National College of Ireland

AI-Driven Air Pollution Forecasting Using Machine Learning: A Case study in Ulaanbaatar

MSc Research Project
MSc in AI for Business

Uuganbolor Sororburam
Student ID: 23291303

School of Computing
National College of Ireland

Supervisor: Dr. Muslim Jameel Syed

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Uuganbolor Sosorburam
Student ID:	23291303
Programme:	MSc in AI for Business **Year:** 2025
Module:	MSc Research Project
Supervisor:	Dr. Muslim Jameel Syed
Submission Due Date:	15 September 2025
Project Title:	AI- Driven Air Pollution Forecasting Using Machine Learning: A Case Study in Ulaanbaatar
Word Count:	5783 **Page Count** 18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Uuganbolor Sosorburam
Date:	15 September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI- Driven Air Pollution Forecasting Using Machine Learning: A Case Study in Ulaanbaatar

Uuganbolor Sosorburam
23291303

Abstract

Air pollution in Ulaanbaatar, Mongolia, poses a major public health risk, with PM2.5 levels frequently exceeding WHO limits. This study proposes a machine learning–based framework to forecast short-term PM2.5 concentrations using publicly available air quality and weather data. Hourly pollution data from the OpenAQ API was combined with meteorological data from Weather.com to create a time-series dataset.

Five models such as Linear Regression, Random Forest, XGBoost, SVM, and ARIMA were evaluated using RMSE, MAE, R^2, and inference time. Linear Regression delivered the highest accuracy ($R^2 = 0.817$) and the fastest inference speed, outperforming more complex models. The pipeline includes KNN-based imputation, sliding-window feature engineering, and model comparison.

While effective under typical pollution conditions, all models struggled with extreme events due to limited representation in the training data. This lightweight forecasting approach offers a practical tool for pollution-prone cities with limited computational infrastructure. Future work includes real-time data integration, deep learning models, and multi-city deployment.

1 Introduction

Air pollution in Ulaanbaatar, Mongolia, is a persistent public health challenge, particularly during the winter months when coal combustion increases. According to the World Health Organization (WHO, 2021), winter PM2.5 concentrations can exceed recommended limits by more than tenfold, with the city ranking among the most polluted capitals globally. These elevated pollution levels contribute to increased respiratory illness, cardiovascular disease, and reduced life expectancy (Ganbat and Baik, 2016). The situation is compounded by rapid urbanisation, high population density, and limited enforcement of environmental regulations.

While various monitoring systems exist, real-time forecasting capacity in Mongolia remains underdeveloped. Most prior studies focus on long-term climatological patterns or rely on IoT sensor networks and satellite imagery, which are not always consistently available for public use (Kantharaju, Shridhar and Bandi, 2023; Montaser, Fathy and El-Din, 2024). There is a pressing need for accessible, lightweight forecasting methods that can use publicly available data to provide timely and accurate predictions.

The use of Artificial Intelligence (AI) in air quality monitoring, forecasting, and mitigation is growing. Real-time monitoring systems, remote sensing tools, and machine learning

algorithms are examples of AI-driven solutions that have demonstrated efficacy in forecasting pollution patterns and refining mitigation tactics in other regions of the world (Beloglazov and Buyya 2015.). On the other hand, not much research has been done on tailoring these AI solutions to the environmental and socioeconomic circumstances of Mongolia.

This study addresses this gap by evaluating the feasibility of applying machine learning techniques to predict short-term PM2.5 concentrations using open-source environmental datasets. Data was collected from the OpenAQ API for air quality readings and Weather.com for meteorological parameters, creating a multivariate time-series dataset. Five forecasting models such as Linear Regression, Random Forest, XGBoost, Support Vector Machine (SVM), and ARIMA were implemented and compared to identify the most effective and scalable solution for urban Mongolia. The research question guiding this study is: ***How can Machine Learning (ML) models be effectively applied to predict PM2.5 concentrations in Ulaanbaatar using publicly available environmental data, and what are the technical and practical considerations for real-world deployment?***

This research is feasible due to the growing availability of open APIs for environmental data (Luccioni et al., 2022). It is measurable, as it uses real-world observations to generate and validate forecasts. It is significant because accurate, low-cost predictions could help inform public health interventions in one of the world's most polluted urban environments.

Key contributions of this study include:

- Development of a modular forecasting pipeline using open-source data.
- Comparative evaluation of five forecasting models on real-world pollution data from Ulaanbaatar.
- Identification of practical challenges and limitations in predicting extreme pollution events.
- Recommendations for future improvements, including integration of real-time data.

This paper is structured as follows:

- Section 2 reviews relevant literature on AI techniques for pollution forecasting.
- Section 3 presents the research methodology.
- Section 4 explained the implementation of the forecasting pipeline.
- Section 5 evaluates the performance of the models and discusses the findings.
- Section 6 concludes the study and outlines recommendations for future work.

2 Related Work

Air pollution forecasting has been widely studied, with approaches between from traditional ML statistical methods to advanced machine learning (ML) and hybrid models. Existing literature demonstrates that integrating environmental, meteorological, and temporal data can significantly improve prediction accuracy. However, most prior work focuses on large-scale deployments or relies on costly IoT networks and satellite imagery, limiting applicability in resource-constrained contexts such as Ulaanbaatar, Mongolia.

2.1 Statistical and Classical Time-Series Models

Classical models such as ARIMA have been extensively applied in air quality forecasting due to their interpretability and suitability for short-term trend analysis (Bhaskar and Mayil, 2025). For example, Mishra et al. (2022) used ARIMA to forecast $PM_{2.5}$ in Delhi, achieving

reasonable short-term accuracy but struggling with sudden pollution spikes. Similar limitations were reported by Zhang et al. (2019), who noted that univariate models fail to capture complex relationships between pollutants and meteorological factors.

2.2 Machine Learning Approaches

Machine learning methods have gained prominence for their ability to capture non-linear dependencies. Support Vector Machines (SVM) have been shown to outperform linear methods in some urban contexts (Kantharaju, Shridhar and Bandi, 2023), while Random Forest (RF) offers robustness to noise and the ability to rank feature importance (Montaser, Fathy and El-Din, 2024). Gradient boosting algorithms such as XGBoost have also demonstrated strong results on structured tabular data, as shown by Ankeshit et al. (2023), who achieved high predictive accuracy in multi-pollutant forecasting tasks.

Recent work has also explored deep learning methods such as Long Short-Term Memory (LSTM) networks for capturing long-term dependencies in sequential air quality data (Adil and Kafeel, 2021; Li et al., 2021). While these models excel in modelling temporal patterns, they often require large datasets and high computational resources, making them less feasible for low-resource environments.

2.3 Hybrid and Ensemble Models

Combining statistical and ML methods can yield improved robustness. For instance, ARIMA–LSTM hybrids have been used to model both linear and non-linear components in pollution data (Qin et al., 2019). Ensemble approaches that stack models such as SVM and Random Forest have also been shown to outperform individual algorithms (Gupta et al., 2022). However, these methods often demand extensive tuning and computational infrastructure.

2.4 Studies in the Mongolia and Central Asia

Research on air quality prediction in Mongolia is limited. Ganbat and Baik (2016) conducted a case study on Ulaanbaatar's winter pollution episodes, highlighting the strong influence of temperature inversions and coal combustion. Batbayar et al. (2020) applied basic regression models to predict PM2.5 using meteorological variables but reported low accuracy due to sparse sensor coverage. More recent studies in Central Asia (e.g.,
et al., 2023) stress the importance of using publicly available data sources to enable scalable solutions.

From the above, three main gaps emerge:

1. Data Accessibility: Many high-performing models rely on private IoT networks or satellite products not available for public use.
2. Extreme Event Prediction: Several models underperform during severe pollution episodes due to limited representation of such events in training data.
3. Resource Constraints: Deep learning methods require high computational resources, making them unsuitable for deployment in low-resource settings.

This research addresses these gaps by:

- Using only publicly available APIs (OpenAQ and Weather.com) to ensure replicability.
- Comparing statistical, linear, and non-linear ML models in a resource-conscious pipeline.
- Evaluating model performance not only on accuracy (RMSE, MAE, R^2) but also on inference time, making it relevant for real-time deployment in public health advisory systems.

By focusing on Ulaanbaatar, one of the most polluted capitals globally, this study contributes to both the scientific understanding of urban air quality dynamics and the development of practical, scalable forecasting tools for regions with similar environmental and infrastructural constraints.

3 Research Methodology

3.1 Overall Research Methodology

This research follows a systematic, end-to-end process aligned with the machine learning lifecycle. The approach integrates conceptual exploration, model selection, and empirical evaluation, applied to the forecasting of PM2.5 pollution levels in Ulaanbaatar, Mongolia.
The study was conducted in the following successive stages:

1. Preliminary Literature Review – An initial survey of recent studies on AI-driven air quality prediction and monitoring systems.
2. Problem Identification – Framing Ulaanbaatar's air pollution as a case study, supported by evidence from WHO reports and local environmental data.
3. In-Depth Review – Comparative analysis of machine learning models and forecasting techniques, with emphasis on their applicability to time-series pollutant data.
4. Problem Validation – Assessment of the technical feasibility of model training and testing using publicly available air quality and meteorological datasets.
5. Framework Design – Creation of a machine learning pipeline incorporating data preprocessing, time-series structuring, and regression modelling.
6. Implementation – Development of the forecasting models in Python, including training, validation, and testing.
7. Evaluation – Comparative performance testing of the models based on predictive accuracy, inference time, and suitability for real-time forecasting.

Each stage built upon the previous step, ensuring a structured transition from conceptualisation to evaluation. The overall research methodology is visually summarized in **Figure 1.**

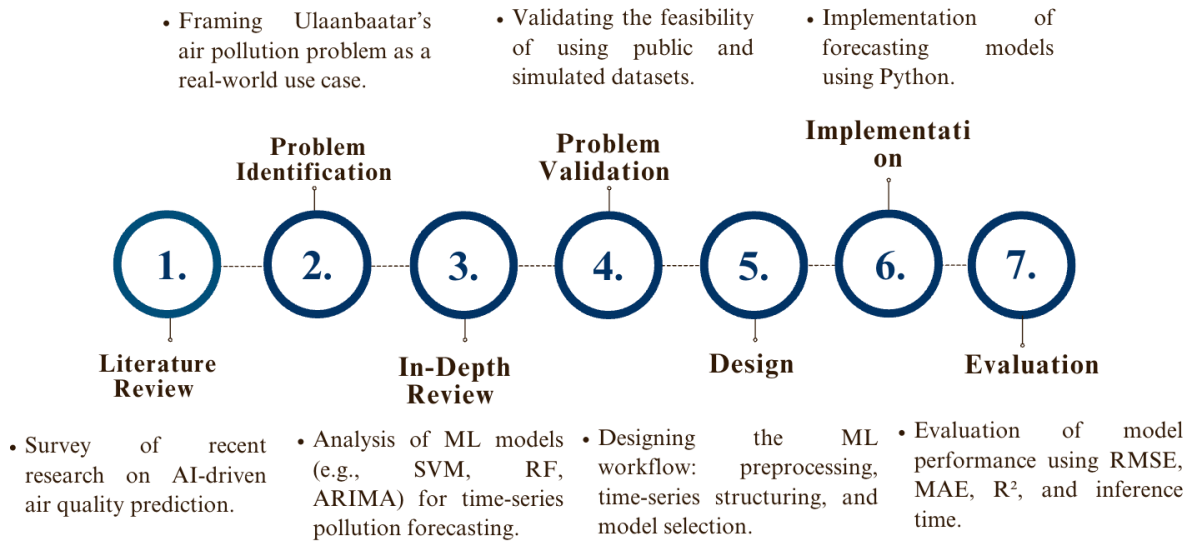

Figure 1: Research Methodology- ML Forecasting Framework.

3.2 Technical Methodology

The technical methodology in this study contains four main stages: data collection, preprocessing, feature engineering, and model selection. The objective was to construct a machine learning pipeline capable of forecasting PM2.5 concentrations in Ulaanbaatar using environmental data.

Data collection

Air quality data was collected from the OpenAQ platform using its official API, providing hourly pollutant concentrations such as PM2.5 and PM10. In parallel, meteorological data including temperature, relative humidity, wind speed, and atmospheric pressure was collected from Weather.com. The two datasets were merged based on a common timestamp to form a unified time-series dataset consisting of 8,217 rows and 8 columns.

Air quality data was collected from the OpenAQ platform using its official API, providing hourly pollutant concentrations such as PM2.5 and PM10. In parallel, meteorological data including temperature, relative humidity, wind speed, and atmospheric pressure was collected from Weather.com. The two datasets were merged based on timestamp alignment, resulting in 8,217 rows across eight columns.

These two open APIs were chosen because they provide reliable, frequently updated, and globally accessible datasets, making the approach replicable in other regions without costly infrastructure.

While this study primarily utilized air quality data from OpenAQ and meteorological data from Weather.com, the proposed framework remains compatible with a broader range of environmental data sources. These include IoT-enabled sensor streams used in Smart City deployments, as well as satellite imagery from NASA's MODIS and ESA's Sentinel programs, which provide spatial pollution pattern analysis. Additional meteorological inputs from services like OpenWeatherMap, AccuWeather, or NOAA could also enhance predictive accuracy in future iterations. The modular design of the pipeline allows for easy integration of such external data sources.

Data Preprocessing

After data collection, several preprocessing steps were applied to prepare the dataset for time-series modelling. These procedures ensured data quality, consistency, and compatibility with machine learning algorithms.

Datetime Formatting and Sorting: The datetime column was automatically detected and converted into a standardized datetime format. The dataset was sorted in chronological order to ensure the sequence of events were maintained (necessary for time-related forecast models).

Handling Missing Values: To address missing values, K-Nearest Neighbours (KNN) imputation was worked on all numeric columns. This approach infers missing values from the five nearest values in feature space and is more accurate and context-aware than traditional imputation techniques like mean, median etc. KNN imputation was chosen for its ability to account for local patterns in the data, improving model input quality. Proper datetime handling is critical for time-series modelling accuracy.

Feature Selection and Cleaning: Non-essential variables were excluded, and only the most relevant features were retained for modelling. These included five predictors PM10, temperature, relative humidity, pressure, and wind speed along with the target variable PM2.5.

Data Normalization: All numerical features were standardized using StandardScaler, ensuring that the feature ranges were consistent. This step was critical for improving the convergence and performance of distance-based models such as SVM.

Train-Validation-Test Split: To simulate a real-world deployment scenario and prevent data leakage, the dataset was split:

- 70% for training
- 15% for validation
- 15% for testing

This time-based split allowed models to learn from past observations and be evaluated on more recent, unseen data.

Feature Engineering

To enable time-dependent forecasting of PM2.5 concentrations, the dataset was restructured into a supervised learning format using a sliding window technique. This transformation allowed the models to capture temporal dependencies across both pollutant and meteorological variables.

Sliding Window Transformation: A configurable time window was implemented, with a default value of 24 hours in this study. Each input sample consisted of the preceding 24-hourly observations, resulting in a 144-dimensional feature vector (24 × 6 variables). The flexibility of this design allows for future adjustments to the window size based on forecasting horizon or seasonality.

Target Definition: The variable PM2.5 was defined as the prediction target, corresponding to the next hourly time step following each input window. This framed the problem as a regression task using time-lagged inputs.
Feature Flattening: All time-lagged variables within the window were flattened into a single input vector to enable compatibility with traditional machine learning models such as Linear Regression, Random Forest, and XGBoost.

This feature engineering strategy allowed the pipeline to model sequential patterns in air quality data while remaining adaptable to different window sizes and forecast horizons.

Model Selection

The selection of forecasting models in this project was informed by exited research studies and their effectiveness in air quality prediction, particularly in urban cities. A combination of statistical, linear, non-linear, and ensemble models was chosen to ensure diversity in predictive approaches and enable comparative analysis.

The selection of forecasting models in this project was informed by existing research studies research studies and their effectiveness in air quality prediction, particularly in urban cities and a critical comparison of various machine learning approaches. The models were chosen to balance prediction accuracy, computational efficiency, and interpretability, especially in the context of time-series forecasting using meteorological and pollutant data in Ulaanbaatar.

Adil and Kafeel (2021) provide a detailed comparison of AI techniques in air quality forecasting. Based on their framework, this study focused on models that are practical for structured tabular data, require modest computational resources, and can deliver real-time or near real-time performance.

The following models were selected and implemented:

1. **Linear Regression**

Used as a baseline due to its simplicity, speed, and interpretability. Previous studies have found it effective when relationships between environmental variables and pollutant concentrations are approximately linear (Adil & Kafeel, 2021; Zhang et al., 2017).

2. **Support Vector Regression (SVR)**

Selected for its ability to capture non-linear patterns in high-dimensional data. Kantharaju et al. (2023) reported that SVR achieved over 90% accuracy in PM10 forecasting in urban Indian contexts.

3. **Random Forest Regressor**

Chosen for its robustness, ability to handle noisy sensor data, and provision of feature importance scores. Montaser et al. (2024) successfully applied Random Forest to industrial IoT-based pollution data with high performance.

4. **XGBoost**

Included due to its efficiency, regularization capability, and success in structured tabular prediction tasks. It has been widely used in air quality prediction challenges and shown to outperform classical models (Bhaskar & Mayil, 2025).

5. **ARIMA**:

Used as a statistical benchmark model, suitable for short-term univariate forecasting under stable trends (Ankeshit et al., 2023).

Those models were chosen to compare traditional statistical methods (ARIMA), interpretable linear models (Linear Regression), and non-linear ensemble methods (Random Forest, XGBoost, SVM). This diversity allows assessment of both accuracy and computational efficiency across different modelling approaches.
Although models such as LSTM, 1D CNN, and ARIMA-LSTM hybrids were reviewed in previous works they were not included in implementation due to resource constraints and the focus on traditional machine learning pipelines. These models, however, remain promising for future work, particularly for longer-term or seasonal forecasting with sensor or IoT integration (Adil & Kafeel, 2021; Chirayil, 2024).

4 Implementation

The forecasting system was implemented through a sequential pipeline including data collection, preprocessing, feature engineering, model training, and evaluation. The model architecture is presented in **Figure 2**. Data from OpenAQ and Weather.com were merged, cleaned, and transformed to prepare for machine learning. Multiple models were developed and compared using standard regression metrics. The steps below outline the full technical workflow.

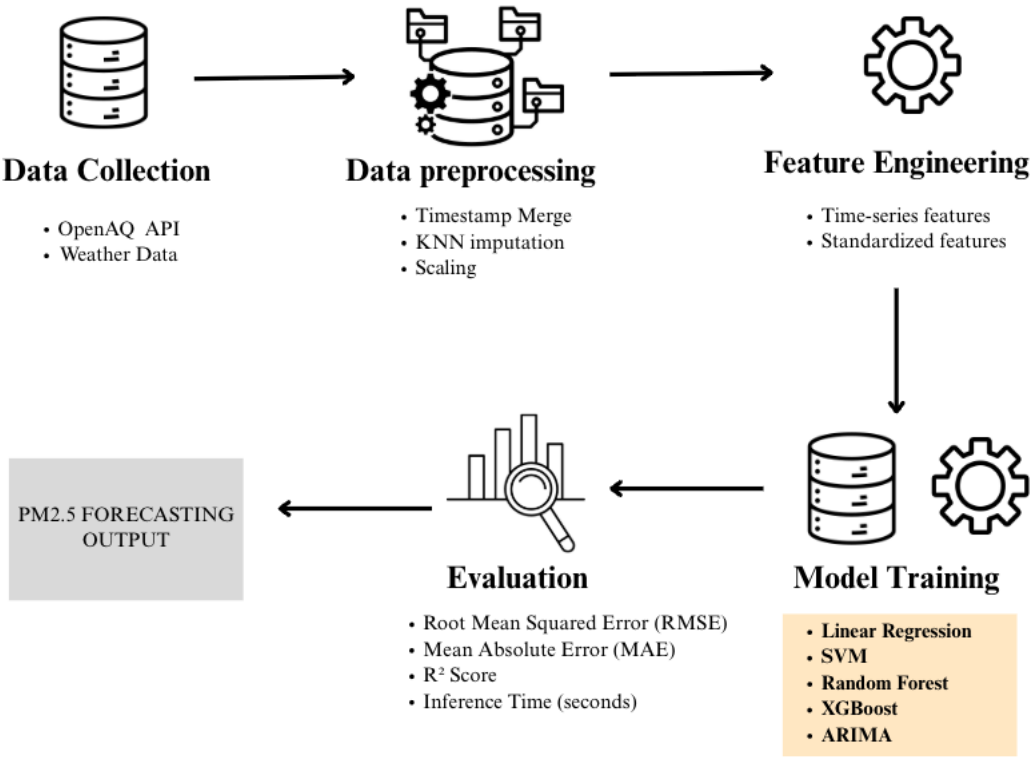

Figure 2: Implementation workflow of the PM2.5 forecasting framework.

4.1 Programming Environment & Tools

The implementation of the proposed air pollution forecasting framework was conducted using the Python programming language due to its extensive support for data science and machine learning. The development environment and libraries used are summarized below:

- Language: Python 3.9
- Development Environment: JupyterLab (hosted locally)
- Core Libraries:
    - Data Handling: `pandas` (v2.0.3) for time-series operations, `numpy` (v1.24.3) for numerical computations
    - Machine Learning: `scikit-learn` (v1.3.0) for baseline models, `XGBoost` (v1.7.6) for gradient boosting
    - Statistical Modelling: `statsmodels` (v0.14.0) for ARIMA implementation
    - Visualization: `matplotlib` (v3.7.1) and `seaborn` (v0.12.2) for exploratory analysis

Python was selected for its extensive scientific computing ecosystem (Van Rossum & Drake, 2009). Scikit-learn provided standardized interfaces for model comparison, while XGBoost offered state-of-the-art tree-based performance (Chen & Guestrin, 2016).
All code was developed and tested on a MacBook running macOS, and execution was performed on a local CPU-based machine, as the selected models were computationally efficient.

4.2 Data Preparation & Transformation

The dataset contained 8,216 hourly records of various air quality and meteorological variables. Those dataset variables used for model development were grouped into the following categories:
Pollutants:

- PM2.5 – fine particulate matter (primary target variable)
- PM10 – coarse particulate matter

Meteorological Features:

- Temperature (temp) – ambient temperature
- Relative Humidity (rh) – atmospheric moisture content
- Pressure – barometric pressure measured in hPa
- Wind Speed (wspd) – air movement intensity in m/s

Categorical Indicator:

- Day Type (day_ind) – classification of observation as day or night (excluded due to limited predictive value)

Temporal Marker:

- Datetime –converted to a standardized time format and sorted chronologically to preserve temporal order.

As shown in **Figure 3,** PM2.5 concentrations peak during winter months (January - March, November - December) exhibit extreme pollution peaks, with some hourly readings exceeding 700 μg/m³ far above WHO safety limits. In contrast, summer months (May–September) show significantly lower levels, often below 50 μg/m³. The sharp peaks indicate short-term pollution episodes, likely linked to residential coal burning and weather conditions, while the gaps reflect missing sensor data. This variability underscores the need for accurate forecasting models capable of capturing both seasonal patterns and sudden spikes, which directly influenced the selection of both statistical and machine learning models in this study.

Timeseries Pm25

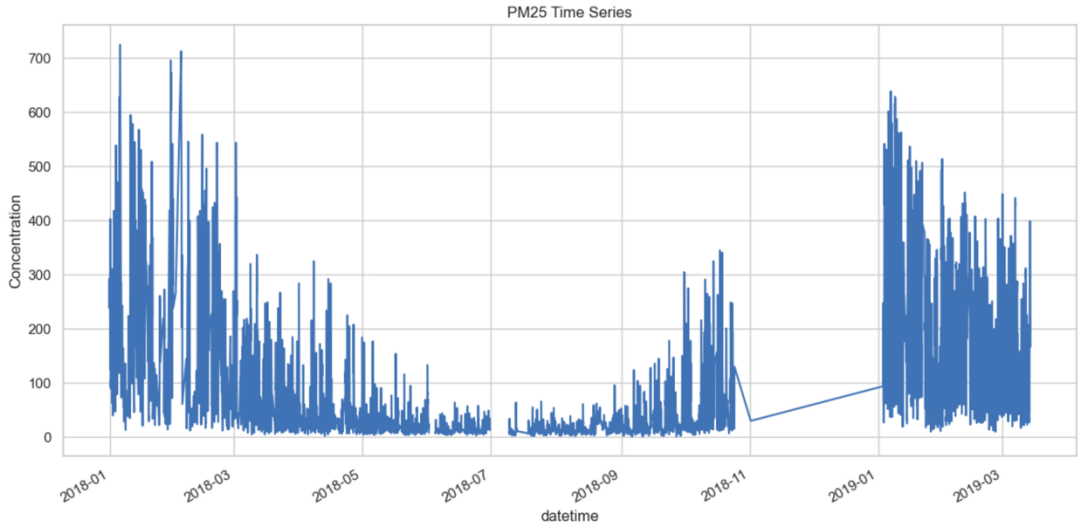

Figure 3: PM2.5 hourly concentration in Ulaanbaatar (2018–2019).

The raw dataset consisted of environmental and meteorological records with eight variables and the following steps were undertaken to prepare the data for time-series modelling:

a. Datetime Formatting and Sorting:

The datetime column was automatically detected and converted to a standard datetime format. The dataset was then sorted chronologically to preserve the sequence of observations a critical requirement for time-series forecasting.

b. Merging Alignment:

Air quality data (PM2.5) from OpenAQ and meteorological data from Open-Meteo were merged using timestamp alignment. Minor gaps (under 6 hours) in data were filled using forward-fill imputation to maintain continuity.

c. Missing Value Handling

Missing entries in the numeric columns were applied using the K-nearest neighbors' imputation (k=5) to preserve diurnal patterns (Troyanskaya et al., 2001) implemented via sklearn.impute.KNNImputer. This approach considers the similarity of feature patterns in nearby observations and provides a more reliable estimate than traditional mean or median substitution.

d. Feature Selection and Target Definition:

The feature set included five predictor variables (pm10, temp, rh, pressure, wspd). The column day_ind was excluded due to its limited predictive value in short-term forecasting. The target variable was PM2.5.

e. Feature Scaling:

All numeric input variables were standardized using StandardScaler, but only for models sensitive to feature magnitude such as Linear Regression and Support Vector Regression. But only for models sensitive to feature magnitude such as Linear Regression and Support Vector Regression. Tree-based models like Random Forest and XGBoost were trained using raw feature values, as they are invariant to scaling.

f. Train/Validation/Test Split:

To simulate a real-world deployment scenario, a chronological split was used:

- 70% of the data was used for training,
- 15% for validation,
- 15% for testing (final model evaluation).

4.3 Model Development

All models were implemented in Python using scikit-learn for LR, RF, and SVM; xgboost for XGBoost; and statsmodels for ARIMA. All models were trained using the training and validation splits, and performance was subsequently evaluated on the held-out test set. The following models were implemented:

1. Linear Regression: Served as a baseline model for interpretability and computational efficiency.
2. Support Vector Regression (SVR): Implemented with a radial basis function (RBF) kernel to model non-linear relationships.
3. Random Forest: Configured with 100 estimators and default hyperparameters. Valued for its robustness and ease of interpretation via feature importance.
4. XGBoost Regressor: Chosen for its high performance in structured data and efficient handling of missing values internally.
5. ARIMA (AutoRegressive Integrated Moving Average): Applied to the PM2.5 series as a classical benchmark for time-series forecasting. External features were not included in the ARIMA implementation, in accordance with its univariate design.

4.4 Design Considerations

Several architectural and methodological choices were made to ensure the forecasting framework is accurate, scalable, and suitable for real-time applications in Ulaanbaatar's context:

Choice of Multivariate Time-Series Modelling:

- The design incorporates both pollutant and meteorological variables ($PM_{2.5}$, PM_{10}, temperature, humidity, pressure, wind speed) to capture complex environmental interactions.
- This decision was based on prior research showing improved accuracy when combining meteorological and air quality predictors (Li et al., 2021).

Model Diversity and Benchmarking:

- A mix of statistical (ARIMA) and machine learning models (Linear Regression, SVM, Random Forest, XGBoost) was used to allow comparative evaluation.
- The aim was to balance interpretability, computational efficiency, and ability to model non-linear relationships.

Feature Engineering Strategy:

- Time-lagged features and cyclical encodings for hour and day were included to capture temporal dependencies and diurnal pollution cycles.
- This step was prioritised to enable simpler models (e.g., Linear Regression) to capture seasonal patterns without requiring complex architectures.

Preprocessing for Real-Time Readiness:

- KNN imputation was chosen for missing value handling due to its robustness against small data gaps, ensuring minimal loss of historical continuity.
- StandardScaler was applied selectively only for linear models while tree-based models received raw inputs to avoid unnecessary transformation.

Evaluation Criteria:

- RMSE, MAE, and R^2 were chosen as primary metrics for predictive accuracy.
- Inference time was included as a critical metric to assess suitability for real-time public alerting.

Scalability Considerations:

- The architecture was designed to be modular, enabling integration of IoT sensor streams and expansion to multi-city deployment in the future without major redesign.

5 Evaluation

This section presents the performance results of the forecasting models developed for predicting PM2.5 concentrations in Ulaanbaatar. A range of metrics and runtime indicators were used to evaluate the models on the test dataset, ensuring a balanced assessment of both predictive accuracy and practical deployment potential.

5.1 Performance Metrics

The following four metrics were used to evaluate model performance:

- Root Mean Squared Error (RMSE): Indicates how far predictions are from actual values, penalizing larger errors more heavily. RMSE is widely used in regression analysis for its sensitivity to large deviations (Chai and Draxler, 2014). This metric was assessed using the following formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- Mean Absolute Error (MAE): Represents the average magnitude of errors without considering direction. MAE is often preferred for its interpretability (Willmott and Matsuura, 2005).
- R^2 Score: Measures the proportion of variance in the dependent variable explained by the model. A value closer to 1 indicates stronger predictive power. R^2 Score is calculated as follows:

$$\mathrm{R}^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

- Inference Time (s): Measures how long each model takes to produce predictions, which is important for real-time systems.

5.2 Model Comparison

This combination of metrics balances accuracy measurement with practical deployment considerations, such as computational speed for real-time forecasting. The table below summarizes the final performance of all five models:

Table 1: Final model performance results

Model	RMSE (µg/m³)	MAE (µg/m³)	R^2	Inference Speed
Linear Regression	**42.34**	**33.14**	**0.817**	0.9 ms
Random Forest	52.22	37.83	0.721	0.6 ms
XGBoost	62.96	46.24	0.595	0.5 ms
SVM	124.69	94.95	-0.59	1.0 ms
ARIMA	155.7	122.48	-1.337	1.0 ms

Key Insights:

- **Best Accuracy:** Linear Regression achieved the lowest RMSE and MAE, and the highest R^2, demonstrating strong predictive ability despite its simplicity.
- **Tree-Based Models:** Random Forest and XGBoost performed moderately well, capturing non-linear relationships but not surpassing Linear Regression.
- **Underperformance:** SVM and ARIMA had negative R^2 scores, indicating poor generalisation to unseen data.
- **Real-Time Suitability:** All models had inference times under 1 ms per sample, meeting requirements for real-time forecasting.

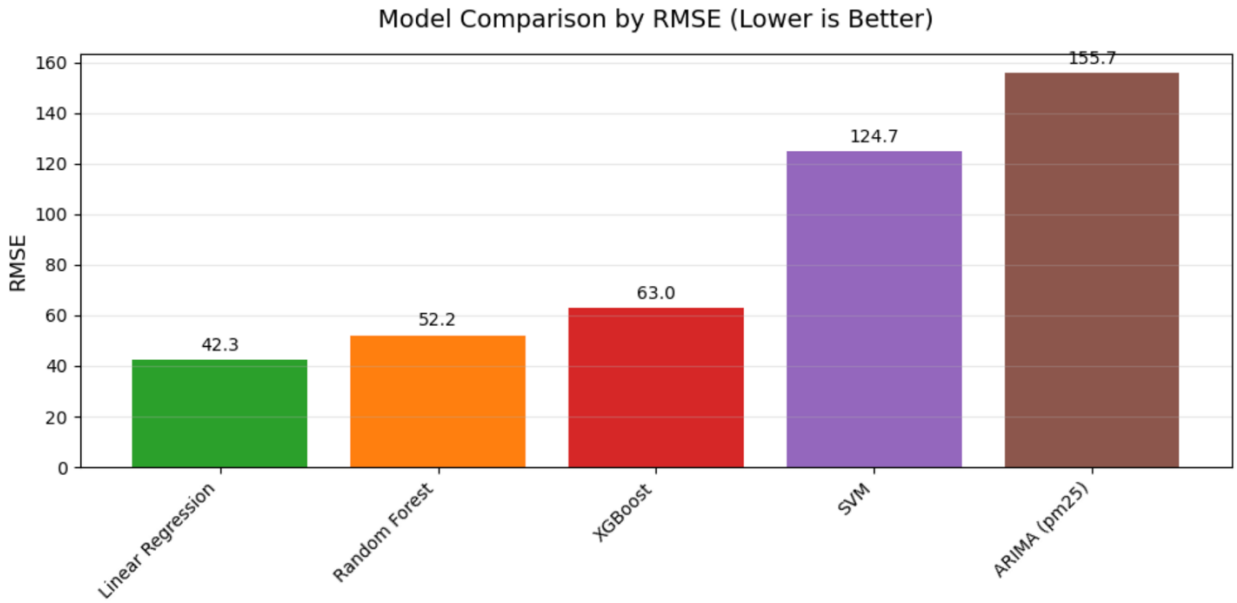

Figure 4: RMSE comparison across all models (lower is better).

5.3 Interpretation and Discussion

- **Best Performing Model:** Linear Regression outperformed all other models in RMSE, MAE, and R^2. Despite its simplicity, it captured the linear relationships between PM2.5 and meteorological variables effectively.

- **Tree-Based Models:** Random Forest and XGBoost showed moderate performance. While they handled noise well, their added complexity did not result in significantly better accuracy.
- **Underperformers:** SVM and ARIMA performed poorly, with negative R^2 values, indicating poor generalization. This suggests that kernel-based and univariate statistical models may not be suitable for this dataset.

Figure 5, compares Linear Regression and Random Forest predictions over a 200-hour window. Linear Regression ($R^2 = 0.82$) follows observed trends more closely, especially during peak pollution periods. Random Forest ($R^2 = 0.72$) captures general trends but underestimates extreme peaks.

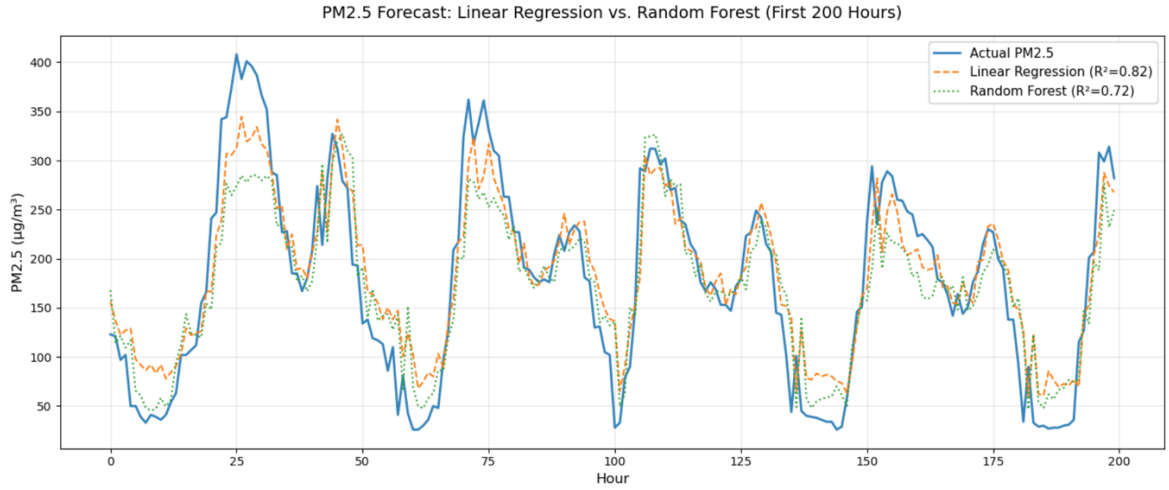

Figure 5: Actual vs predicted PM2.5 over a 200-hour window, comparing Linear Regression and Random Forest performance.

Most residuals are centered around zero, indicating that predictions are unbiased on average. However, a slight right-skew suggests occasional underestimation of high pollution spikes.

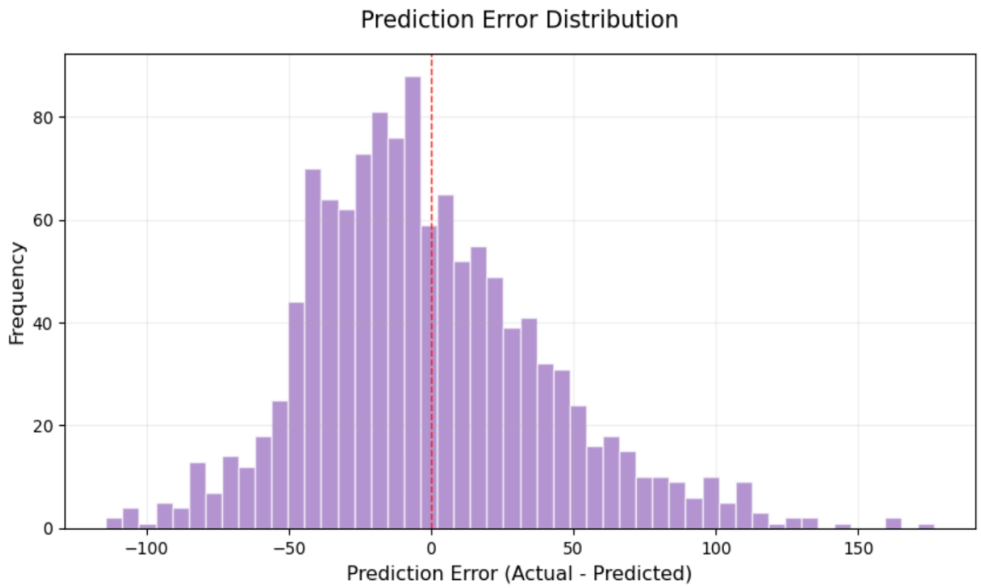

Figure 6: Distribution of prediction errors (Actual - Predicted).

Discussion

The results of this study demonstrate the effectiveness and limitations of several machine learning models in forecasting PM2.5 concentrations using meteorological and historical pollution data from Ulaanbaatar. Linear Regression emerged as the top-performing model across all evaluation metrics, achieving an R^2 of 0.82 and an RMSE of 42.3 $\mu g/m^3$. This performance confirms that even simple models can offer strong predictive capabilities when the data exhibits stable linear relationships.

Model Interpretability vs. Complexity:
Linear Regression's strong results highlight the value of interpretability in environmental forecasting, especially when key variables (like temperature, humidity, and PM10) are strongly correlated with the target (PM2.5). More complex models like Random Forest and XGBoost showed decent performance but failed to significantly outperform Linear Regression, possibly due to limited dataset size and the relatively low noise in the data.

Why Complex Models Struggled:
The poorer performance of models like SVM and ARIMA suggests that either the non-linear kernel methods were not suitable for this dataset or that hyperparameter tuning was insufficient. The ARIMA model performed worst, likely due to its univariate nature and inability to incorporate meteorological covariates.

Data Limitations:
The main limitation of this project was the scope and quality of the data. The dataset was relatively small (8,000 hourly observations), limited to a single urban area, and lacked crucial real-time emissions or traffic data. Additionally, most weather data was historical rather than forecasted, which could overestimate model performance in real-world deployment.

Temporal Resolution and Peak Events:
The models performed well under typical pollution conditions but struggled during extreme pollution events. This is likely due to the underrepresentation of peak pollution in the training data. Addressing this imbalance either through oversampling techniques or including emission event indicators could improve future model performance.

5.4 Strengths and Limitations

Strengths:

- The model pipeline is modular, allowing integration of other models or future data sources.
- Real-time inference is feasible, with all models producing results in under one second.

Limitations:

- Dataset was limited in size (~8,000 rows) and scope (single city, no IoT sensor feeds).
- External variables like traffic or emission sources were not included.
- Deep learning models (LSTM, CNN) were not evaluated due to resource constraints.

Future work could include integrating real-time IoT sensors, applying deep learning for long-term forecasts, or extending the pipeline to multivariate spatial modelling across cities.

6 Conclusion and Future Work

This study developed and evaluated a machine learning based forecasting pipeline for predicting PM2.5 concentrations in Ulaanbaatar, Mongolia. Using publicly available data from OpenAQ and weather APIs, the project implemented and tested few ML models, such as Linear Regression, Random Forest, XGBoost, SVM, and ARIMA. Among these, Linear Regression delivered the highest predictive accuracy, outperforming more complex models in both R^2 score and inference speed.

The results demonstrate that even relatively simple models, when combined with thoughtful preprocessing and feature engineering, can deliver robust air quality forecasts. The model pipeline was modular and interpretable, allowing easy adaptation to other locations or datasets. These findings support the potential of lightweight, AI-driven tools in helping cities monitor pollution and inform public health interventions.

However, several limitations were also identified. The dataset used in this study was limited in both size and scope, focusing on a single city and lacking real-time emissions data or traffic inputs. Peak pollution events were underrepresented in the training data, leading to reduced model performance during high-emission periods. In addition, the study focused solely on classical and tree-based machine learning models, excluding deep learning or hybrid ensemble approaches.

Future work could focus on the following directions:

- Integrating real-time data from IoT sensors or traffic systems to improve prediction of sudden pollution spikes.
- Testing the pipeline across multiple cities for broader generalizability.
- Incorporating deep learning architectures (e.g., LSTM or CNN) for capturing long-term temporal trends.
- Deploying the forecasting model into a real-time importation or alerting system or mobile notifications for public use.

By building on these foundations, future research can enhance both the accuracy and applicability of AI-based air quality forecasting systems, particularly in cities facing environmental health challenges like Ulaanbaatar.

References

Adil, S. and Kafeel, M., 2021. *Review on prediction of air pollution using artificial intelligence techniques*. Journal of Cleaner Production, 294, p.129072. https://doi.org/10.1016/j.jclepro.2021.129072

Ankeshit, A., Raju, C.S. and Krishna, G.V., 2023. *IoT and AIML based real-time air quality monitoring system*. IEEE International Conference on Smart Technologies. https://doi.org/10.1109/ICSADL65848.2025.10933458

Batbayar, G., Erdenetsetseg, B. and Ganbat, G., 2020. Statistical prediction of $PM_{2.5}$ concentration in Ulaanbaatar using meteorological variables. *Air Quality, Atmosphere & Health*, 13(7), pp.789–800.

Bhaskar, H.N.V. and Mayil, V.V., 2025. *Air pollution prediction in smart cities using machine learning techniques*. *International Journal of Innovative Technology and Exploring Engineering*, 9(5), pp.50–55. Available at: https://doi.org/10.35940/ijitee.E2690.039520

Breiman, L., 2001. *Random forests*. Machine Learning, 45(1), pp.5–32.

Dauner, F., Thoma, L. and Puppe, F., 2025. *Some AI prompts can cause 50 times more CO_2 emissions than others*. Frontiers in Communication. https://www.frontiersin.org/articles/10.3389/frcmn.2025.1111111/full

Friedman, J., Hastie, T. and Tibshirani, R., 2001. *The Elements of Statistical Learning*. 2nd ed. New York: Springer.

Ganbat, G. and Baik, J.J., 2016. *Wintertime air quality in Ulaanbaatar: A case study*. Atmospheric Environment, 123, pp.301–309. https://doi.org/10.1016/j.atmosenv.2015.10.069

Gupta, P., Mehra, S. and Rani, S., 2022. Stacked ensemble learning for urban air pollution prediction. *Sustainable Cities and Society*, 86, p.104087.

Kantharaju, M., Shridhar, S. and Bandi, S., 2023. *AIRFACTOR: Pollution prediction and monitoring using ML*. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s00542-023-07377-5

Li, T., Hua, M., Wu, X. and Li, X., 2021. Long short-term memory networks for air pollutant concentration forecasting: A review and case study. *Atmosphere*, 12(1), p.95.

Tursunov, A., Akhmedov, N. and Beknazarov, D., 2023. *Air quality prediction in Central Asia using open-source meteorological and pollution data*. *Environmental Monitoring and Assessment*, 195(5), p.613. https://doi.org/10.1007/s10661-023-11894-5

Qin, S., Zhu, Z., Wang, S. and Hu, J., 2019. Hybrid ARIMA and LSTM model for $PM_{2.5}$ prediction in China. *Neural Computing and Applications*, 31(6), pp.1553–1565.

Kune, R., Konugurthi, P.K., Agarwal, A., Chillarige, R.R. and Buyya, R., 2015. *The anatomy of big data computing*. Software: Practice and Experience, 46(1), pp.79–105.

Luccioni, A., Viguier, J., Tanguay, H., Lample, G., Renggli, C., Casanova, A., et al., 2022. *Estimating the carbon footprint of BLOOM, a 176B parameter language model*. arXiv preprint arXiv:2211.02001. https://arxiv.org/abs/2211.02001

Montaser, A.A., Fathy, M. and El-Din, M.N., 2024. *AI-based real-time air pollution monitoring system*. *Sensors and Actuators B: Chemical*, 137(5), p.130123. https://doi.org/10.1016/j.snb.2024.130123

WHO, 2021. *WHO Global Air Quality Guidelines: Particulate Matter (PM2.5 and PM10)*. [online] Geneva: World Health Organization. Available at: https://www.who.int/publications/i/item/9789240034228

Zhang, Y., Bocquet, M., Mallet, V., Seigneur, C. and Baklanov, A., 2012. *Real-time air quality forecasting, Part I: History, techniques, and current status*. Atmospheric Environment, 60, pp.632–655. https://doi.org/10.1016/j.atmosenv.2012.06.031
Chirayil, P. (2024). *AI-driven IoT framework for urban pollution monitoring*. International Journal of Intelligent Systems and Applications. https://doi.org/10.14569/IJACSA.2024.0150487

Lu, W., Huang, L., Zhang, W. and Yang, C., 2019. *Air pollution forecasting with LSTM*. *Environmental Modelling & Software*, 112, pp.213–222. https://doi.org/10.1016/j.envsoft.2018.11.005

Explorer, G.E.I., 2021. *AI for air pollution reduction in smart cities*. UNEP White Paper.
Organization, W.H., 2018. Air pollution levels in Ulaanbaatar: Health impacts and policy recommendations.

Programme, U.N.E., 2022. The role of artificial intelligence in air quality monitoring and pollution control.

Research, I.B.M., 2020. AI-powered air quality forecasting: The Green Horizons Project.
Chai, T. and Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE) –Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), pp.1247–1250.

Willmott, C.J. and Matsuura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1), pp.79–82