A Comparative Review of LSTM, CNN-LSTM, and XGBoost Models for PM 2.5 Air Quality Prediction

Suhani Pahwa

Abstract. PM 2.5, the fine particulate matter with a diameter of 2.5 micrometres or less, poses severe threats to public health and the environment due to its ability to penetrate deep into the respiratory system and bloodstream. Accurate prediction of PM 2.5 concentrations is essential for timely interventions, urban planning, and the development of smart city infrastructure. This paper provides a comparative review of three advanced models—LSTM, CNN-LSTM, and XGBoost—used for PM 2.5 forecasting. The deep learning-based models (LSTM and CNN-LSTM) excel in capturing temporal dependencies and modelling non-linear relationships. LSTM is relatively simpler but limited in capturing spatial patterns, whereas CNN-LSTM combines the strengths of both spatial and temporal modelling, offering higher accuracy at the cost of increased complexity and computational demand. XGBoost, a robust ensemble-based machine learning method, performs exceptionally well with structured data and offers interpretability and speed but lacks temporal depth. The review analyses model architectures, data preprocessing, and performance metrics, concluding that hybrid deep models show superior performance in dynamic urban environments.

Keywords: PM 2.5 prediction, Air quality forecasting, CNN-LSTM, LSTM, XGBoost, Deep learning, Machine learning, Environmental monitoring, Time series analysis, Smart cities

1. Introduction

PM 2.5 is the particulate matter suspended in air with a diameter of 2.5 micrometres or less. The reason why it is considered so dangerous is because it surpasses body's natural defences due to its small size and easily enters the lungs and then the blood stream [50]. Short term exposure causes coughing, throat irritation and shortness of breath. Long term exposure leads to serious illnesses like cardiovascular diseases, chronic respiratory diseases like COPD, lung cancer, cognitive decline and even premature death [51]. Environmentally too it leads to reduced visibility, damaged ecosystems and even contributes to climate change because of black carbon which is a component of PM 2.5 which absorbs heat. Its composition includes sulphates, nitrates, ammonia, black carbon, heavy metals and other organic compounds. Common sources include vehicle emissions, industrial pollution and biomass burning.

Real time and short-term predictions for PM 2.5 help issue health advisories for people to stay indoors or use a mask while long term trends can influence urban planning decisions. In smart cities, PM 2.5 prediction can be integrated with **IoT-based air quality sensors**, traffic systems, and weather data. Cities like Beijing, Seoul and Delhi are involving AI powered pollution forecasting models in their urban planning.

Machine Learning and Deep Learning models have emerged to be a lot better than traditional statistical and physical models to predict PM 2.5 in air. They capture non-linear patterns, have a high prediction accuracy, work with big datasets and give real time predictions. This paper will compare and individually evaluate three models for prediction, namely CNN-LSTM, LSTM and XGBoost.

DL (Deep Learning) models like LSTM AND CNN-LSTM are great at temporal forecasting making them suitable for both short term and long-term predictions. ML (Machine Learning) models like XGBoost not only predict with high accuracy but also can identify most important features influencing the concentration of PM 2.5, which helps the government take necessary steps to reduce pollution.

2. Review of Models:

2.1 CNN-LSTM model

This model leverages the strength of both CNN (Convolutional Neural Networks) and LSTM (Long Short-Term Memory) neural networks to predict PM 2.5 in air [1]. This approach is better because this pollutant shows strong temporal dependencies and complex spatial features which are both taken into account to make better predictions. The formation of PM 2.5 is very complex and is caused by non-linear characteristics in time and space [2]. The data belongs to time series data [3]. Time series prediction has become a hot topic [4]. There are many mature time series prediction methods, including ARMA [5], ARIMA [5], SARIMA [6], SVR [7], BP neural network [8], Bayesian network [9] and so on. But as the data becomes more in amount and more complex, these methods fall short as they take a lot of training time. Deep learning approaches, including CNN and RNN/LSTM, have emerged as promising solutions for analysing massive environmental data and identifying complex correlations.

- **Convolutional Neural Network (CNN)**
  - Functionality: CNNs are highly effective at **feature extraction** [10]. They can automatically learn and detect specific features in the input data through convolutional layers [11]. For PM2.5 prediction, CNNs are used to extract **spatial features** from input data, such as the spatial effects among monitoring stations or correlations between air quality and meteorological conditions in neighbouring cities [12].
  - Architecture: CNN employs a unique feature called weight sharing. This simplifies training by reducing the number of weights [12]. 2D CNNs are used for image recognition [10] and 3D CNNs are used for medical images [13]. **1D CNNs are particularly well-suited for sequence data processing and time series analysis** [14]**.** For PM 2.5, 1D CNNs are adopted.

    After the input is fed, convolution layers extract features by applying filters or kernels. Then the ReLU activation function is used to introduce non-linearity. Finally, the max pooling layers reduce dimensionality while retaining the important features.
  - Process:

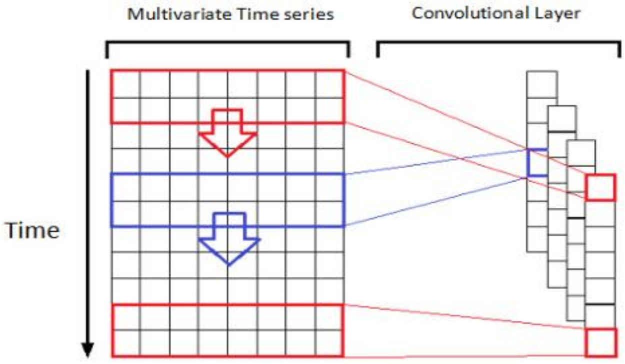


Fig.1. CNN for feature extraction

The left of Figure 1 is the input time series data which is a multi-dimensional matrix, which is convoluted from top to bottom as shown by the arrow in Figure 1, and the red represents a filter. The number of the extracted feature dimensions is N*1 after convolution with a filter, where N is related to the number of input data dimensions, the size of filter and convolution step length. The blue indicates another filter, which can be followed by other filters. Suppose the number of filters is M, and the extracted feature dimension will be N* M. [1]

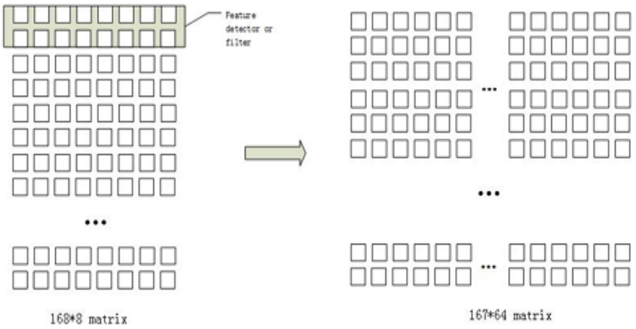


Fig.2 Describing 1-D convolution

Figure 2 shows the process of a one-dimensional convolution. Suppose the input is a 168*8 matrix, the output will be a 167*64 matrix after convolution with 64 filters of size 2. [1]

- **LSTM Network**
  - Functionality: LSTM [15] is a specialized type of RNN and it is better than RNN at predicting time series data as there is no issue of the exploding/vanishing gradient.
  - Architecture: LSTM has a unique structure, which includes memory cells within the hidden layer and controllable gates: a forget gate, an input gate and an output gate [16]. These gates regulate the flow of information, determining how much of the previous cell's "memory" is retained, how much current input is saved, and how much information is outputted.

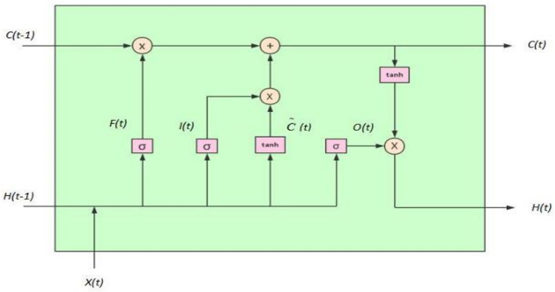


Fig.3 LSTM

In Figure 3, σ is the sigmoid function shown in equation (7), whose output is a value between 0 and 1. Here, 0 means let nothing pass while 1 means let everything pass. Then the hyperbolic tangent function illustrated in Equation (8), is used to overcome the problem of gradient disappearance. The input and output of the network structure of the LSTM in Figure 6 can be described as Eqns. (1) – (8):

1. $F(t) = \sigma\left(W_f \cdot \left(H_{(t-1)}, X_t\right) + b_f\right)$

2. $I(t) = \sigma\left(W_i \cdot \left(H_{(t-1)}, X_t\right) + b_i\right)$

3. $\hat{C}(t) = tanh\left(W_c \cdot \left(H_{(t-1)}, X_t\right) + b_c\right)$

$$4.\ C(t) = F(t) \times C_{(t-1)} + I(t) \times \hat{C}(t)$$

$$5.\ O(t) = \sigma\left(W_o \cdot \left(H_{(t-1)}, X_t\right) + b_o\right)$$

$$6.\ H(t) = O(t) \times tanh\left(C(t)\right)$$

$$7.\ sigmoid(x) = 1/\left(1 + e^{(-x)}\right)$$

$$8.\ tanh(x) = \left(e^{x} - e^{(-x)}\right)/\left(e^{x} + e^{(-x)}\right)$$

where W_f, W_i, W_c and W_o are input weights, b_f, b_i, b_c and b_o are bias weights, t represents the current time state, and t-1 is the previous time state, X represents input; H represents output and C is the cell status; F(t) is the forget gate, I(t) is the input gate, $\hat{C}(t)$ is the candidate cell state, C(t) is the cell state update, O(t) is the output gate, and H(t) is the hidden state. [1]

- **A Hybrid Model**

Let us now discuss the structure of an example of the hybrid model.

The inputs to this model are the PM 2.5 concentrations, wind speed and hours of rain over the last 24 hours. The output will be the predicted PM 2.5 concentration for the next hour. The first half of the model is CNN used for feature extraction and the second part is LSTM forecasting which uses the feature selection done by CNN. The CNN part contains 3 1-D convolutional layers. To increase efficiency, batch normalisation is done after the second and third layers.

Scaled Exponential Linear Units (SELU) is used instead of Rectified Linear Unit (ReLU). This is because, compared to ReLU, SELU has better convergence and can effectively avoid the problem of gradient vanishing [17]. The output of LSTM goes through the fully-connected architecture and the sigmoid activation function to produce the final output.[18]

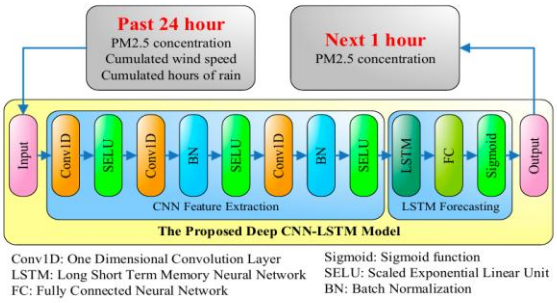


Fig.4 The proposed model

- **Methodology**
  - The **data source** is hourly PM 2.5 concentration and meteorological data from Beijing [18] and Shanghai [12].
  - **Data preprocessing:**
    Handling Missing Values: Missing values in the datasets are filled, typically with zeros.
    Feature Encoding: Wind direction is often encoded and converted into digital values.
    Normalization: Values of features are normalized to fall within a specific range, commonly 0-1, using methods like Min-Max normalization, to improve prediction accuracy and prevent bias towards certain dimensions.
    Data Splitting: Datasets are usually divided into training and test sets, for example, the first 80% for training and the remaining 20% for testing.

- **Overfitting Prevention**: Common techniques to avoid overfitting in deep neural networks are applied, including Dropout [1], regularization (specifically Elastic Net combining L1 and L2 regularization) [12] and early stopping [18].
- **Performance metrics**:

  The **multivariate CNN-LSTM model performed best** with the lowest average MAE of **13.9697** and RMSE of **17.9306** over ten randomly selected samples.

  Multivariate models showed significantly lower MAE values than univariate models, and CNN-LSTM models generally had lower MAE and RMSE values compared to single LSTM models. [1]

  The model achieved the **lowest average MAE (14.63446)** and **RMSE (24.22874)**

  It also yielded the **highest average Pearson correlation coefficient (0.959986)** and **Index of Agreement (0.97831).**

  It outperformed SVM, RF, DT, MLP, CNN (alone), and LSTM (alone), confirming that combining CNN for feature extraction with LSTM for time-series analysis is highly effective for PM2.5 forecasting. [18]

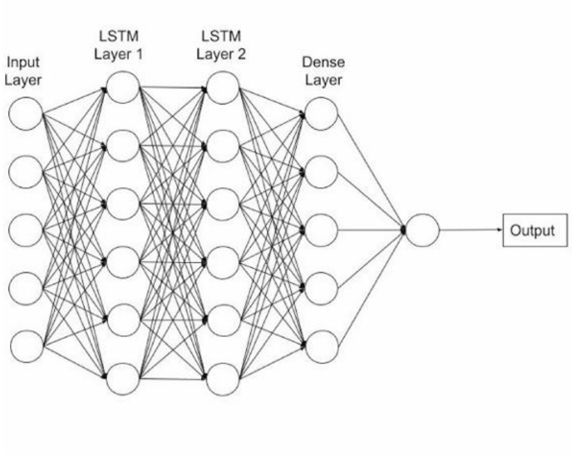
2.2 LSTM and its Variations

Long Short-Term Memory:

LSTM was introduced by Hochreiter and Schmidhuber [15] to overcome the shortcomings of traditional RNN. LSTM utilizes the concepts of gates (forget, input, and output gate) that enable the network to learn necessary information from the present time step and either forget or retain the information from past time steps. The memory of present and past data is stored and passed through the hidden layer activation so that the future activation has decencies of the past data and its sequence.

The proposed network: [19]

A many-to-one RNN structure is used to predict future PM 2.5 concentrations by using the past concentrations and other important meteorological factors. A combination of LSTM and the connected dense layer is expected to be very effective in learning and predicting highly non-linear sequences. The deep LSTM neural network employed in the study consists of the input layer, LSTM-1 layer, LSTM-2 layer, a dense layer, and a lambda layer. The LSTM-1 layer consists of 30 units and utilizes ReLU activation [22] to return a sequence while the LSTM-2 layer consists of 15 units and utilizes ReLU activation without returning a sequence. The dense layer consists of n units and utilizes a sigmoid activation function [23]. Lambda layer scales the final dense layer values with a factor of 200 and assists in maintaining the values from the dense layer below 1. This complies with the output from sigmoid activation which is in the range of 0 to 1.



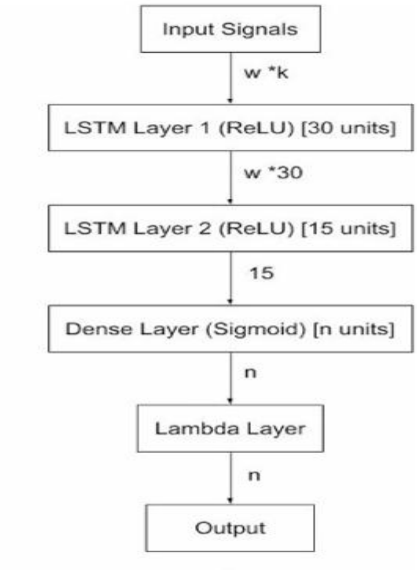

Input Signals
w *k
LSTM Layer 1 (ReLU) [30 units]
w *30
LSTM Layer 2 (ReLU) [15 units]
15
Dense Layer (Sigmoid) [n units]
n
Lambda Layer
n
Output

Fig.5 Showing the basic model structure

Fig.6

In this figure, k is the number of input features, n is the number of steps of future predictions, and w is the window size (in days)

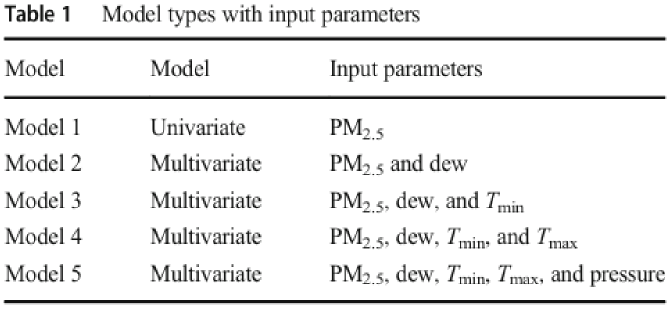
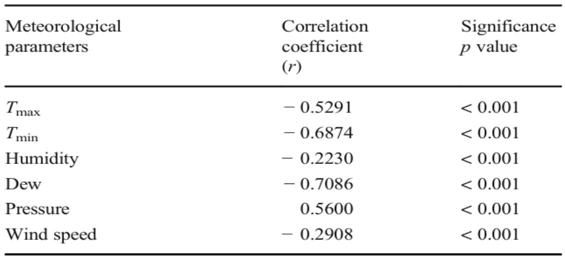


Table 1 Model types with input parameters

Model	Model	Input parameters
Model 1	Univariate	$PM_{2.5}$
Model 2	Multivariate	$PM_{2.5}$ and dew
Model 3	Multivariate	$PM_{2.5}$, dew, and T_{min}
Model 4	Multivariate	$PM_{2.5}$, dew, T_{min}, and T_{max}
Model 5	Multivariate	$PM_{2.5}$, dew, T_{min}, T_{max}, and pressure

Multivariate models perform better to predict PM 2.5 than univariate models [20]. Correlation analysis can be done to select the dominant meteorological factors [21]. Following is an example from the correlation analysis done for the year 2019 in Kathmandu Valley, Nepal. [19]



Meteorological parameters	Correlation coefficient (r)	Significance p value
T_{max}	− 0.5291	< 0.001
T_{min}	− 0.6874	< 0.001
Humidity	− 0.2230	< 0.001
Dew	− 0.7086	< 0.001
Pressure	0.5600	< 0.001
Wind speed	− 0.2908	< 0.001

Table 2 Correlation analysis

From this table we can also deduce the temporal variations.

Performance and Output:

Five models were developed with different input combinations [Table 1]. **Model 2, which used past PM2.5 data and dew as inputs with single-step prediction, was identified as the best performing deep LSTM model**, achieving a Root Mean Square Error (RMSE) of 13.04 µg/m3 and a Mean Absolute Error

(MAE) of 10.81 µg/m37. The deep LSTM model significantly outperformed the Seasonal AutoRegressive Integrated Moving Average (SARIMA) model, which yielded an RMSE of 19.54 µg/m3 and MAE of 15.21 µg/m3 for the test data. Correlation analysis revealed that dew, Tmin, Tmax, and pressure were strongly correlated with PM2.5 concentration [Table 2]. [19]

The LSTM model achieved an accuracy of 72–79% across 19 districts in Korea [24]. The RMSE was 5–9 µg/m3, about half of the CMAQ (Community Multiscale Air Quality) forecasts. The Area Under the ROC Curve (AUC) was 0.87–0.93, indicating good forecast skill. LSTM showed lower False Alarm Rates (FAR) in more polluted regions and seasons. Inputs included observed air quality variables (PM10, PM2.5, O3, SO2, NO2, CO) and meteorological variables (pressure, temperature, dew point temperature, relative humidity, horizontal wind). It also utilized CMAQ air quality variables, WRF meteorological variables at multiple atmospheric levels, cosine similarity (spatial similarity to meteorological fields during high PM2.5 events) [25], and back-trajectory cluster values from the FLEXPART model [26]. Data preprocessing involved regional and seasonal grouping (12 groups of three consecutive months), bagging ensembles (40 learners per regional/seasonal model, totalling 2880 models), and normalization of datasets (0-1 range). However, the study notes that AI models have inherent limitations, such as being "black boxes" where decision-making processes are not interpreted, and their high dependency on numerical model inputs. Therefore, they are best used as *additional references* for human forecasters.

Bidirectional LSTM:

Unlike traditional unidirectional LSTMs, Bi-LSTMs [28] [30] can use both past and future information leading to better predictions. In a traditional unidirectional LSTM, each time step of the hidden layer can only access information from the past, limiting its ability to capture long-term dependencies in the sequence. To address this limitation, the bidirectional LSTM introduces an additional inverse layer that considers both past and future information.

Attention mechanism:

Attention mechanisms [29] are used to enhance the expressive power of neural network models. This mechanism enables the model to focus on the most relevant parts of the input by dynamically assigning different weights to different positions in the input sequence. A common implementation of the attention mechanism includes dot product attention or weighted average attention. Dot product attention calculates attention weights based on the similarity between inputs and the query, while weighted average attention performs a weighted sum based on the importance of the inputs. Including an attention mechanism in the model makes us focus on the most relevant inputs, improve performance over long sequences, increase interpretability and boosting accuracy.

ATT-Bi-LSTM:

This model combines LSTM with attention mechanism. It has two independent LSTM layers, one which processes the input sequence from front to back and the other from back to front in the opposite direction. Subsequently, with the introduction of the attention mechanism, it aggregates the information in the input sequence by computing a vector of weights indicating the importance of different input positions to the current hidden state and then using these weights to weigh the information in the input sequence. After feature selection and weight aggregation, the selected data is again fed to Bi-LSTM. Finally, a dropout layer is used to prevent overfitting and a fully connected layer is used for the output. [27]

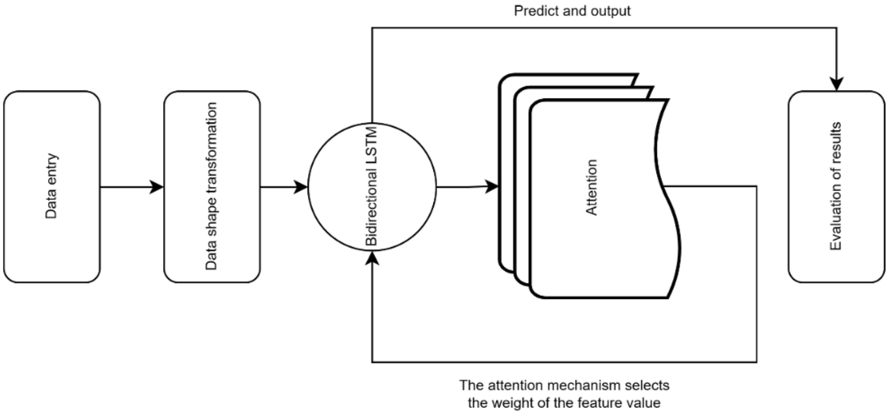

Predict and output
Data entry
Data shape transformation
Bidirectional LSTM
Attention
Evaluation of results
The attention mechanism selects the weight of the feature value

Fig.7 Depicting a flowchart of the model

Performance and Output:

The proposed ATT-Bi-LSTM model demonstrated superior performance compared to ATT-LSTM, BI-LSTM, and BI-ATT-LSTM models when evaluated using MSE, RMSE, MAE, and R2. The model showed consistent stability and low errors, even in unstable air quality conditions like Pingxiang, while maintaining a high level of fit (R2). Input data was the hourly air quality data from eleven cities in Jiangxi Province, China, covering February 2, 2017, to December 22, 2018. A time-sliding window of 24 hours was used, inputting PM2.5, PM10, SO2, NO2, CO, and O3 as eigenvalues, with PM2.5 as the target. Missing values were removed, and data was normalized to between (-1, 1). [27]

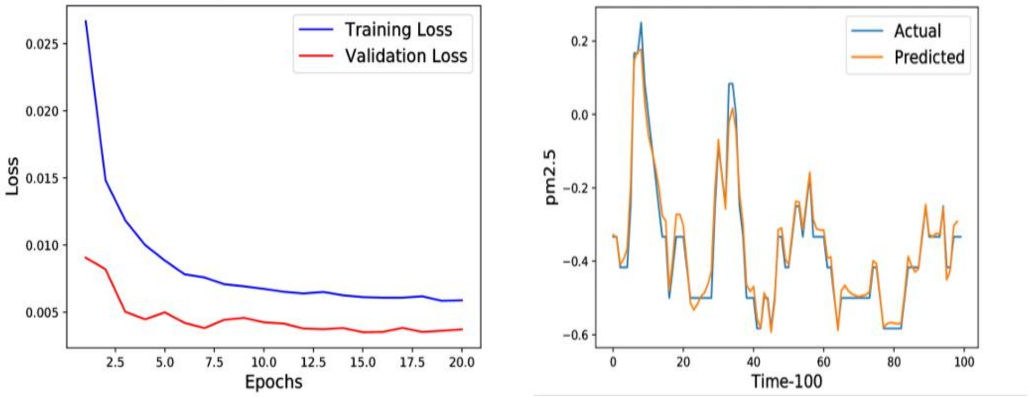


Fig.8 Depicting the LOSS function and the model prediction

2.3 XGBoost

The XGBoost model is a powerful yet simple machine learning technique that has been successfully applied to PM2.5 concentration prediction to improve air quality forecasting, particularly in urban areas such as Shanghai [31], Tehran [32], Tianjin [33], and Macau [34]. This paper focuses on these 4 studies.

Data Collecting and Preprocessing:

These studies involve collecting diverse data in different ways and then processing them.

- **Study locations and periods:**
  Shanghai, China: Data from January 1, 2015, to December 31, 2018, was used. The study focused on Shanghai, a mega city in eastern China.

Tehran, Iran: Data spanned from January 1, 2015, to the end of 2018. Tehran is the capital city of Iran, located at the southern slope of Alborz Mountain.

Tianjin, China: Hourly PM2.5 concentration prediction was based on data from nineteen air-monitoring stations in Tianjin from December 1, 2016, to December 30, 2016.

Macau: Daily measurements from 2016 to 2021 were obtained to forecast PM2.5, PM10, and CO concentrations.

- **Data sources and types:**
    - **Air Pollutant Concentrations**: Ground-measured hourly PM2.5 mass concentrations are primary inputs. Other pollutants like PM10, SO2, NO2, CO, and O3 are also collected. Sources include the National Urban Air Quality Real-Time Release Platform (China), Tehran's Municipality ICT website and Department of Environment's Air Pollution Monitoring System platform (Iran), and Macau Meteorological and Geophysical Bureau (Macau).

    - **Meteorological Conditions**: Hourly meteorological measurements are vital. Parameters include atmospheric pressure (P), air temperature (T, Tmax, Tmin), relative humidity (RH), precipitation (Prs/Rainfall), wind direction, wind speed, visibility, and dew point. Sources include meteorological bureaus (China), Iran Meteorological Organization, and Hong Kong Observatory (for Macau).

    - **Atmospheric Chemical-Transport Model Outputs**: Forecasts from numerical prediction systems like **WRF-Chem (Weather Research and Forecasting-Chemistry model)** are integrated. WRF-Chem provides hourly PM2.5 mass concentration forecasts, as well as high-altitude weather forecast data at various pressure layers (e.g., 500 hPa, 700 hPa, 850 hPa, 925 hPa, 1000 hPa).

    - **Satellite Remote Sensing Data**: **Aerosol Optical Depth (AOD)** from MODIS instruments (Aqua satellite) at 3 km (AOD03) and 10 km (AOD10) spatial resolutions are used in some studies [32]. AOD products are obtained from the NASA Atmosphere Archive & Distribution System (LAADS) archive.

    - **Geographical and Auxiliary Data**: Latitude, longitude, altitude of monitoring stations, day of year, day of week, and season are incorporated.

- **Data preprocessing and feature selection:**
    - **Matching and Aggregation**: Air quality and meteorological data, often from different locations, are matched to the nearest observatories. AOD values are sampled for APM site locations. Daily or hourly averages are derived, and all data is merged based on sampling date.

    - **Handling Missing Values**: Missing values are a common problem. Interpolation may be used for some climatic data. However, high rates of missing values, particularly for PM2.5 (around 54.11% in Tehran study) and AOD (e.g., AOD03 at 94.09% in Tehran study), lead to exclusion of records during training. [35]

    - **Feature engineering:**

- **Lag Data**: Historical pollutant observations (e.g., PM2.5_lag1, PM2.5_lag2 for previous day concentrations) and rainfall with lag (Rainfall_lag1, Rainfall_lag2) are crucial features.
- **Derived Factors**: Differences between different layers of meteorological factors (e.g., Tc_700-Tc_500, Td_700-Td_500) are used to represent vertical variations and atmospheric stability.
- **Temporal Features**: Day of year, day of week, and season are added to the dataset.

- **Normalization**: Features are standardized by deducting the mean and scaling with the variance (e.g., to a range of -1 to 1), especially for deep learning methods, to ensure stability during training.

- **Dataset Splitting**: Datasets are typically shuffled and split into training (e.g., 70%) and testing/validation (e.g., 30%) sets.

- **Feature Selection**: To reduce overfitting and improve model generalization, techniques like **Lasso regression [36] [37]** are used to select the most important factors. In the Shanghai study, Lasso regression retained 36 key factors. Other methods include **SHAP (Shapley Additive Explanations) values** [38] and **recursive feature elimination** to identify and remove less important features, optimizing model performance and cost.

XGBoost Model

The Extreme Gradient Boosting algorithm is based on Gradient Boosting Decision Trees (GBDT).

Core Principle: XGBoost is an **ensemble learning model** composed of multiple decision trees. It builds trees sequentially, where each new tree is added to **complement the already built ones** by **fitting and learning the residuals** (errors) of the previous tree's predictions. The final prediction is the sum of the effects of all regression trees.[39]

Objective Function: XGBoost minimizes a **regularized objective function**. This function includes:

- A **loss function**: Measures the total prediction error between observed and predicted values.

- A **regularization term (Ω)**: This penalty term is added to prevent **overfitting** by controlling the complexity of the model. It penalizes models that are too complex based on the number of leaves and the magnitude of leaf weights in each tree.

Key enhancements and optimisations:

◦**Regularization**: It employs both **LASSO (L1)** and **Ridge (L2) regularization** to penalize more complex models and explicitly prevent overfitting.

◦**Parallelization**: XGBoost is designed for efficient use of hardware. It approaches sequential tree building with a **parallelized implementation**, interchanging loop order by employing global scans and sorting with parallel threads, which significantly increases runtime.

◦**Tree Pruning**: Unlike other methods, XGBoost uses a "max depth" parameter and prunes trees backward (a "depth-first" approach), which improves computational performance.

◦**Hardware Optimization**: Achieved through **caching awareness** (allocating internal buffers in each thread for gradient statistics) and "out-of-core" computing for handling large datasets that don't fit into memory.

◦**Sparsity Awareness**: It can naturally handle **sparse features** in input data by automatically "learning" the best missing value based on training loss.

◦**Weighted Quantile Sketch**: Uses a distributed weighted quantile sketch algorithm to efficiently find optimal split points in weighted datasets.
◦**Built-in Cross-validation**: The algorithm includes a cross-validation method at each iteration, removing the need for explicit programming of this search and for determining the exact number of boosting iterations. [40] [41]
Hyperparameter Tuning: Effective PM2.5 prediction with XGBoost requires **tuning hyperparameters**. Common parameters include learning rate, sample subsampling rate, characteristic subsampling rate, maximum tree depth, and L1/L2 Regular Coefficients. Grid search with k-fold cross-validation is often used to find optimal values for parameters.

Results:
•**Shanghai Case Study**:
◦The **modified XGBoost model** (which combined XGBoost with Lasso linear regression for feature selection) provided **significantly better daily forecasting of PM2.5 than the WRF-Chem model**.
◦It achieved **correlation coefficients (R) that were higher by 50–100%** and **standard deviations that were lower by 14–24 µg m–3** compared to WRF-Chem.
◦The model also showed improved consistency with observations, **better reflecting variations over time**, and **avoiding false peaks and valleys** often seen in WRF-Chem predictions.
◦For the full concentration range, the R values were 0.51 (WRF-Chem), 0.73 (Lasso), and **0.77 (modified XGBoost)**. The RMSE of the modified XGBoost model was 26.1 µg m–3, about **41% lower than WRF-Chem**.
◦It performed especially well at **high PM2.5 concentrations** (exceeding 50 µg m–3 or 75 µg m–3), where its predictive correction ability was stronger.
◦The modified XGBoost model also showed a **correcting effect on monthly WRF-Chem forecasts across all seasons**, notably during heavy winter pollution.
◦**Error Analysis**: The prediction error (residual) of the modified XGBoost model for PM2.5 showed a **strong negative correlation with actual PM2.5 values** (R = –0.65), indicating it tended to overestimate low concentrations and underestimate high ones, but its RMSE was consistently lower than other models.

•**Tehran Case Study**:
◦XGBoost demonstrated the **highest model performance** with **R2 = 0.8** (R = 0.894), **MAE = 10.0 µg/m3**, and **RMSE = 13.62 µg/m3** when AOD03 data was excluded.
◦It also boasted a **very low time cost of 19 seconds**.
◦While all three tested ML methods (RF, XGBoost, Deep Learning) performed similarly in terms of R2 (0.77 to 0.81 when AOD03 was excluded), XGBoost showed the best overall performance.
◦**Feature Importance**: Historical PM2.5 observations (PM2.5_lag1) and visibility consistently ranked as significantly important features in both RF and XGBoost models. Other important features include wind speed, day of the year, altitude, and temperature. Interestingly, AODs (especially AOD03) did not improve model performance due to high missing values, suggesting other features substituted their influence.

•**Tianjin Case Study**:
◦The XGBoost algorithm **outperformed Random Forest, Multiple Linear Regression, Decision Tree Regression, and Support Vector Machines for regression** models in hourly PM2.5 concentration prediction.
◦Its evaluation metrics were **RMSE = 17.298, MAE = 11.774, and R2 = 0.9520**, which were all better than the other four models.
◦The correlation analysis showed PM2.5 had a high correlation with PM10 (R2=0.918), CO (R2=0.457), and O3 (R2=0.536), but a very low correlation with SO2 (R2=0.026), indicating that SO2 concentration did not contribute significantly to PM2.5 prediction.

•**Macau Case Study**:

◦XGBoost, along with ANN, RF, SVM, and MLR, successfully forecasted 24-h and 48-h pollutant concentrations in Macau.
◦For 24-hour PM2.5 prediction in 2020, XGBoost achieved an R2 of 0.83, RMSE of 4.42, MAE of 3.51, and BIAS of 2.41104. While RF and SVM models often showed slightly better R2 for PM2.5 and PM10 in Macau (e.g., RF with R2 of 0.88 for PM2.5 in 2020), XGBoost remained a strong performer.
◦For 48-hour PM2.5 prediction, XGBoost had an R2 of 0.43105, indicating a drop in performance compared to 24-hour predictions, which was expected for longer forecast horizons. Feature selection based on SHAP values was found to be critical for enhancing accuracy, especially for 48-hour forecasts.
◦Similar to other studies, lagged pollutant concentrations (e.g., PM25_16D1, PM25_23D1) were identified as the **most significant features** influencing predictions. Meteorological features like temperature, wind direction, dew point, and geopotential height also contributed to the models.

3. Challenges and Future Directions

Challenges:

Data Quality and Availability: the reliance on satellite data or low-cost sensors for data collection often leads to inconsistent or missing data, which negatively impacts the accuracy of models.

Model Interpretability: explaining policy predictions is difficult owing to the nature of deep learning models, as they are viewed as “black boxes”.

Generalization Across Regions: Environmental and meteorological differences mean that models trained in one geographic region face difficulties being applied to another, impacting accuracy.

Computational Complexity: the training duration alongside the need for high computational resources makes deep models like CNN-LSTM more demanding.

Temporal and Spatial Resolution: capturing and predicting patterns in spatial and temporal terms is difficult within real-time contexts.

Future Directions:

Integration with IoT and Edge Devices: in the frameworks of smart cities, real-time sensors can be leveraged alongside models to enable air quality estimation in a more continuous and localized setting. [42]

Explainable AI (XAI): decision-making based on deep models can be improved through designing explainable modules, which add clarity for interpreting model outcomes. [43]

Transfer Learning: adapting pre-trained models to learn different cities and seasons can be achieved through transfer learning approaches. [44] [45]

Multi-Modal Data Fusion: prediction can be enhanced through the integration of satellite imagery and traffic, meteorological, and social data. [46] [47]

Hybrid and Attention-Based Architectures: Exploring advanced architectures such as transformer models [48] or attention-based CNN-LSTM hybrids for enhanced performance [49].

4. Conclusion

Predicting PM 2.5 concentration accurately is a critical component in managing air pollution and safeguarding public health. This review demonstrates how LSTM, CNN-LSTM, and XGBoost models have advanced air quality forecasting beyond traditional methods.

- **LSTM** models are strong in handling time series and long-term dependencies but may fall short in capturing spatial complexities.
- **CNN-LSTM** models offer the best of both worlds by combining spatial feature extraction and temporal forecasting, achieving superior accuracy, especially with multivariate data; however, they require significant computational resources and expertise to implement.
- **XGBoost**, on the other hand, is fast, interpretable, and efficient for structured data, making it ideal for practical deployments, though it may not model sequential dependencies as effectively as deep learning models.

Each model presents unique advantages and trade-offs, and the choice depends on the specific requirements—such as interpretability, accuracy, data complexity, or real-time application. Future efforts should focus on developing more interpretable, transferable, and computationally efficient hybrid models that leverage the strengths of each approach. With continued innovation, AI-driven forecasting systems can become indispensable tools in building healthier and smarter cities.

References:

1) Li, T., Hua, M., & Wu, X. U. (2020). A hybrid CNN-LSTM model for forecasting particulate matter (PM2. 5). *Ieee Access*, *8*, 26933-26940.

2) Wang, J. L., Zhang, Y. H., Shao, M., Liu, X. L., Zeng, L. M., Cheng, C. L., & Xu, X. F. (2004). Chemical composition and quantitative relationship between meteorological condition and fine particles in Beijing. *Journal of Environmental Sciences (China)*, *16*(5), 860-864.

3) Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., ... & Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, *454*(1971), 903-995.

4) Rani, S. (2014, September). Review on time series databases and recent research trends in Time Series Mining. In *2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)* (pp. 109-115). IEEE.

5) Jenkins, G. M., & Box, G. E. (1976). Time series analysis: forecasting and control. *(No Title)*.

6) Williams, B. M., Durvasula, P. K., & Brown, D. E. (1998). Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record*, *1644*(1), 132-141.

7) Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*, 199-222.

8) Haipeng, C., Ping, S., & Shengguo, H. (2008). Study of aircraft hard landing diagnosis based on neural network. *Computer Measurement & Control*, *16*(7), 906-908.

9) Sun, S., Zhang, C., & Yu, G. (2006). A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, *7*(1), 124-132.

10) LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (2002). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

11) Chen, Y. (2015). *Convolutional neural network for sentence classification* (Master's thesis, University of Waterloo).

12) Qin, D., Yu, J., Zou, G., Yong, R., Zhao, Q., & Zhang, B. (2019). A novel combined prediction scheme based on CNN and LSTM for urban PM 2.5 concentration. *Ieee Access*, *7*, 20050-20059.

13) Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, *35*(5), 1285-1298.

14) Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of sound and vibration*, *388*, 154-170.

15) Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

16) Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, *12*(10), 2451-2471.

17) Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in neural information processing systems*, *30*.

18) Huang, C. J., & Kuo, P. H. (2018). A deep CNN-LSTM model for particulate matter (PM2. 5) forecasting in smart cities. *Sensors*, *18*(7), 2220.

19) Dhakal, S., Gautam, Y., & Bhattarai, A. (2021). Exploring a deep LSTM neural network to forecast daily PM 2.5 concentration using meteorological parameters in Kathmandu Valley, Nepal. *Air Quality, Atmosphere & Health*, *14*, 83-96.

20) Muller, K. E., & Stewart, P. W. (2006). *Linear model theory: univariate, multivariate, and mixed models*. John Wiley & Sons.

21) Pandey, R., Dhoundiyal, M., & Kumar, A. (2015, April). Correlation analysis of big data to support machine learning. In *2015 Fifth International Conference on Communication Systems and Network Technologies* (pp. 996-999). IEEE.

22) Eckle, K., & Schmidt-Hieber, J. (2019). A comparison of deep networks with ReLU activation function and linear spline-type methods. *Neural Networks*, *110*, 232-242.

23) Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, *6*(12), 310-316.

24) Ho, C. H., Park, I., Kim, J., & Lee, J. B. (2023). PM2. 5 forecast in korea using the long short-term memory (LSTM) model. *Asia-Pacific Journal of Atmospheric Sciences*, *59*(5), 563-576.

25) Hur, S. K., Oh, H. R., Ho, C. H., Kim, J., Song, C. K., Chang, L. S., & Lee, J. B. (2016). Evaluating the predictability of PM10 grades in Seoul, Korea using a neural network model based on synoptic patterns. *Environmental Pollution*, *218*, 1324-1333.

26) Stohl, A., Forster, C., Frank, A., Seibert, P., & Wotawa, G. (2005). The Lagrangian particle dispersion model FLEXPART version 6.2. *Atmospheric Chemistry and Physics*, *5*(9), 2461-2474.

27) Huang, X., Hong, X., Liu, Z., Zhang, Q., Huang, Q., & Liu, Z. (2024, September). PM2. 5 prediction based on attention mechanism and Bi-LSTM. In *Fifth International Conference on Green Energy, Environment, and Sustainable Development (GEESD 2024)* (Vol. 13279, pp. 845-852). SPIE.

28) Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019, December). The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International conference on big data (Big Data)* (pp. 3285-3292). IEEE.

29) Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, *452*, 48-62.

30) Prihatno, A. T., Nurcahyanto, H., Ahmed, M. F., Rahman, M. H., Alam, M. M., & Jang, Y. M. (2021). Forecasting PM2. 5 concentration using a single-dense layer BiLSTM method. *Electronics*, *10*(15), 1808.

31) Ma, J., Yu, Z., Qu, Y., Xu, J., & Cao, Y. (2020). Application of the XGBoost machine learning method in PM2. 5 prediction: A case study of Shanghai. *Aerosol and Air Quality Research*, *20*(1), 128-138.

32) Zamani Joharestani, M., Cao, C., Ni, X., Bashir, B., & Talebiesfandarani, S. (2019). PM2. 5 prediction based on random forest, XGBoost, and deep learning using multisource remote sensing data. *Atmosphere*, *10*(7), 373.

33) Pan, B. (2018, February). Application of XGBoost algorithm in hourly PM2. 5 concentration prediction. In *IOP conference series: earth and environmental science* (Vol. 113, p. 012127). IOP publishing.

34) Lei, T. M., Ng, S. C., & Siu, S. W. (2023). Application of ANN, XGBoost, and other ml methods to forecast air quality in Macau. *Sustainability*, *15*(6), 5341.

35) Nabavi, S. O., Haimberger, L., & Abbasi, E. (2019). Assessing PM2. 5 concentrations in Tehran, Iran, from space using MAIAC, deep blue, and dark target AOD and machine learning algorithms. *Atmospheric Pollution Research*, *10*(3), 889-903.

36) Ranstam, J., & Cook, J. A. (2018). LASSO regression. *Journal of British Surgery*, *105*(10), 1348-1348.

37) Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *58*(1), 267-288.

38) García, M. V., & Aznarte, J. L. (2020). Shapley additive explanations for NO2 forecasting. *Ecological Informatics*, *56*, 101039.

39) Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., ... & Zhou, T. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, *1*(4), 1-4.

40) Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

41) Nielsen, D. (2016). *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?* (Master's thesis, NTNU).

42) Loh, B. G., & Choi, G. H. (2017). Development of IoT-based PM2. 5 measuring device. *Journal of the Korean Society of Safety*, *32*(1), 21-26.

43) Chakraborty, S., Misra, B., & Dey, N. (2024). Explainable Artificial Intelligence (XAI) for Air Quality Assessment. In *Design Studies and Intelligence Engineering* (pp. 333-341). IOS Press.

44) Ni, J., Chen, Y., Gu, Y., Fang, X., & Shi, P. (2022). An improved hybrid transfer learning-based deep learning model for PM2. 5 concentration prediction. *Applied Sciences*, *12*(7), 3597.

45) Gupta, S., Park, Y., Bi, J., Gupta, S., Züfle, A., Wildani, A., & Liu, Y. (2024, August). Spatial Transfer Learning for Estimating PM 2.5 in Data-Poor Regions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 385-400). Cham: Springer Nature Switzerland.

46) Li, K., Bai, K., Li, Z., Guo, J., & Chang, N. B. (2022). Synergistic data fusion of multimodal AOD and air quality data for near real-time full coverage air pollution assessment. *Journal of environmental management*, *302*, 114121.

47) Kalajdjieski, J., Zdravevski, E., Corizzo, R., Lameski, P., Kalajdziski, S., Pires, I. M., ... & Trajkovik, V. (2020). Air pollution prediction with multi-modal data and deep neural networks. *Remote Sensing*, *12*(24), 4142.

48) Yu, M., Masrur, A., & Blaszczak-Boxe, C. (2023). Predicting hourly PM2. 5 concentrations in wildfire-prone areas using a SpatioTemporal Transformer model. *Science of The Total Environment*, *860*, 160446.

49) Li, S., Xie, G., Ren, J., Guo, L., Yang, Y., & Xu, X. (2020). Urban PM2. 5 concentration prediction via attention-based CNN–LSTM. *Applied Sciences*, *10*(6), 1953.

50) Marshall, J. (2013). PM 2.5. *Proceedings of the National Academy of Sciences*, *110*(22), 8756-8756.

51) Sharma, S., Chandra, M., & Kota, S. H. (2020). Health effects associated with PM 2.5: a systematic review. *Current Pollution Reports*, *6*, 345-367.