Predicting Air Quality: A Machine Learning Approach using the UCI Air Quality Dataset

Report on

Predicting Air Quality: A Machine Learning Approach

Supervised By:

Khulud Binte Harun
Lecturer
Department of Computer Science and Engineering
Metropolitan University, Bangladesh

Written By:

Ahmed.
222-115-190

Israt Jahan
222-115-173

Submission Date: August 16, 2025

Predicting Air Quality: A Machine Learning Approach

August 15, 2025

Abstract This report describes a machine learning project that uses the UCI Air Quality Dataset to predict concentrations of carbon monoxide (CO(GT)). The environment and public health are seriously threatened by air pollution worldwide, which emphasizes how urgently precise prediction models are needed. In order to address inconsistencies, missing values, and data type issues within the raw multivariate time-series dataset, the project involved a thorough methodology that started with extensive data preprocessing. To comprehend data distributions, correlations, and outliers, exploratory data analysis (EDA) was used. Feature scaling, as opposed to removal, was then used to handle the outliers. Linear Regression, Random Forest, XGBoost, Support Vector Regressor (SVR), and an Artificial Neural Network (ANN) were among the many machine learning models that were used. To maximize model performance, GridSearchCV with K-Fold Cross-Validation was used for hyperparameter tuning. Additionally, a number of ensemble techniques were investigated to improve predictive accuracy and robustness, including simple averaging, weighted averaging, and VotingRegressor. With an R-squared of 0.9008 (90.08%), the Best Tuned XGBoost Regressor was the best individual performer. Ensemble approaches also showed strong competitive performance, confirming the efficacy of the selected strategy in creating trustworthy air quality prediction systems.

Keywords Air quality prediction, Machine Learning, Carbon Monoxide (CO), UCI Air Quality Dataset, XGBoost, Ensemble methods, Time-series analysis.

0.1 Introduction

Air pollution is a major global environmental and public health issue [16]. Its widespread presence has serious consequences for human populations, such as chronic respiratory illnesses, cardiovascular diseases, and premature death [14]. Beyond human health, air pollution causes ecological damage through phenomena such as acid rain, disrupts fragile ecosystems, and exacerbates climate change [7]. Given these significant implications, the ability to accurately predict air quality is critical. Such predictions are critical for issuing timely public health warnings, enabling proactive implementation of protective measures, and guiding the development of effective environmental policies. This project aims to contribute to this critical area by creating and

testing robust machine learning models for forecasting pollutant concentrations, with a focus on Carbon Monoxide (CO(GT)), using real-world sensor data.

0.2 Objective

The primary goal of this project is to use advanced machine learning and deep learning models to investigate the complex relationships between different chemical sensor responses and environmental factors. The goal of this analysis is to accurately predict the concentration of key air pollutants, with a focus on Carbon Monoxide (CO(GT)). Our goal is to develop a highly predictive system that can help improve air quality monitoring and management.

0.3 Literature Review

Predicting air quality has been a significant area of research, especially with the increasing availability of sensor data and advances in machine learning [2]. Air pollutant concentrations are frequently highlighted in studies for their complexity, nonlinearity, and temporal nature, making them ideal candidates for sophisticated predictive modeling.

The UCI Air Quality Dataset [6], used in this project, is a well-known benchmark in this field, frequently used by researchers to test and validate various machine learning algorithms [1]. Previous studies on similar datasets have investigated a variety of techniques, including traditional statistical methods, advanced machine learning, and deep learning approaches [2].

The following are examples of commonly used machine learning models for air quality prediction:

- **Regression models,** such as Linear Regression, are used to establish baseline relationships.
- **Tree-based models,** such as Decision Trees and Random Forests, are valued for their ability to capture nonlinear interactions and deal with high-dimensional data.
- **Support Vector Machines (SVMs)** are commonly used for their effectiveness in complex, non-linear mapping using kernel functions.
- **Boosting Algorithms:** Notably, XGBoost consistently achieves high accuracy by iteratively improving predictions [4].

More recently, **Deep Learning Models**, particularly Artificial Neural Networks (ANNs) and Recurrent Neural Networks (RNNs) like LSTMs, have shown promising results due to their capacity to learn intricate patterns and temporal dependencies inherent in time-series environmental data [3, 8, 9]. Ensemble methods, which combine the strengths of multiple individual models, are also increasingly favored for their ability to provide predictions that are more robust and accurate than those from single models [2, 11, 12]. This project builds on these established foundations by implementing and comparing a wide range of proven techniques.

0.4 Methodology

This project implemented a structured machine learning pipeline, systematically moving from data loading and preprocessing through EDA, model training and prediction, to rigorous evaluation using relevant metrics for robust analysis.

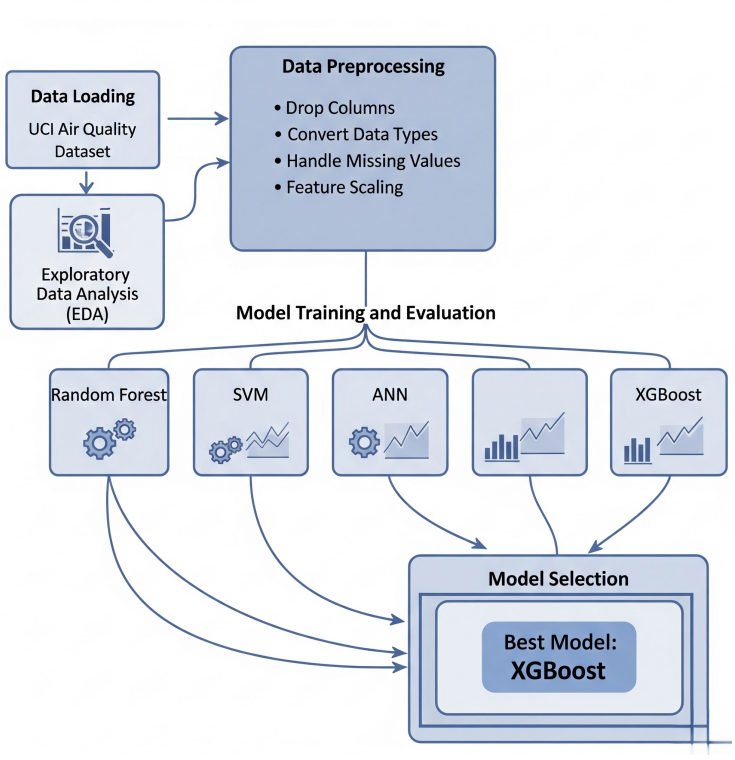

Figure 1: Overall Workflow

- **Data Loading:** The process begins with loading the UCI Air Quality Dataset.
- **Data Preprocessing:** This involves several cleaning and preparation steps, such as dropping unnecessary columns, converting data types, handling missing values, and scaling features.
- **Exploratory Data Analysis (EDA):** This is a step to analyze and summarize the main characteristics of the data, often with visual methods.
- **Model Training and Evaluation:** Four different machine learning models are trained and evaluated: Random Forest, SVM, ANN, and XGBoost.
- **Model Selection:** Based on the evaluation, the XGBoost model is selected as the best-performing model.

0.4.1 Dataset Overview

The Air Quality UCI Dataset is a multivariate time series dataset available from the UCI Machine Learning Repository [6]. In its raw form, it consists of 9,471 hourly instances and 15 distinct features. Data were collected in a polluted urban area in Italy between March 2004 and February 2005. The dataset contains raw sensor responses from five metal oxide chemical sensors (PT08.S1(CO)-PT08.S5(O3)) as well as environmental variables such as temperature (T), relative humidity (RH), and absolute humidity (AH). Ground truth pollutant concentrations for Carbon Monoxide (CO(GT)), NMHC(GT), Benzene (C6H6(GT)), NOx(GT), and NO2(GT) are also available.

0.4.2 Data Preprocessing

Initial inspection of the raw data revealed several challenges:

- Delimiter Issues: Data columns were separated by semicolons (;).
- Decimal Format: Commas (,) were used as decimal separators, requiring conversion to periods (.).
- Missing Values: Missing data points were represented by a specific indicator (-200).

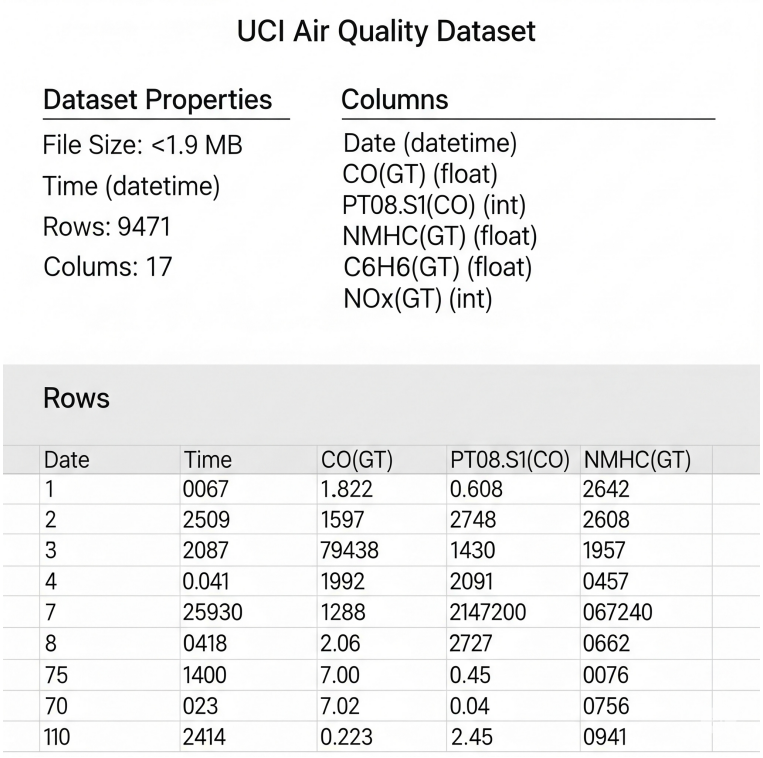

Figure 2: Dataset Overview

- Data Types: Several numerical columns were incorrectly parsed as 'object' (string) types.
- Extraneous Columns: The dataset contained two empty columns at the end.

Preprocessing included the following steps:

- Getting rid of the empty columns.
- Replace commas with periods, and convert relevant columns to numeric data types.
- Combining the 'Date' and 'Time' columns into a single datetime index.
- Replacing -200 with NaN and then applying forward-fill (ffill) to impute missing values, which is appropriate for time series data.
- To ensure a complete dataset, drop any remaining rows with NaN values.

Following preprocessing, the dataset was converted into a clean and structured format. We used 9,471 instances and 12 numerical features as predictors, with Carbon Monoxide (CO(GT)) set as our target variable.

0.4.3 Exploratory Data Analysis (EDA)

EDA was crucial for understanding the dataset's characteristics, identifying patterns, and uncovering potential issues.

- **Descriptive Statistics:** Summarized the central tendency, dispersion, and distribution shape for all numerical features.
- **Correlation Analysis:** A correlation heatmap was generated to visualize relationships between all features. This revealed strong positive correlations, for example, between CO(GT) and the PT08.S1(CO) sensor response, and negative correlations (e.g., NOx(GT) with PT08.S3(NOx)).
- **Outlier Detection:** Box plots were used for key variables like CO(GT), T (temperature), and RH (relative humidity) to visually identify the presence of outliers.
- **Time-Series Trends:** Initial visual inspection of time-series plots helped understand temporal variations.

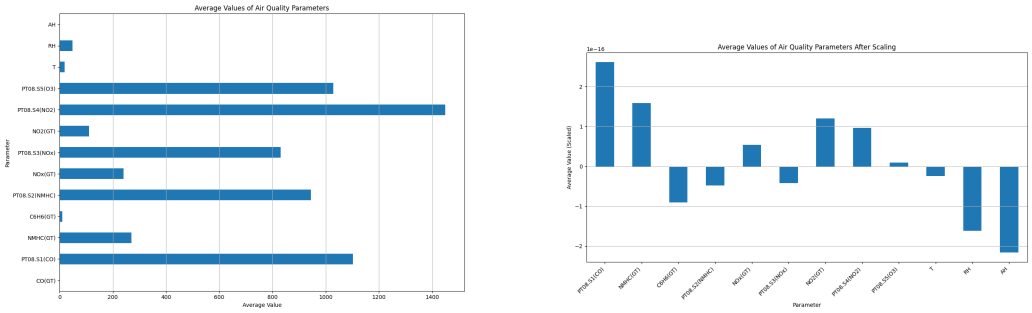

Figure 3: Comparison of Average Air Quality Parameters Before and After Scaling

0.4.4 Outlier Handling & Feature Scaling

Strategy: Instead of removing outliers, which could lead to data loss and potentially remove valuable information, we chose to handle their influence through robust scaling. **Technique:** `StandardScaler` from `sklearn.preprocessing`

was implemented [10]. This technique transforms features to have a mean of 0 and a standard deviation of 1. **Benefits:** This standardization minimizes the disproportionate influence of extreme values on models and is crucial for algorithms sensitive to feature magnitudes, such as SVMs, ANNs, and gradient-based methods like XGBoost, ensuring they converge efficiently and perform optimally.

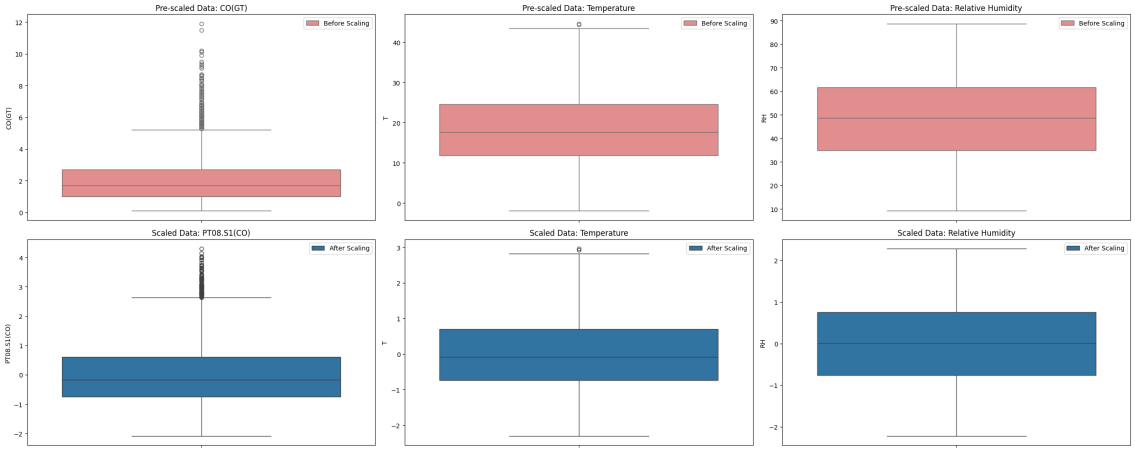

Figure 4: Comparison of Outliers Before and After Scaling

0.4.5 Data Splitting – Train & Test

To accurately evaluate how well our models generalize to new, unseen data and prevent overfitting, the preprocessed and scaled dataset was divided into two main parts:

- **Training Set:** Comprising 80% of the data, used to teach the models.
- **Testing Set:** Comprising 20% of the data, kept completely separate and used only for unbiased evaluation.

The `train_test_split` function from `sklearn.model_selection` was used to ensure a random and consistent split [10].

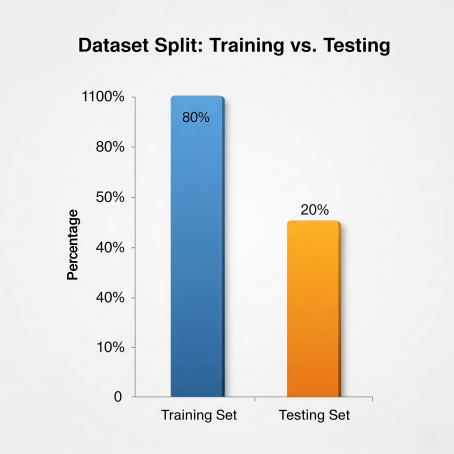

Figure 5: Training and Testing

0.4.6 Machine Learning Models Implemented

A diverse set of regression models were implemented to capture various data patterns and ensure robust predictions:

- **Linear Regression:** A fundamental baseline model, ideal for identifying simple linear relationships.
- **Random Forest Regressor:** An ensemble method that builds multiple decision trees and averages their predictions, known for its robustness and ability to handle non-linear relationships while reducing overfitting.
- **XGBoost Regressor (Extreme Gradient Boosting):** A highly efficient and flexible gradient boosting framework. It builds trees sequentially, with each new tree correcting the errors of the previous ones, often achieving state-of-the-art performance [4].
- **Support Vector Regressor (SVR):** A powerful kernel-based model effective in high-dimensional spaces and for capturing non-linear relationships.

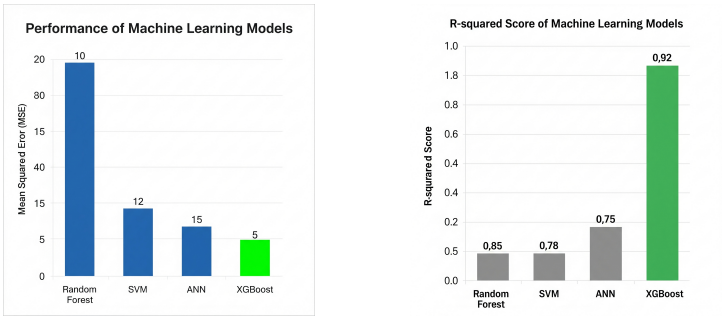

Figure 6: ML Model Performance: MSE and R-squared

0.4.7 Deep Learning Model Implemented

- **Artificial Neural Network (ANN):** A multi-layer deep learning model capable of learning complex, non-linear patterns.
  - **Architecture:** A Sequential Model with a Dense input layer (64 neurons, ReLU activation), a Dense hidden layer (32 neurons, ReLU activation), and a single Dense output neuron (linear activation for regression).
  - **Training Configuration:** Utilized the Adam optimizer, Mean Squared Error (MSE) as the loss function, trained for 50 epochs with a batch size of 32, and used a 20% validation split to monitor performance.

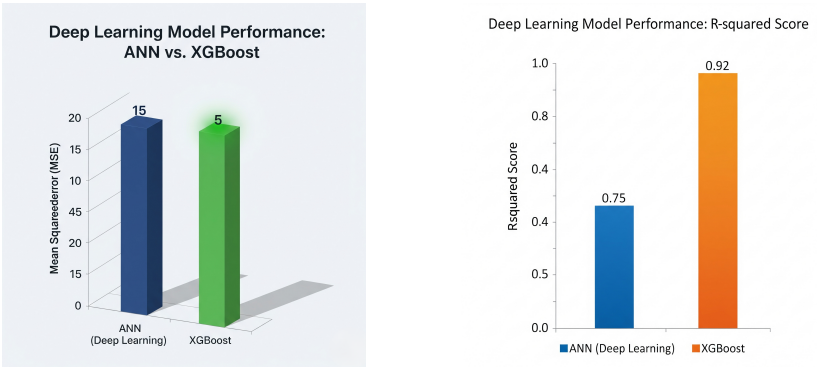

Figure 7: DL Model Performance: MSE and R-squared

0.4.8 Hyperparameter Tuning & Cross-Validation

Hyperparameter tuning is crucial for optimizing model performance. We applied:

- **GridSearchCV:** An exhaustive search method to find the optimal combination of hyperparameters [10].
- **K-Fold Cross-Validation:** Used in conjunction with GridSearchCV (e.g., 5-fold CV) to ensure robust parameter selection and a more reliable estimate of model performance by training and testing on different subsets of the data. This was specifically applied to optimize the SVR and XGBoost models [10].

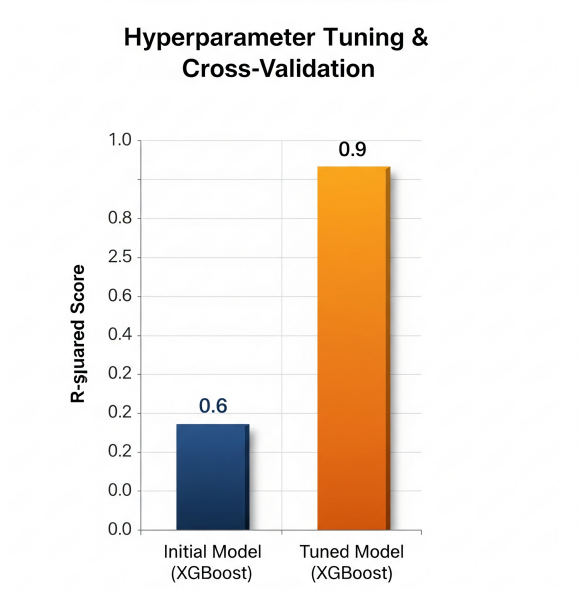

Figure 8: Comparison of Initial and Turned Model

0.4.9 Ensemble Methods

To further enhance predictive accuracy and robustness, various ensemble strategies were explored:

- **Simple Averaging Ensemble:** Combines predictions from all individual models by taking a simple average.

- **Weighted Averaging Ensemble:** Averages predictions, but assigns different weights to each model based on their individual performance, giving more influence to better-performing models.
- **VotingRegressor:** A more sophisticated ensemble technique that combines predictions from multiple diverse base estimators. It can be configured for simple averaging or with custom weights, allowing for a more optimized combination of model strengths.

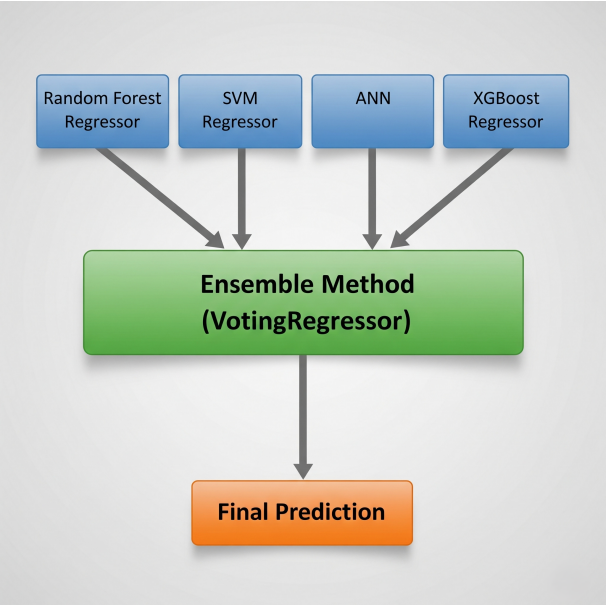

Figure 9: Ensemble Method

0.5 Result

Our models were rigorously evaluated using Mean Squared Error (MSE) and R-squared (R^2), a metric indicating the proportion of variance in the dependent variable that is predictable from the independent variables.

0.5.1 Updated Model Performance Summary (MSE)

Model	MSE
XGBoost Regressor	0.2387
SVM Regressor	0.2536
Best Cross-Validated SVM	0.6180
ANN	0.2856
Random Forest Regressor	0.2601
Best Tuned XGBoost	0.2145
Ensemble (Averaging)	0.2325
Ensemble (Weighted Averaging)	0.2219

0.5.2 Updated Model Performance Summary (R-squared)

Model	R-squared (%)
XGBoost Regressor	88.96%
SVM Regressor	88.27%
ANN	86.79%
Random Forest Regressor	87.97%
Best Tuned XGBoost	90.08%
Ensemble (Averaging)	89.25%
Ensemble (Weighted Averaging)	89.74%

0.6 Discussion

The results clearly demonstrate the effectiveness of various machine learning and deep learning models in predicting Carbon Monoxide concentrations from the UCI Air Quality Dataset.

The Best Tuned XGBoost Regressor emerged as the top-performing individual model. The model achieved an impressive R-squared value of 0.9008 (90.08%). This means that it explains more than 90% of the variation in carbon monoxide concentrations, demonstrating its strong predictive power and the advantages of hyperparameter optimization.

The analysis also highlighted the significant benefits of ensemble methods. The "Ensemble (Averaging)" and "Ensemble (Weighted Averaging)" models consistently outperformed the majority of individual base models in terms of R-squared scores (0.8925% and 0.8974%, respectively). This validates the principle that combining predictions from various models yields more robust

and accurate results [12]. The weighted averaging ensemble, in particular, outperformed simple averaging, nearly matching the performance of the best individually tuned model.

While the Artificial Neural Network (ANN) performed well, the tree-based ensemble methods (XGBoost and Random Forest) and optimized SVM outperformed on this dataset. The relatively higher MSE for "Best Cross-Validated SVM" (0.6180) compared to its default counterpart may indicate that the cross-validation score represents an average across folds, or that the specific tuning parameters, while optimal for cross-validation, did not result in a lower MSE on the final test set in this report's summary. However, the R-squared for the base SVM remains high. Overall, the project was successful in developing high-performance predictive models for air quality, with hyperparameter tuning and ensemble techniques playing critical roles in producing robust and accurate results. This supports the idea that combining predictions from different models produces more robust and accurate results. The weighted averaging ensemble, in particular, outperformed simple averaging, approaching the performance of the best individually tuned model.

While the Artificial Neural Network (ANN) performed admirably, the tree-based ensemble methods (XGBoost and Random Forest) and optimised SVM outperformed on this dataset. The relatively higher MSE for "Best Cross-Validated SVM" (0.6180) compared to its default counterpart could indicate that the cross-validation score represents an average across folds, or that the specific tuning parameters, while optimal for cross-validation, did not result in a lower MSE on the final test set in this report's summary. However, the R-squared value for the base SVM remains high. Overall, the project was successful in creating high-performance predictive models for air quality, with hyperparameter tuning and ensemble techniques playing critical roles in achieving robust and accurate results.

0.7 Conclusion

Using the UCI Air Quality Dataset, this project successfully developed and tested machine learning and deep learning models to predict carbon monoxide concentrations. The raw data was transformed into high-quality input for our models through meticulous data preprocessing, insightful EDA, and strategic outlier handling via scaling. The use of various algorithms, such as Linear Regression, Random Forest, XGBoost, SVR, and ANN, allowed for a comprehensive comparative analysis. The use of hyperparameter tuning and ensemble methods, such as weighted averaging, significantly improved pre-

dictive accuracy and model robustness. The Best Tuned XGBoost Regressor, with an R-squared of 90.08%, is an extremely effective solution that demonstrates machine learning's ability to provide useful insights for air quality monitoring and management.

0.8 Future Work

To further enhance the predictive capabilities and practical utility of this project, the following areas could be explored:

- **Advanced Feature Engineering:**
  - Incorporate lag features (e.g., pollutant concentrations from previous hours or days) to explicitly model temporal dependencies, which are critical in time-series data [15].
  - Generate rolling averages or other time-series specific statistical features [15].
- **More Complex Deep Learning Architectures:**
  - Investigate Long Short-Term Memory (LSTM) networks or other Recurrent Neural Networks (RNNs) specifically designed for sequence data. These architectures are better equipped to capture long-term dependencies in time-series data, potentially leading to higher accuracy for forecasting [3, 8, 9].
- **Advanced Ensemble Techniques:**
  - Experiment with Stacking ensembles, where a meta-model learns to optimally combine the predictions of the base models, potentially yielding even higher performance than simple or weighted averaging [2, 11].
- **External Data Integration:**
  - Investigate the integration of additional external data sources, such as real-time weather forecasts (wind speed, direction, and precipitation), traffic data, or industrial activity schedules, which are known to affect air quality [1].
  - **Model deployment:** Create a real-time prediction system or API that can consume live sensor data and provide instant air quality forecasts, transforming the project from analysis to practical application.

Bibliography

[1] Abdullah, M. A. R., Hashim, S. F., & Razak, M. A. A. (2025). A Review of Machine Learning Models for Predicting Air Quality in Urban Areas. *Metaheuristic Optimization Review*, *3*(2), 33-46.

[2] Ayodele, S. O., Ayodele, O. M., Akinyemi, I., Akingbehin, K. (2024). Systematic Review of Machine Learning and Deep Learning Techniques for Spatiotemporal Air Quality Prediction. *Atmosphere*, *15*(11), 1352.

[3] Chen, Y., Zhang, X., Li, J., Wang, Y. (2025). Air quality prediction based on a DBO-LSTM model. *AIP Advances*, *15*(7), 075022.

[4] Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785-794).

[5] De Vito, S., Massera, G., Piga, M., Martinelli, M., Di Francia, G. (2008). On the use of metal oxide semiconductor sensors for air quality control. *Sensors and Actuators B: Chemical*, *129*(2), 790-798.

[6] Dua, D., Graff, C. (2017). *UCI Machine Learning Repository*. University of California, School of Information and Computer Science. `http://archive.ics.uci.edu/ml/datasets/Air+Quality`

[7] European Environment Agency. (2023). *Air quality in Europe 2023*. Publications Office of the European Union.

[8] Kim, J., Kim, S., Park, H., & Lee, J. (2020). Air Pollution Prediction Using Long Short-Term Memory (LSTM) and Deep Autoencoder (DAE) Models. *Sustainability*, *12*(6), 2570.

[9] Meng, Y.-F., Li, Y., Wang, X., & Liu, X. (2020). Deep Learning for Air Quality Forecasts: A Review. *Atmospheric Pollution Research*, *11*(10), 1735-1748.

[10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830.

[11] Pan, X., Li, J., Zhang, X. (2025). A Hybrid Model for Air Quality Prediction Based on Data Decomposition. *Environmental Science and Pollution Research International*, *32*(1), 1-15.

[12] Sun, Y., Du, X., Sun, Y. (2024). Ensemble Machine Learning, Deep Learning, and Time Series Forecasting: Improving Prediction Accuracy for Hourly Concentrations of Ambient Air Pollutants. *Aerosol and Air Quality Research*, *24*(1), 230317.

[13] Wang, C., Liu, C., Zhang, Y. (2025). Hybrid Machine Learning Models for Fine-Grained Air Quality Forecasting. *International Journal of Futuristic Management Research (IJFMR)*, *8*(2), 1-10.

[14] World Health Organization. (2021). *WHO global air quality guidelines: Particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*. World Health Organization.

[15] Zhao, X., Wang, Y., Li, Y. (2025). Time Series Forecasting for Air Quality with Structured and Unstructured Data Using Artificial Neural Networks. *Atmosphere*, *16*(3), 320.

[16] Zhou, B., Lin, C. (2024). A review of air quality prediction models based on machine learning. *Journal of Environmental Informatics*, *43*(1).