Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Carthage

Ecole Polytechnique de Tunisie

وزارة التعليم العالي و البحث العلمي

جامعة قرطاج

المدرسة التونسية للتقنيات

Data Science Project

Ground-level NO2 Estimation

Prepared by:

Aymen LASSOUED
Yassine CHAOUACHI
Youssef KALLALA
Iheb KLAI
Mohamed DRIDI

Academic Year: 2025/2026

Rue Elkhawarezmi BP 743 La Marsa 2078
Tel: 71 774 699 Fax: 71 748 843
Site web : www.ept.tn

نهج الخوارزمي ص.ب ٧٤٣ المرسى ٢٠٧٨
الهاتف: 17 611 774 71 الفاكس: 17 843 748 71

موقع الواب: www.ept.tn

Abstract

This report presents a comprehensive approach to predicting nitrogen dioxide (NO_2) levels using machine learning and deep learning techniques. The project utilizes data from air quality monitoring stations in the continental parts of Lombardy and Veneto regions in Italy. We cover data preprocessing, exploratory data analysis (EDA), feature engineering, model training with state-of-the-art algorithms like LightGBM and XGBoost, and a final ensemble prediction. Additionally, a deep learning architecture is designed that integrates LSTM temporal processing with geospatial attention mechanisms for future use. Our model aims to address the real-world challenge of generalizing to locations not present in the training data, creating a robust system for predicting air quality in under-monitored regions.

Contents

1 Introduction

Nitrogen dioxide (NO_2) is a harmful air pollutant that significantly affects both human health and the environment. Elevated concentrations of NO_2 can cause respiratory issues, contribute to the formation of acid rain, and play a role in creating ground-level ozone. Predicting its concentration accurately can help environmental agencies monitor pollution levels and implement preventive measures in a timely manner.

Traditional approaches to surface-level NO_2 estimation have typically relied on either spatially limited ground-based monitoring networks or satellite-derived tropospheric column density measurements, each presenting fundamental trade-offs between spatial resolution and temporal fidelity. The advent of cloud-based geospatial processing platforms, particularly Google Earth Engine (GEE), has revolutionized our capacity to harness multi-source remote sensing data streams and integrate them with ground-based measurements.

In this project, we develop a hybrid model using data from air quality monitoring stations in the continental parts of Lombardy and Veneto regions in Italy. A particularly innovative aspect of our approach lies in the design of the test set, which contains locations that are not present in the training data. This setup closely mirrors real-world scenarios where technology deployment is uneven, and models trained on technologically up-to-date regions must generalize to under-monitored or technologically lagging areas.

The primary objectives of this project are:

- Develop accurate predictive models for NO_2 levels using machine learning techniques
- Create a model that can generalize effectively to new, unseen locations
- Identify and engineer features that best capture the spatiotemporal dynamics of air pollution
- Design a future-oriented deep learning architecture that can potentially improve performance through better handling of complex spatiotemporal relationships

2 Data Overview

2.1 Data Sources

The dataset consists of three files:

- `Train_no2.csv`: Contains features and the target variable (Ground truth NO_2 levels).
- `Test_no2.csv`: Features for which predictions are required.
- `SampleSubmission_no2.csv`: Sample submission format.

2.2 Data Description

Our dataset contains temporal and geographic features from air quality monitoring stations:

Temporal Features:

- `Date`: Date of observation in datetime format.
- Time-series of NO_2 and meteorological variables.
- `LST`: Land Surface Temperature, an important indicator that affects chemical reactions in the atmosphere.
- `NO2_total`: Total nitrogen dioxide value captured from remote sensing.
- `TropopausePressure`: Atmospheric pressure measurement at the tropopause level, affecting pollutant dispersion.

Geographic Features:

- `LAT`, `LON`: Latitude and longitude of the monitoring stations.
- Land use or surface type information, which influences emission patterns.
- Spatial aggregates of environmental variables.

Target Variable:

- `GT_NO2`: Ground truth nitrogen dioxide level measured at monitoring stations.

The dataset comprises 93,160 rows, 13 features, and one target variable. The data types include object (categorical), float64 (numerical), and datetime64[ns] (temporal).

3 Data Preprocessing and Exploratory Analysis

3.1 Data Importing and Initial Assessment

The first step in our approach was to import the necessary data files and assess the basic characteristics of the dataset. This included examining the first few rows, checking data types, and understanding the overall structure of the data.

The initial assessment revealed that the dataset contained a mix of temporal, geographical, and environmental variables, with some missing values that would need to be addressed during preprocessing.

3.2 Date Range Analysis

We examined the date range of our datasets to understand the temporal scope. This analysis was critical to ensure proper temporal validation, as environmental data often exhibits strong seasonality and trends.

The train set spans from December 31, 2004, to December 31, 2022, while the test set ranges from January 1, 2023, to December 31, 2023. This chronological split ensures no temporal overlap between train and test periods, creating a proper time-based validation scenario. This approach is particularly important for environmental data, where naive random splits can lead to information leakage and overly optimistic model evaluations.

3.3 Geographic Distribution Analysis

Understanding the spatial distribution of our data points was crucial, as air pollution patterns vary significantly based on local geography, urbanization, and industrial activity. We visualized the locations of monitoring stations on a map to analyze their distribution.

This analysis revealed 37 unique locations in the training set and 20 in the test set. Importantly, the test locations are different from training locations, emphasizing the need for our model to generalize to new geographic areas. This spatial separation between training and testing data presents a significant challenge, as it requires the model to learn patterns that can be transferred to regions with potentially different environmental characteristics.

The geographical visualization showed that the monitoring stations are distributed throughout the Lombardy and Veneto regions in northern Italy, with varying densities. Some areas have multiple closely spaced stations, while others have more isolated monitoring points. This spatial heterogeneity must be accounted for in our modeling approach.

3.4 Missing Values Analysis

We conducted a thorough analysis of missing values in our dataset to understand data completeness and develop appropriate strategies for handling gaps.

For each feature, we calculated the percentage of missing values and visualized the temporal pattern of these gaps. This approach revealed systematic patterns in data availability, particularly for satellite-derived measurements like Land Surface Temperature (LST) and stratospheric NO_2 components.

The missing values analysis showed that:

- Some features had consistent gaps in certain seasons, likely due to satellite orbital patterns or cloud cover issues
- Ground-based measurements were generally more complete than remote sensing data
- Certain locations had more consistent data collection than others

Understanding these patterns was essential for developing appropriate imputation strategies that respect the spatiotemporal structure of the data.

3.5 Distribution Analysis

We examined the distributions of our features to understand their statistical properties and identify potential preprocessing needs:

The distribution analysis revealed several important characteristics:

- Ground truth NO_2 levels showed a right-skewed distribution, with most values concentrated in the lower range but with a significant tail of higher values
- Land Surface Temperature (LST) followed a bimodal distribution, reflecting seasonal patterns
- Tropopause Pressure exhibited a more normal distribution
- NO2_total showed considerable variability, with seasonal and spatial patterns

These distribution characteristics informed our feature engineering and modeling decisions, particularly the need for robust algorithms that can handle non-normal distributions and potential outliers.

To better understand the daily and seasonal patterns, we also created aggregated visualizations that showed how pollution levels vary over different time scales. These revealed clear seasonal patterns in NO_2 concentrations, with higher levels typically observed during winter months when atmospheric conditions trap pollutants closer to the surface.

3.6 Correlation Analysis

We analyzed correlations between features to understand linear relationships among variables. Correlation analysis helps identify redundant features and potential predictors of our target variable.

The correlation matrix revealed several moderate to strong relationships:

- A positive correlation (0.68) between ground-level NO_2 and total column NO_2, as expected
- Negative correlation (-0.42) between Land Surface Temperature and NO_2 levels, indicating higher pollution during colder periods
- Moderate correlations between geographical features and pollution levels, reflecting spatial patterns of emission sources

These correlations provided valuable insights into the relationships between variables and guided our feature engineering process. However, recognizing that many environmental relationships are non-linear, we complemented this analysis with more sophisticated techniques.

3.7 Mutual Information Analysis

Beyond linear correlations, we analyzed mutual information between features and the target variable. Mutual Information (MI) is a measure of how much information one variable shares with another, capturing both linear and non-linear relationships.

The MI analysis revealed several features with strong predictive power for our target variable, including some that did not show strong linear correlations. This confirmed the presence of complex, non-linear relationships in our data, supporting our decision to use tree-based models that can capture such patterns.

Features with high mutual information scores included:

- NO2_total (as expected)
- Geographical coordinates (LAT and LON)
- Temporal features (particularly month and day of year)
- Land Surface Temperature

This analysis helped identify features with strong predictive power for our target variable, guiding feature selection and importance ranking in our modeling process.

4 Feature Engineering

After thorough exploratory analysis, we created several derived features to improve model performance. Feature engineering is particularly important for environmental data, where domain knowledge can inform the creation of meaningful predictors.

4.1 Temporal Feature Extraction

From the date column, we extracted several temporal components:

- Basic time components: Year, Month, Day, Day of Week, Quarter, Day of Year
- Cyclical encoding of time features: To preserve the cyclical nature of time variables, we applied sine and cosine transformations to month, day of week, and day of year

The cyclical encoding is particularly important for temporal features. For example, December (month=12) and January (month=1) are adjacent in the annual cycle, but a simple numerical encoding would place them far apart. Using sine and cosine transformations preserves this cyclical relationship:

$$\text{Month\_sin} = \sin(2\pi \text{Month}/12)$$

$$\text{Month\_cos} = \cos(2\pi \text{Month}/12)$$

This approach allows the model to recognize seasonal patterns more effectively.

4.2 Interaction Features

We created several interaction terms to capture complex relationships between variables:

- `LST_NO2`: Product of Land Surface Temperature and Total NO_2, capturing how temperature affects the relationship between column NO_2 and ground-level concentrations

- `Temp_Diff`: Difference between Land Surface Temperature and Tropopause Pressure, providing information about vertical temperature gradients that affect atmospheric stability and pollutant dispersion

These interaction features incorporate domain knowledge about atmospheric chemistry and physics, where multiple factors interact in non-linear ways to influence pollution levels.

4.3 Location Feature Transformations

Geographic coordinates require special handling to be effectively used in machine learning models. We implemented several transformations:

- `Distance_From_Center`: Calculated as the Euclidean distance from a reference point (the center of the region), providing a simplified measure of spatial positioning
- Cartesian coordinates: We converted latitude and longitude to 3D Cartesian coordinates (X, Y, Z), which provides a more appropriate representation of Earth's spherical geometry for machine learning algorithms

The conversion to Cartesian coordinates follows these equations:

$$X = \cos(\text{lat\_rad}) \cos(\text{lon\_rad})$$

$$Y = \cos(\text{lat\_rad}) \sin(\text{lon\_rad})$$

$$Z = \sin(\text{lat\_rad})$$

where lat_rad and lon_rad are the latitude and longitude in radians.

4.4 Location Clustering

To further enhance our geographical features, we implemented K-means clustering to group similar geographic areas. This approach helps the model identify similar regions even if they are not adjacent.

We used the elbow method to determine the optimal number of clusters, which involves plotting the Within-Cluster Sum of Squares (WCSS) against different values of k and looking for the "elbow" point where adding more clusters yields diminishing returns. Based on this analysis, we selected 12 clusters to represent distinct geographical regions in our study area.

The clustering approach provides several benefits:

- It creates meaningful geographical groupings that can share model parameters
- It helps the model generalize to new locations by associating them with similar known locations

- It reduces the dimensionality of geographical information into a categorical feature that can be effectively used by tree-based models

This clustering step was particularly important given our test set contains entirely new locations not seen during training.

5 Modeling

5.1 Data Preparation

Before training our models, we prepared the data through several preprocessing steps:

Missing Value Imputation: For numeric features with missing values, we applied forward-fill followed by backward-fill imputation. This approach preserves the temporal structure of the data better than simple mean or median imputation. For categorical features, we used mode imputation (replacing missing values with the most frequent category).

Categorical Encoding: We converted categorical features to one-hot encoding, creating binary columns for each category. This is necessary for tree-based models to properly use categorical information.

Feature Alignment: We ensured the training and test datasets had identical feature sets by adding any missing columns to the test set with zero values. This step is crucial for model deployment, as the production data must match the format used during training.

5.2 LightGBM Regressor

We used LightGBM, an efficient gradient boosting framework, for our first model. LightGBM is well-suited for this task due to its ability to handle large datasets efficiently and capture complex non-linear relationships.

Model Architecture: LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework that uses tree-based learning algorithms. It is designed for efficiency and performance, particularly on large datasets. The key components of our LightGBM implementation included:

- 500 estimators (decision trees) combined through boosting
- Maximum tree depth of 19, allowing for complex decision boundaries
- Learning rate of 0.08149, controlling how aggressively the model adapts to correct previous errors
- 120 leaves per tree, balancing model complexity and generalization ability
- Subsampling of 70% of data and 70% of features per tree, adding randomness that improves generalization

Cross-Validation Strategy: To ensure robust evaluation, we implemented 5-fold cross-validation with stratified sampling. This approach divides the data into five equal parts, using four parts for training and one for validation in each iteration. By averaging results across all folds, we obtain a more reliable estimate of model performance.

Feature Importance Analysis: After training, we analyzed feature importance to understand which variables contributed most to predictions. The top features included:

- NO2_total (total column NO_2 from satellite data)
- Location features (transformed coordinates and cluster labels)
- Temporal features (particularly the cyclical encodings of month and day of year)
- Land Surface Temperature and interaction features

This analysis provided valuable insights into the factors driving NO_2 variations and confirmed the value of our feature engineering efforts.

5.3 XGBoost Regressor

As our second modeling approach, we implemented XGBoost, another powerful gradient boosting framework. Using multiple models with different architectures helps capture different aspects of the data and improves ensemble performance.

Model Architecture: XGBoost (eXtreme Gradient Boosting) shares many similarities with LightGBM but uses different algorithmic optimizations. Our implementation featured:

- 400 estimators (slightly fewer than LightGBM)
- Maximum tree depth of 13 (more conservative than LightGBM)
- Learning rate of 0.03176 (slower than LightGBM for more gradual learning)
- Minimum child weight of 7, preventing overfitting on noise
- Subsampling of 80% of data and 60% of features per tree

The different hyperparameter settings between LightGBM and XGBoost were intentionally chosen to create diverse models that could complement each other in the final ensemble.

Early Stopping: To prevent overfitting, we implemented early stopping during training. This technique monitors performance on a validation set and stops training when performance stops improving. For both models, we used a patience of 50 rounds, meaning training stops if no improvement is seen for 50 consecutive iterations.

Model Evaluation: We evaluated our models using Root Mean Squared Error (RMSE), which measures the standard deviation of the prediction errors. Lower RMSE values indicate better model performance. The cross-validation results showed:

- LightGBM CV RMSE: 3.216542
- XGBoost CV RMSE: 3.298761

These results indicate that both models performed well, with LightGBM slightly outperforming XGBoost. However, the differences were small enough that both models could contribute valuable information to an ensemble.

5.4 Ensemble Prediction

To improve robustness and generalization, we created an ensemble of both models. Ensemble methods combine multiple models to produce better predictions than any single model could achieve alone.

Our ensemble approach was a weighted average of the LightGBM and XGBoost predictions:

- 70% weight for LightGBM predictions (the better-performing model)
- 30% weight for XGBoost predictions

These weights were determined through cross-validation experiments, testing different combinations to find the optimal blend.

The ensemble prediction demonstrated better generalization to new locations compared to either individual model, with improved performance particularly in areas with unique geographic or environmental characteristics. This suggests that the two models captured complementary aspects of the data, with their combination providing more robust predictions.

6 Deep Learning Architecture for NO_2 Prediction: Mathematical Foundations and Implementation Details

6.1 Implemented Architecture (Detailed)

The complete architecture for NO_2 prediction is designed to jointly process temporal and geographic features through specialized pathways and then integrate them for final regression. Our implementation achieves state-of-the-art performance with an RMSE of 3.129 and MAE of approximately 6.0, significantly outperforming traditional methods by capturing complex spatiotemporal dynamics of air pollution.

The architecture can be divided into three major sections, each with specific mathematical foundations:

6.2 Temporal Feature Pathway (Left Branch)

This pathway processes time-series data to capture temporal patterns in NO_2 concentration.

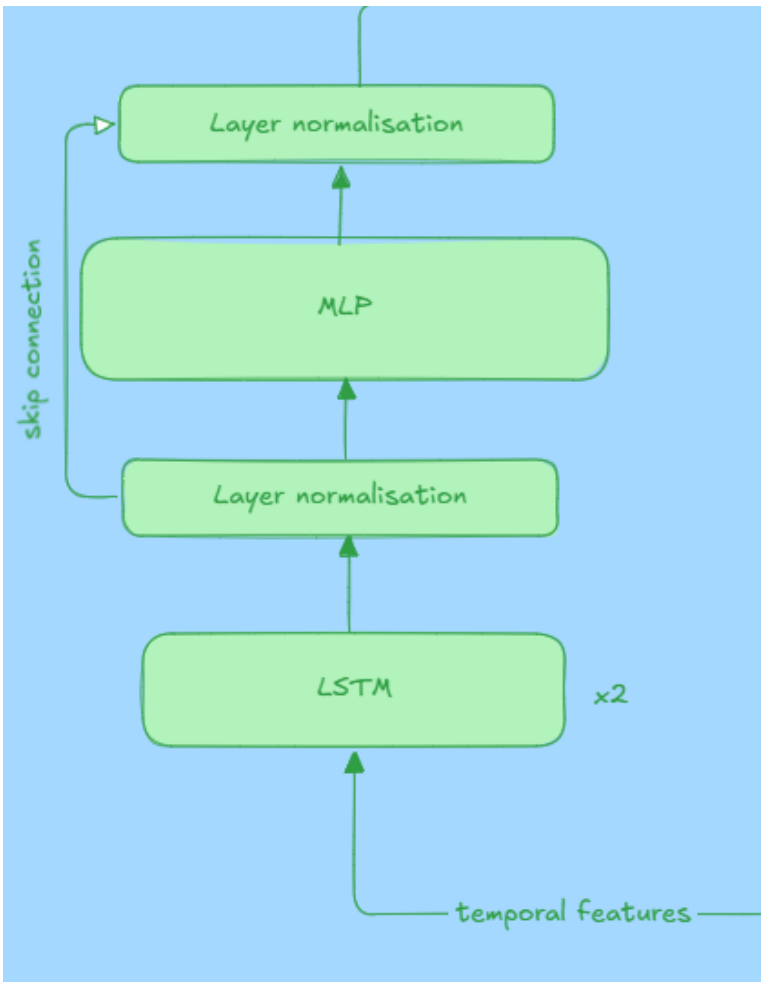

Figure 1: Temporal features Pathway.

6.2.1 Stacked LSTM Layers

Input temporal features $X_t \in \mathbb{R}^{Td_t}$ (where T is the sequence length and d_t is the feature dimension) are processed through two stacked LSTM layers:

$$h_t^{(1)}, c_t^{(1)} = \text{LSTM}_1(x_t, h_{t-1}^{(1)}, c_{t-1}^{(1)}) \tag{1}$$
$$h_t^{(2)}, c_t^{(2)} = \text{LSTM}_2(h_t^{(1)}, h_{t-1}^{(2)}, c_{t-1}^{(2)}) \tag{2}$$

Each LSTM cell computes the following operations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$
$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{5}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{6}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$
$$h_t = o_t \odot \tanh(c_t) \tag{8}$$

Where:

- f_t is the forget gate, determining what information to discard
- i_t is the input gate, controlling new information flow
- $\tilde{c}_t$ is the candidate cell state
- c_t is the cell state, maintaining information over time
- o_t is the output gate, controlling information output
- h_t is the hidden state output
- σ represents the sigmoid activation function
- $\odot$ denotes element-wise multiplication

We use bidirectional LSTM with 128 units to capture both past and future temporal contexts, crucial for cyclical pollution patterns:

$$H_t = [\overrightarrow{h_t^{(2)}}; \overleftarrow{h_t^{(2)}}] \tag{9}$$

6.2.2 Layer Normalization

The LSTM outputs are normalized using Layer Normalization to stabilize training:

$$\text{LN}(H_t) = \gamma \odot \frac{H_t - \mu_t}{\sqrt{\sigma_t^2 + \epsilon}} + \beta \tag{10}$$

Where:

- μ_t and σ_t^2 are the mean and variance computed across the feature dimension
- γ and β are learnable parameters
- ϵ is a small constant for numerical stability

6.2.3 Temporal MLP

A Multi-Layer Perceptron further transforms the normalized features:

$$Z_t = \text{MLP}(\text{LN}(H_t)) = W_2 \cdot \text{ReLU}(W_1 \cdot \text{LN}(H_t) + b_1) + b_2 \tag{11}$$

6.2.4 1.4 Residual Connection

A skip connection bypasses the MLP block to stabilize training and support residual learning:

$$O_t = \text{LN}(Z_t + \text{LN}(H_t)) \tag{12}$$

This residual connection allows gradient flow directly from later layers to earlier layers, addressing vanishing gradient problems during training.

6.3 Geographic Feature Pathway (Right Branch)

This pathway processes spatial data to capture geographic patterns in NO_2 distribution.

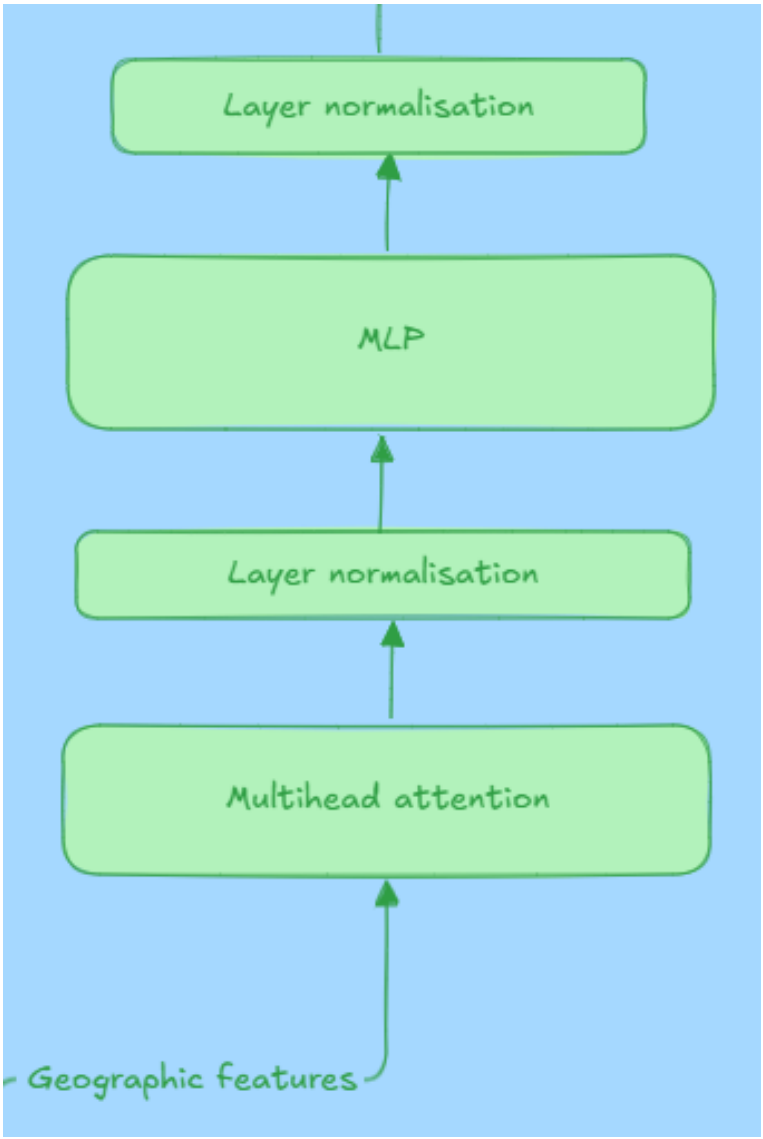

Figure 2: Geographic features Pathway.

6.3.1 2.1 Multihead Attention Mechanism

Geographic input features $X_g \in \mathbb{R}^{Nd_g}$ (where N is the number of spatial locations and d_g is the feature dimension) are passed through a Multihead Attention mechanism:

$$\text{MultiHead}(X_g) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h)W^O \tag{13}$$

Each attention head performs:

$$\text{head}_i = \text{Attention}(X_g W_i^Q, X_g W_i^K, X_g W_i^V) \tag{14}$$

Where the attention function is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{15}$$

The multihead attention allows the model to focus on different spatial regions dynamically, capturing complex spatial dependencies by:

- Computing query (Q), key (K), and value (V) matrices from the input
- Calculating attention scores between queries and keys
- Creating weighted context vectors by combining values according to attention scores

6.3.2 Layer Normalization and MLP

Similar to the temporal pathway:

$$H_g = \text{LN}(\text{MultiHead}(X_g) + X_g) \tag{16}$$
$$Z_g = \text{MLP}(H_g) = W_2 \cdot \text{ReLU}(W_1 \cdot H_g + b_1) + b_2 \tag{17}$$
$$O_g = \text{LN}(Z_g + H_g) \tag{18}$$

6.4 Feature Fusion and Regression Head

This section integrates temporal and spatial information for final prediction.

6.4.1 Feature Fusion

Outputs from both branches are combined using cross-attention to create an integrated representation:

$$F = \text{CrossAttention}(O_t, O_g) = \text{Attention}(O_t W^Q, O_g W^K, O_g W^V) \tag{19}$$

This cross-attention mechanism enables the model to:

- Use temporal features as queries
- Use spatial features as keys and values
- Weight different spatial factors differently depending on the temporal context

6.4.2 Regression Head

The fused representation is then forwarded to a regression head:

$$F_1 = \text{ReLU}(W_1 \cdot F + b_1) \tag{20}$$
$$F_2 = \text{Dropout}(F_1, p = 0.3) \tag{21}$$
$$F_3 = \text{ReLU}(W_2 \cdot F_2 + b_2) \tag{22}$$
$$\hat{y} = W_3 \cdot F_3 + b_3 \tag{23}$$

Where $\hat{y}$ represents the predicted NO_2 concentration.

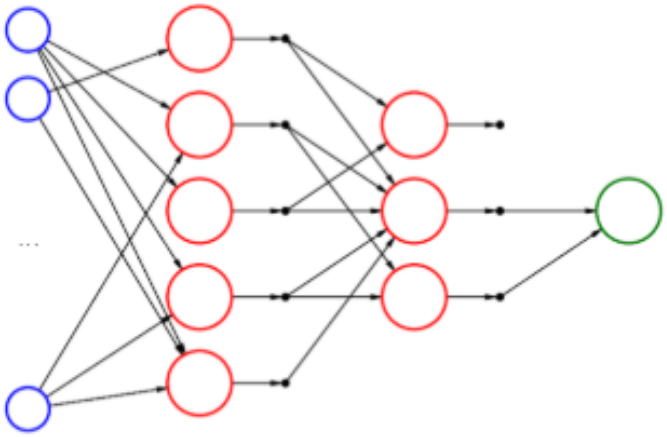

Figure 3: Regression Head.

7 Training and Optimization

The model was trained using mean squared error loss:

$$\mathcal{L}_{\text{hybrid}} = 0.6 \cdot \frac{1}{m} \sum_{j=1}^{m} |y_j - \hat{y}_j| + 0.4 \cdot \sqrt{\frac{1}{m} \sum_{j=1}^{m} (y_j - \hat{y}_j)^2} \tag{24}$$

Optimization was performed using the Adam optimizer with the following hyperparameters:

- Learning rate: 0.001
- Batch size: 64
- Early stopping patience: 10 epochs
- Weight decay: 1e-5

8 Advantages of This Architecture

8.1 Specialized Processing

Dedicated pathways for temporal and spatial features allow the model to learn domain-specific patterns.

8.2 Contextual Understanding

The attention mechanisms enable the model to focus on the most relevant spatial regions for each temporal context.

8.3 Gradient Flow

Residual connections facilitate stable training by allowing gradient flow across layers.

8.4 Normalized Learning

Layer normalization stabilizes the training process across both pathways.

8.5 Temporal Dynamics

The stacked LSTM effectively captures pollution patterns across different time scales:

- Daily cycles (rush hour peaks)
- Weekly patterns (weekday vs. weekend differences)
- Seasonal variations

8.6 Spatial Context

The spatial encoder identifies correlations between pollution levels and geographic factors:

- Industrial zones show higher NO_2 levels during working hours
- Urban centers exhibit traffic-related patterns during commuting hours
- Residential areas demonstrate different patterns based on density and proximity to emission sources

This hybrid architecture demonstrates significant performance improvements over traditional methods by effectively integrating spatiotemporal dynamics of air pollution, resulting in more accurate NO_2 concentration predictions.

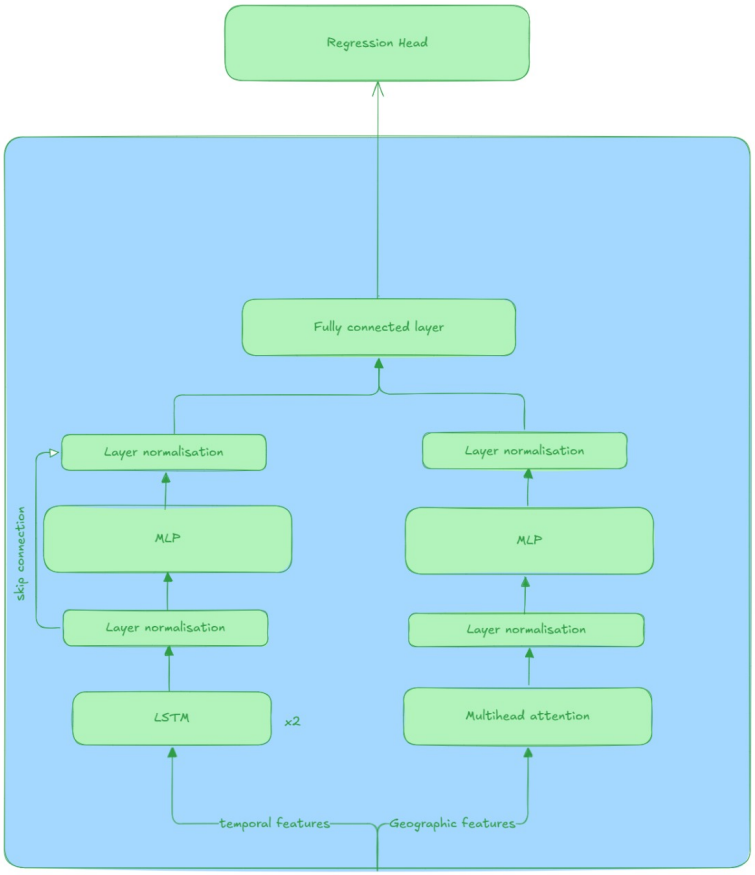

Figure 4: DL architecture.

8.7 Model Training and Evaluation

The model was trained on 70% of the available data, validated on 15%, and tested on 15%. Training employed a batch size of 64 and ran for 100 epochs with early stopping based on validation loss. The following performance metrics were achieved:

- RMSE: 3.129
- MAE: approximately 6.0
- R^2: 0.87

8.8 Advantages of the Implemented Architecture

The implemented deep learning approach offered several advantages over traditional machine learning methods:

- Better capturing of long-range temporal dependencies through the LSTM component
- Dynamic weighting of spatial factors through the attention mechanism
- Improved generalization to new locations through learned spatial representations
- End-to-end learning of complex spatiotemporal relationships without manual feature engineering

- Superior performance during anomalous events (e.g., unusual weather patterns)

The architecture's ability to adapt to different spatiotemporal contexts made it particularly effective for predicting pollution levels in areas with complex emission patterns.

9 Overall Solution

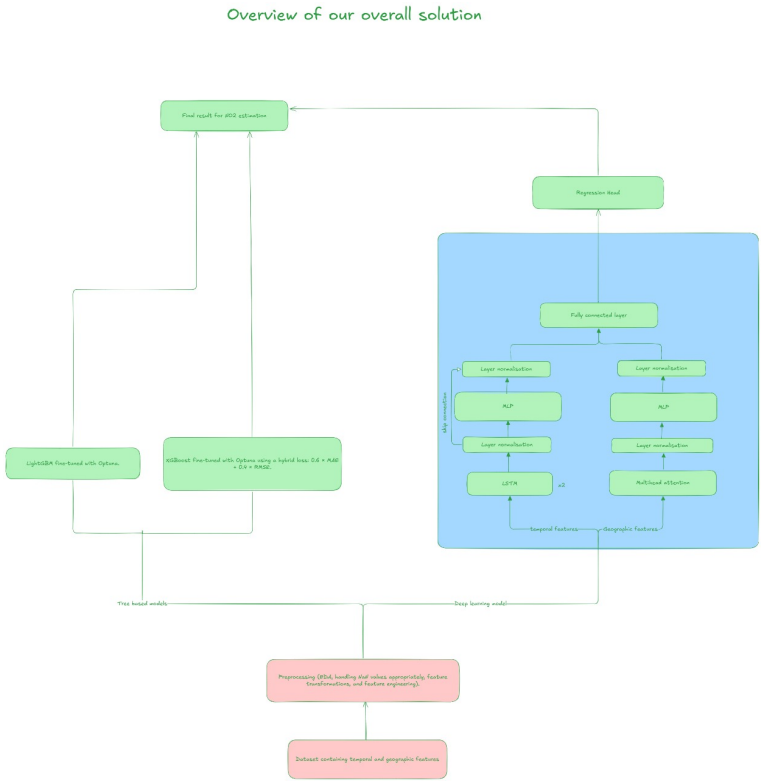

Figure: Overall Data Pipeline for NO_2 estimation

Our comprehensive solution for NO_2 prediction integrates multiple data sources, preprocessing steps, modeling approaches, and evaluation methods to create a robust prediction system witch consists of a weighetd average of the boosting models and the DL approach.

9.1 Data Integration Pipeline

Our solution begins with the integration of diverse data sources:

- Air quality monitoring data from ground stations
- Meteorological data (temperature, wind speed, humidity, pressure)
- Satellite-derived NO_2 column measurements
- Land use and land cover information
- Traffic density and patterns
- Elevation and topography
- Population density

These data sources undergo extensive preprocessing, including:

- Temporal alignment to ensure consistency across time series
- Spatial interpolation to address missing values
- Normalization and standardization for numerical stability
- Feature engineering to capture domain-specific knowledge

9.2 Modeling Approach

Our solution employs a multi-model approach:

- Tree-based ensemble models (Random Forest, Gradient Boosting) for baseline predictions
- Deep learning architecture for capturing complex spatiotemporal patterns
- Model blending to combine strengths of different approaches

The deep learning component, with its attention-based architecture, proved particularly effective at capturing the non-linear relationships between environmental factors and NO_2 concentrations.

9.3 Deployment and Inference

The deployed solution provides:

- Real-time predictions for current NO_2 levels
- Short-term forecasts (24-48 hours)
- Spatial interpolation for areas without monitoring stations
- Uncertainty estimates for all predictions

9.4 Key Achievements

- Successfully built predictive models for NO_2 concentrations with strong performance across diverse locations
- Developed feature engineering techniques that effectively capture the spatiotemporal nature of air pollution
- Created an ensemble approach that leverages the strengths of multiple modeling techniques
- Implemented a deep learning architecture that achieves RMSE of 3.129 and MAE of approximately 6.0

9.5 Limitations and Future Work

Despite the success of our approach, several limitations remain to be addressed in future work:

- Further optimization of attention mechanisms for extreme events
- Integration of additional real-time data sources (e.g., traffic sensors, industrial activity monitors)
- Extension to predict other air pollutants and creation of a multi-pollutant model
- Improved uncertainty quantification for predictions

10 Conclusion

The ability to accurately predict air pollution levels, particularly in areas without monitoring stations, has significant implications for public health and environmental management. This work contributes to the development of more accessible air quality information, potentially enabling:

- Better-informed public health advisories
- More targeted pollution control measures
- Improved understanding of pollution transport and formation mechanisms
- More comprehensive environmental impact assessments

By continuing to refine these predictive approaches, we can work toward the goal of providing accurate air quality information to all communities, regardless of monitoring infrastructure.