Comparative Analysis of Machine Learning Models for Forecasting Urban Air Pollutants

Martina Ivanova
Politecnico di Milano (Italy)
martina.ivanova@polimi.it

Alberto Celani
Politecnico di Milano (Italy)
alberto.celani@polimi.it

Luca Mottola
Politecnico di Milano (Italy), RI.SE (Sweden), and Uppsala University (Sweden)
luca.mottola@polimi.it

***Abstract*—We comprehensively compare thirteen machine learning models for forecasting urban air pollutants. However, the accuracy of existing prediction models varies as a function of what specific pollutant is predicted, as well as the nature and size of the training set. We examine the performance of thirteen machine learning models using fifteen years of IoT sensor data, including both meteorological and pollutant data representative of a rural industrial urban environment in the heart of the Lombardy region (Italy). While prior studies have applied machine learning models to urban air pollution forecasting [3,4,7], few have systematically compared a diverse set of models using a long-term, 15-year dataset across multiple pollutants and training data scenarios. In this work, we benchmark thirteen models, revealing how pollutant-specific characteristics and training history affect forecasting performance. Ensemble tree-based models, particularly LightGBM, XGBoost, and Random Forest, consistently outperform others, especially for pollutants with strong temporal patterns such as NO_2 and NO. Conversely, pollutants like NH_3 and CO prove more challenging to predict, due to irregular dynamics and weaker correlation with meteorological features. Our analysis also reveals that increasing the proportion of training data generally enhances model accuracy as expected, though improvements diminish beyond a 70–80% split w.r.t test data.**

***Index Terms*—Air quality forecasting, machine learning, ensemble models, environmental monitoring, pollutant prediction, data-driven modeling.**

I. Introduction

The ability to evaluate and predict the concentration of air pollutants represents a crucial step in addressing air pollution, which remains a major global environmental and public health concern. Among the most harmful pollutants are nitrogen dioxide (NO_2), ozone (O_3), ammonia (NH_3), and nitric oxide (NO), due to their significant adverse effects on both human health and the environment. Prolonged exposure to these substances is linked to adverse health effects and declining air quality. The Sustainable Development Goals of the United Nations General Assembly [1] establish a strong correlation between air pollution and a number of different goals.

Effective pollutants forecasting allows for early warnings, better pollution control measures, and informed policymaking to mitigate environmental damage. The rise of the Internet of Things (IoT) has facilitated dense networks of low-cost air quality sensors, generating high-volume environmental data in real time. These IoT-driven data streams open new opportunities for Machine Learning (ML) models to learn complex patterns and provide timely forecasts of pollutant levels. However, selecting appropriate ML algorithms and understanding their performance for different pollutants remains an open challenge [2].

Prior research [3] has applied various ML techniques for air quality prediction, including linear regression models, tree-based ensembles, support vector machines, and neural networks, with mixed results depending on pollutant and context [4]. Yet, comparative evaluations across a wide range of models under identical conditions are limited.

To fill this gap, we conduct a comprehensive comparison of thirteen ML models for forecasting hourly concentrations of five common urban air pollutants: CO, NO, NH_3, O_3, NO_2. These pollutants exhibit different emission sources and temporal dynamics. For example, NO and NO_2 typically follow strong diurnal cycles tied to traffic rush hours and photochemical processes, whereas NO_2 emissions (from agriculture, waste, etc.) may be sporadic or seasonally driven, potentially leading to irregular patterns. By evaluating a diverse set of models on each pollutant, we identify which approaches work best for which pollutant and why.

Another key element we explored in this study was how different train/test split strategies influence model performance. We tested five variations to evaluate how much historical data is necessary for reliable predictions and how the choice of training data impacts accuracy.

In this analysis we used a rich fifteen-year historical dataset of meteorological variables and pollutant concentrations, representative of a rural industrial urban area in the heart of Lombardy region, with IoT sensor coverage. We intend to release the complete dataset and accompanying code used in this study upon the manuscript's acceptance.

The remainder of the paper is structured as follows: Section II reviews related work. Section III outlines the materials and methods we used. Section IV presents the results and offers a detailed interpretation of the findings. Section V discusses key insights and practical implications, emphasizing the relevance of the outcomes. Finally, Section VI concludes the paper.

II. Related Work

Earliest efforts in air quality prediction date back to the mid-20th century, when the use of statistical and physical modeling techniques provided foundational techniques for simulating pollutant dispersion and temporal trends [5].

A key turning point came in the 1990s when ML—particularly Artificial Neural Networks (ANNs)—for air quality forecasting were introduced as an alternative method to the traditional simple regression models and time-series approaches [6], such as ARIMA. However, these methods often suffer from high computational cost and sensitivity to boundary conditions, which can hinder real-time deployment in urban environments.

Early adopters of ML in environmental contexts, demonstrated that Artificial Neural Networks (ANNs) could outperform linear regression models in capturing non-linear dependencies between meteorological variables and pollutant levels, especially NO_2 [7]. This marked a turning point, setting the stage for a broader adoption of ML in environmental sciences.

Subsequent studies in the 2000s expanded the application of ML to various pollutants, such as PM2.5, O_3, and NO_2. For instance, decision trees and Support Vector Machines (SVM) applied to predict air pollution episodes in Portugal, reporting improved accuracy and interpretability [7].

The 2010s saw exponential growth in this field, driven by the availability of large environmental datasets and the rising influence of IoT technology. Sensors now offer high-resolution spatiotemporal data, which ML models can utilize effectively. Techniques such as Random Forests, Gradient Boosting Machines, XGBoost, and LightGBM became increasingly popular for their ability to handle non-linearities, collinearity among predictors, and large datasets with minimal preprocessing. For example, in 2015 ensemble learning models for hourly NO_2 prediction were successfully applied in Beijing, demonstrating their superior performance compared to ANN and SVM [8].

IBM's *Green Horizons* project in China [9] utilized ML and cognitive computing to forecast air pollution and provide actionable insights for city planners. On a commercial scale, *BreezoMeter* applies ensemble and deep learning methods for hyperlocal air quality forecasting, offering forecasts up to 96 hours in advance.

More recent literature [10,11] has turned attention to multi-pollutant and hybrid modeling approaches, using spatiotemporal deep learning architectures to incorporate both local conditions and spatial correlations across sensor networks. Yet, these models demand significant computational resources and may struggle to generalize under changing emission patterns, as evidenced during anomalous periods like COVID-19 lockdowns [12].

Our study addresses the comparative performance of multiple ML algorithms in pollutants concentration forecasting, incorporating train-test variability through five different split ratios. Our approach directly tackles the lack of consistency in prior studies, offering insight into how the size of training data affects model performance. By using a consistent dataset and evaluation metrics across all models, we establish a rigorous benchmarking framework that enabled meaningful comparisons. Our approach contributes to the discussion on the generalizability and robustness of ML techniques in air quality forecasting.

III. Materials and Methods

We illustrate the key features of the area we study, of the dataset we use and of the ML models we consider. We conclude with a description of the methodology we adopt.

Study area. Codogno is a small town in the Lombardy region, Italy, which serves as the target for this work. Its strategic position in the Po Valley gives rise to significant air pollution due to industrial activities, vehicular emissions, and meteorological conditions that contribute to pollutant accumulation [13].

The monitoring stations in this study are operated by ARPA Lombardia (Agenzia Regionale per la Protezione dell'Ambiente), which provides reliable air quality data across the region, on different time scales, thanks to the numerous IoT measurement points situated across the region surrounding Codogno, which monitor not only meteorological conditions but also the concentration of up to ten different air pollutants.

Datasets. We used historical air quality data spanning fifteen years, from Jan 2010 through 2024, including hourly records of pollutant concentrations and meteorological variables. The data include four target pollutants— NO_2, NH_3, O_3, and NO—and environmental features such as temperature, solar radiation, precipitation, wind speed, and humidity.

For model evaluation, we define a 30-day forecast period corresponding to Jan 1–30, 2025 at hourly resolution. We treat this period as an out-of-sample test set, includes wintertime conditions with typical daily pollutant cycles. Actual pollutant concentrations for this period were provided as "future" observations, for comparison with forecasts. We chose a contiguous 30-day horizon deliberately, with the scope of mimicking a realistic forecasting scenario.

ML models. To ensure a comprehensive evaluation of the ML algorithms deployed, we employ five different train-test data splits, allowing for an in-depth analysis of the influence of the amount of data over model performance, using random sampling with a fixed random seed for reproducibility. All models were ultimately applied to predict the entire 30-day period. This approach allowed for each model to be trained five times—once for each training fraction—and then to generate five sets of forecasts for Jan 2025.

The ML models we consider are selected based on their diversity across modeling paradigms (linear, tree-based, boosting, kernel-based, and non-parametric), as well as their widespread adoption in time series forecasting and environmental data analysis. This set of thirteen models includes:

1) **Linear Regression** (linear model): assumes a linear relationship between the regressors and the dependent variable. It estimates coefficients that minimize the difference between predicted and actual values, serving as a simple baseline for comparison;
2) **Ridge Regression** (linear model): a linear model that prevents large coefficient values, reducing the risk of overfitting and is particularly useful when features (in this case meteorological) are highly correlated;
3) **Lasso Regression** (linear model): a linear regression model, which selectively shrinks some feature coeffi-

cients to zero, performing feature selection with the goal of improving interpretability;

4) **ElasticNet** (linear model): combines Lasso and Ridge Regression, balancing feature selection and regularization. This model is effective when dealing with multicollinearity [14], that is, when several independent variables are correlated, among meteorological inputs;
5) **DecisionTree** (tree-based model): splits the data into hierarchical decision rules ("leaves" and "branches") based on meteorological factors. It captures nonlinear relationships but may overfit without proper pruning;
6) **Random Forest** (tree-based model): an ensemble of multiple decision trees that reduces variance by averaging predictions. It is robust against overfitting and captures complex interactions in air quality data;
7) **ExtraTrees** (tree-based model): similar to Random Forest with additional randomization in feature selection and splitting criteria, enhancing robustness;
8) **AdaBoost** (boosting model): sequentially corrects errors from previous weak learners, improving predictive performance, especially for volatile levels;
9) **Gradient Boosting** (boosting model): builds trees iteratively, optimizing residual errors. It is claimed to be effective in modeling complex dependencies in meteorological and air quality data;
10) **XGBoost** (boosting model): an optimized version of Gradient Boosting that incorporates regularization and handling of missing values, claimed to be one of the best performing models;
11) **LightGBM** (boosting model): a gradient boosting model optimized for speed and efficiency. It processes large datasets quickly while maintaining high accuracy;
12) **K-Nearest Neighbors (KNN)** (non-parametric model): predicts the target variables based on the values of the nearest features observations;
13) **Support Vector Regressor (SVR)** (kernel-based model): maps input data into a higher-dimensional space to identify complex patterns, particularly useful for handling nonlinear relationships.

We use the regular implementation of all models as found in standard Python libraries. We use the `scikit-learn.` implementation for most models, and pick XGBoost and LightGBM from their respective packages [15-17]. We use default hyperparameters, except the number of trees in ensemble models which was set to 100 for Random Forest, Extra Trees, GBM, XGBoost, and LightGBM to ensure adequate learning capacity. To retain consistency and avoid overfitting across a broad comparison, we did not perform hyperparameter tuning. While this decision allows for fairer benchmarking of default model behavior, it may disadvantage algorithms like SVR and AdaBoost, which are known to be highly sensitive to their configuration settings [3,4].

A. ML Pipeline and Methodology

We trained each model separately for each pollutant. The input features to the models included recent meteorological variables and time features. Specifically: Temperature, Radiation, Precipitation, Wind Speed, Humidity, and temporal indicators (Hour, Day, Month, Weekday) are provided as predictors.

We deliberately excluded lagged pollutant values to simulate forecasting from exogenous features alone, aligned with approaches aiming for real-time deployability without reliance on prior pollutant measurements. Although autoregressive inputs can improve accuracy, this restriction allows evaluating model performance solely from environmental predictors [2,3]. This design simulates real-world constraints where historical pollutant data may be unavailable or delayed. Although lag features can improve accuracy through autocorrelation [2], this study isolates exogenous predictive power to better assess model generalizability.

After selecting a target variable (y), we construct the feature set (x) by excluding time information and the target variable. The dataset is split into training and testing subsets, according to multiple train-test splits: 50%, 60%, 70%, 80%, and 90%.

Forecast. After training on the historical data, we predict pollutant concentrations hour-by-hour over the forecast period. We assumed that the required features, for example, meteorological forecasts, for the period at stake were available or could be estimated; in practice, one would use actual weather forecasts as input. In our case, we used the actual observed meteorological data for Jan 2025 as input to isolate model errors. The result was, for each pollutant, 65 forecast series, that is 13 models times 5 training scenarios, of 720 hours each, alongside the actual observed series.

IV. Results

The following subsections present the results separately for each pollutant through summary tables containing performance metrics, indicating the best performing train/test ratio of each model for each pollutant, in terms of accuracy. We also graphically illustrate typical forecast behavior. The results are based on predictions for the forecast period compared against the actual observed concentrations for that period. We evaluate the forecast models using standard performance metrics: R^2, RMSE, MAE, and (Accuracy (%) $= \left(1 - \frac{\text{MAE}}{\overline{Y}_{\text{actual}}}\right) \times 100$).

A. Carbon Monoxide (CO)

As seen in Fig. 1, CO concentrations during the forecast period were low in magnitude (mostly 0.5–1.5 units) with a relatively flat diurnal profile and a regular fluctuation. The lack of strong upward or downward long-term trend lead to limited variability proving challenging for models to capture improvements over a naive prediction. *Table I* summarizes the CO forecasting performance metrics for each model, ranking the best performed train/test split among each algorithm.

Among all tested configurations, XGBoost with the 90% training split (CO_XGBoost_90) demonstrated the best overall performance, indicating a strong balance between prediction precision and generalization. The LightGBM model (CO_LightGBM_80) followed with a respectable R^2 and

TABLE I
PERFORMANCE OF CO PREDICTION MODELS

Model	R^2	*RMSE*	*MAE*	*Accuracy (%)*
CO_XGBoost_90	**0.5560**	**0.2537**	**0.1966**	**78.03**
CO_LightGBM_80	**0.4634**	**0.2789**	**0.2260**	**74.74**
CO_ExtraTrees_90	**0.3661**	**0.3031**	**0.2493**	**72.13**
CO_RandomForest_90	**0.3358**	**0.3103**	**0.2530**	**71.72**
CO_AdaBoost_50	0.1707	0.3467	0.2793	68.78
CO_DecisionTree_80	-0.0259	0.3856	0.2966	66.84
CO_KNN_80	0.0493	0.3712	0.3096	65.40
CO_GradientBoosting_70	0.1083	0.3595	0.3130	65.02
CO_ElasticNet_50	-0.1052	0.4002	0.3497	60.92
CO_Lasso_50	-0.1331	0.4053	0.3534	60.50
CO_SVR_50	-0.1578	0.4096	0.3594	59.82
CO_Ridge_60	-0.2383	0.4237	0.3671	58.96
CO_LinearRegression_60	-0.2383	0.4237	0.3671	58.96

[a]All values rounded to 4 decimal places. Accuracy is expressed as a percentage.

TABLE II
PERFORMANCE OF NO PREDICTION MODELS

Model	R^2	*RMSE*	*MAE*	*Accuracy (%)*
NO__RandomForest_80	**0.8138**	**22.2040**	**14.0604**	**78.31**
NO__ExtraTrees_80	**0.8033**	**22.8231**	**14.4405**	**77.72**
NO__LightGBM_70	**0.8012**	**22.9462**	**14.4598**	**77.69**
NO__XGBoost_50	**0.7868**	**23.7641**	**15.0071**	**76.85**
NO__GradientBoosting_70	0.7541	25.5190	16.0295	75.27
NO__KNN_90	0.7254	26.9650	16.2958	74.86
NO__ElasticNet_80	0.6744	29.3656	16.8102	74.07
NO__Lasso_80	0.6793	29.1441	16.9541	73.85
NO__Ridge_80	0.6786	29.1758	18.3979	71.62
NO__LinearRegression_80	0.6786	29.1759	18.3983	71.62
NO__DecisionTree_50	0.6560	30.1838	18.7037	71.15
NO__SVR_90	0.1814	46.5614	26.4179	59.25
NO__AdaBoost_50	0.5495	34.5397	27.6603	57.33

[a]All values are rounded to 4 decimal places. Accuracy is expressed as a percentage.

slightly higher RMSE and MAE, suggesting its predictions were somewhat more dispersed compared to XGBoost. Ensemble-based methods like Extra Trees and Random Forest with 90% training split also performed reasonably well, though with slightly reduced accuracy.

On the other hand, AdaBoost models, for example, CO_AdaBoost_50, showed relatively lower performance, possibly due to their sensitivity to noise or underfitting in this particular dataset. Overall, boosted tree-based models, especially XGBoost and LightGBM, consistently outperformed other algorithms, underlining their robustness against time series prediction tasks involving CO concentration levels.

B. Nitric Oxide (NO)

For NO, the highest concentration peaks seen in Fig. 2 (up to 382 µg/m³) posed forecasting challenges, but overall performance was strong for the top performing models, as seen in *Table II*. RandomForest achieved the lowest RMSE and highest R², indicating it explained 81% of NO variance– the best R² among all pollutants. ExtraTrees and LightGBM were next, followed by XGBoost. GradientBoosting and KNN have slightly higher RMSE but still maintain R² of 0.73–0.75, indicating decent performance. The Linear and Ridge models, which again produced similar results, and Lasso were mid-tiers, while SVR and AdaBoost were the worst.

All top 8 models show $R^2 > 0.67$, indicating they captured the majority of variance in NO levels, likely due to NO's diurnal seasonality, which is easier to model when using tree ensembles that capture nonlinear interactions. RandomForest emerged as the best-performing model for NO, closely followed by ExtraTrees and LightGBM. Simpler models performed worse and AdaBoost continued to exhibit instability.

C. Ammonia (NH_3)

Model performance for forecasting NH_3 concentrations demonstrated great variability across different algorithms and parameter settings, as shown on Fig. 3. NH_3 had a wider dynamic range, posing a more complex forecasting task. Forecasting NH_3 was challenging not only due to its complex and dynamic behavior, but also due to heterogeneous emission sources, chemical reactions occurring in the atmosphere, which together introduce significant nonlinearity in its concentration levels. Among the best-performing models, as shown in *Table III*, ensemble-based approaches such as Gradient Boosting and LightGBM emerged with higher predictive accuracy, where the former achieved the highest R² value of 0.050, indicating a modest but superior explanatory power compared to others.

Models like Extra Trees and Random Forest with a 70-minute forecast horizon showed negative R² values, suggesting poor generalization and predictive capability in those configurations. SVR, although generally underperforming in terms of R² and RMSE, demonstrated reasonable MAE values, indicating consistent albeit biased forecasts.

TABLE III
PERFORMANCE OF NH_3 PREDICTION MODELS

Model	R^2	*RMSE*	*MAE*	*Accuracy (%)*
NH3_LightGBM_60	**-0.0213**	**15.4757**	**10.1091**	**62.11**
NH3_GradientBoosting_80	**0.0504**	**14.9227**	**10.5408**	**60.49**
NH3_SVR_90	**0.0253**	**15.1186**	**11.0149**	**58.71**
NH3_ExtraTrees_70	**-0.3460**	**17.7664**	**11.5523**	**56.70**
NH3_RandomForest_70	-0.3583	17.8476	11.6669	56.27
NH3_XGBoost_60	-0.5935	19.3309	13.0289	51.16
NH3_DecisionTree_80	-0.7206	20.0876	13.8389	48.12
NH3_Lasso_60	-0.1601	16.4943	14.0059	47.50
NH3_ElasticNet_60	-0.1742	16.5937	14.1307	47.03
NH3_KNN_50	-0.5049	18.7862	14.3479	46.22
NH3_Ridge_60	-0.1765	16.6100	14.3554	46.19
NH3_LinearRegression_60	-0.1765	16.6099	14.3554	46.19
NH3_AdaBoost_60	-4.3080	35.2813	31.7938	-19.18

[a]All values rounded to 4 decimal places. Accuracy is expressed as a percentage.

Overall, NH_3 forecasts proved difficult: only ensemble and kernel methods contained errors to 15 units, while linear models and AdaBoost failed to generalize, showing large errors and negative accuracy. In this case, considering a pollutant with a more dynamic nature, models proved well performing at predicting trends, at the expense of accuracy, finding it difficult to reach the exact observed values.

D. Ozone (O_3)

O_3 forecasting was also challenging for most models, as *Table IV* demonstrates. LightGBM emerged as the best, with $R^2 = 0.47$, meaning it explained approximately 47% of the variability. ExtraTrees and XGBoost followed closely, achieving only 52-55% accuracy, since O_3 concentrations up

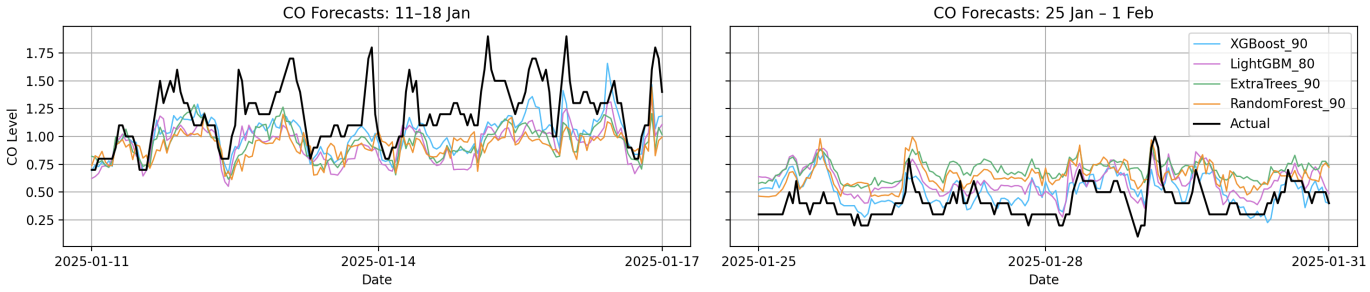

Fig. 1. Comparison of actual CO concentrations (black line) with predictions from the 4 best performing models, over 2 different 7-day periods in Jan 2025.

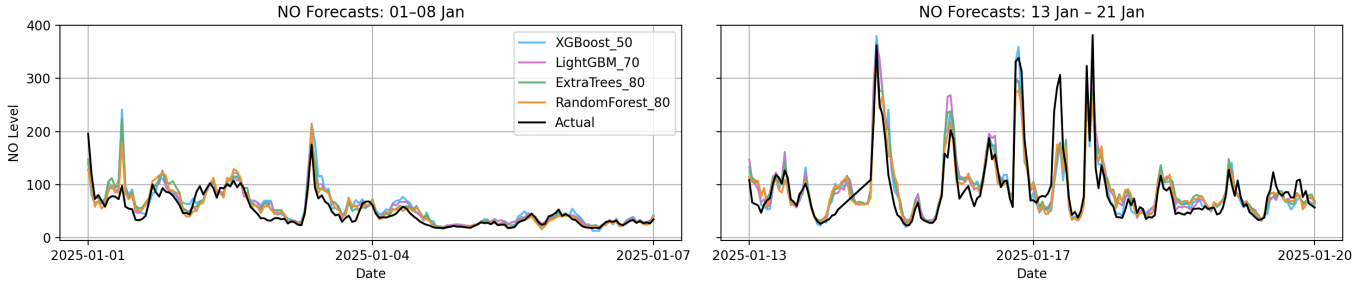

Fig. 2. Comparison of actual NO concentration levels (black line) with predictions from the 4 best performed train/test splits over time. The plot highlights the models' ability to capture peak concentrations and general trends in NO levels throughout Jan 2025.

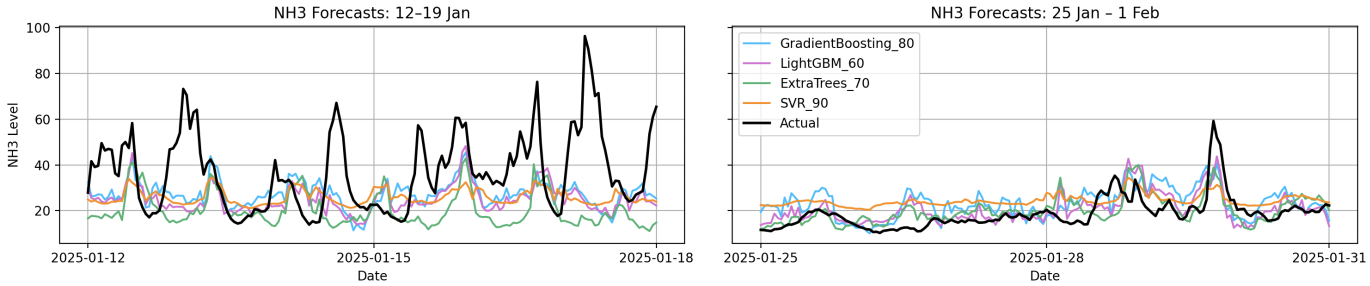

Fig. 3. Comparison of observed NH_3 concentrations (black line) with forecasted values from the 4 top-performing model variations in 2 snippets of 7 days each in the forecast period.

to ∼60 µg/m^3) often had low absolute values, making percentage errors high. Many models showed negative accuracy for O_3, indicating MAPE $>$ 100%. The forecast comparison is visualized in Fig. 4.

Still, even the best models had $R^2 < 0.5$, meaning large relative errors. This is partly due to very low overnight O_3 values, when actual O_3 is near zero. This effect explains why accuracy percentages are low or negative despite moderate RMSE values. Moreover, regression models yielded similar results, as did ElasticNet and Lasso, highlighting that little regularization was helpful for O_3.

In summary, LightGBM provided the most reliable O_3 forecasts among the tested models, but even it left substantial errors, whereas models like SVR and AdaBoost severely overfit or mis-predicted O_3 levels, resulting in unacceptably high errors. Ozone proved to be one of the most challenging pollutants to forecast accurately–many models yielded negative R² and low accuracy, indicating large errors relative to its variability. Simpler ML models appear not to perform sufficiently well to forecast values with high variability and low seasonality, thus requiring more advanced models, additional

TABLE IV
PERFORMANCE OF O_3 PREDICTION MODELS

Model	R^2	*RMSE*	*MAE*	*Accuracy (%)*
O3__LightGBM_70	**0.4718**	**8.6796**	**6.4855**	**55.11**
O3__XGBoost_90	**0.4607**	**8.7703**	**6.5682**	**54.53**
O3__ExtraTrees_70	**0.4523**	**8.8383**	**6.6185**	**54.19**
O3__RandomForest_50	**0.4132**	**9.1483**	**6.8706**	**52.44**
O3__GradientBoosting_70	0.3513	9.6190	7.5330	47.86
O3__ElasticNet_50	0.2427	10.3926	8.7501	39.43
O3__Lasso_70	0.2420	10.3973	8.7584	39.37
O3__LinearRegression_80	0.2246	10.5163	8.7929	39.14
O3__Ridge_80	0.2246	10.5163	8.7929	39.14
O3__KNN_50	-0.0112	12.0093	8.9154	38.29
O3__DecisionTree_90	-0.2254	13.2199	9.2601	35.90
O3__SVR_90	-0.2588	13.3988	11.8704	17.83
O3__AdaBoost_60	-2.5026	22.3506	20.4434	-41.51

[a]All values rounded to 4 decimal places. Accuracy is expressed as a percentage.

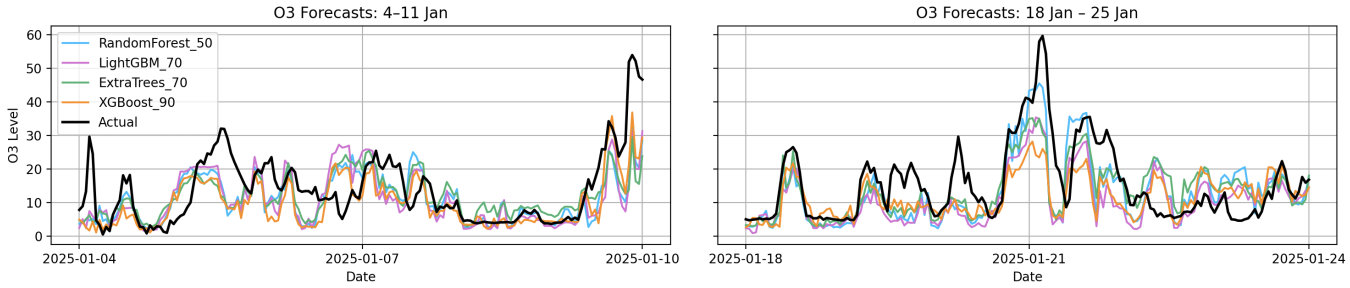

Fig. 4. Comparison of observed O_3 concentrations (black line) with forecasted values from the 4 top-performing models over the 2 snippets, each of 7-day periods in the month of Jan.

TABLE V
PERFORMANCE OF NO_2 PREDICTION MODELS

Model	R^2	RMSE	MAE	Accuracy (%)
NO2_LightGBM_50	**0.8015**	**5.4493**	**4.0730**	**86.84**
NO2_ExtraTrees_50	**0.7990**	**5.4840**	**4.1383**	**86.63**
NO2_RandomForest_50	**0.7759**	**5.7907**	**4.2828**	**86.17**
NO2_XGBoost_90	**0.7806**	**5.7292**	**4.4008**	**85.79**
NO2_KNN_50	0.7285	6.3737	4.7284	84.73
NO2_GradientBoosting_70	0.7207	6.4641	4.8540	84.32
NO2_SVR_50	0.6936	6.7708	4.9269	84.09
NO2_DecisionTree_60	0.6224	7.5160	5.3228	82.81
NO2_ElasticNet_50	0.6517	7.2193	5.3508	82.72
NO2_Lasso_50	0.6483	7.2544	5.3779	82.63
NO2_Ridge_60	0.6340	7.4003	5.5797	81.98
NO2_LinearRegression_60	0.6340	7.4003	5.5797	81.98
NO2_AdaBoost_50	0.2958	10.2647	8.5766	72.30

[a]All values rounded to 4 decimal places. Accuracy is expressed as a percentage.

predictors or hyperparameters.

E. Nitrogen Dioxide (NO_2)

The NO_2 forecasting results, shown in *Table V*, demonstrate that ensemble tree models excel there. LightGBM attained the best performance, explaining approximately 79% of NO_2 variance. ExtraTrees is a close second, followed by XGBoost and RandomForest. These top four models, plotted in Fig. 5, have high accuracy (84–86%), indicating their predictions are very close to actual NO_2 levels.

Linear models show somewhat higher errors and R^2 values, still positive, implying they capture some signal, but are clearly inferior to the ensembles. Further, AdaBoost's errors are nearly double those of the best model, yielding $R^2 = 0.295$ and accuracy 62%, making it an outlier again.

Notably, all models achieved >60% accuracy for NO_2, and most surpassed 80% of accuracy, indicating that NO_2 was comparatively easier to predict than NH_3 or O_3. In summary, LightGBM and ExtraTrees provided excellent NO_2 forecasts, while single-tree and boosting methods without sufficient ensemble diversity (DecisionTree, AdaBoost) performed significantly worse. The strong performance of ensemble methods is likely due to their ability to capture complex relationships in the NO_2 time series.

From the visual patterns in Fig. 6, we observe that NO_2 and NO consistently exhibit high R^2 values across nearly all models and training splits, suggesting that their temporal and/or spatial patterns are well-captured by both linear and non-linear regression techniques. Ensemble-based methods, particularly ExtraTrees, GradientBoosting, RandomForest, LightGBM, and XGBoost, demonstrate robust and stable performance in modeling these species, often yielding R^2 values exceeding 0.8. Even conventional linear models maintain strong predictive capabilities, indicating a relatively linear and structured relationship between the input features and the target variable.

In contrast, the models perform only moderately well on CO. While some ensemble methods such as GradientBoosting and XGBoost manage to achieve reasonable R^2 scores, most models struggle to accurately capture the variability in CO concentrations. This suggests a potentially more complex or less stable relationship between CO levels and the available predictor variables, or possibly a higher sensitivity to noise or temporal fluctuations.

NH_3, on the other hand, proves highly resistant to accurate modeling. Negative R^2 values are observed across nearly all models and data splits, signifying that these models perform worse than a simple mean-based baseline predictor. Such consistently poor performance indicates that NH_3 may exhibit a high degree of non-linearity, variability, or dependence on unmeasured or external factors, such as agricultural practices, industrial activities and soil characteristics [18], not represented in the dataset.

O_3 demonstrates an intermediate level of predictability. Model performance at lower training ratios (50–60%) is generally poor, but improves steadily as the proportion of training data increases. This trend is especially evident among ensemble methods, which begin to show moderate predictive power at higher splits, suggesting that O_3 modeling may benefit significantly from larger datasets.

V. TAKE AWAYS

The heatmap in Fig. 6 illustrates the predictive performance of various ML models on air pollutant concentrations across different data splits, measured by the R^2 score. Ensemble models—especially Extra Trees, Gradient Boosting, LightGBM, Random Forest, and XGBoost—consistently perform best, particularly for NO and NO_2, reflecting strong feature relationships. CO shows moderate predictability, while NH_3 and O_3 are harder to model, with most models producing low or negative R^2 values. Simpler models like Linear Regression, SVR, and KNN generally underperform. Overall, the heatmap

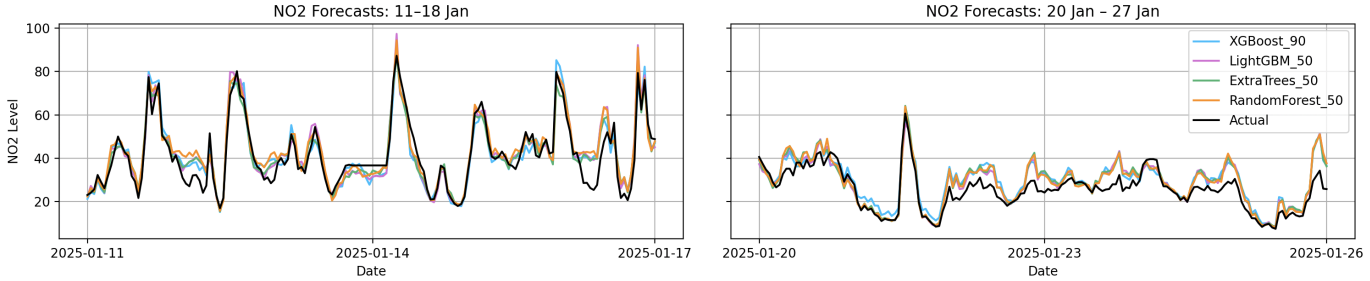

Fig. 5. Comparison of actual NO_2 concentrations with forecasts from ML models in 2 snippets of 7 days each from the month of Jan 2025. The black line represents the true NO_2 levels, while colored lines show predictions from the 4 top-performing models. The chart highlights model accuracy and their ability to capture temporal patterns and spikes in pollution.

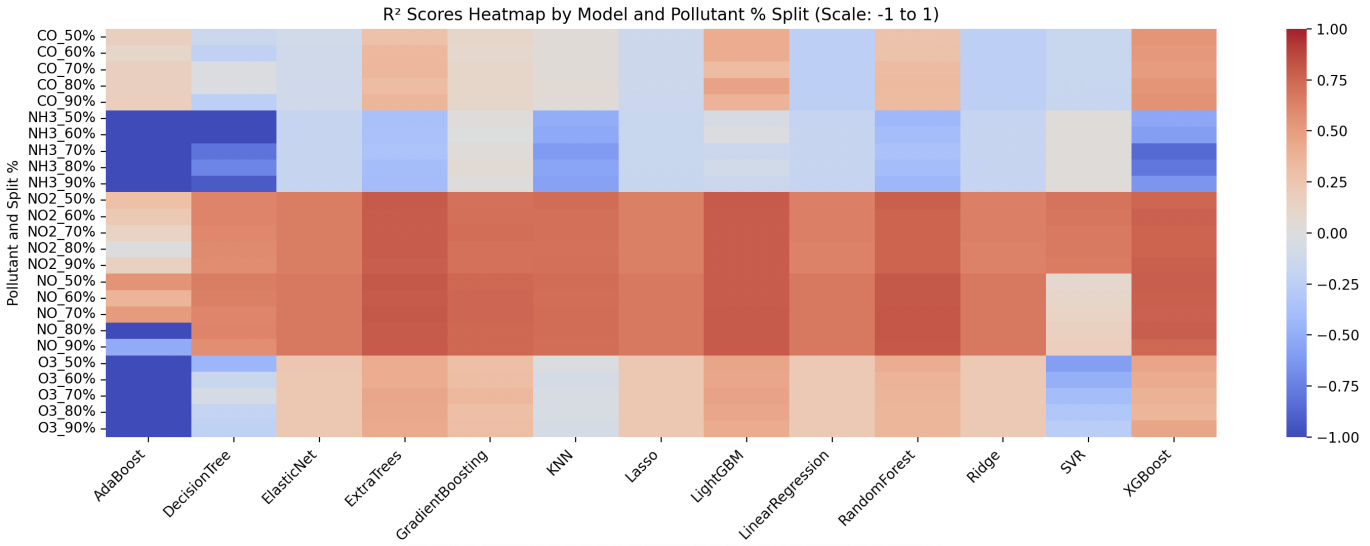

Fig. 6. Visualizing the R^2 scores of the 13 different machine learning models across the five pollutants (CO, NH_3, NO_2, O_3, NO) and the five different train/test split ratios (50% to 90%). Stronger red represents better accuracy.

underscores the strength of ensemble methods for predicting nitrogen oxides, while highlighting the challenges of modeling more variable pollutants like ammonia and ozone.

Generally, we draw several key observations:

Model selection matters. The choice of ML model has a pronounced impact on forecast accuracy. Ensemble tree-based models, especially LightGBM and Random Forest, delivered the best overall performance. These models captured complex temporal patterns, like daily cycles and sudden emission spikes, far better than simpler models. In contrast, some other techniques, such as AdaBoost and SVR, consistently underperformed and should likely be avoided for uni-variate pollutant time-series forecasting unless tailored/tuned extensively.

Training data matters, but not too much. The results of the amount of historical data needed investigation indicated that, for most pollutants and models, performance improvements beyond the 50–70% range were marginal. In several instances, optimal performance was observed at intermediate splits: for example, LightGBM achieved peak performance at 50% for NO_2 and 70% for O_3.

Generally, increasing the volume of training data did not degrade performance, apart from minor fluctuations potentially due to overfitting. These findings suggest that beyond a certain threshold, additional data primarily reinforce patterns already learned, such as weekday/weekend differences and seasonal emission variations. An important trend observed across nearly all pollutants is the positive correlation between training set size and model performance. Larger training datasets generally lead to higher R^2 values, underscoring the importance of sample size in improving generalizability and predictive stability, especially in the context of environmental modeling where noise and variability are inherent.

The pollutant matters. Different pollutants pose different forecasting challenges. NO_2, NO and O_3 showed somewhat predictable patterns, leading to higher R^2 scores, whereas NH_3 and CO were harder to predict.

This implies that we should not adopt a one-size-fits-all modeling approach. For instance, a LightGBM model can be confidently used for O_3 and NO_2 forecasting, enabling useful predictions for exposure management. However, the same model applied to CO might yield no improvement over a baseline. NH_3 forecasts remained challenging: likely due

to highly localized and intermittent emission sources, such as agricultural practices and fertilizer use, which are not captured by meteorological predictors alone. Prior work has shown that NH3 emissions are spatially heterogeneous and influenced by surface conditions and seasonality [18]. Improving predictions may require integrating spatiotemporal emission models or domain-specific proxies for agricultural activity.

Trend explanation. A deeper analysis of historical data reveals why model performance varies across pollutants. For example, NH_3 lacks a consistent daily cycle. On specific days, concentrations remain nearly constant from afternoon through the night, even if other pollutants exhibit significant variation. This behavior implies that temporal features provide little predictive value for NH_3. A slight midday decline may be linked to chemical processes in daylight, but such mechanisms were not explicitly modeled to test the algorithms impartially.

As a result, models tend to regress toward the mean, leading to large errors when actual values deviate. Consequently, models can reliably predict timing but often struggle with magnitude, leading to moderate overall performance.

Expectations. The most accurate models achieved MAE in the range of 5 to 10 ppb (parts per billion) for NO_2 and O_3, and in the order of a few tenths of a part per million (ppm) for CO. These levels of error are generally acceptable for air quality monitoring, where regulatory and public health thresholds tend to be defined in broader concentration ranges.

For instance, an average error of approximately 5 ppb for NO_2 enables reliable distinction between scenarios such as 20 ppb and 40 ppb, which is particularly useful for identifying moderate pollution events and triggering appropriate responses. Similarly, an MAE of around 5 ppb for O_3 is well within tolerable limits when considering the U.S. EPA 8-hour ozone standard of approximately 70 ppb, making the model suitable for early-warning applications.

VI. Conclusion

This study shows that combining IoT data with advanced machine learning models enables accurate forecasting of air pollutant concentrations, outperforming simpler methods. Success depends on selecting robust ensemble models—such as LightGBM or Random Forest—and tailoring them to the unique behavior of each pollutant. These findings provide practical guidance for deploying IoT-based air quality forecasting systems: use ensemble tree models for strong performance, avoid overfitting by managing training history length, and account for pollutant-specific dynamics. With such models, IoT networks can go beyond monitoring to deliver short-term forecasts, issue timely pollution alerts, and help reduce exposure risks.

References

[1] United Nations General Assembly, "Transforming our world: the 2030 Agenda for Sustainable Development," UN General Assembly Resolution A/RES/70/1, pp. 1–35, October 2015 [Adopted at the UN Sustainable Development Summit, New York, September 25, 2015].

[2] I. Essamlali, H. Nhaila, and M. El Khaili, "Supervised Machine Learning Approaches for Predicting Key Pollutants and for the Sustainable Enhancement of Urban Air Quality: A Systematic Review," *Sustainability*, vol. 16, no. 3, Art. no. 976, pp. 1–26, Jan. 2024. [Online]. Available: https://doi.org/10.3390/su16030976

[3] A. K. Rad, M. J. Nematollahi, A. Pak, and M. Mahmoudi, "Predictive modeling of air quality in the Tehran megacity via deep learning techniques," *Scientific Reports*, vol. 15, Art. no. 1367, pp. 1–18, Jan. 2025. [Online]. Available: https://doi.org/10.1038/s41598-024-84550-6

[4] M. Méndez, M. G. Merayo, and M. Núñez, "Machine learning algorithms to forecast air quality: a survey," *Artificial Intelligence Review*, vol. 57, Art. no. 10424, pp. 1–56, Feb. 2023. [Online]. Available: https://doi.org/10.1007/s10462-023-10424-4

[5] F. Gifford Jr., "An Alignment Chart for Atmospheric Diffusion Calculations," *Bulletin of the American Meteorological Society*, vol. 34, no. 3, pp. 101–105, Mar. 1953.

[6] J. Yi and V. R. Prybutok, "A neural network model forecasting for prediction of daily maximum ozone concentration in an industrialized urban area," *Environmental Pollution*, vol. 92, no. 3, pp. 349–357, 1996. [Online]. Available: https://doi.org/10.1016/0269-7491(95)00078-X

[7] Y. Liu, Q. Zhu, D. Yao, and W. Xu, "Forecasting Urban Air Quality via a Back-Propagation Neural Network and a Selection Sample Rule," Atmosphere, vol. 6, no. 7, pp. 891–907, July 2015 [Online]. Available: https://www.mdpi.com/2073-4433/6/7/891

[8] J. Kukkonen, L. Partanen, A. Karppinen, J. Ruuskanen, H. Junninen, M. Kolehmainen, H. Niska, S. Dorling, T. Chatterton, R. Foxall, *et al.*, "Extensive evaluation of neural network models for the prediction of NO_2 and PM_{10} concentrations, compared with a deterministic modelling system and measurements in central Helsinki," *Atmos. Environ.*, vol. 37, pp. 4539–4550, 2003.

[9] IBM Corporation, "IBM and the Environment Report 2015," IBM Corporate Environmental Affairs, pp. 1–52, 2015. [Online]. Available: https://www.ibm.com/ibm/environment/annual/

[10] Y. Li and R. Li, "A hybrid model for daily air quality index prediction and its performance in the face of impact effect of COVID-19 lockdown," *Process Safety and Environmental Protection*, vol. 176, pp. 673–684, June 2023. [Online]. Available: https://doi.org/10.1016/j.psep.2023.06.021

[11] Y. El Mghouchi, M. T. Udristioiu, and H. Yildizhan, "Multivariable air-quality prediction and modelling via hybrid machine learning: A case study for Craiova, Romania," *Sensors*, vol. 24, no. 5, Art. no. 1532, pp. 1–23, Feb. 2024. [Online]. Available: https://doi.org/10.3390/s24051532

[12] M. Ghahremanloo, Y. Lops, Y. Choi, J. Jung, S. Mousavinezhad, and D. Hammond, "A Comprehensive Study of the COVID-19 Impact on PM2.5 Levels Over the Contiguous United States: A Deep Learning Approach," *Atmospheric Environment*, vol. 268, p. 118944, Jan. 2022.

[13] European Space Agency, "Air pollution fluctuations over the Po Valley," ESA, Feb. 13, 2024. [Online]. Available: https://www.esa.int/Applications/Observing_the_Earth/Air_pollution_fluctuations_over_the_Po_Valley

[14] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005. [Online]. Available: https://doi.org/10.1111/j.1467-9868.2005.00503.x

[15] Scikit-learn developers, "Scikit-learn: Machine Learning in Python," Version 1.4.2, [Online]. Available: https://scikit-learn.org/stable/

[16] XGBoost Developers, "XGBoost Documentation," Version 3.0.0, [Online]. Available: https://xgboost.readthedocs.io/en/release_3.0.0/, Accessed: Apr. 8, 2025.

[17] LightGBM Developers, "LightGBM Documentation," [Online]. Available: https://lightgbm.readthedocs.io/en/stable/, Accessed: Apr. 8, 2025.

[18] Z. Luo, Y. Zhang, W. Chen, M. Van Damme, P.-F. Coheur, and L. Clarisse, "Estimating global ammonia (NH_3) emissions based on IASI observations from 2008 to 2018," *Atmospheric Chemistry and Physics*, vol. 22, pp. 10375–10388, Aug. 2022. [Online]. Available: https://doi.org/10.5194/acp-22-10375-2022