COMP9417 PROJECT REPORT:
FORECASTING AIR POLLUTION WITH MACHINE LEARNING

Abdullah Ariyanto (z5543164), Jeremia Kevin Raja Siahaan (z5493216),
Riasat Ahmed Chowdury (z5527294), and Zafran Akhmadery Arif (z5603811)

1. Introduction

Air pollution forecasting plays an important role in environmental science and public health policy. A reliable forecast is crucial for government and public health authorities to issue warnings, design mitigation strategies, and evaluate some policies. To create such systems, authorities need an accurate prediction of pollutant concentrations in the air. Furthermore, accurate and reliable forecasts would help reduce health risks that are associated with exposure to high pollution levels, especially for children and people with respiratory conditions.

One of the main challenges in air quality forecasting is the complex nature of atmospheric processes. Pollutants in the air are mostly influenced by emission sources such as vehicles and industrial activities. Other than that, other variables such as temperature, humidity, and wind patterns could also influence the concentration of the pollutants. Machine learning (ML) helped us in creating such models to learn the pollutant behavior by learning the temporal patterns and relationships among chemicals and other variables. By learning from historical measurements, ML models can analyze some hidden dependencies across time and variables to create more accurate predictions.

For this project, we analyzed the Air Quality dataset from the UCI Machine Learning Repository consisting of 9,358 hourly sensor readings. This dataset contains hourly air quality records in Italy from March 2004 to February 2005. This dataset also contains readings for the pollutants such as carbon monoxide (CO), benzene (C6H6), nitrogen oxides (NOx), nonmethane hydrocarbon (NMHC), and nitrogen dioxide (NO2). Other than that, we also have some other variables such as temperature and humidity.

2. Data Analysis

During our initial inspection, the raw data had several issues associated with real-world environmental monitoring, such as missing values, inconsistent formatting, and redundant columns. As we continue our data exploration, we found out that the dataset used semicolons as separators and commas as decimal markers. Hence, we need to deal with it first before proceeding with further analysis.

We found out that a lot of values were missing, but the dataset is encoded using the sentinel value (-200) and not the NaN notation. These missing values were also not uniformly distributed across variables. For instance, at least 90% of the variable NMHC(GT) were missing, which would be very unreliable for analysis. Moreover, since the useful information under this variable is very limited, we removed this variable entirely from the cleaned dataset. For the other variables, after we identified the missing values, we handled it by imputing the median values to preserve the integrity of the time series.

In addition to handling the missing data, we also converted the dataset into a usable temporal format during preprocessing. This was done because the raw dataset stored date and time as separate string fields with inconsistent formatting. We merged these into a single timestamp variable and converted them into proper values so that we can use this data properly. We removed the rows with invalid timestamps since we can't position them within the time series. Finally, we sort the cleaned dataset in ascending temporal order.

3. Methodology

3.1. Regression Modeling

We began by loading the cleaned dataset, defining the prediction targets and forecast horizons, and applying a chronological split in which data from 2004 was used for training and data from 2005 for testing. To capture temporal dependencies in pollutant behavior, we generated lag features (t, t-1). Furthermore, we normalize the numerical inputs using *StandardScaler* to ensure that linear regression and boosting methods have normalized data. For the models, we selected Linear Regression as a baseline to validate basic data relationships (Karatzas et al., 2018), Random Forest (RF) to handle non-linear pollution spikes (Sathvika et al., 2024), and Gradient Boosting (XGBoost) to optimize residual errors and handle missing data native to the sensors (Tırınk, 2025). We also use GridSearch to systematically tune key hyperparameters, such as tree depth and the number of estimators, ensuring our models outperform standard default configurations.

3.2. Classification Modeling

For classification modeling, we began the initial steps in the same way as in regression modeling. However, for the lag features, we generated more (t-1, t-6, t-12, t-24) to capture the temporal dependencies better, as well as the future features (t+1, t+6, t+12, t+24) to be used as target variables. The CO(GT) values were discretized into three pollution categories: low, mid, and high. The resulting classes were encoded using both one-hot encoding and label encoding. For the models, we selected Decision Tree (DT), which is well-known for its interpretability; we use this as a baseline to classify multiple values, Support Vector Machine (SVM) for its effectiveness to classify in complex boundaries, and Logistic Regression (LR) with Ensemble to classify using a linear model. Same as before, we used *StandardScaler* to normalize numerical inputs for the DT and SVM models, and also used GridSearch for all models as well.

3.3 Anomaly Detection

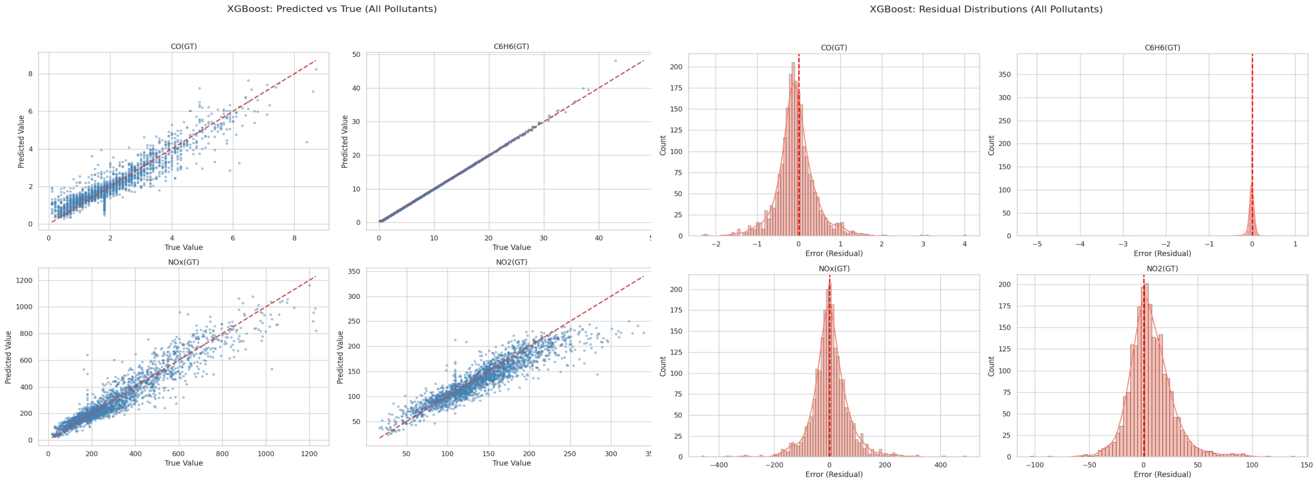
For anomaly detection, we followed a residual-based approach to detect the predicted values for meteorological and calendar features. After training the model, we generated predicted values to capture the difference between the observed and predicted concentrations. Then, we examine the distribution of the residuals and determine the 99th percentile value as a threshold so that we can see the difference between normal variation and the anomaly. Any data that has an absolute residual greater than the 99th percentile will be detected as an anomaly.

4. Results

4.1 Regression Model Result

Pollutant	Model	MAE	RMSE	R2
C6H6(GT)	Linear Regression	3.992	4.110	0.586
	Random Forest	2.005	2.510	0.846
	Gradient Boosting	0.048	0.153	0.999
CO(GT)	Linear Regression	0.779	0.882	0.564
	Random Forest	0.523	0.706	0.721
	Gradient Boosting	0.330	0.465	0.879
NO2(GT)	Linear Regression	15.848	21.802	0.815
	Random Forest	26.834	36.693	0.476
	Gradient Boosting	15.000	20.980	0.829
NOx(GT)	Linear Regression	60.630	85.948	0.831
	Random Forest	50.613	73.714	0.876
	Gradient Boosting	46.309	67.092	0.897

(Table 1: Test set performance table)

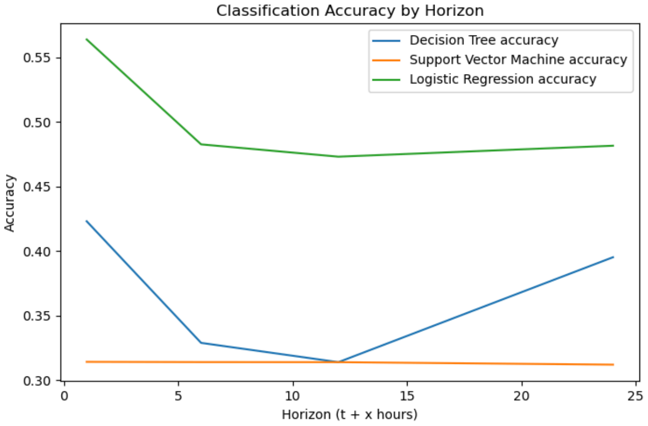


(Figure 1: showcasing Gradient boosting for all pollutants)

4.2 Classification Model Result

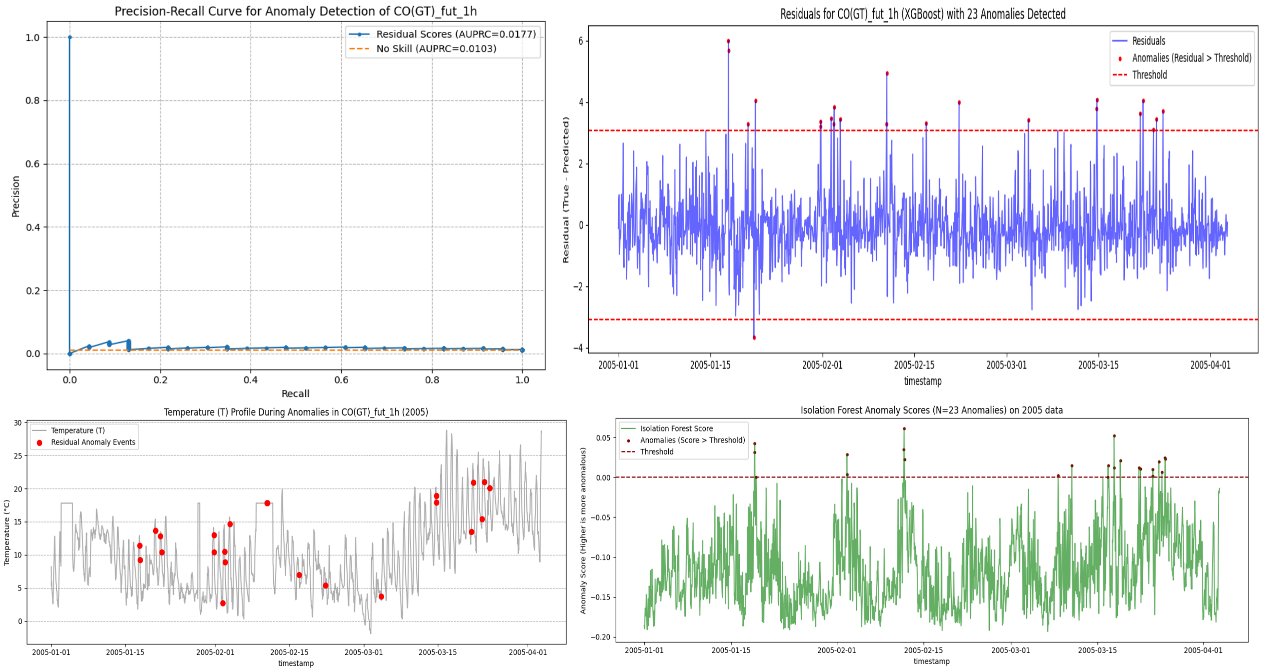
Horizon	Decision Tree	Support Vector Machine	Logistic Regression
1	0.42304	0.31413	0.56391
6	0.32882	0.31394	0.48263
12	0.31388	0.31388	0.47309
24	0.39518	0.31196	0.48158

(Table 2: Accuracy table)



(Figure 2: Classification model accuracy)

4.3 Anomaly Detection Result



(Figure 3: Anomaly detection results)

5. Discussion

The results from the regression model show that model performance varies across different pollutants. Gradient Boosting generally performed the best, especially strong for both C6H6(GT) and CO(GT), and good for NOx(GT). Random Forest has excellent performance for both pollutants as well, where the performances are very close to XGBoost's performance. Surprisingly, Linear Regression was very effective for NOx(GT) and even outperformed other models when predicting for NO2(GT). Overall, the results emphasize that model effectiveness depends on each pollutant.

Based on the classification model result, we can see that Logistic Regression is the most reliable model as it predicts better than the other predictors. The higher accuracy from this model suggests that it learns better to capture the underlying patterns in CO concentration over time. We can say that the simpler linear models are more effective for this dataset compared to more complex approaches.

The analysis on the anomaly detection shows that the model's failures are concentrated during colder weekday periods. This means the anomalies are affected by extreme weekday commuting events that coincide with unfavorable meteorological conditions, which trap emissions that could lead to high CO values, exceeding the model's expectation.

6. Conclusion

The findings from both the regression and classification models showed that model performance in air quality forecasting is very dependent on the predicted pollutant. There's a trend on the regression model where Gradient Boosting and Random Forest produced a similar output at capturing complex patterns and non-linear relationships, while Linear Regression is superior when predicting some certain variables. This trend is also reflected in the classification model results, where LR gave us the highest accuracy compared to the other complex models, such as DT and SVM. In conclusion, this project just showed us that simpler models can be equally or even better than advanced ensemble models. Also, selecting models based on pollutant characteristics instead of overall complexity is important to get more accurate results.

References

Karatzas, K., Katsifarakis, N., Orlowski, C. and Sarzyński, A. (2018). Revisiting urban air quality forecasting: a regression approach. *Vietnam Journal of Computer Science*, 5(2), pp.177–184. doi:https://doi.org/10.1007/s40595-018-0113-0.

Sathvika, G., Poojitha, P., Rakesh, K., Tadepalli, L. and Chaitanya, K. (2024). Issue 6 www.jetir.org (ISSN-2349-5162). *JETIR2406312 Journal of Emerging Technologies and Innovative Research*, [online] 11(1). Available at: https://www.jetir.org/papers/JETIR2406312.pdf.

Tırınk, S. (2025). Machine learning-based forecasting of air quality index under long-term environmental patterns: A comparative approach with XGBoost, LightGBM, and SVM. *PLOS One*, 20(10), p.e0334252. doi:https://doi.org/10.1371/journal.pone.0334252.