Air Pollutant Concentration Forecasting Using Machine Learning Methods

INTRODUCTION

Our daily life is being affected by air pollution, which has become an important problem. Therefore, we hope to learn the laws of air quality changes by analyzing the collected data. In this way, we can predict air pollution and take timely measures.

The UCI air quality data set used in this project contains 9,358 hourly time series records, including real pollutant concentration (CO, NMHC, C6H6, NOx, NO_2), sensor response signals and meteorological variables (temperature, humidity, etc.). Since the dataset contains missing values, it requires preprocessing.

To build machine learning models that learn time patterns and environmental relationships, we apply regression models such as Linear Regression, Random Forest, and XGBoost to forecast future pollutant levels. And we classify air quality levels using Random Forest, XGBoost, and Logistic Regression. We hope that our air quality forecasts can help to protect the environment.

DATA ANALYSIS

The dataset has some clear quality problems. Many entries use a value like "−200" to indicate missing sensor readings, and the NMHC series in particular shows long periods where the values are flat and unchanged, which likely reflects sensor faults or signal loss rather than real atmospheric behaviour.

The time-series charts reveal that pollutants show both seasonal and daily patterns. For example, after November 2004 the concentrations of NOx and NO_2 increase markedly, which is likely linked to higher winter heating emissions and unfavourable dispersion

conditions. The intraday fluctuation patterns also match the typical morning and evening traffic peaks in an urban setting.

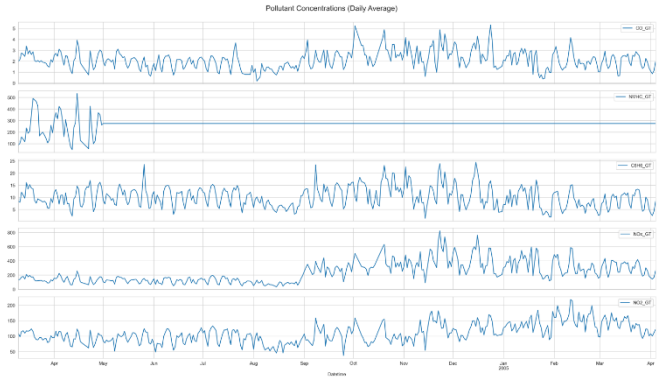

Figure 1 timing diagram

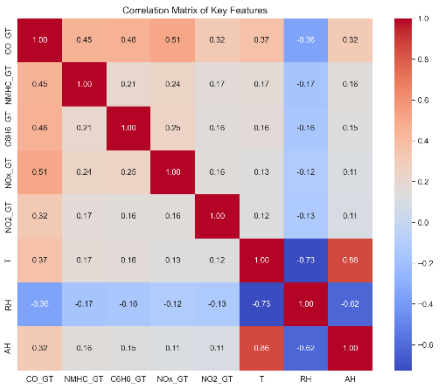

Figure 2 correlation matrix

From the correlation matrix, CO, C_6H_6 and NOx show strong positive correlations ($r \approx 0.79–0.82$), suggesting a common source such as road traffic. Meteorological variables are also positively correlated, especially temperature (T) and absolute humidity (AH) with $r \approx 0.66$. In contrast, NO_2 is negatively correlated with AH ($r \approx -0.32$), indicating that drier conditions may favour the accumulation of this pollutant.

METHODOLOGY

We first combined the original date and time columns into a single pandas Datetime index. Blank cells, the sentinel value −200 and any non-numeric entries were converted to np.nan. Rows or columns with too many missing values were removed, and the remaining gaps were filled using forward and backward filling. Finally, all variables were scaled so that pollutants and meteorological features, which are on very different numerical ranges, are on a comparable scale.

For feature engineering, we used sinusoidal encodings of hour and month to capture temporal cycles. In the classification task, we added autoregressive features (lags of 1, 6, 12 and 24 hours) and 3- and 6-hour rolling means so models could exploit short-term dynamics. CO(GT) was discretized into three levels (low <1.5, medium, high >2.5), and Isolation Forest on pollutant concentrations was used to flag anomalous readings. Following the project specification, 2004 data was used for training and 2005 for testing,

enforcing a chronological split and avoiding leakage; for each regression target, other pollutant columns were removed and future targets were created with shift(-horizon) for 1, 6, 12 and 24-hour forecasts.

For regression, we compared linear regression, random forest, gradient boosting and XGBoost against a naive baseline that predicts the next value as the current one. Random forests were used as a robust perturb-and-combine method [1], XGBoost for its scalability and empirical performance [2], and linear regression as a simple benchmark [3]. For CO classification, we used logistic regression, random forest and a tuned XGBoost classifier (600 trees, max depth 6, subsample 0.8). Regression models were evaluated mainly with RMSE, while classification performance was assessed using overall accuracy and macro-averaged F1 to better reflect minority-class performance under imbalance.

RESULTS

4.1 Regression Results

In our experiments, machine-learning regression models outperform the persistence baseline. They show lower RMSE and MAE, and their mean R^2 values are positive, meaning they capture useful temporal patterns. The baseline works well only for a few 1-hour predictions (CO, NOx, NO_2) due to strong autocorrelation, but its overall R^2 remains negative.

Linear Regression is the most stable model, with an average R^2 of about 0.26 across pollutants and horizons. Longer horizons reduce accuracy: results are best at 1 and 6 hours, and clearly worse at 12 and 24 hours. CO and NOx are easier to forecast, while NMHC, C6H6, and NO_2 are much harder.

Model	Average RMSE	Average MAE	Average R^2
Naive Baseline	67.763	50.196	-0.116
Linear Regression	57.306	42.837	0.262
Random Forest	63.929	49.277	0.138

XGBoost	65.330	49.580	0.051

Table 1 Average regression performance across all pollutants and horizons

4.2 Classification Results

In practice, the learned classifiers usually outperform the naive rule that just repeats the current class at the next time step. Among them, Logistic Regression performs best, with F1 scores of roughly 0.71 for the 1-hour forecast and about 0.54 for 24 hours ahead; Random Forest and XGBoost are a little worse but show the same general pattern. The naive baseline only keeps up at the extreme horizons (1 h and 24 h) and clearly lags behind for the intermediate ones. As the prediction horizon grows, both accuracy and macro-F1 tend to decline.

Model	Average Accuracy	Average F1
LogisticRegression	0.609	0.603
RandomForest	0.597	0.574
XGBoost	0.593	0.577
Naive_Baseline	0.543	0.544

Table 2. Average classification performance across all horizons

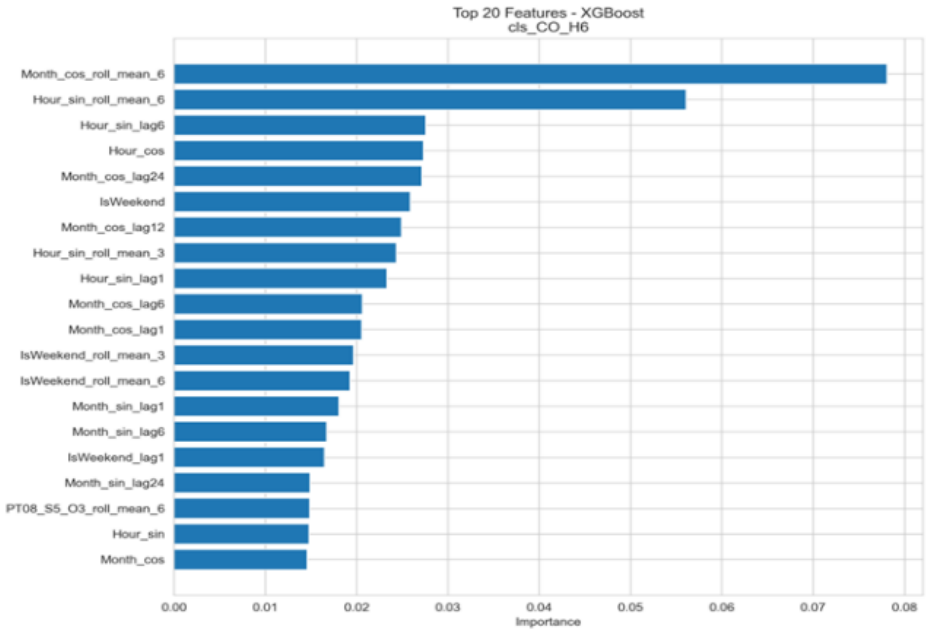

Figure 3. Top 20 feature importances for XGBoost CO classification at 6-hour horizon.

Figure 3 shows that XGBoost mainly relies on temporal and seasonal variables (hour-of-day sine/cosine and month-of-year encodings), plus a few sensor-based features.

4.3 Hyper-parameters and design choices

For the tree-based models and XGBoost, we tuned key hyperparameters such as the number of trees, maximum depth and learning rate using Bayesian optimisation with time-series cross-validation. In the Random Forests, we also varied the number of randomly selected features at each split (mtry): lower mtry values make trees more different and less correlated, which improves stability when aggregating, but individual trees perform worse on average because they are built on more limited and sometimes suboptimal variables [4]. XGBoost not only optimises the boosting algorithm but also uses CPU multithreading to run calculations in parallel, which speeds up training and often gives strong predictive performance [5]. Overall, however, the tuned configurations provide only modest gains over reasonable defaults, and the good performance of Linear Regression suggests that relatively simple models already offer a reasonable balance between accuracy and complexity.

DISCUSSION

The three learning algorithms behaved quite differently across regression and classification. For 1-hour regression, tree-based models – especially Random Forest – often performed best for pollutants such as C6H6 by capturing nonlinear relationships and achieving the lowest RMSE and highest R^2 in those cases. However, over all targets and horizons, linear regression was the most stable and competitive overall, particularly for CO, which behaves more like a linear function of the inputs. As the horizon increased to 12–24 hours, performance degraded for all models and the naive "copy-last-value" baseline became comparable, suggesting that long-range forecasts are dominated by slowly varying background levels and noise. A similar pattern appeared in the CO classification task: at 1 hour the naive baseline (predicting the current class)

achieved the best accuracy because labels change slowly, at 6 and 12 hours the machine-learning models clearly outperformed it, and by 24 hours the gap largely closed again. Evaluation metrics also shaped how we interpreted performance. For regression, RMSE was the primary metric because it penalises large errors and is critical when underestimating extreme pollution is costly, while MAE provided a more interpretable "typical error" and R^2 summarised explained variance. For classification, accuracy was intuitive but could be misleading under class imbalance, so the weighted F1 score gave a fairer view by balancing precision and recall, and ROC/precision–recall curves highlighted how models trade off false alarms against missed high-pollution events. Finally, our results suggest several improvements: moving beyond simple temporal encodings to sequence models such as LSTMs or GRUs, using multi-output or multitask models to exploit correlations between pollutants, incorporating external drivers (e.g. traffic and weather forecasts), and applying more systematic hyper-parameter tuning to balance accuracy, robustness and interpretability.

CONCLUSION

In summary, our results suggest that linear models, tree-based ensembles and a simple persistence baseline each play a different role in urban air-quality prediction. Tree-based methods are good at picking up short-term nonlinear patterns, linear regression provides a stable and interpretable point of comparison, and the naive baseline can still perform well when pollutant levels change slowly, especially at very short or very long horizons. With appropriate temporal features, suitable evaluation metrics, and possible extensions to richer inputs and sequence models, this setup can form a practical basis for forecasting pollution levels and supporting real-world air-quality monitoring.

REFERENCES

[1] L. Breiman, "Arcing Classifiers", Annals of Statistics, 1998.

[2] T. Chen, C. Guestrin, "XGBoost: A Scalable Tree Boosting System", Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[3] D. Maulud, A. M. Abdulazeez, "A Review on Linear Regression Comprehensive in Machine Learning", 2020.

[4] P. Probst, et al., "Hyperparameters and tuning strategies for random forest", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018.

[5] M. Ma, et al., "XGBoost-based method for flash flood risk assessment", Journal of Hydrology, 2021.