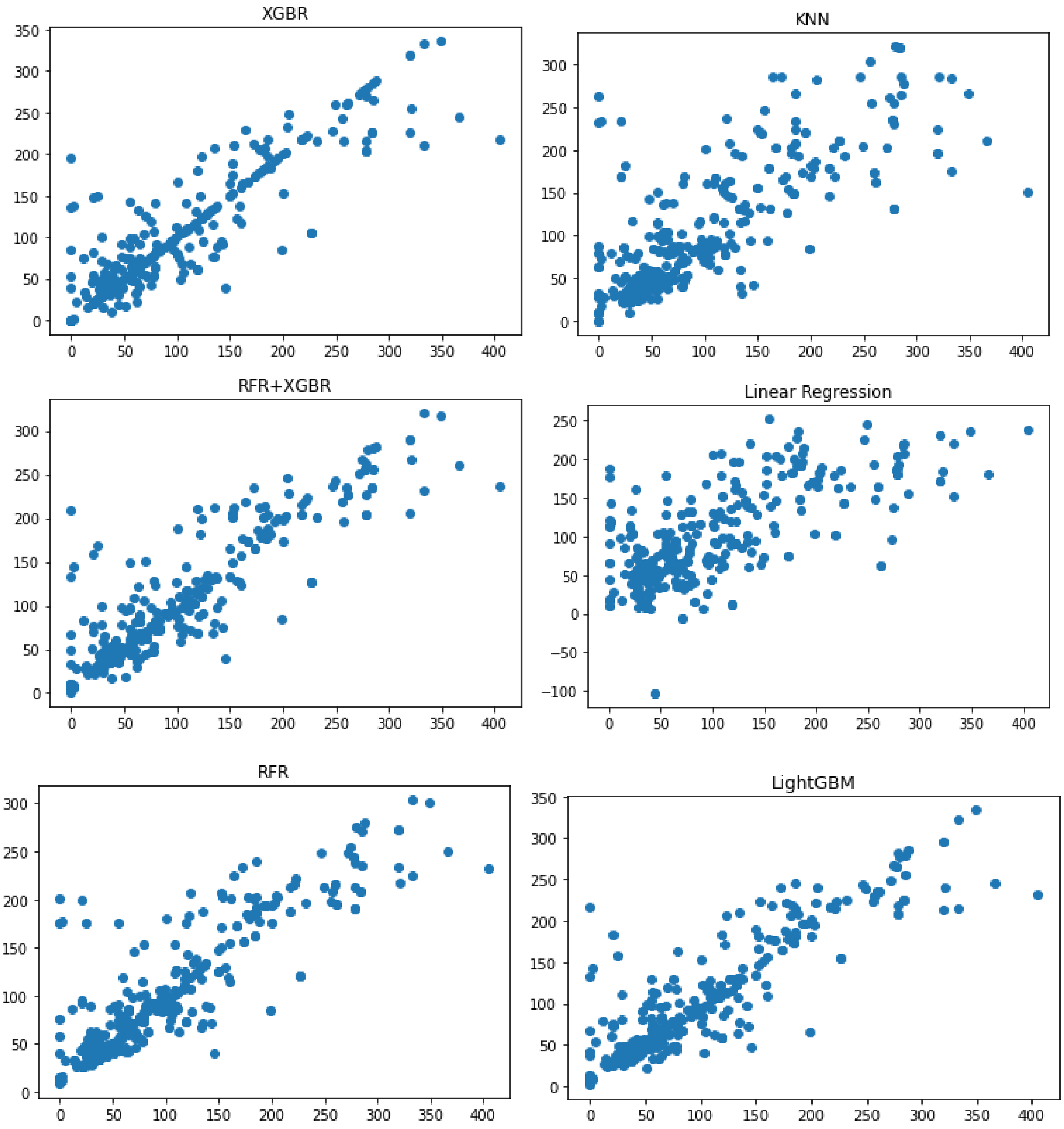
7. *Artificial Neural Networks (ANN)*:

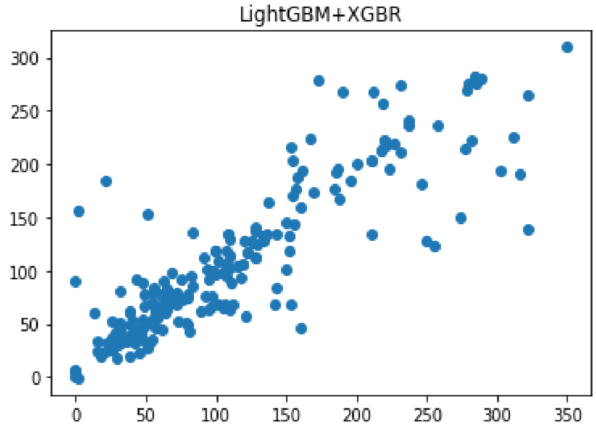
- ANN’s performance was relatively underwhelming, with *38% higher errors* compared to XGBoost and a lower \(R^2 \) score, suggesting it struggled with the dataset compared to tree-based models.

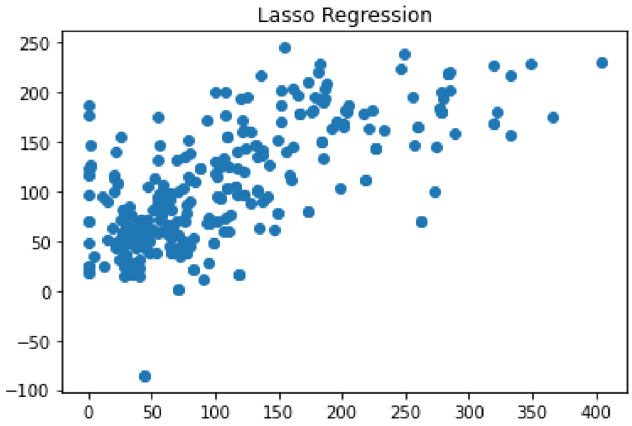
Final Recommendation

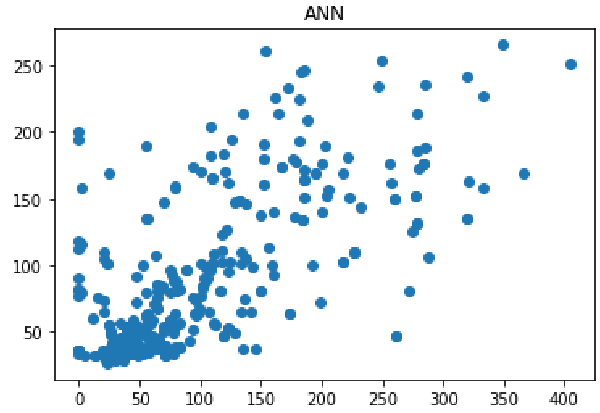
XGBoost is the recommended model for its consistent accuracy and efficiency. It achieved the lowest error rates (MAE: 19.02, RMSE: 36.81) and the highest \(R^2 \) score (0.81), proving to be the most reliable choice for this regression task. However, LightGBM remains a close second and could be considered in scenarios where computational efficiency is a priority.

Figure Scatterplot of various algorithms below:









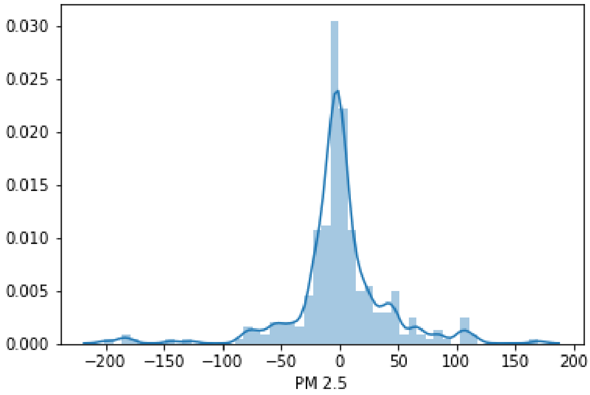
This aerial perspective allows for mapping air quality across various altitudes and locations, providing a more detailed and comprehensive assessment of air pollution levels.

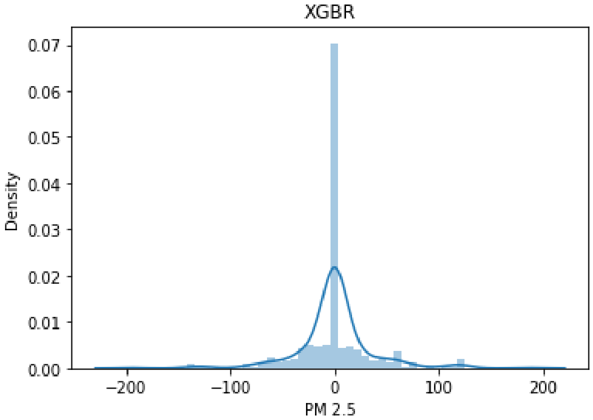
Expanding the scope of the project, another potential avenue is the development of sensor networks specifically tailored for gas source detection in lithium-ion batteries. By leveraging sensor technology, it is possible to detect and monitor gas emissions from batteries, contributing to the improvement of battery safety and performance.

These proposed directions not only offer increased accuracy in air quality prediction but also have significant real-world impact. The integration of IoT, hybrid data approaches, UAVs, and specialized sensor networks has the potential to revolutionize air quality monitoring, providing valuable insights for decision-makers, researchers, and the public.

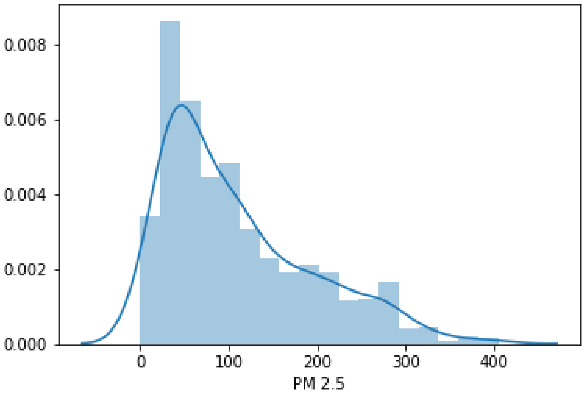
Figure Distplot of various algorithms are as follows:

1.) RFR

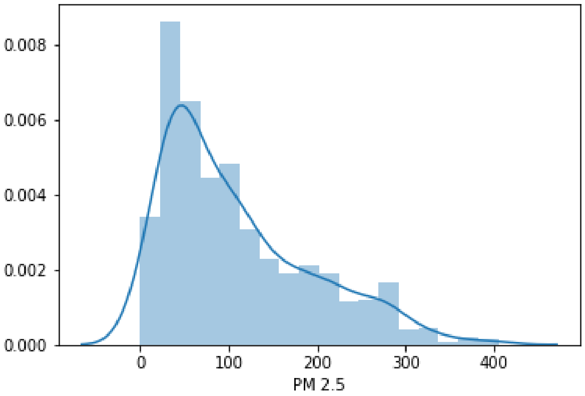




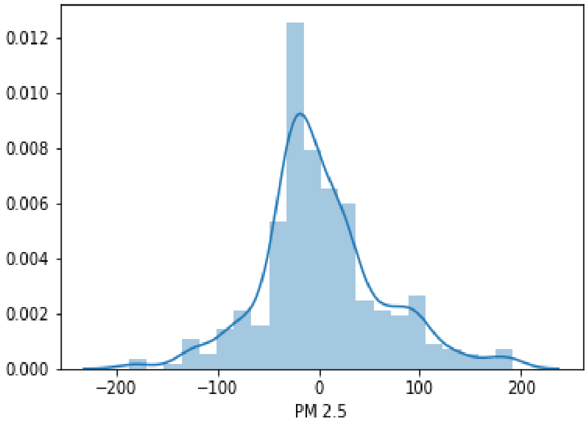
3.) ANN



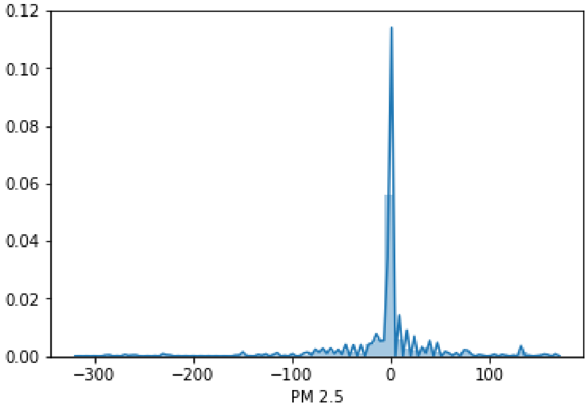
4.) LINEAR REGRESSION



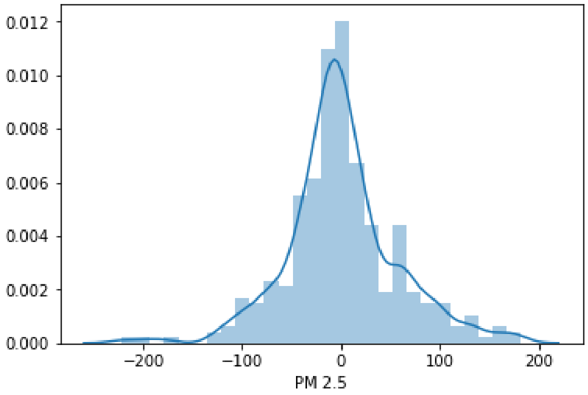
5.) LASSO



6.) KNN



7.) DT



8. CONCLUSION

In conclusion, our work highlights the efficacy of several machine learning algorithms in air quality prediction. Notably, algorithms such as Random Forest, Gradient Boosting, and XGBoost exhibit excellent performance in this context. Additionally, the combination of ensemble techniques, such as Random Forest + XGBoost and Gradient Boosting + XGBoost, further enhances predictive accuracy.

However, it is worth noting that our analysis revealed that Artificial Neural Networks (ANN) did not perform as well as anticipated. This observation can be attributed to the limited size of our dataset.

Neural networks typically require a substantial amount of data to generalize effectively and capture complex relationships accurately. In our case, the relatively small dataset may have hindered the ANN's ability to learn and generalize from the available information. Despite the suboptimal performance of ANN in our study, the success of other machine learning algorithms demonstrates their robustness and effectiveness in air quality prediction tasks. As we acquire a larger dataset in the future, it would be worthwhile to revisit the application of neural networks and assess their performance under improved data conditions. This would enable a more comprehensive evaluation and comparison of different algorithms.

Ultimately, our findings underscore the importance of dataset size in achieving optimal performance with machine learning techniques. Additionally, the successful combination of ensemble techniques highlights the potential for leveraging the strengths of multiple algorithms to enhance predictive accuracy.

This knowledge can inform future research and contribute to the development of more accurate and reliable air quality prediction models.

9. FUTURE WORK

In future work, improving the accuracy of air quality prediction can be achieved by incorporating a larger dataset. Real-time data collection through IoT offers numerous advantages. By continuously monitoring air quality in real-time, it becomes possible to capture fluctuations and variations that occur throughout the day. This dynamic data provides a more comprehensive understanding of air pollution patterns, enabling timely interventions and decision-making.Additionally, the utilization of Unmanned Aerial Vehicles (UAVs) present an exciting opportunity for advancing air pollution monitoring systems. By equipping UAVs with appropriate sensors, it becomes feasible to create a three-dimensional (3D) air pollution monitoring system.

10. REFERENCES

1. Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, ‘‘Forecasting fine-grained air quality based on big data,’’ in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD, 2015, pp. 2267–2276.

2. M. Alvarado, F. Gonzalez, P. Erskine, D. Cliff, and D. Heuff, ‘‘A methodology to monitor airborne PM10 dust particles using a small unmanned aerial vehicle,’’ Sensors, vol. 17, no. 2, p. 343, 2017.

3. I. Kok, M. U. Simsek, and S. Ozdemir, ‘‘A deep learning model for air quality prediction in smart cities,’’ in Proc. IEEE Int. Conf. Big Data (Big Data), Dec. 2017, pp. 1983–1990.

4. S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, ‘‘Realtime air quality monitoring through mobile sensing in metropolitan areas,’’ in Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput. UrbComp, 2013, p. 15.

5. Y.-C. Hsu, P. Dille, J. Cross, B. Dias, R. Sargent, and I. Nourbakhsh, ‘‘Community-empowered air quality monitoring system,’’ in Proc. CHI Conf. Hum. Factors Comput. Syst., May 2017, pp. 1607–1619.

6. A. C. Rai, P. Kumar, F. Pilla, A. N. Skouloudis, S. Di Sabatino, C. Ratti, A. Yasar, and D. Rickerby, ‘‘End-user perspective of low-cost sensors for outdoor air pollution monitoring,’’ Sci. Total Environ., vols. 607–608, pp. 691–705, Dec. 2017.

7. S. R. Garzon, S. Walther, S. Pang, B. Deva, and A. Küpper, ‘‘Urban air pollution alert service for smart cities,’’ in Proc. 8th Int. Conf. Internet Things, Oct. 2018, p. 8.

8. B. Maag, Z. Zhou, and L. Thiele, ‘‘W-Air: Enabling personal air pollution monitoring on wearables,’’ Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol., vol. 2, no. 1, p. 24, 2018.

9. R. Zhao, X. Gu, B. Xue, J. Zhang, and W. Ren, ‘‘Short period PM2.5 prediction based on multivariate linear regression model,’’ PLoS ONE, vol. 13, no. 7, 2018, Art. no. e0201011.

10. K. P. Moustris, P. T. Nastos, I. K. Larissi, and A. G. Paliatsos, ‘‘Application of multiple linear regression models and artificial neural networks on the surface ozone forecast in the greater athens area, greece,’’ Adv. Meteorol., vol. 2012, pp. 1–8, Jul. 2012.

11. S. Fotouhi, M. H. Shirali-Shahreza, and A. Mohammadpour, ‘‘Concentration prediction of air pollutants in tehran,’’ in Proc. Int. Conf. Smart Cities Internet Things SCIOT, 2018, pp. 1–7.

12. Dan Zhang and Simon S. Woo, “Real Time Localized Air Quality Monitoring and Prediction Through Mobile and Fixed IoT Sensing Network”.