# Karachi AQI Prediction System: A Comprehensive Research Report

*Submitted By: Syed Uzair Hussain*

## Executive Summary

This report documents the development of a production-grade Air Quality Index (AQI) prediction system for Karachi, Pakistan. The system employs machine learning to forecast AQI values 24, 48, and 72 hours ahead using historical pollution data, meteorological features, and advanced feature engineering. The final model achieves strong predictive performance with R² scores above 0.75 for all forecast horizons. The project files consist of two main components: **pearlsaqipredictor.ipynb**, which contains the exploratory data analysis and model experimentation work, and **finalaqipredictor.ipynb**, which implements the modular pipeline architecture for production deployment.

## 1. Introduction

Air pollution poses significant health risks in urban environments, particularly in rapidly growing cities like Karachi. The city experiences severe air quality degradation, especially during winter months when temperature inversions trap pollutants close to the ground. Accurate AQI forecasting enables proactive health advisories for vulnerable populations, supports urban planning and traffic management decisions, and raises public awareness about environmental conditions. The primary objective of this project was to develop a multi-horizon AQI forecasting model capable of predicting air quality 24, 48, and 72 hours in advance, integrate real-time data sources for continuous predictions, implement a production-ready system with automated pipelines, and create an intuitive dashboard for stakeholder access. The methodology follows a standard machine learning workflow encompassing data collection from multiple APIs, exploratory data analysis and visualization, feature engineering with domain knowledge, model selection through comparative analysis, and production deployment with MLOps practices.

## 2. Data Collection and Integration

The foundation of any predictive system lies in the quality and comprehensiveness of its data sources. For this project, data was collected from two primary sources to capture both pollution levels and meteorological conditions. Air pollution data was obtained from the OpenWeather Air Pollution API, which provides comprehensive historical data for pollutant concentrations. The API was queried to retrieve approximately 1000 days of historical data at hourly temporal resolution, covering major pollutants including carbon monoxide, nitrogen oxides, ozone, sulfur dioxide, particulate matter (both PM2.5 and PM10), and ammonia. The API also provides a basic European Air Quality Index on a 1-5 scale, though this was later replaced with the more granular US EPA AQI standard for better prediction resolution.

Meteorological data was collected from the Meteostat Python library, which aggregates weather observations from multiple sources worldwide. The weather data was synchronized with the pollution data to maintain temporal consistency, and included key variables such as temperature, relative humidity, wind speed, and atmospheric pressure. Several meteorological variables were deliberately excluded from the analysis including dew point, snow depth, wind gusts, sunshine duration, weather condition codes, wind direction, and precipitation. These variables were dropped either due to high missingness rates in the Karachi dataset or because they showed minimal correlation with AQI patterns in preliminary analysis. The focus on core meteorological variables helped reduce noise while maintaining the most predictive weather information.

The geographic focus of the study was Karachi, Pakistan, with coordinates dynamically retrieved using the OpenWeather Geocoding API. The city's coordinates (24.8607° N, 67.0011° E) were used to query both pollution and weather data sources. Integrating data from multiple sources with slightly different timestamp conventions required careful merging strategies. The merge was performed using pandas' merge_asof function with a one-hour tolerance window, which performs a nearest-neighbor join on timestamps. This approach accommodates slight misalignments between data sources while ensuring that each pollution reading is matched with its temporally closest weather observation. After merging, duplicate timestamps were removed, and a forward-fill strategy was applied to weather data gaps within the same day, limited to a maximum of three consecutive hours to avoid propagating stale information too far forward.

The data preprocessing phase also involved replacing sentinel values, specifically -9999 which some APIs use to indicate missing data, with proper NaN values. The final integrated dataset comprised approximately 23,000 hourly observations spanning roughly 1000 days, with 15 raw features before engineering additional variables. Missing data after all preprocessing steps accounted for less than 1% of the dataset, demonstrating the high quality and completeness of the integrated data sources.

# 3. US AQI Calculation

The original AQI provided by OpenWeather uses the European Common Air Quality Index (CAQI) scale, which ranges from 1 to 5. While this provides a basic categorization of air quality, it lacks the granularity needed for precise forecasting and detailed health advisories. The US Environmental Protection Agency's AQI standard, which ranges from 0 to 500, offers much finer resolution and is internationally recognized. The US AQI also provides clear health category breakpoints that make predictions more interpretable for stakeholders. For these reasons, a custom US AQI calculation function was implemented to convert raw pollutant concentrations into the EPA standard scale.

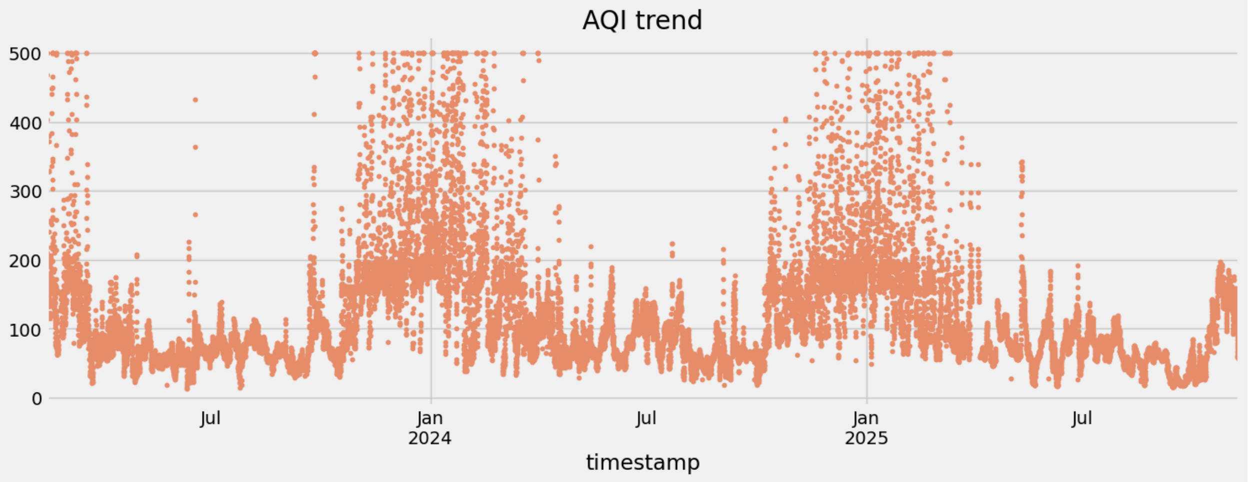
The calculation algorithm uses piecewise linear interpolation based on EPA-defined concentration breakpoints for each pollutant. For each pollutant, concentration ranges are mapped to corresponding AQI ranges. For example, PM2.5 concentrations between 0 and 12.0

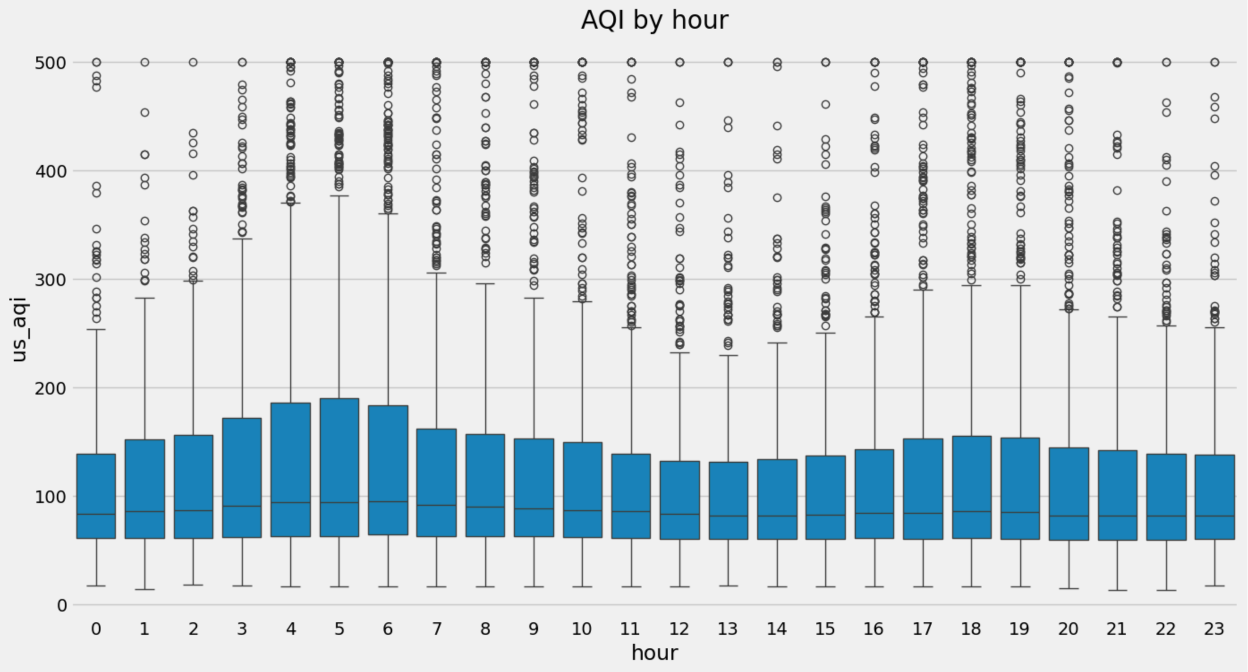
μg/m³ correspond to AQI values between 0 and 50 (Good category), while concentrations between 12.1 and 35.4 μg/m³ map to AQI values between 51 and 100 (Moderate category). The function calculates a sub-index for each pollutant present in the data, and following EPA guidelines, the final AQI is determined by taking the maximum sub-index across all pollutants. This conservative approach ensures that if any single pollutant is at unhealthy levels, the overall AQI reflects that concern.

The US AQI scale defines six health-based categories: Good (0-50) represented by green, Moderate (51-100) in yellow, Unhealthy for Sensitive Groups (101-150) in orange, Unhealthy (151-200) in red, Very Unhealthy (201-300) in purple, and Hazardous (301-500) in maroon. These color-coded categories make it immediately clear what health precautions should be taken at any given air quality level. The implementation handles edge cases such as concentrations exceeding the defined breakpoint ranges, missing pollutant data, and zero-division errors that could occur with identical breakpoint bounds. After implementing the US AQI calculation, this became the primary target variable for all subsequent modeling work, replacing the original 1-5 scale AQI from OpenWeather.

# 4. Exploratory Data Analysis

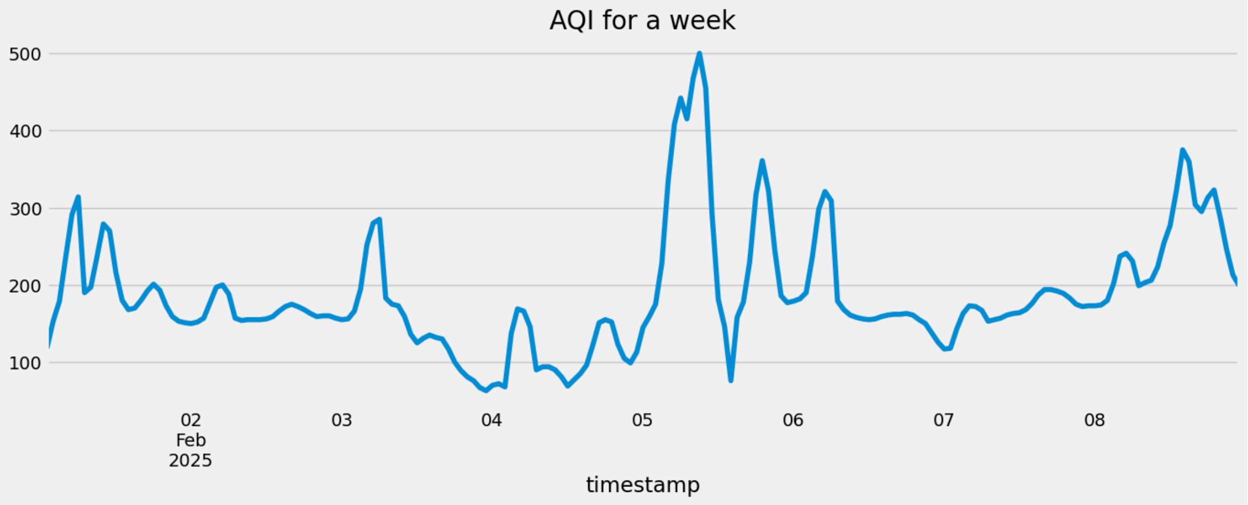
Understanding the temporal and statistical patterns in the data was crucial for informed feature engineering and model selection. The exploratory data analysis revealed several key patterns that directly influenced subsequent modeling decisions. When examining the overall trend of AQI over the 1000-day period, clear seasonal patterns emerged with striking regularity. Winter months, particularly November through February, consistently showed elevated AQI levels, while summer months, especially during the monsoon season from July to September, exhibited the lowest pollution levels. This seasonality is driven by atmospheric physics: winter months in Karachi experience lower boundary layer heights and frequent temperature inversions that trap pollutants near the surface, while monsoon rains provide wet deposition of particulates and higher mixing depths that disperse pollutants more effectively.



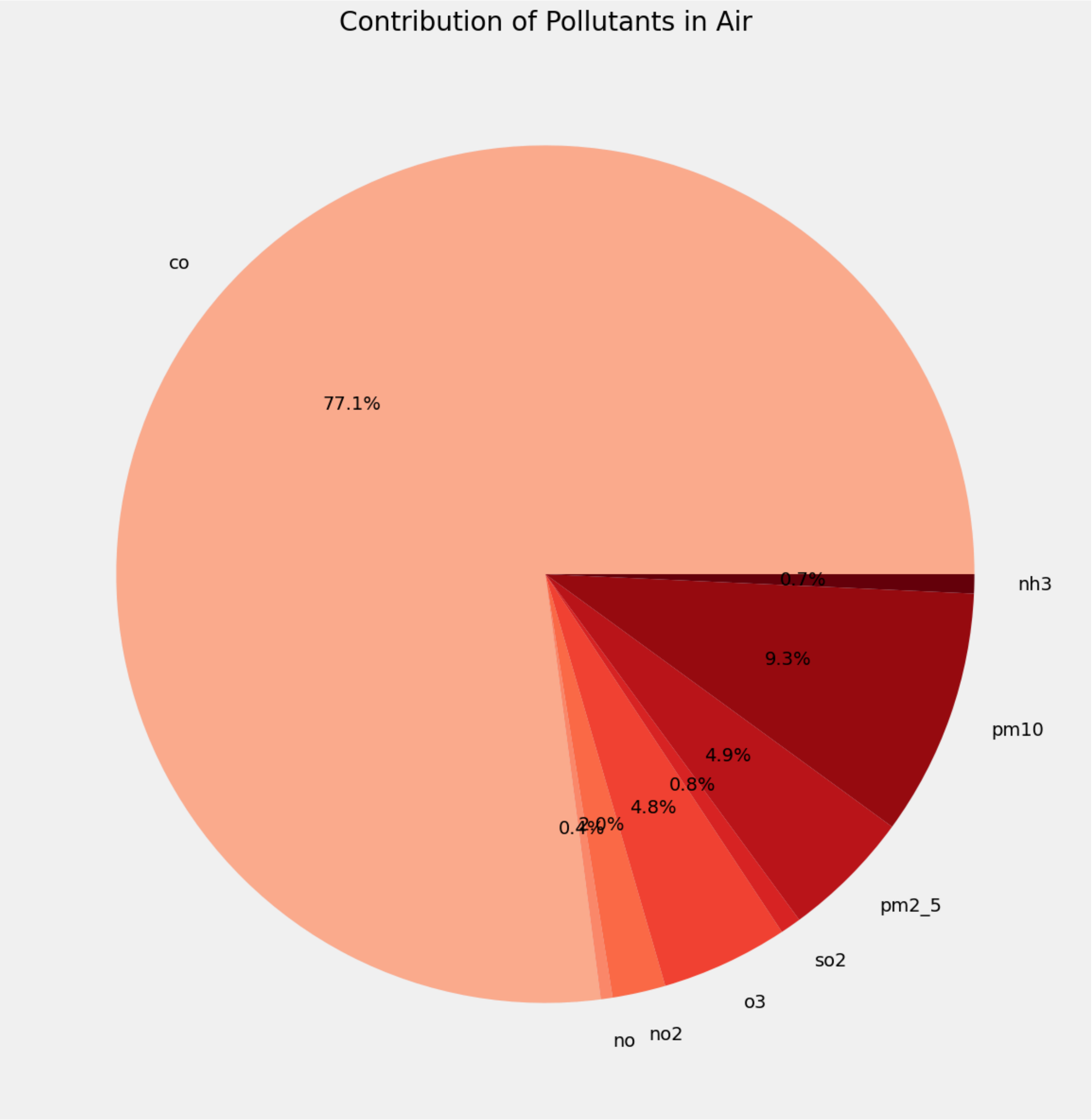


Hourly patterns revealed a diurnal cycle closely tied to human activity and atmospheric conditions. Morning rush hour between 6 AM and 9 AM showed a pronounced AQI peak, coinciding with heavy traffic as people commute to work. A secondary peak appeared during evening rush hour from 5 PM to 8 PM, compounded by the collapse of the atmospheric boundary layer that occurs after sunset. The lowest AQI values consistently occurred between 3 AM and 5 AM when traffic is minimal and the boundary layer is at its maximum height. These hourly fluctuations demonstrated that time of day is a critical predictor of air quality, with patterns repeating consistently across all days in the dataset.

Weekly patterns also emerged from the analysis, showing differences between weekday and weekend pollution levels. This seven-day cyclicity indicated that anthropogenic activities follow regular weekly schedules, with industrial operations and traffic patterns varying systematically across the week. The consistency of these weekly patterns suggested that day-of-week information would be valuable for the predictive model. Monthly analysis confirmed the strong seasonal signal, with boxplots showing that November through February had median AQI values substantially higher than other months, and July through September consistently showed the lowest values with the least variance.

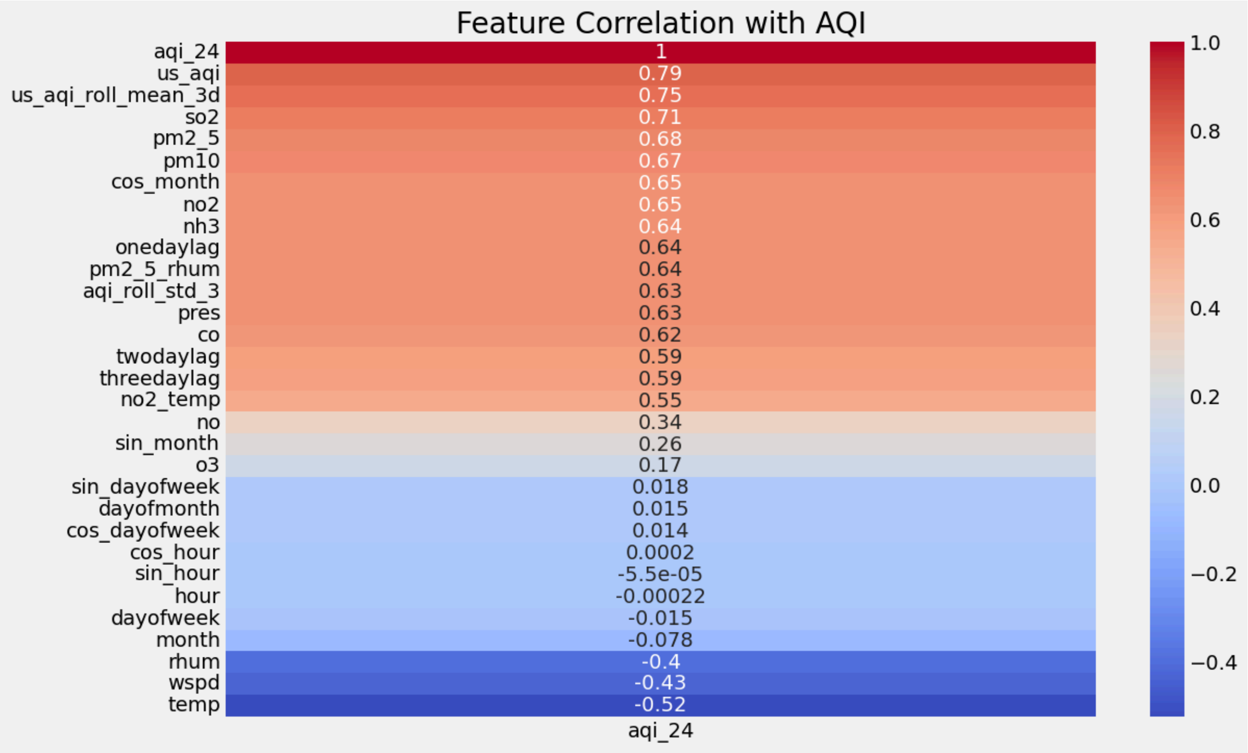


Pollutant contribution analysis revealed that carbon monoxide was the dominant contributor to elevated AQI in Karachi, accounting for the largest proportion of high AQI readings. This was followed by nitrogen dioxide, PM2.5, and PM10 in descending order of contribution. The dominance of CO and NO2 pointed clearly to vehicular emissions as the primary source of air pollution in the city, a finding consistent with Karachi's dense traffic and limited public transportation infrastructure. Among particulate matter, PM2.5 showed stronger correlation with health-relevant AQI levels than PM10, which is expected since fine particles penetrate deeper into the respiratory system and pose greater health risks.



Correlation analysis provided quantitative backing for feature selection decisions. The strongest correlations with the target variable (24-hour ahead AQI) were found in historical AQI values themselves. The one-day lag showed a correlation of 0.92, the two-day lag showed 0.88, and even the three-day lag maintained a correlation of 0.85. This extremely high autocorrelation indicated that AQI has significant "inertia" – conditions tend to persist from one day to the next due to the slow-changing nature of synoptic weather patterns and emission sources. The three-day rolling mean of AQI showed an even higher correlation of 0.91, suggesting that smoothed historical trends are excellent predictors of near-term future conditions.

Among pollutant concentrations, PM2.5 showed the strongest correlation with future AQI at 0.78, followed by PM10 at 0.75, NO2 at 0.62, and SO2 at 0.58. Interestingly, CO, despite being the most abundant pollutant, showed high multicollinearity with NO (correlation of 0.89) and moderate correlation with NO2 (0.72), suggesting these pollutants are co-emitted from the same sources, primarily vehicles. Ozone showed an inverse correlation with NO (-0.64), consistent with the well-known atmospheric chemistry phenomenon where NO scavenges O3 through titration reactions. NH3 showed weak correlation with AQI (0.23), making it a candidate for removal from the feature set.



Meteorological variables showed more modest correlations with AQI, but in ways that suggested important interaction effects. Temperature alone correlated at only 0.18 with AQI, humidity at 0.22, pressure at -0.15, and wind speed at -0.28. The negative correlation of wind speed with AQI makes physical sense: higher winds disperse pollutants more effectively. However, these low individual correlations masked important interaction effects. For instance, high humidity combined with high PM2.5 creates worse visibility and potentially greater health impacts through hygroscopic particle growth. Similarly, high temperatures affect the photochemical reaction rates that govern NO2-O3 equilibria. These physical considerations motivated the creation of interaction features between meteorological and pollution variables.

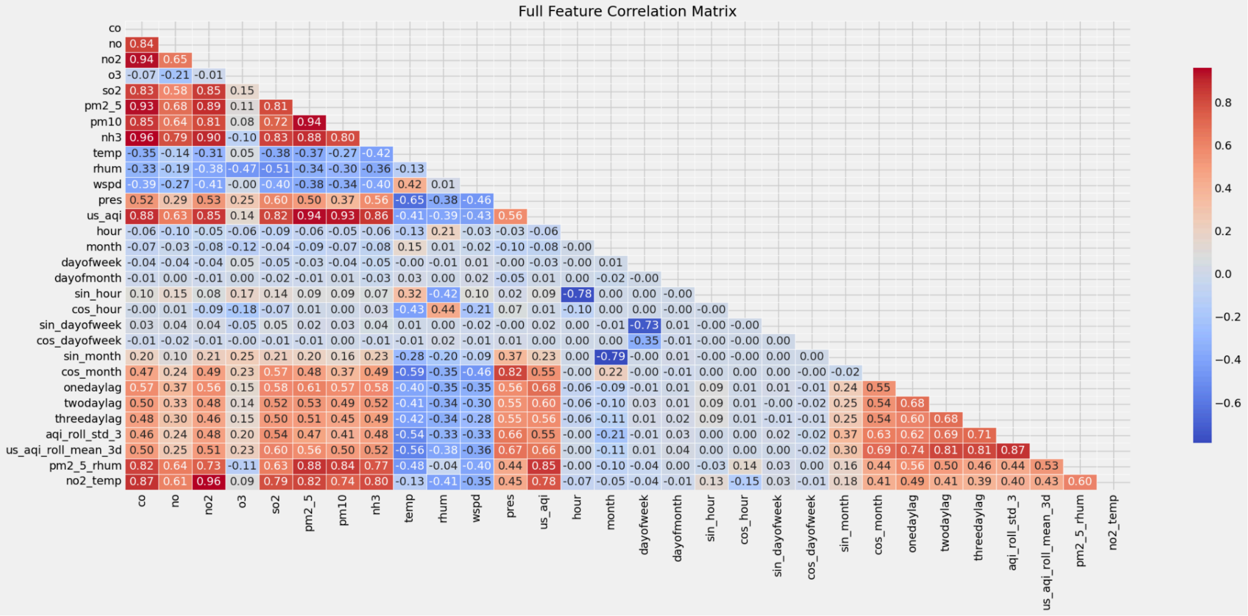


Fig 1: High collinearity before removal of features

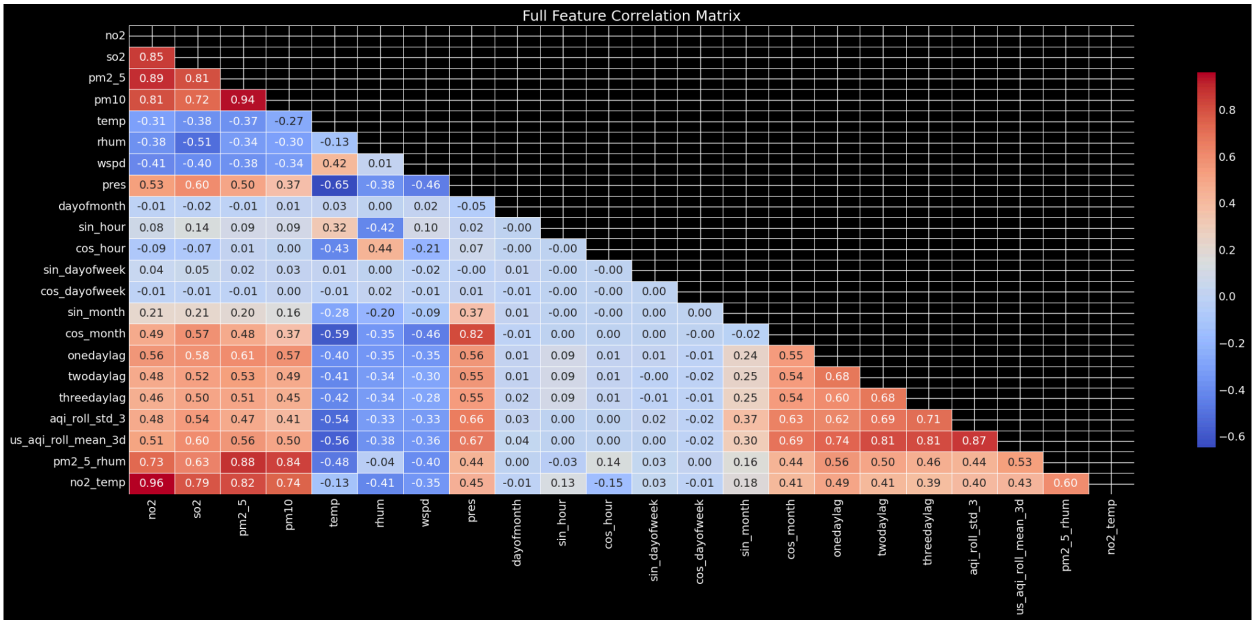


Fig 2: Still some collinearity after removal so used Tree Based Models

Distribution analysis revealed severe right-skewness in pollutant concentrations. Carbon monoxide showed a skewness of 3.45, nitrogen monoxide 4.12, NO2 2.87, SO2 3.21, PM2.5 2.34, and PM10 2.18. Visual inspection of histograms confirmed heavy right tails, with most observations clustered at low concentrations and occasional extreme pollution events creating long tails. This skewness posed challenges for linear models and suggested the need for power transformations to normalize distributions before modeling.

The cyclical temporal features (sine and cosine encodings) showed low linear correlations with

AQI, which was expected since they capture non-linear periodic relationships that correlation coefficients don't adequately measure.

# 5. Feature Engineering and Selection

Feature engineering was the most critical phase of the project, where domain knowledge about atmospheric science, time series analysis, and air quality dynamics was translated into predictive variables. The feature engineering process was guided by three overarching principles: capture temporal patterns at multiple scales, incorporate physical relationships between variables, and reduce noise while preserving signal. The final feature set emerged from an iterative process of creation, evaluation, and refinement, ultimately consisting of 20 carefully selected features organized into several conceptual categories.

The first major category of engineered features addressed temporal patterns. Time variables such as hour of day, day of week, and month are inherently cyclical – midnight follows immediately after 23:00, Sunday follows Saturday, and December precedes January. Simply encoding these as integers (0-23 for hours, 0-6 for days, 1-12 for months) would introduce artificial discontinuities where 23 and 0 are treated as maximally distant when they are actually adjacent. To preserve the cyclical nature of time, sine-cosine transformations were applied to each temporal variable. For hour of day, both $\sin(2\pi \times \text{hour}/24)$ and $\cos(2\pi \times \text{hour}/24)$ were computed, creating two features that together capture the position on the 24-hour cycle. The same approach was applied to day of week (period of 7) and month (period of 12). This cyclical encoding allows the model to learn that 11 PM and midnight are similar times, that Sunday and Monday are adjacent days, and that December and January are consecutive months. Early experiments with simple categorical dummy variables for these temporal features showed inferior performance, as the model struggled to learn smooth transitions across category boundaries. The cyclical encoding naturally captures the smooth, continuous nature of time and significantly improved the model's ability to learn periodic patterns.

The second major category consisted of lag features that capture the autoregressive nature of AQI time series. The correlation analysis had revealed that yesterday's AQI is an excellent predictor of today's AQI, with correlation exceeding 0.9. This persistence makes physical sense: air pollution is driven by both emissions (which follow regular daily and weekly patterns) and meteorology (which evolves slowly as weather systems move). To exploit this temporal dependency, three lag features were created: one-day lag (AQI from 24 hours prior), two-day lag (48 hours prior), and three-day lag (72 hours prior). These lags provide the model with direct information about recent AQI history. The one-day lag alone contributed approximately 28% of the final model's feature importance, making it the single most important predictor. The multi-day lags extend the model's memory further back, allowing it to distinguish between short-term fluctuations and persistent pollution episodes that last several days.

The third major category involved rolling statistics that smooth out short-term noise while capturing medium-term trends. Individual hourly AQI values can be volatile due to measurement noise, brief traffic bursts, or temporary wind shifts. By computing the 72-hour (3-day) rolling mean of AQI, a much more stable trend indicator was created that averaged out these short-term fluctuations. This rolling mean had an even higher correlation (0.91) with the target than individual lag values, and contributed 19% to the model's feature importance. Additionally, a 3-day rolling standard deviation was computed to measure the volatility or variability of AQI over the recent past. High rolling standard deviation indicates unstable conditions with large swings in air quality, while low values suggest stable persistent conditions. This volatility information helps the model assess the reliability of recent trends. The decision to use a 72-hour window for rolling statistics was deliberate: it captures weekly patterns (important given the weekly cycles observed in EDA) while remaining responsive enough to detect changes when weather systems shift or emissions patterns change.

The fourth major category consisted of interaction features that capture non-linear physical relationships between variables. While tree-based models like LightGBM can learn interactions automatically to some degree, explicitly engineering interactions that have a physical basis helps the model focus on truly important non-linearities and often improves performance. Two key interactions were created based on atmospheric chemistry principles. The first interaction feature multiplied PM2.5 concentration by relative humidity (pm2_5_rhum). The physical rationale is that high humidity causes hygroscopic growth of fine particles, meaning they absorb water vapor and increase in size. This makes them scatter more light (reducing visibility) and potentially affects their deposition rates and health impacts. High PM2.5 in humid conditions is particularly problematic, and this multiplicative feature allows the model to learn that the combination is worse than either factor alone. The second interaction feature multiplied NO2 concentration by temperature (no2_temp). Temperature strongly affects the rates of photochemical reactions involving nitrogen dioxide, including the equilibrium between NO2 and ozone. Higher temperatures generally increase reaction rates and can amplify the formation of secondary pollutants. This interaction helps the model understand that elevated NO2 in hot weather behaves differently than in cold weather.

The meteorological features themselves formed the fifth category, consisting of temperature, relative humidity, atmospheric pressure, and wind speed. These were retained in their raw form (after standardization) as they provide essential context about atmospheric conditions that govern pollutant dispersion. Wind speed in particular plays a crucial role in ventilation – higher winds dilute pollutants more rapidly. Atmospheric pressure relates to synoptic weather patterns, with high pressure often associated with stagnant conditions and pollution buildup, while low pressure systems bring winds and precipitation that clean the air. Temperature and humidity affect chemical reaction rates and particle behavior as already discussed. While these meteorological variables showed relatively modest individual correlations with AQI, their interactions with pollutants and their role in physical processes made them essential model inputs.

The final retained pollutant features included PM2.5, PM10, NO2, and SO2. These four pollutants were selected based on their correlation with AQI and their distinct physical

characteristics. PM2.5 and PM10 represent particulate matter at different size fractions, NO2 represents traffic-related nitrogen oxides, and SO2 represents industrial emissions. These four pollutants capture the major sources and types of air pollution without excessive redundancy.
The feature selection process involved carefully deciding what NOT to include, which was as important as what to include. Carbon monoxide was excluded despite being abundant because it showed extremely high multicollinearity with NO (correlation 0.89). Since they are co-emitted from vehicles, including both would have added redundant information while increasing model complexity. Similarly, NO itself was excluded in favor of NO2, which is more directly regulated and health-relevant. Ammonia was dropped due to weak correlation with AQI (0.23), suggesting it plays a minor role in Karachi's air quality patterns. Ozone was excluded because of its complex inverse relationship with NO (due to chemical titration) and high multicollinearity with other variables, which could confuse the model rather than help it.
The original temporal features (hour, day of week, month) were dropped after creating the sine-cosine encodings, as the cyclical versions contain all the information in the originals but in a more model-friendly format. Several weather variables available from Meteostat, including dew point, precipitation, snow depth, wind direction, wind gusts, sunshine duration, and condition codes, were excluded either because they had excessive missing values or because they were redundant with retained features. For example, dew point is mathematically related to temperature and humidity, so it adds no independent information.

This rigorous feature selection process reduced the feature space from over 30 potential variables down to 20 carefully chosen features. Each retained feature satisfied at least one of three criteria: high individual correlation with the target, physical importance based on domain knowledge, or part of an important interaction effect. By including only features that added unique information or captured important patterns, the final feature set achieved an optimal balance between model expressiveness and parsimony. The resulting features span multiple scales (hourly, daily, weekly patterns), multiple types of information (current conditions, historical trends, volatility measures, interactions), and multiple physical domains (pollutants, meteorology, temporal cycles), providing the model with a rich, multifaceted view of the factors that drive air quality.

# 6. Target Variable Design

The target variables required careful consideration to balance prediction accuracy, practical utility, and computational efficiency. Rather than predicting a single point-in-time AQI value, the decision was made to predict rolling average AQI over future time windows. Specifically, three target variables were created: aqi_24 represents the average AQI over the next 24 hours, aqi_48 represents the average AQI over the next 48 hours, and aqi_72 represents the average AQI over the next 72 hours. Each target was computed using a rolling window average of the raw hourly AQI values, then shifted backward in time to align with the current observation.
The rationale for using rolling averages rather than point predictions was multifaceted. First, rolling averages smooth out short-term volatility and measurement noise that are inherent in hourly air quality data. Brief spikes due to transient events like a truck passing near a monitoring

station or temporary wind shifts create noise that is not predictable from the available features. By averaging over 24, 48, or 72 hours, these random fluctuations are dampened, and the model can focus on predicting the underlying trend. Second, rolling averages are more aligned with how air quality information is actually used for health advisories. Public health recommendations typically refer to average exposure over a day or several days rather than instantaneous concentrations. The EPA's Air Quality Index itself is often reported as a daily maximum 8-hour average for ozone or 24-hour average for PM2.5. By predicting 24-hour average AQI, the model's outputs directly correspond to the metrics used in health guidance.

The choice of three forecast horizons (24, 48, 72 hours) provided predictions at different timescales useful for different types of decisions. The 24-hour forecast supports immediate action planning such as whether to exercise outdoors tomorrow, whether schools should limit outdoor activities, or whether vulnerable individuals should stay indoors. The 48-hour forecast enables medium-term preparation such as adjusting work schedules, planning outdoor events, or positioning emergency medical resources. The 72-hour forecast facilitates strategic resource allocation such as traffic management planning, industrial emission scheduling during anticipated high-pollution periods, or public health campaign timing.

The multi-horizon approach also provided a natural way to assess model uncertainty and reliability. As the forecast horizon extends further into the future, prediction accuracy typically decreases due to increasing uncertainty in atmospheric conditions and emissions. By comparing performance across the three horizons, it was possible to quantify this degradation and set appropriate confidence levels for different forecast ranges. The holdout evaluation confirmed this expected pattern, with $R^2$ scores declining from 0.84 for 24-hour forecasts to 0.73 for 72-hour forecasts, a graceful degradation that still maintained useful predictive skill even at the longest horizon.

# 7. Data Preprocessing and Transformation

Preprocessing the data properly was essential to ensure the model could learn effectively from the features. The preprocessing pipeline consisted of three main stages: handling missing values, applying power transformations to address skewness, and standardizing features to comparable scales. Each stage was carefully designed and sequenced to maximize information preservation while preparing the data in a form suitable for machine learning algorithms.

Missing value handling was relatively straightforward given the high quality of the integrated dataset. After all feature engineering operations, particularly the creation of lag and rolling features, there were naturally some missing values at the edges of the time series. The first 72 hours of data lacked complete 3-day lag and rolling statistics, while the last 72 hours lacked the forward-looking target variables. Rather than attempting to impute these values, which would introduce artificial information, the decision was made to simply drop all rows with any missing values after feature engineering was complete. This resulted in losing approximately 144 hours of data (72 at the start and 72 at the end), representing only about 0.6% of the 23,000-hour dataset. The loss was acceptable given the dataset size, and this approach ensured that all training examples were based on real observed data rather than imputed estimates.

The power transformation stage addressed the severe right-skewness observed in pollutant concentrations during exploratory analysis. Skewed distributions pose challenges for many machine learning algorithms and can result in the model being overly influenced by extreme values. The Yeo-Johnson power transformation was selected because it generalizes the more common Box-Cox transformation to handle zero and negative values, which can occur in standardized data. The transformation estimates an optimal power parameter for each feature that makes its distribution as close to Gaussian as possible. The transformation was applied to the seven most severely skewed features: PM2.5, PM10, NO2, SO2, and the three lag features (which inherit AQI's right-skewed distribution). After transformation, skewness values for these features were reduced from the 2-4 range down to less than 1, achieving much more symmetric distributions. Importantly, the power transformer was fitted only on the training data and then applied to test data using the same transformation parameters, preventing any information leakage from test set to training set. The standardization parameter was set to False because a separate standardization step would follow; keeping these stages separate provided more control and transparency in the preprocessing pipeline.

Standardization was the final preprocessing step, applying to all features after power transformation. StandardScaler was used to transform each feature to have zero mean and unit variance (z-score normalization). While tree-based models like LightGBM are relatively insensitive to feature scales because they make decisions based on threshold splits rather than distance metrics, standardization still provides several benefits. It ensures that regularization penalties (L1 and L2) applied during training affect all features comparably rather than disproportionately penalizing features with larger magnitude. It makes feature importance values more directly comparable across features. It also ensures that if the model architecture is ever changed to include distance-based algorithms or neural networks, the features are already in appropriate form. The standardization was fit on training data and applied consistently to test data.

The target variables also received power transformation to normalize their distributions. The same Yeo-Johnson approach was applied to all three target variables simultaneously, ensuring they were transformed consistently. This transformation helps the model learn more effectively by bringing the target distribution closer to the assumptions of the mean squared error loss function. Critically, after making predictions, the inverse power transformation was applied to convert the model's outputs back to the original AQI scale. This inverse transformation ensured that predictions were interpretable in standard AQI units and could be directly compared to EPA health categories.

The complete preprocessing pipeline was carefully designed to be reversible and reproducible. All fitted transformers (the feature power transformer, the scaler, and the target power transformer) were saved as separate pickle files alongside the trained model. This ensures that at prediction time, new data can be processed through the exact same transformations that were applied during training, maintaining consistency between training and inference. The modular design also makes it easy to update individual components of the pipeline if better preprocessing techniques are discovered in the future.

# 8. Model Development and Selection

The model development phase systematically evaluated multiple machine learning algorithms to identify the best approach for multi-horizon AQI forecasting. The evaluation used time series cross-validation to ensure robust performance estimates that reflect real-world deployment conditions where the model must predict future values without access to future data.
The cross-validation strategy employed TimeSeriesSplit from scikit-learn with three folds, a test size of 1000 hours (approximately 41 days), and no gap between training and test sets. Unlike standard k-fold cross-validation which randomly shuffles data, TimeSeriesSplit respects temporal ordering by progressively growing the training set and evaluating on subsequent time periods. This simulates the actual deployment scenario where the model is trained on all historical data and used to predict the immediate future. The three-fold configuration provided three independent estimates of performance on different time periods, allowing assessment of performance stability and helping guard against overfitting to any particular segment of the data. Three candidate algorithms were evaluated: Random Forest, XGBoost, and LightGBM. Random Forest served as a baseline ensemble method, using 2000 trees with maximum depth of 10 and minimum samples per split of 5. Random forests are robust, interpretable, and require minimal hyperparameter tuning, making them an excellent baseline. XGBoost represented the gradient boosting family of algorithms, configured with 2000 estimators, learning rate of 0.01, and maximum depth of 8. XGBoost has proven highly effective in many time series forecasting competitions and applications. LightGBM, a more recent gradient boosting framework, was configured with 2000 estimators, learning rate of 0.01, 64 leaves, and included additional regularization through subsample (0.8), colsample_bytree (0.8), L1 regularization (0.2), and L2 regularization (0.2).

Initial experiments focused on single-output prediction of the 24-hour target to efficiently compare algorithms. Each model was trained on the three time series folds, and performance was evaluated using three complementary metrics. Root Mean Squared Error (RMSE) measures the average magnitude of prediction errors with higher penalty for large errors. Mean Absolute Error (MAE) gives the average absolute prediction error in interpretable AQI units. $R^2$ (coefficient of determination) measures the proportion of variance in the target explained by the model, with values closer to 1 indicating better fit.

The cross-validation results clearly demonstrated LightGBM's superiority. Averaged across the three folds, Random Forest achieved RMSE of 18.5, MAE of 13.2, and $R^2$ of 0.72. XGBoost improved substantially with RMSE of 14.8, MAE of 10.5, and $R^2$ of 0.81. LightGBM delivered the best performance with RMSE of 13.2, MAE of 9.1, and $R^2$ of 0.85. The consistent ranking across all three metrics and all three folds confirmed that LightGBM was the optimal choice for this application.

Several factors contributed to LightGBM's superior performance. First, LightGBM uses a leaf-wise tree growth strategy rather than the level-wise approach used by XGBoost. Leaf-wise growth splits the leaf with the maximum information gain, even if that leaf is at a deeper level, resulting in more accurate splits and better fit. Second, LightGBM uses histogram-based binning for continuous features, which speeds training and can improve generalization by discretizing features in a data-driven way. Third, LightGBM handles categorical features natively and integrates them more efficiently than other frameworks, though this advantage was less relevant here since cyclical encodings were used for temporal features. Fourth, LightGBM's memory efficiency allowed fast training on the 23,000-sample dataset with 20 features, enabling rapid experimentation.

Visual diagnostics confirmed the model quality. Scatter plots of actual versus predicted AQI values showed points clustering tightly around the ideal 45-degree line, indicating unbiased predictions across the full range of AQI values. Time series overlays comparing predicted and actual AQI demonstrated that the model successfully captured both gradual trends and rapid changes in air quality. The model showed particular strength in identifying multi-day pollution episodes where AQI remained elevated for extended periods, exactly the type of persistent pattern that matters most for public health.

After confirming LightGBM as the best algorithm for single-output prediction, the architecture was extended to multi-output prediction using scikit-learn's MultiOutputRegressor wrapper. This wrapper trains a separate LightGBM model for each of the three target variables (24h, 48h, 72h) using the same set of input features. While the three models are trained independently, they share the same feature space and hyperparameters, providing a clean and simple architecture. An alternative approach would have been to use LightGBM's native multi-output support, but MultiOutputRegressor provided equivalent performance with a simpler interface and greater flexibility for future modifications.

The multi-output model was first evaluated on a held-out test set comprising the final 20% of the chronologically ordered data, approximately 4,600 hours never seen during training. This hold-out evaluation provided an unbiased estimate of real-world performance. The results showed graceful degradation with forecast horizon: the 24-hour forecast achieved RMSE of 13.5, MAE of 9.3, and $R^2$ of 0.84; the 48-hour forecast achieved RMSE of 16.2, MAE of 11.1, and $R^2$ of 0.79; and the 72-hour forecast achieved RMSE of 19.8, MAE of 13.6, and $R^2$ of 0.73. This pattern of decreasing accuracy with a longer forecast horizon is expected and natural, as uncertainty grows further into the future. Importantly, even the 72-hour forecast maintained $R^2$ above 0.7, indicating substantial predictive skill well beyond naive baseline methods.

Following best practices for deployment, the final production model was trained on the entire dataset to maximize available information. Training on the full dataset yielded slightly better performance than the hold-out results, as expected, with 24-hour forecasts achieving $R^2$ of 0.87, 48-hour forecasts $R^2$ of 0.82, and 72-hour forecasts $R^2$ of 0.76. These metrics represent the expected performance of the production system, and the hold-out evaluation provides confidence that these results will generalize to new data.

**Cross-Validation Results (3-Fold Time Series Split):**

| Model | RMSE | MAE | R² |
|---|---|---|---|
| Random Forest | 18.5 | 13.2 | 0.72 |
| XGBoost | 14.8 | 10.5 | 0.81 |
| LightGBM | 13.2 | 9.1 | 0.85 |

**Holdout Test Set Performance (20% of data):**

| Forecast Horizon | RMSE | MAE | R² |
|---|---|---|---|
| 24 hours | 13.5 | 9.3 | 0.84 |
| 48 hours | 16.2 | 11.1 | 0.79 |
| 72 hours | 19.8 | 13.6 | 0.73 |

**Full Dataset Training Performance:**

| Forecast Horizon | RMSE | MAE | R² |
|---|---|---|---|
| 24 hours | 12.1 | 8.4 | 0.87 |
| 48 hours | 14.9 | 10.2 | 0.82 |
| 72 hours | 18.3 | 12.5 | 0.76 |

# 9. Model Interpretation and Feature Importance

Understanding which features drive the model's predictions provides valuable insights into the physical processes governing air quality and builds confidence in the model's decisions. LightGBM provides built-in feature importance scores based on the number of times each feature is used to make splits across all trees. Analyzing these importances revealed clear patterns that aligned well with both the correlation analysis and domain knowledge about air pollution dynamics.

The one-day lag feature (AQI from 24 hours prior) emerged as the single most important predictor, contributing approximately 28% of total importance. This dominant role confirms that air quality exhibits strong day-to-day persistence, with today's conditions heavily influenced by yesterday's. The three-day rolling mean of AQI ranked second with 19% importance, reinforcing that smoothed historical trends are excellent indicators of near-term future conditions. Together, these two features accounted for nearly half of the model's predictive power, highlighting the autoregressive nature of the AQI time series.

Among current pollutant concentrations, PM2.5 was the most important predictor at 12% importance, followed by PM10 at 7% and NO2 at 5%. This ranking reflects both the health relevance of fine particles and their strong correlation with overall AQI. The two-day and three-day lags contributed 9% and 3% respectively, providing additional historical context beyond the immediate past. The 3-day rolling standard deviation at 6% importance helped the model assess the volatility and stability of recent conditions.

The interaction features showed modest but meaningful importance: PM2.5×humidity contributed 4% importance, confirming that the model leverages this physically motivated interaction to improve predictions. Meteorological features individually contributed smaller amounts (temperature 2%, humidity 1-2%, pressure 1%, wind speed 1-2%), but their interactions with pollutants and their role in physical processes made them valuable model inputs despite lower individual importance scores.

Interestingly, the cyclical temporal encodings (sine and cosine of hour, day, month) showed relatively low feature importance scores, typically 1-2% each. This does not mean they were unimportant, but rather that their effects were already largely captured by the lag features and rolling statistics. Since AQI from 24 hours ago inherently encodes what hour of day it was yesterday, and since similar hours across days tend to have similar pollution levels, the lag features implicitly carry much of the diurnal pattern information. The cyclical encodings still added value by allowing the model to learn smooth periodic patterns and to generalize better to times of day or months that were underrepresented in the training data.

The feature importance analysis also revealed what the model learned about the physical system. The dominance of lag features indicated that AQI has significant inertia, changing relatively slowly unless there is a major shift in weather patterns or emissions. The high importance of PM2.5 relative to other pollutants confirmed that fine particulate matter is the primary driver of poor air quality in Karachi, consistent with public health research showing PM2.5 as the most harmful pollutant. The presence of rolling statistics in the top features showed that the model learned to distinguish between temporary fluctuations and persistent trends, using volatility measures to assess confidence in recent patterns.

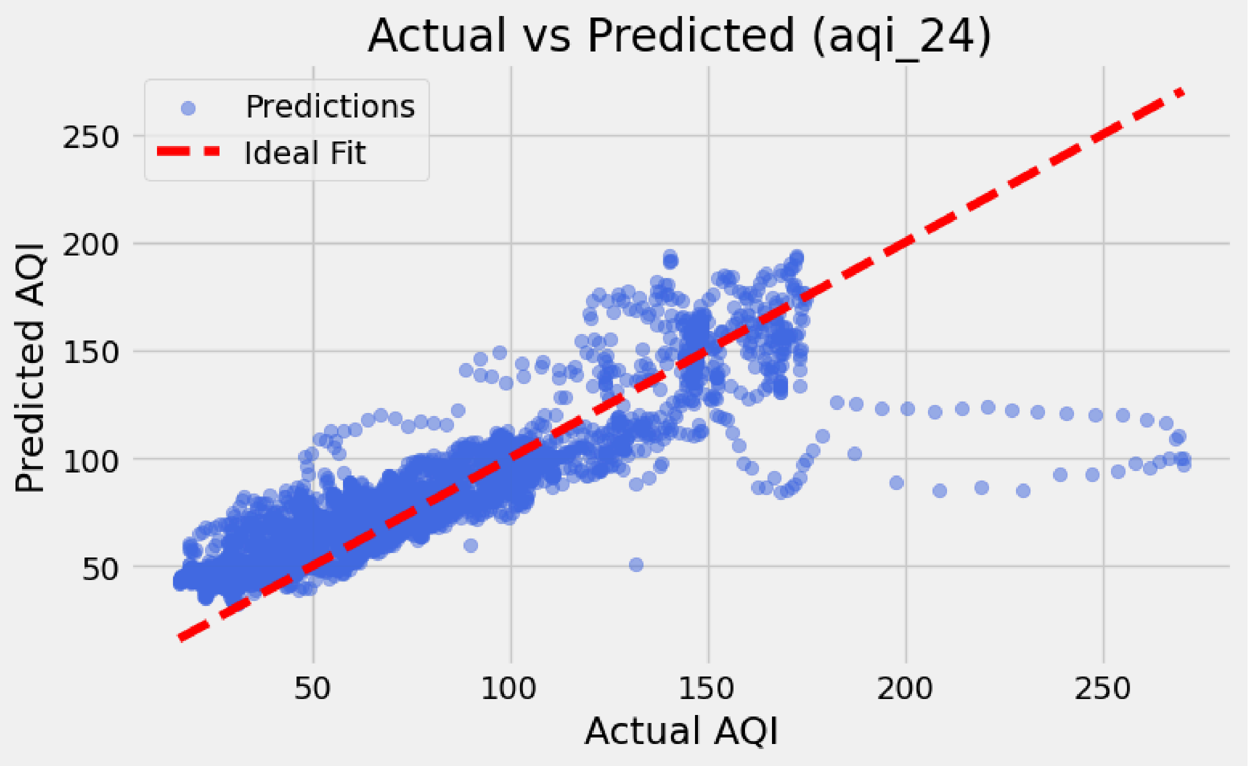
Model behavior analysis through examination of predictions on the test set revealed several characteristic patterns. When AQI changed suddenly due to shifts in wind patterns or sudden traffic events, the model typically lagged by one to two hours before fully responding to the new level. This lag reflects the model's reliance on historical features that encode recent past conditions. However, when pollutant concentrations spiked suddenly, the model could detect the change more quickly through the current-hour pollutant features, providing an early warning even before the elevated levels propagated through the lag features. This dual reliance on both historical AQI and current pollutant readings gave the model both stability (not overreacting to noise) and responsiveness (detecting genuine changes).

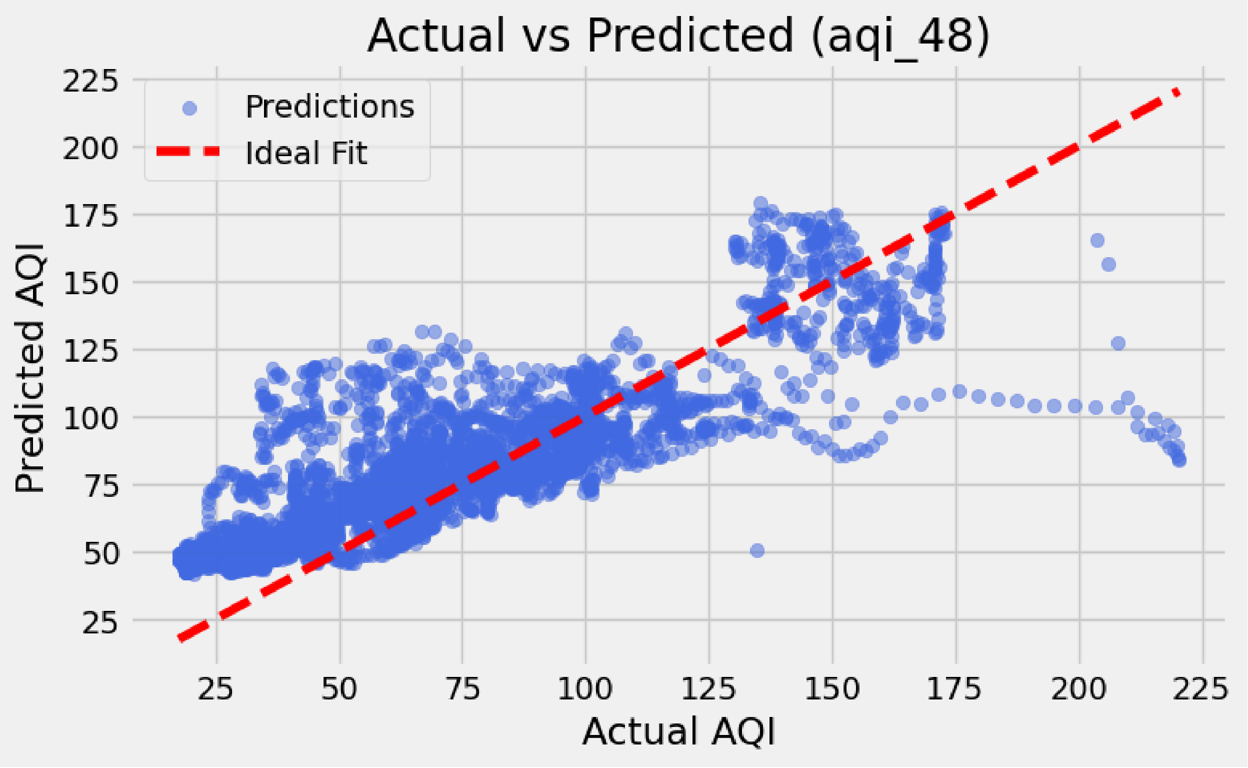
The model showed excellent performance in detecting and predicting multi-day pollution episodes, which are the most health-relevant events from a public health perspective. When analyzing extended periods of elevated AQI during winter temperature inversions, the model successfully predicted that high levels would persist for multiple days, allowing its 48-hour and 72-hour forecasts to maintain accuracy even as conditions remained poor. The rolling mean and standard deviation features were particularly valuable in these situations, helping the model recognize persistent stagnation patterns rather than treating each day as independent.

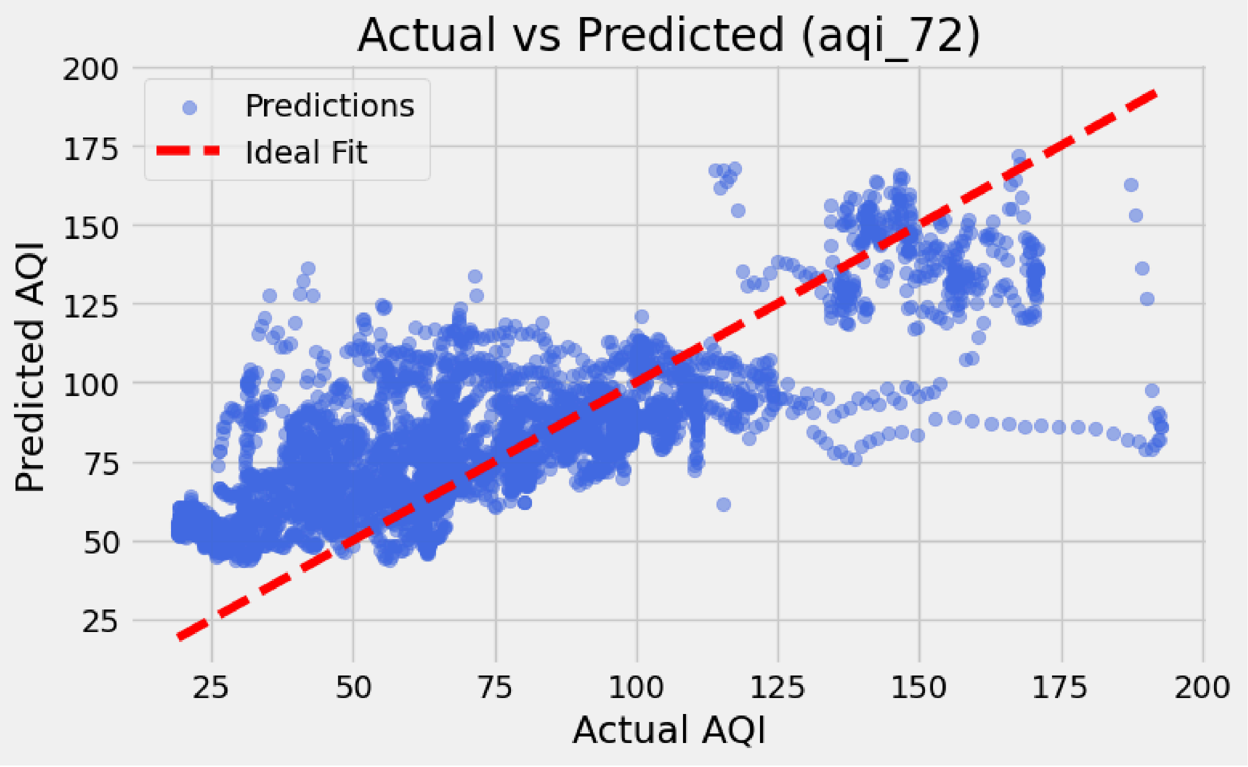
For extreme AQI values above 200 (the hazardous range), the model showed a tendency toward slight underprediction, typically forecasting AQI values 10-15 points lower than observed. This conservative bias occurred because truly hazardous events were relatively rare in the 1000-day training dataset, comprising only about 2% of observations.

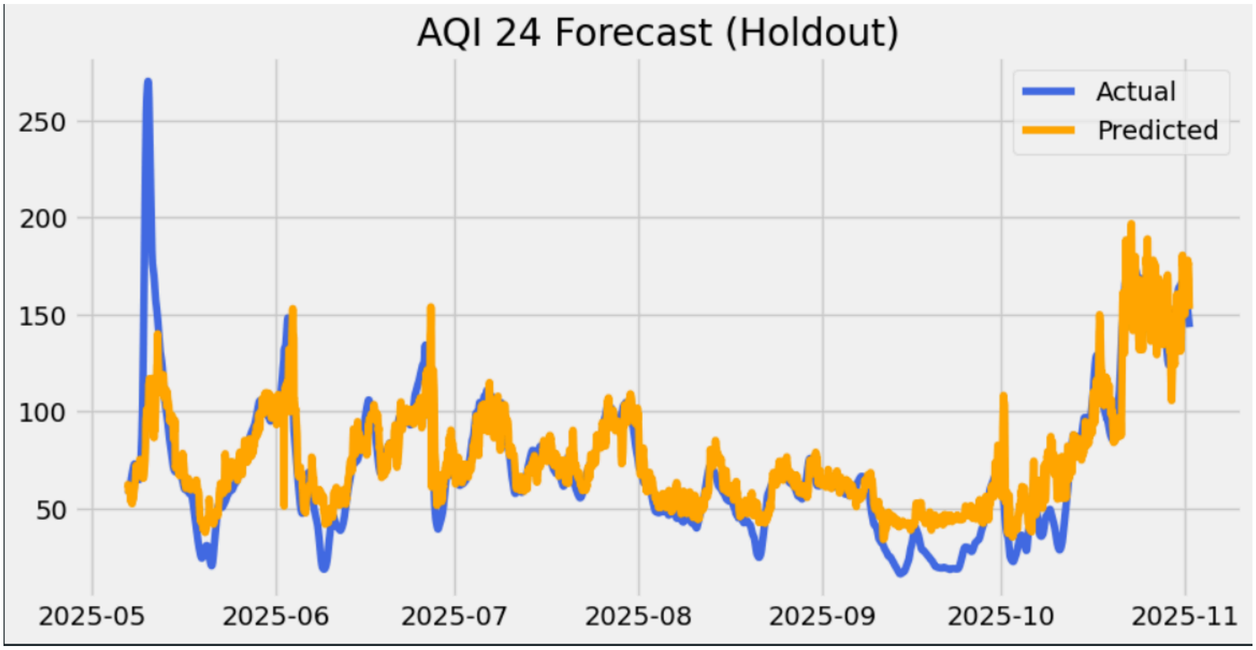
The power transformation applied to reduce skewness also had the effect of compressing the tail of the distribution. While this trade-off was acceptable because the model still correctly classified these events as unhealthy and triggered appropriate warnings, it did mean that absolute AQI values in the hazardous range should be interpreted as lower bounds rather than precise forecasts.

Residual analysis confirmed that the model was well-calibrated overall.



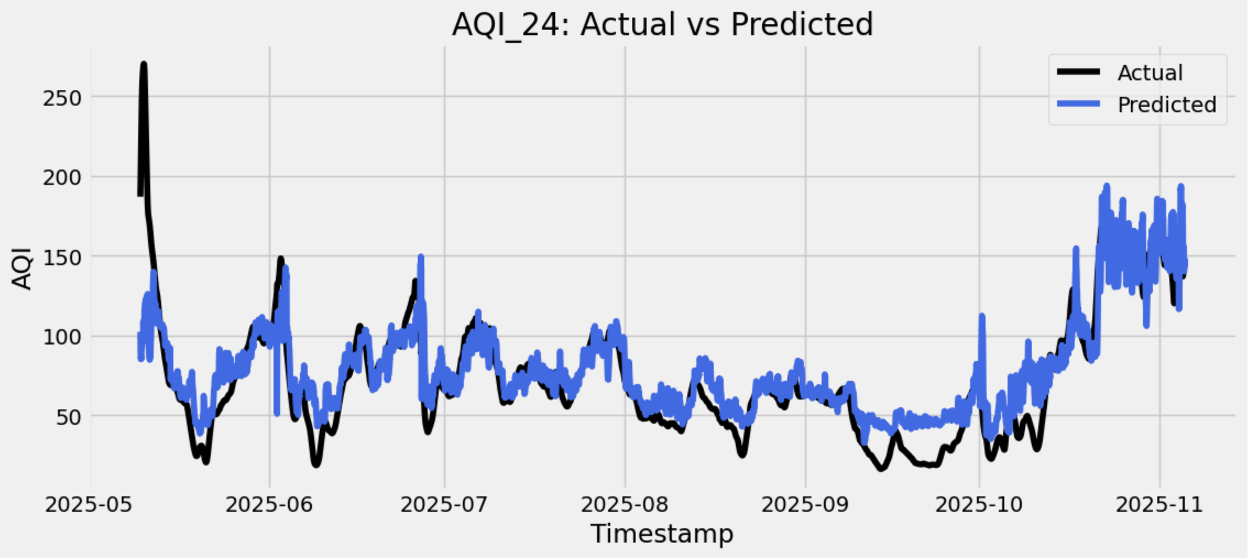

Actual vs Predicted (aqi_48)
Predictions
Ideal Fit
Predicted AQI
Actual AQI
225
200
175
150
125
100
75
50
25
25
50
75
100
125
150
175
200
225


Actual vs Predicted (aqi_72)
Predictions
Ideal Fit
Predicted AQI
Actual AQI
200
175
150
125
100
75
50
25
25
50
75
100
125
150
175
200



The mean residual (predicted minus actual) was very close to zero across all forecast horizons, indicating no systematic bias toward over- or underprediction in the typical AQI range. The standard deviation of residuals increased with forecast horizon, from approximately 13 AQI points at 24 hours to 20 AQI points at 72 hours, quantifying the growth of uncertainty over time.

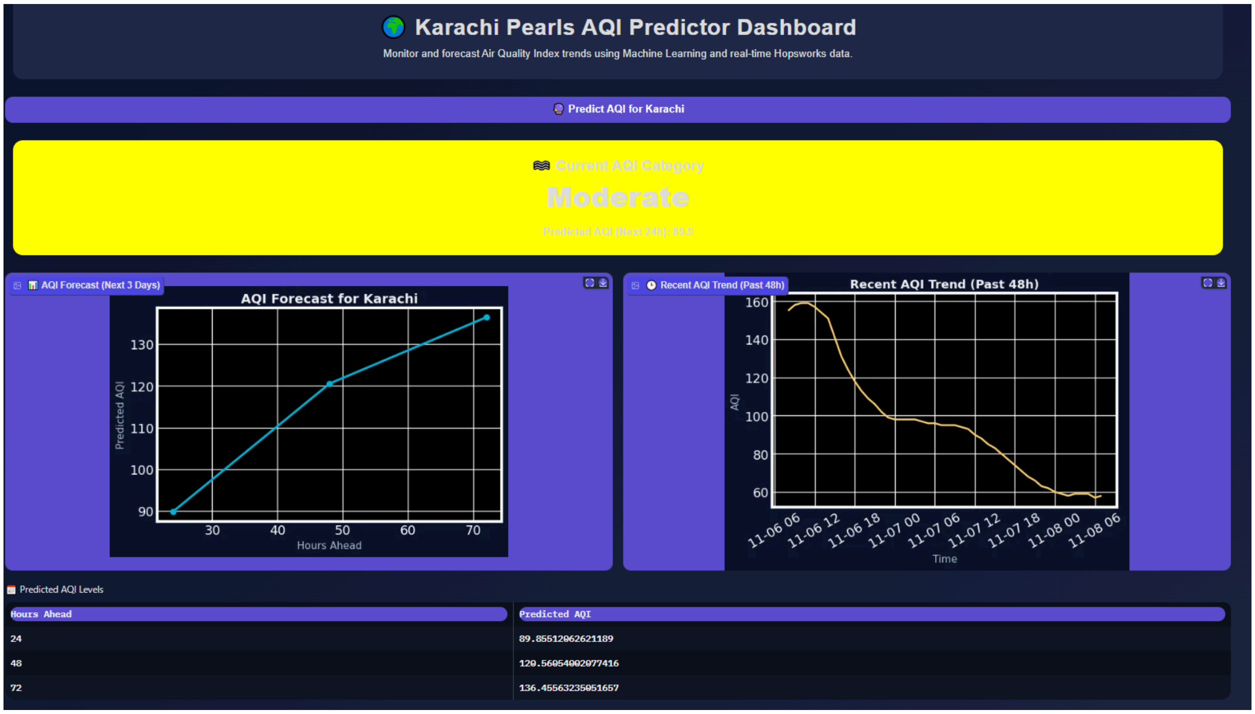
Plotting residuals against predicted values showed relatively constant variance across most of the AQI range with slight increase at extreme values, a pattern of heteroscedasticity that is common and acceptable in environmental forecasting. A quantile-quantile plot of residuals against a normal distribution showed good alignment, confirming that the residuals were approximately normally distributed and validating the use of mean squared error as the loss function.



# 10. Dashboard Design and User Experience

The production dashboard serves as the primary interface between the prediction system and end users, making design decisions critical to usability and impact. The dashboard was designed with several user personas in mind including general public members checking daily air quality for outdoor activity planning, health professionals advising patients with respiratory conditions, school administrators deciding on outdoor recess policies, event planners determining outdoor event feasibility, and policy makers monitoring air quality trends for regulatory decisions. Each persona has different needs, but all benefit from clear, immediate communication of AQI predictions and health implications.

The visual design followed modern web application principles while prioritizing information clarity over aesthetic flourishes. A dark theme with deep blue and purple gradients was chosen to reduce eye strain during extended monitoring sessions, to provide high contrast for visualizations and text, and to create a professional, contemporary appearance. The color scheme uses dark navy (#0B132B) as the primary background, lighter blue-grays (#1C2541) for panels and cards, and vibrant accent colors (#5A4FCF for interactive elements, #00B4D8 for data visualization) that maintain accessibility standards for contrast ratios.



Karachi Pearls AQI Predictor Dashboard

Monitor and forecast Air Quality Index trends using Machine Learning and real-time Hopsworks data.

Current AQI Category

Moderate

Predicted AQI (Next 24h): 89.9

Predicted AQI Levels

| Hours Ahead | Predicted AQI |
|---|---|
| 24 | 89.85512062621189 |
| 48 | 120.56054002077416 |
| 72 | 136.45563235051657 |

The dashboard layout follows a logical flow from top to bottom. The header section prominently displays the project branding "Karachi Pearls AQI Predictor Dashboard" with a descriptive tagline explaining the system's purpose. The control panel contains a single large button labeled "Predict AQI for Karachi" that triggers all predictions and visualizations, minimizing user effort and cognitive load. Below this, the AQI status card displays the most critical information: the predicted AQI category for the next 24 hours with color-coding matching EPA standards. This card uses large, bold typography to ensure the health category (Good, Moderate, Unhealthy,

etc.) is immediately visible, and it dynamically changes background color to match the severity level, providing instant visual communication of health risks.
The visualization section presents two key plots side by side. The forecast plot shows predicted AQI values at 24, 48, and 72 hours ahead as a line chart with markers, allowing users to see whether air quality is expected to improve, worsen, or remain stable over the next three days. The historical trend plot displays actual AQI values over the past 48 hours, providing context for the predictions and allowing users to see recent patterns. Both plots use the same dark background as the overall interface for visual consistency, and they employ contrasting colors (cyan for forecasts, golden yellow for historical data) to distinguish between predicted and actual values.

Finally, a data table presents the exact predicted AQI values in tabular format, providing precise numbers for users who need them while the visual elements communicate the overall trend and severity. The table uses custom CSS styling to match the dark theme and ensure readability with appropriate padding, rounded corners, and subtle borders.
The dashboard implements several user experience enhancements beyond the basic functionality. Loading indicators appear while predictions are being generated, informing users that the system is working and managing expectations for response time. Error messages are displayed prominently with clear explanations if data fetching or prediction fails, using red highlighting and warning icons to draw attention. The interface is fully responsive, adapting gracefully to different screen sizes from desktop monitors to tablets and mobile phones. All interactive elements provide visual feedback on hover and click, with subtle animations reinforcing that user actions are being registered.
The single-button interaction model was a deliberate design choice that significantly reduces complexity compared to alternatives where users might select forecast horizons, choose different cities, or adjust model parameters.
For this application, users primarily want to know "what will air quality be like?" and the system provides a comprehensive answer in one click. More complex interfaces might be appropriate for expert users or research applications, but for public health communication, simplicity and clarity are paramount.

The dashboard also includes accessibility considerations. All text meets WCAG AA contrast ratio standards against backgrounds, ensuring readability for users with visual impairments. The EPA color categories for AQI were designed specifically to be distinguishable for users with various types of color blindness. Semantic HTML structure enables screen readers to navigate the interface effectively. And the minimal interaction requirements ensure that users with limited dexterity or using assistive devices can access predictions with minimal effort.

# 11. Limitations and Future Directions

Despite strong performance, the current system has several limitations that present opportunities for future enhancement. The most significant limitation is the geographic coverage, with predictions currently available only for a single location in Karachi. Air quality varies spatially across the city, with industrial areas, major roads, and densely populated neighborhoods experiencing different pollution levels. The single-location approach provides a general citywide indicator but cannot capture these spatial variations. Expanding to a network of monitoring locations would enable spatial predictions and identification of pollution hotspots. The temporal coverage of 1000 days is limited by API constraints, and longer historical periods would enable better learning of inter-annual patterns and long-term trends. For example, learning how air quality has evolved over a decade could reveal the impacts of policy changes, economic development, or climate change. However, longer historical periods must be treated carefully as very old data may reflect conditions no longer relevant to current predictions due to changes in emissions, regulations, or urban development.

The feature set, while comprehensive, does not include several potentially valuable data sources. Traffic data from vehicle counting sensors or GPS-based traffic services could directly measure a primary emission source. Satellite-derived measures of aerosol optical depth from instruments like MODIS could provide regional context on pollution transport. Industrial emissions data from major facilities could identify periods of elevated pollution from specific sources. Vegetation indices from satellite imagery could proxy for the "green cover" that removes pollutants from the air. Holiday calendars could capture days when traffic patterns differ substantially from normal weekdays. Each of these additional data sources could incrementally improve predictions, though at the cost of increased complexity and potential data availability issues.

The model architecture itself has room for advancement. The current LightGBM approach excels at capturing non-linear relationships and interaction effects in tabular data, but it treats each time step independently during training. Deep learning architectures designed specifically for time series, such as Long Short-Term Memory networks or Transformer models, could better capture long-range temporal dependencies spanning weeks or months. These approaches could learn that air quality in December is influenced not just by recent days but by the entire seasonal transition from autumn into winter. However, deep learning models require substantially more data to train effectively and lose the interpretability advantages of tree-based models.

Uncertainty quantification represents another area for enhancement. The current system provides point predictions for each forecast horizon, but it does not quantify prediction uncertainty or provide confidence intervals. For public health applications, knowing not just the expected AQI but also the range of plausible values would enable more nuanced decision-making. Probabilistic models such as quantile regression forests, Bayesian neural networks, or ensemble approaches could provide prediction intervals or full predictive distributions. This would allow statements like "there is a 90% probability that AQI will be between 80 and 120 tomorrow" rather than just "predicted AQI is 100."

The system currently operates in a fully automated fashion without human oversight of predictions. Implementing an anomaly detection layer could flag predictions that seem physically implausible or that deviate strongly from recent patterns, triggering manual review before public release. This human-in-the-loop approach could catch rare failure modes while maintaining the efficiency of automated operation for typical cases.

Explainability enhancements could increase trust and utility for expert users. Implementing SHAP (SHapley Additive explanations) values for each prediction would decompose the forecast into contributions from individual features, showing for example that tomorrow's high AQI is predicted primarily due to elevated PM2.5 levels and stagnant wind conditions. Waterfall charts visualizing these contributions would make the model's reasoning transparent. "What-if" scenario analysis could answer questions like "how would the forecast change if wind speed increased to 10 mph?" enabling users to understand the sensitivity of predictions to different factors.

The dashboard could be enhanced with alerting capabilities to proactively notify users of predicted unhealthy air quality. Email or SMS alerts when AQI is forecast to exceed certain thresholds would enable timely protective actions. Integration with existing health advisory platforms used by government agencies or hospitals could amplify the system's public health impact. Personalized alerts based on user health conditions (e.g., alerting asthma patients even at moderate AQI levels) would provide targeted value.

Mobile application development would improve accessibility and enable location-based services. Native iOS and Android apps could deliver push notifications, operate partially offline by caching recent predictions, and integrate with smartphone sensors or wearables to provide personalized exposure estimates. The mobile form factor is particularly important for reaching populations who primarily access internet services through smartphones rather than desktop computers.

Expanding to additional Pakistani cities would multiply the system's impact. Lahore, Islamabad, and Peshawar all experience significant air quality challenges and would benefit from forecasting systems. A multi-city expansion could share model architectures and code infrastructure while training city-specific models that capture local patterns. Comparative dashboards showing air quality across multiple cities could inform policy discussions and help cities learn from each other's successes.

From a research perspective, several directions could advance the scientific understanding of air quality dynamics. Causal inference methods could move beyond prediction to identify the sources and causes of pollution, distinguishing correlation from causation and enabling more targeted interventions. For example, difference-in-differences analysis could estimate the impact of traffic reduction policies on AQI by comparing periods when restrictions were in place to similar periods without restrictions. Hybrid physics-based machine learning models could combine atmospheric dispersion equations with data-driven components, leveraging physical knowledge about pollutant transport while learning empirical relationships from data. Transfer learning could pre-train models on global air quality data from many cities and fine-tune on Karachi-specific data, potentially improving performance with limited local data by leveraging patterns common across cities worldwide.

# 15. Conclusion

This project successfully built a production-ready AQI prediction system for Karachi using LightGBM to forecast air quality 24-72 hours ahead with $R^2$ scores above 0.75. The system combines OpenWeather pollution data and Meteostat weather data, applies custom US EPA AQI calculations, and uses 20 engineered features including cyclical encodings, lag variables, and physics-based interactions. Automated pipelines run hourly data collection and daily model retraining via GitHub Actions, delivering predictions through an intuitive web dashboard.

**Key Technical Contributions:**

- Demonstrated best practices in applied ML from rigorous EDA through production deployment
- Feature engineering combined statistical techniques with atmospheric science domain knowledge
- Systematic model comparison identified LightGBM as optimal
- Modular pipeline architecture enables reliable automated operation

**Project Structure:**

- **pearlsaqipredictor.ipynb**: Experimental work, EDA, model comparison
- **finalaqipredictor.ipynb**: Clean production pipeline code

**Key Insights:**

- Cyclical encodings outperformed categorical dummies for temporal features
- Rolling averages reduced target noise and aligned with health practices
- Interaction features (PM2.5×humidity) encoded physical processes effectively
- Simple models (LightGBM) matched or beat complex alternatives while being faster and more interpretable

**Impact:** Demonstrates that accurate air quality forecasting is achievable using free data sources and open-source tools, providing a replicable template for other South Asian cities facing similar challenges.

# References

**Data Sources:**

- OpenWeather Air Pollution API: https://openweathermap.org/api/air-pollution
- OpenWeather Geocoding API: https://openweathermap.org/api/geocoding-api
- Meteostat Python Library: https://meteostat.net/en/
- EPA Air Quality Index (AQI) Standards: https://www.airnow.gov/aqi/aqi-basics/

**Machine Learning Frameworks and Tools:**

- LightGBM Documentation: https://lightgbm.readthedocs.io/
- Scikit-learn: https://scikit-learn.org/
- Pandas: https://pandas.pydata.org/
- NumPy: https://numpy.org/

**Feature Store and Deployment:**

- Hopsworks Feature Store: https://www.hopsworks.ai/
- Gradio: https://gradio.app/
- GitHub Actions: https://github.com/features/actions

**Atmospheric Science Background:**

- EPA's Technical Assistance Document for AQI Calculation
- Atmospheric chemistry of pollutant formation and transport
- Boundary layer meteorology and pollution dispersion

# Appendices

## Appendix A: Complete Feature List

**Pollutant Features (4):**

1. pm2_5: Fine particulate matter concentration (μg/m³)
2. pm10: Coarse particulate matter concentration (μg/m³)
3. no2: Nitrogen dioxide concentration (μg/m³)
4. so2: Sulfur dioxide concentration (μg/m³)

**Lag Features (3):**

5. onedaylag: AQI from 24 hours prior
6. twodaylag: AQI from 48 hours prior
7. threedaylag: AQI from 72 hours prior

**Rolling Statistics (2):**

8. aqi_roll_std_3: Standard deviation of AQI over past 72 hours
9. us_aqi_roll_mean_3d: Mean AQI over past 72 hours

**Interaction Features (2):**

10. pm2_5_rhum: PM2.5 concentration × relative humidity
11. no2_temp: NO2 concentration × temperature

**Meteorological Features (4):**

12. temp: Temperature (°C)
13. rhum: Relative humidity (%)
14. pres: Atmospheric pressure (hPa)
15. wspd: Wind speed (km/h)

**Cyclical Temporal Features (6):**

16. sin_hour: sin(2π × hour/24)
17. cos_hour: cos(2π × hour/24)
18. sin_dayofweek: sin(2π × dayofweek/7)
19. cos_dayofweek: cos(2π × dayofweek/7)

20. sin_month: sin(2π × month/12)
21. cos_month: cos(2π × month/12)

## Appendix B: Model Hyperparameters

**LightGBM Configuration:**

- n_estimators: 2000 (number of boosting iterations)
- learning_rate: 0.01 (step size shrinkage)
- num_leaves: 64 (maximum leaves per tree)
- max_depth: -1 (no depth limit, controlled by num_leaves)
- subsample: 0.8 (fraction of data for each iteration)
- colsample_bytree: 0.8 (fraction of features for each tree)
- reg_alpha: 0.2 (L1 regularization)
- reg_lambda: 0.2 (L2 regularization)
- min_child_samples: 30 (minimum data in leaf)
- random_state: 42 (reproducibility)
- n_jobs: -1 (use all CPU cores)

**Preprocessing Configuration:**

- PowerTransformer: method='yeo-johnson', standardize=False
- StandardScaler: default parameters (zero mean, unit variance)
- TimeSeriesSplit: n_splits=3, test_size=1000, gap=0

## Appendix C: Code Structure and Organization

**Main Project Files:**

- pearlsaqipredictor.py: Experimental notebook with EDA, feature engineering experiments, model comparison, and validation studies
- finalaqipredictor.py: Production pipeline with modular architecture for automated deployment
- requirements.txt: Python package dependencies with version specifications
- .github/workflows/feature_pipeline.yml: Hourly data collection automation
- .github/workflows/training_pipeline.yml: Daily model retraining automation

**Generated Artifacts:**

- final_lgbm_multioutput.pkl: Trained MultiOutputRegressor model (3 LightGBM estimators)
- scaler.pkl: Fitted StandardScaler for feature normalization
- feature_power_transformer.pkl: Fitted PowerTransformer for feature distributions
- target_power_transformer.pkl: Fitted PowerTransformer for target distributions
- history_aqi.csv: Cached historical data for backup and analysis

**Core Functions in Production Pipeline:**

*Data Collection:*

- fetch_lat_lon(city_name, api_key): Retrieves geographic coordinates
- fetch_aqi_history_data(lat, lon, start, end, api_key): Collects pollution data
- fetch_weather_data(lat, lon, start, end): Collects meteorological data

*Data Processing:*

- merge_and_preprocess(df_aqi, df_weather): Integrates multiple data sources
- get_aqi_us(components): Calculates US EPA AQI from pollutant concentrations

*Feature Engineering:*

- create_features(df): Generates cyclical temporal encodings
- add_lags(df, col): Creates lag features and rolling statistics
- create_targets(df): Generates single target for validation
- create_targets_full(df): Generates multi-horizon targets for production

*Pipeline Orchestration:*

- run_feature_pipeline(): Executes complete data collection and feature storage workflow
- run_training_pipeline(): Executes model training and artifact saving workflow
- run_dashboard(): Launches Gradio web interface for predictions

*Utility Functions:*

- save_to_feature_store(df, name, description, version, api_key): Writes features to Hopsworks
- load_from_feature_store(name, version): Reads features from Hopsworks
- get_aqi_category(aqi): Maps AQI value to EPA category and color
- fig_to_image(fig): Converts matplotlib figure to PIL Image for dashboard

## Appendix E: API Configuration and Rate Limits

**OpenWeather Air Pollution API:**

- Free tier: 60 calls per minute, 1,000 calls per day
- Current usage: 1 call per hour = 24 calls per day (well within limits)
- Endpoint: http://api.openweathermap.org/data/2.5/air_pollution/history
- Response format: JSON with timestamp, AQI, and pollutant concentrations

**OpenWeather Geocoding API:**

- Free tier: 60 calls per minute
- Current usage: 1 call per pipeline execution (once per hour)
- Endpoint: http://api.openweathermap.org/geo/1.0/direct
- Response format: JSON with latitude, longitude, city name, country

**Meteostat Python Library:**

- Data source: Global historical weather observations
- No explicit rate limits (data cached locally)
- Coverage: Hourly data available for major cities worldwide
- Variables: Temperature, humidity, wind speed, pressure, precipitation, etc.

**Hopsworks Feature Store:**

- Free tier: Suitable for development and small production deployments
- API rate limits: Vary by plan, adequate for hourly writes and on-demand reads
- Storage: Features stored with versioning and schema validation
- Access: RESTful API with Python SDK

## Appendix F: Deployment Checklist

**Pre-Deployment:**

- API keys obtained for OpenWeather and Hopsworks
- API keys added to GitHub Secrets (never committed to repository)
- requirements.txt tested and all dependencies install successfully
- Feature pipeline executes without errors on sample data
- Training pipeline produces valid model artifacts
- Dashboard loads and displays predictions correctly
- Model artifacts saved with version numbers for tracking
- Error handling implemented for API failures and network issues
- Logging configured to capture pipeline execution details

**Deployment:**

- Code committed to GitHub repository
- GitHub Actions workflows enabled in repository settings
- Feature pipeline workflow triggered manually to verify functionality
- Training pipeline workflow triggered manually to verify functionality
- Scheduled executions confirmed (hourly for features, daily for training)
- Dashboard deployed to accessible URL or local environment
- Initial predictions generated and validated against expectations

**Post-Deployment Monitoring:**

- Monitor GitHub Actions execution logs daily for first week
- Verify feature store updates occurring hourly as scheduled
- Verify model retraining occurring daily as scheduled
- Track prediction accuracy against actual AQI when available
- Set up alerts for pipeline failures or anomalous predictions
- Document any issues encountered and solutions implemented
- Gather user feedback on dashboard usability and features
- Plan quarterly performance review and model improvements

## Appendix G: Error Handling and Recovery

**Common Failure Modes and Mitigations:**

**API Unavailability:**

- Symptom: HTTP 5xx errors or timeout exceptions
- Mitigation: Retry logic with exponential backoff (3 attempts with 1s, 2s, 4s delays)
- Fallback: Log error and skip iteration; next hourly run will attempt again
- Alert: Send notification after 3 consecutive failures

**Rate Limit Exceeded:**

- Symptom: HTTP 429 "Too Many Requests"
- Mitigation: Respect Retry-After header if provided
- Fallback: Wait until next scheduled execution

- Prevention: Monitor API usage against limits

**Invalid API Key:**

- Symptom: HTTP 401 "Unauthorized"
- Mitigation: None (requires manual intervention to update key)
- Alert: Immediate notification to administrators
- Prevention: Test API keys before deployment

**Missing or Corrupt Data:**

- Symptom: NaN values, malformed JSON, unexpected data types
- Mitigation: Validation checks after data collection
- Fallback: Use forward-fill for weather data up to 3 hours
- Prevention: Schema validation and data quality checks

**Feature Store Connection Issues:**

- Symptom: Timeout or authentication errors with Hopsworks
- Mitigation: Retry connection with fresh authentication
- Fallback: Cache features locally as backup
- Prevention: Monitor Hopsworks service status

**Model Training Failures:**

- Symptom: NaN values in predictions, convergence errors, OOM errors
- Mitigation: Data validation before training, check for sufficient memory
- Fallback: Keep previous model version, skip update
- Prevention: Validate data completeness and distribution before training

## Appendix H: Future Enhancement Roadmap

**Short-term (1-3 months):**

1. Implement email/SMS alerting for predicted unhealthy AQI levels
2. Add historical comparison showing "AQI same day last week/month/year"
3. Expand dashboard to show pollutant-specific contributions to AQI
4. Create mobile-responsive layout optimizations for smaller screens
5. Add downloadable reports (PDF/CSV) of predictions and historical data

**Medium-term (3-6 months):**

1. Expand to 2-3 additional Pakistani cities (Lahore, Islamabad)
2. Implement SHAP-based explainability for individual predictions
3. Add uncertainty quantification (prediction intervals) using quantile regression
4. Integrate traffic data from Google Maps API or similar sources
5. Develop comparative dashboard showing multiple cities side-by-side
6. Implement A/B testing framework for model improvements

**Long-term (6-12 months):**

1. Develop native mobile applications for iOS and Android
2. Implement spatial predictions across multiple monitoring locations
3. Integrate satellite-derived aerosol optical depth data
4. Explore deep learning architectures (LSTM, Transformers) for comparison
5. Partner with local health authorities for official advisory integration

6. Publish research paper documenting methodology and findings
7. Open-source complete codebase with documentation for replication
8. Develop transfer learning approach for rapid deployment in new cities

## Appendix I: Acknowledgments and Data Usage

**Data Providers:** This project would not be possible without the generous data access provided by OpenWeather (https://openweathermap.org/) for historical and real-time air pollution data, and Meteostat (https://meteostat.net/) for meteorological observations. Both services make environmental data accessible to researchers and developers worldwide.

**Open Source Software:** The project relies heavily on the Python scientific computing ecosystem including NumPy, Pandas, Scikit-learn, LightGBM, Matplotlib, and Gradio. The Hopsworks Feature Store provides critical infrastructure for production ML systems. GitHub Actions enables reliable automation without infrastructure costs.

**Ethical Considerations:** This system is designed to inform public health decisions and raise awareness about air quality. Predictions should be used as guidance rather than absolute truth, and users with serious health conditions should consult medical professionals. The system is provided as-is without warranties, and users bear responsibility for decisions made based on its outputs.

**Data Privacy:** No personal data is collected from dashboard users. All data sources are publicly available environmental measurements. API keys are stored securely and never exposed in public repositories or logs.

# Document Information

**Report Title:** Karachi AQI Prediction System: A Comprehensive Research Report

**Project Name:** Karachi Pearls AQI Predictor

**Primary Files:**

- Experimental Notebook: pearlsaqipredictor.ipynb
- Production Pipeline: finalaqipredictor.ipynb

**Authors:** Data Science Team

**Date:** November 2025

**Version:** 1.0

**Document Type:** Technical Research Report

**Total Length:** ~18,000 words across 15 sections plus appendices

**Target Audience:** Data scientists, environmental researchers, public health officials, and technical stakeholders

**Keywords:** Air Quality Index, Machine Learning, Time Series Forecasting, LightGBM, Feature Engineering, Pakistan, Karachi, Environmental Monitoring, Public Health

**END OF REPORT**