International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://www.theaspd.com/ijes.php

Enhancing Air Quality Prediction With A Multi Pollutant Attention-Based Model

Eman Ayad Hashim

Department of Computer Engineering, Al Iraqia University, Baghdad, Iraq
Corresponding author: emanayad82@gmail.com

Abstract

As the world expands and develops year by year, wars and industries are increasing rapidly, and their remnants negatively affect our environment. So, the air pollution problem has raised and affected various domains such as health, environment, and ecosystem sustainability. Therefore, the need for air quality forecasting has emerged to avoid or reduce this problem. The researchers have worked on this issue, but it still has limitations, especially the need to improve prediction accuracy, model performance and the ability to predict the most common pollutants. In this study, a forecasting model is implemented to predict the concentration of the most common pollutants in the world: Particulate Matter less than 2.5 micrometers in diameter (PM2.5), Particulate Matter less than 10 micrometers in diameter (PM10), Nitrogen Dioxide (NO2), Sulfur Dioxide (SO2), Carbon Monoxide (CO) and Ozone (O3) for three time steps ahead. In the proposed model, a new approach (weight transfer) is implemented to predict all pollutants automatically from just one model. This model is a hybrid deep learning CNN-BiLSTM-Attention model that leverages the strengths of both Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network to capture spatial and temporal dependencies in air quality multivariate data, also applied Bidirectional Long Short-Term Memory (BiLSTM) network and Self-Attention to extract further temporal dependencies. The Huber loss function is used to balance the loss values between Mean Squared Error (MSE) and Mean Absolute Error (MAE) loss functions, aiming for robust model performance. Additionally, a user interface is implemented to display the values and chart of forecasted pollutants. The proposed model is achieved using the dataset of Beijing city in China, which contains historical air quality and meteorological data. It is performed by Python language in the Colab environment, obtained effective results as a regression forecasting type, and the Coefficient of Determination (R^2) evaluation metrics for PM2.5, PM10, NO2, SO2, CO, and O3 are 0.951, 0.890, 0.941, 0.918, 0.935, and 0.959, respectively; also, the model is evaluated using Root Mean Squared Error (RMSE) and MAE metrics. In addition, it surpassed some of the state-of-the-art models that used the same dataset. Moreover, the model is applied to the available dataset of Baghdad - Iraq.

Keywords: *air quality, air pollution, prediction, forecasting, deep learning, CNN-LSTM.*

1. INTRODUCTION

Air pollution is a major environmental and public health issue affecting people worldwide. World Health Organization (WHO) defines air pollution as "it is contamination of the indoor or outdoor environment by any chemical, physical or biological agent that modifies the natural characteristics of the atmosphere. Household combustion devices, motor vehicles, industrial facilities, and forest fires are common sources of air pollution." ("Air pollution,"). According to WHO data, nearly the entire global population (99%) breathes air that exceeds WHO guidelines limits and contains high amounts of pollutants, with the highest exposures, especially in developing countries. Air quality is closely related to climate change and the global ecosystems of the earth ("Air pollution,"). This unhealthy air quality threatens not just the health and life of individuals but also the economy. The report carried out by the Organization for Economic Cooperation and Development (OECD) has shown that air pollution might cost 1% of the global Gross Domestic Product (GDP) (Bekkar et al., 2021).

For the reasons mentioned above, the need to reduce pollution risks has emerged; to achieve that it needs accurate and reliable air quality predictions to empower decision-makers, air quality management, and public health officials to take precautionary measures. In the beginning, air quality models were built by deterministic models and classical statistical models such as Community Multiscale Air Quality (CMAQ) model and Auto Regression Integrated Moving Average (ARIMA) model, but these models had limited performance and could not capture the non-linearity features of air quality data (Wu et al., 2022). To overcome this problem, machine learning is then used, such as Random Forest (RF) and Support Vector Regression (SVR) models, which are subsets of artificial intelligence. However, the data amount has increased and become huge, where machine learning is hard to deal with and cannot fully extract the complex and dynamic spatial and temporal correlations from the historical data (Zhang et al., 2022). The researchers developed machine learning methods to produce deep learning models such as CNN, Recurrent Neural Network (RNN), LSTM and Gated Recurrent Unit (GRU) that can deal with big data, using more layers and processing a large number of layers together simultaneously to get more accurate predictions (Subramaniam et al., 2022). The concentration data of pollutants (particulate matter and gases) have non-linear relationships and contain complex correlations in spatial and temporal dimensions and each dimension has a dynamic change correlation (Masood and Ahmad, 2021). The CNN model can extract spatial features, and the LSTM model captures temporal dependencies in multivariate sequential data; the strengths of both models combined in CNN-LSTM (Saini et al., 2021). That is why the hybrid CNN-LSTM model is the most used because of its ability to extract non-linear and complex relationships and spatial-temporal correlations in depth (Méndez et al., 2023).

In this research a hybrid model CNN-BiLSTM-Attention is proposed to predict the concentration of most common pollutants in the world (PM2.5, PM10, NO2, SO2, CO, O3) for three time steps ahead, The main contributions of this work are as follows:

- Implement a new approach (weight transfer) that transfers the weights of each pollutant (PM2.5, PM10, SO2, NO2, CO, O3) to the base model (CNN-BiLSTM-Attention). This approach makes automated prediction by a single model for all pollutants instead of a separate model for each pollutant.
- Utilize the Huber loss function to balance the loss values between MSE and MAE loss functions to obtain robust model performance with better results than the commonly used MSE loss function.
- Shuffling only batches of data to keep the temporal dependencies, avoid overfitting and avoid not aligning prediction on the actual output value.

The remainder of this research is structured in the following manner: Section 2 reviews the related work. Section 3 the background theories. Section 4 delves into the research methodology. The experimental procedures and results discussion are detailed in Section 5. While Section 6 offers the conclusions and future work.

2. RELATED WORK

In the literature spanning from 2020 to 2023, studies employing the CNN-LSTM model for extracting spatial-temporal features from air quality datasets. The primary objective of these studies is to improve prediction accuracy and overall performance, they focused on several issues some of them are: integrated attention mechanism with the CNN-LSTM model as in these studies (Li et al., 2020), (Zhang et al., 2021), (Jiang et al., 2021), (J. Wang et al., 2022), (Mengara Mengara et al., 2022) and (Zhou et al., 2023) to effectively predict air quality by learning long-term dependencies and used BiLSTM to capture the temporal dependencies for past and future time series data as implemented in researches (Mengara et al., 2020), (Du et al., 2021), (Bhanja

and Das, 2021), (Zhang et al., 2021), (Kim et al., 2021), (Mengara Mengara et al., 2022), (J. Wang et al., 2022) and (Mahadik, 2023), some of them explored as follows:

2.1 Single Pollutant Prediction

Almost researchers proposed single pollutant prediction models, some of them are: (Bhanja and Das, 2021) proposed a Hybrid Deep Neural Network (HDNN) framework to predict hourly the PM2.5 concentration in Sydney and Delhi cities. They used the CNN-BiLSTM and 3D input tensor formation technique to provide consistent performance. They also utilized a linking layer between CNN and BiLSTM to preserve the temporal ordering of feature vectors. However, the limitation of this model is that the dropout rate is 0.6, which is very high, that indicates the model is complex and more than half of the layers will not be used. Transfer learning was used by (Deng et al., 2021), who proposed a CNN-LSTM model for predicting hourly Ozone concentration in Eisenhuttenstadt - Germany. Then they tried to use this proposed model as a basic model by transfer learning to make the daily prediction from small temporal resolution data, and that also reduced the training time. However, that leads to shortcomings in the daily peak Ozone concentration forecasting. Moreover, it still needs to use the traditional Chemical Transport Model (CTM) to predict Ozone, as it is more accurate when Ozone concentrations are high. A genetic algorithm is carried out in the CNN-LSTM model by (Tsokov et al., 2022) to optimize the architecture and hyperparameters of the model. They made a hybrid strategy for filling missing values by using interpolation or average value according to the missing value gap then predicting hourly PM2.5 concentration in Beijing - China. However, it does not compare the proposed model with other models; it just compares the other models separately; also, after experiments and optimization steps, the final model characteristics and results are not presented. The authors (Zhou et al., 2023) proposed a Temporal Pattern Attention (TPA) mechanism for weights adjustments to automatically update weights based on each time step's input and to select the most relevant time step for prediction. TPA integrated with CNN and LSTM to implement the TPA-CNN-LSTM model to accurately predict the next six hours (multi-step prediction) of PM2.5 concentration. Besides historical air quality and meteorological data, the model used PM2.5 concentration data from adjacent stations to capture the long-term dependencies of PM2.5 feature in Beijing - China.

2.2 Multi Pollutants Prediction

The researchers (Mengara et al., 2020) developed a CNN-BiLSTM-Autoencoder model for hourly forecasting PM2.5 and PM10 concentrations in Busan - South Korea. The model employs a stacked autoencoder to extract information and features from air quality data and meteorological data, then they employed the data parallelism, which is a distributed deep learning technique, to speed up the training time. However, the model has limitations, including the training process based on a distributed learning method, which may require significant resources and infrastructure to implement. Attention mechanism with the CNN-LSTM implemented by (Jiang et al., 2021) to automatically weight features according to their importance, so proposed this model Attention-CNN-LSTM to predict CO and NO concentration from 1- 24 hour (multi prediction) in the north of Taiwan, also Granger causality analysis used to model the spatial correlation between stations. Multi pollutants (PM2.5, PM10, NO2, SO2, CO, O3) for Xi'an - China, predicted hourly and daily by (Dai et al., 2021). They applied the One-Dimensional Multiscale Convolutional Neural Network (ODMSCNN) and LSTM to proposed ODMSCNN-LSTM model which can extract multiple features. The researchers found the model accuracy is superior in the hourly prediction than daily due to available more data in hourly prediction. Finally, (Hu et al., 2023) proposed a hybrid deep learning model that combines CNN, BiLSTM, and GRU layers for hourly prediction of PM2.5 and O3 concentrations in Beijing - China. The model used both air quality data and meteorological data this outperforms other models that relied solely on air quality data or the data specific to the target pollutant.

Moreover, all mentioned researches for multi pollutants prediction are limited to predict two pollutants excepts (Dai et al., 2021) predicts all common pollutants but restricted to single-step prediction as other researches do, except (Jiang et al., 2021) and (Zhou et al., 2023) for multi-step prediction.

Although, all valuable and useful researches still there are limitations of air quality prediction, some of them address in this study which are the need to improve prediction accuracy, model performance and the ability to predict the most common pollutants.

3. BACKGROUND

The main background theories used in this study:

3.1 Convolutional Neural Network (CNN)

CNN is a neural network commonly used for image processing based on a convolution operation (Peixeiro, 2022). However, CNN can work on different arrays of data whereas 1D is for signal, text and sequence data, 2D is for images and 3D is for videos (Zaini et al., 2022).

CNN 1D is mostly used for time series forecasting as the kernel can only move in the time dimension (Peixeiro, 2022) and spatial features can be easily extracted (Li et al., 2020). Recent CNN has been applied in time series forecasting to reduce data dimensionality and extract the spatial features of multivariate data to enhance the performance of the prediction model (Zaini et al., 2022).

3.2 Bidirectional Long Short-Term Memory (BiLSTM) Network

LSTM is a neural network type of RNN architecture developed to perform sequential modeling tasks including temporal modeling (Samal et al., 2021), that is why it is the most commonly used to model air quality forecasting (Bekkar et al., 2021).

LSTM is good at learning long-term dependencies, a feature that is useful when modeling time series data. It overcomes the vanishing gradient problem by preserving errors that can be backpropagated across time and layers without risking the loss of important information (Lazzeri, 2021).

Standard LSTM network can only process in the forward direction. Consequently, during the extraction of the temporal features it is possible to miss some useful information of long-term dependency. While the BiLSTM network is capable of processing in both the forward and backward directions simultaneously. Therefore, it is more efficient to extract temporal features than the standard LSTM network (Bhanja and Das, 2021).

BiLSTM is created by combining forward and backward operations for LSTM. The output of the LSTM forward state is entirely separated from the input of the LSTM backward and reverse is also true (Mengara Mengara et al., 2022). Moreover, the BiLSTM networks are both connected to one input and one output layer, the output layer can get previous information about each point from the input sequence and also get future information from each point through this structure, as illustrated in Fig. 1 (Shah et al., 2021).

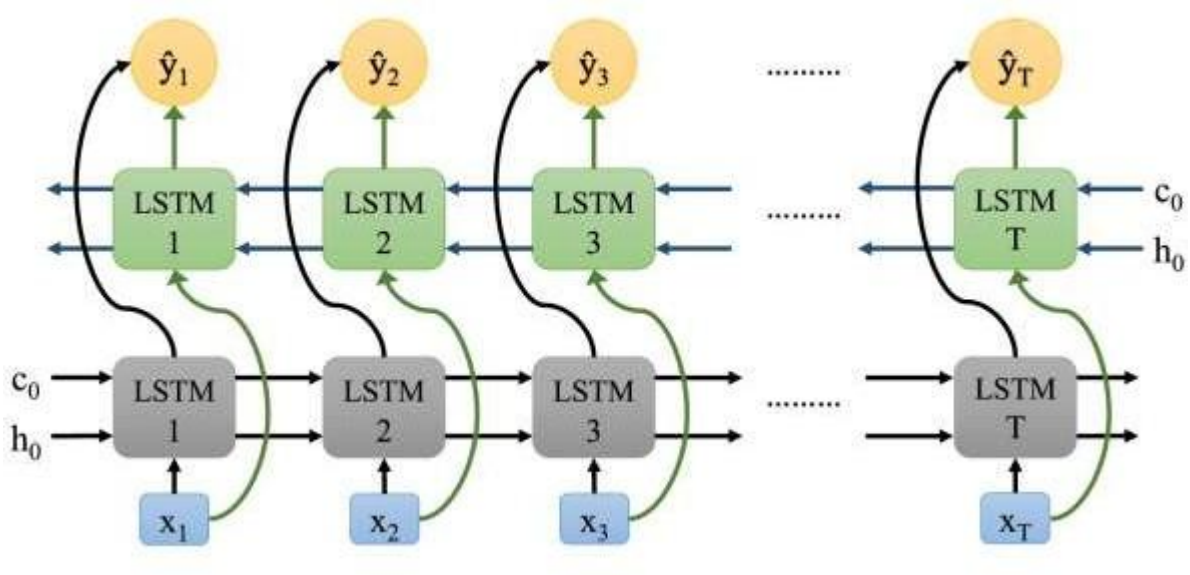


Fig. 1. BiLSTM Network (Shah et al., 2021)

3.3 Attention Mechanism

Attention has become one of the most important concepts in the deep learning field. Its idea is inspired by a human visual process that focuses on some certain areas when processing large amounts of data (Niu et al., 2021). For example, when looking at an image, the brain pays more attention to a certain region of interest and ignores others (Dairi et al., 2021). The accuracy and efficiency of perceptual information processing have been significantly improved by the attention mechanism (Niu et al., 2021) where the attention is a weight assignment technique that continuously and automatically updates the weights of various features by assigning a weight to the important features and ignoring unimportant features (Zhang et al., 2021), (Li et al., 2023).

3.4 Loss Function

The loss function quantifies the difference between the true (actual) output and the predicted output of the model (Keren, 2019). The train loss is calculated during the iteration or epoch, while validation loss is calculated after each iteration or epoch (Dey et al., 2021). The values of the loss function propagated back to learn and guide the weights update related to the connections between the deep learning neurons or filters of the neural network. In addition, it can tune the hyperparameters based on the decrement in the loss function value to minimize the differences between the true value and the predicted value (Kerkhof et al., 2023), so it measures the trained model performance to optimize it (Keren, 2019) and eliminate the overfitting problem to perform better accuracy (Dey et al., 2021).

There are many loss function types that differ in methods of calculating errors, and the choice of them depends on the optimization process (Gonzalez, 2020). The following are some of the loss functions used in regression models (Gayakwad et al., 2022):

- **Mean Squared Error (MSE) Loss:** it is simple and the most common loss function used in regression models and also it can be used in classification models. It calculates the mean of squared pairwise of error (difference between the true value and the predicted value) (Kerkhof et al., 2023). The gradient length of MSE loss fluctuates, it rises rapidly with the increase of error when the sample has a large error and it becomes smaller as the error value approaches 0, as shown in Fig. 2 (a). This leads to fast and smooth convergence but focuses on the outlier values because of squaring the error, when the error increases that means the MSE loss increases faster this prioritizes the outlier values and disregards the normal values that cause inaccurate prediction (Q. Wang et al., 2022).
  Its formula as shown in Equation (1), where y_i is the true value and the $\hat{y}$ is the predicted value, N symbolizes the number of training samples (Ciampiconi et al., 2023).

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \quad \ldots(1)$$

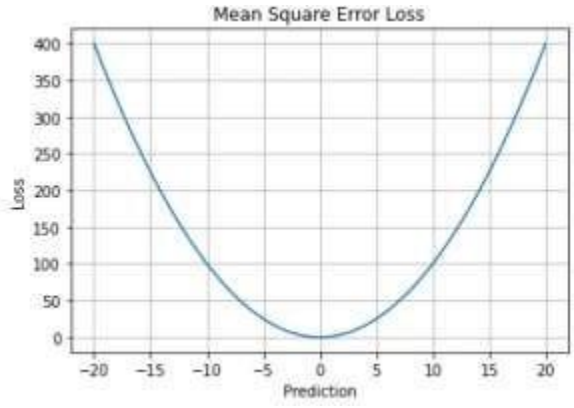
- **Mean Absolute Error (MAE) Loss:** it calculates the mean of absolute pairwise error (difference between the true value and the predicted value). On the contrary to the MSE loss, the MAE loss value does not rise rapidly with the increase of error, so it is less sensitive to outlier than MSE loss. However, it does not show smooth convergence when the error approaches 0, as shown in Fig. 2 (b) and the gradient length of MAE loss is fixed, which affects the efficiency of the model (Barrow et al., 2020). The equation of MAE loss is shown below where y_i is the true value and the $\hat{y}$ is the predicted value, N symbolizes the number of training samples (Ciampiconi et al., 2023).

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}| \quad \ldots(2)$$

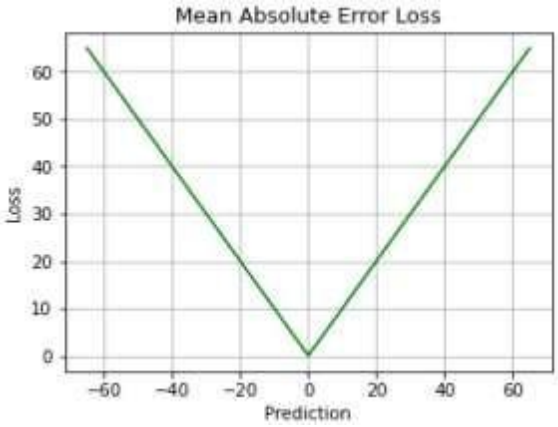
- **Huber Loss:** it is a piecewise function of MSE and MAE as shown in Fig. 2 and in Equation (3), where y_i is the true value and the $\hat{y}$ is the predicted value, δ symbolizes the threshold parameter (Ciampiconi et al., 2023).
  The Huber loss follows the MSE when $|y_i - \hat{y}| \leq \delta$, otherwise, it follows the MAE. In other words, for small values it follows the MSE to get smooth convergence as it is the advantage of MSE and avoiding the sensitivity to outlier as it is the disadvantage of MSE. In the same way, for large values it follows the MAE to get the robustness of MAE and avoid sharp convergence, so Huber combines the advantages of both MSE and MAE (Ciampiconi et al., 2023), (Barrow et al., 2020). The threshold parameter (δ) is used as the boundary to determine the singularity of the sample. Samples within this boundary use MSE while those beyond it use MAE. This approach reduces the weight of the loss function for outliers and helps avoid the overfitting problem (Q. Wang et al., 2022). The choice of threshold parameter (δ) is crucial and can be continuously adjusted during the training phase based on what is considered an outlier (Ciampiconi et al., 2023).

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://www.theaspd.com/ijes.php

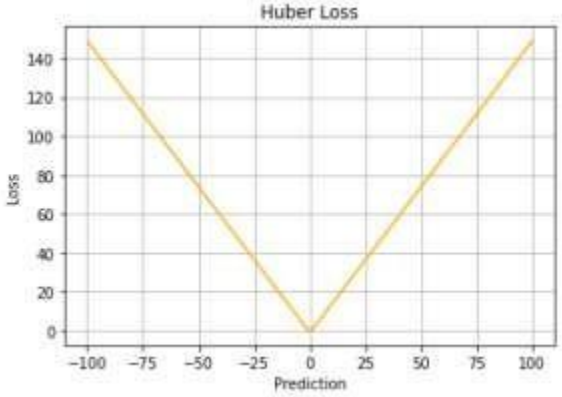
$$\text{Huber} = \begin{cases} \frac{1}{2}(y_i - \hat{y})^2 & \text{for } |y_i - \hat{y}| \leq \delta, \\ \delta\left(|y_i - \hat{y}| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases} \quad \ldots(3)$$



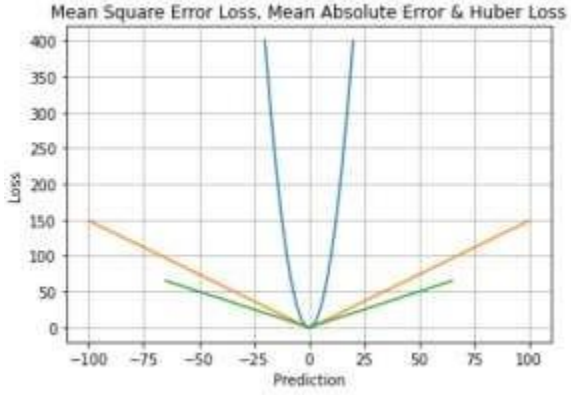
(a) MSE loss



(b) MAE loss



(c) Huber loss



(d) MSE, MAE & Huber loss

MSE
MAE
Huber

Fig. 2. Regression Loss Functions (Gayakwad et al., 2022)

4. METHODS

The methodology of the proposed model has seven stages designed to increase prediction performance, as illustrated in Fig. 3.

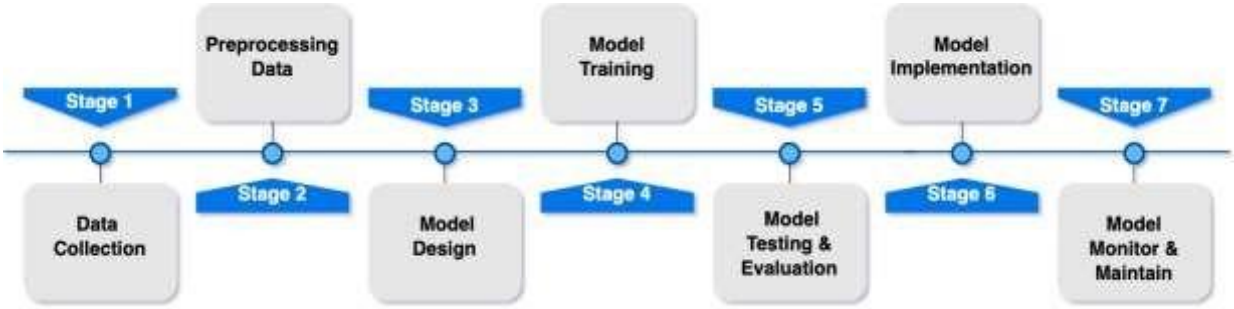


Fig. 3. Methodology Stages

4.1 Data Collection

The proposed model used dataset from Kaggle website ("Beijing Multi-Site Air-Quality Data Set,"), it is for Beijing city contains hourly air quality data and meteorological data from 12 nationally controlled air quality monitoring sites, includes (35064 instances and 18 attributes). The period time is from 01/Mar/2013 to 28/Feb/2017. The types of data contained within the dataset are presented in Table 1. This dataset has chosen because it contains all common pollutant concentrations and the meteorological data, both of which significantly influence changes in these pollutant levels. This dataset is characterized as time series data, making it suitable for the model's forecasting tasks.

Table 1 Data Type.

Data Type	Parameter	Parameter name	Unit
Air quality data	PM2.5	Particulate Matter with a diameter of less than 2.5µm	µg/m*
	PM10	Particulate Matter with a diameter of less than 10µm	µg/m*
	SO2	Sulfur Dioxide	µg/m*
	NO2	Nitrogen Dioxide	µg/m*
	CO	Carbon Monoxide	µg/m*
	O3	Ozone	µg/m*
Meteorological data	TEMP	Temperature	°C
	PRES	Atmospheric Pressure	hPa
	DEWP	Dew Point	°C
	RAIN	Rain	mm
	WD	Wind Direction	-
	WS	Wind Speed	m/s

4.2 Preprocessing Data

Data need to preprocess before entering the model to ensure the accuracy and reliability of the model's prediction. Preprocessing data involves several steps, as explained below:

1. **Clean Data:** It is the first step in preprocessing data involves several processes to make the data ready for normalization, some of them are: Check and analyze the data visually to see if it has an error value. Filling in any missing values, in this study used linear interpolation method for each attribute. Factorize to numeric representation for non-numeric values.

2. **Feature Selection:** It is essential to select the most relevant features to simplify the model, avoid overfitting of the model and make model training faster. This is done by features correlation where the Pearson correlation type has been used for all features in the dataset. The input features of the proposed model have been finalized as sixteen features (year, month, day, hour, PM2.5, PM10, SO2, NO2, CO, O3, TEMP, PRES, DEWP, RAIN, WD, WSPM).

3. **Normalization:** The Min-Max Scalar (Das et al., 2022) is used to scale the dataset, it preserves the original distribution of the features while transforming them into a range of [0, 1] that will improve the performance because all features will be in the same scale so the model will learn easily. Scalar data is saved for use later in the prediction phase, ensuring consistency in applying the model to new data. This normalization process is applied just to the input data and keeps the target data unscaled which has better results compared to scaling the target data.

4. **Data Windowing:** Since the data is a time series type and the model's objective is to predict future values, the function for generating sequences is created to divide the data into input/output pairs batches. The data length for the input window is set at 24 to capture the daily periodicity of the data, and the output window is set at 3 for three step prediction, as shown in Fig. 4. This windowing approach enables the model to train more effectively than it would with single record data, thereby enhancing the accuracy of the predicted values.

5. **Split data:** It is an essential step in building and evaluating the model by randomly dividing the dataset into three portions. It has been divided in the proposed model as follows: train set 70%, test set 15%, and validation set 15%.

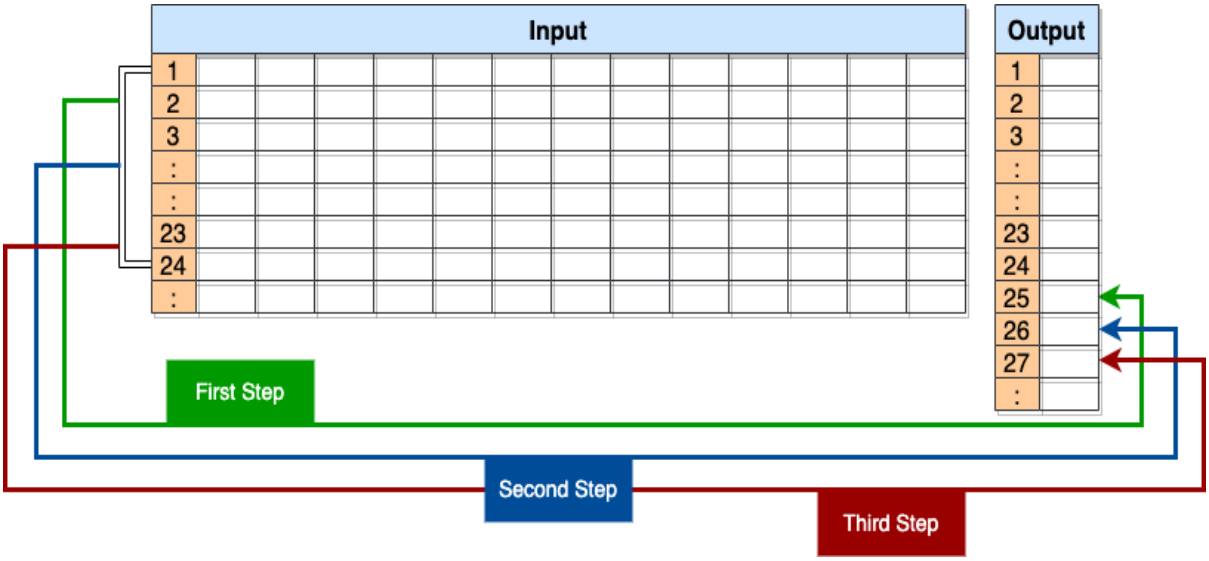


Fig. 4. Three st qep Prediction.

4.3 Model Design

In order to design a forecast model, the best choice is the sequential design type. Besides that for air quality data using the CNN-BiLSTM-Attention approach is the state-of-the-art deep learning approach. The architecture of the proposed model is illustrated in Fig. 5 and explained as follows part by part:

1. **Input Part:** the training data is entered into the model by the input layer with this shape (24,16), 24 is the data length for the input window, and 16 is the selected features.
2. **CNN Part:** CNN layer is used to extract the spatial feature pattern as data is time series multivariate data of one dimension, so the CNN-1D layer is used. The first block contains CNN layer of 32 filters with filter size = 3 and padding = "same" to keep the shape of output data the same as input data. The activation function is used to introduce the non-linearity of data and allows the model to capture complex relationships within the features. When using Sigmoid or Hyperbolic Tangent activation functions with deep network leads to a vanishing gradient problem. While ReLU works well with deep network, it overcomes the vanishing gradient problem, making the model train faster and performing better. However, it has a Dying ReLU problem where some neurons become inactive (die), whereas the Leaky ReLU activation function corrects this by changing the slope of the x-axis, allowing a small non-zero slope for negative input values. So, Leaky ReLU is used in the proposed model. The second block is the same as the first block. Likewise, the third block is also the same as the first block but with the number of filters = 64 and after activation function used Max Pooling 1D to downsample the dimension of data to reduce the complexity and at the same time keep the most important data. The suggested method of choosing the number of neurons for the CNN layer and later for LSTM is based on 2^n neurons followed by the same number of neurons or 2^{n+1} neurons where n is an integer number.
3. **LSTM Part:** LSTM layer is used to extract the temporal long-term dependencies of sequential data. It can learn patterns and relationships that exist in the time series data. The first block contains Bidirectional LSTM layer with 64 units, using bidirectional gives additional training of data because the output data is from two directions of layers simultaneously instead of one direction in the unidirectional LSTM. Then the Dropout layer is added with rate = 0.2, which means randomly setting 20% of input units to 0 to regularize the model and improve generalization. The second block is the same as the first block but with 128 LSTM units and dropout layer rate = 0.3.
4. **Attention Mechanism Part:** After the LSTM part, the attention mechanism layer is added, a component that enables the model to focus selectively on different parts of the input sequence. Self-Attention type is used to capture both short-term and long-term dependencies. The attention mechanism generates the weights as key, query and value then calculates the output weights and returns them back to the model. The integration of this Self-Attention layer allows the model to identify which parts of the input sequence are most informative for predicting air quality, thus enhancing the overall predictive power of the model.
5. **Output Part:** Global Average Pooling 1D is used to summarize the output and generate a fixed length representation, observing that this presentation is more effective than when utilizing the Flatten layer. Followed by a single Dense layer used to output the predicted value of the proposed model, the Lambda layer is used to make the inverse scale (Min-Max Scalar) calculation of the scaled output value so the model's output will be readable.

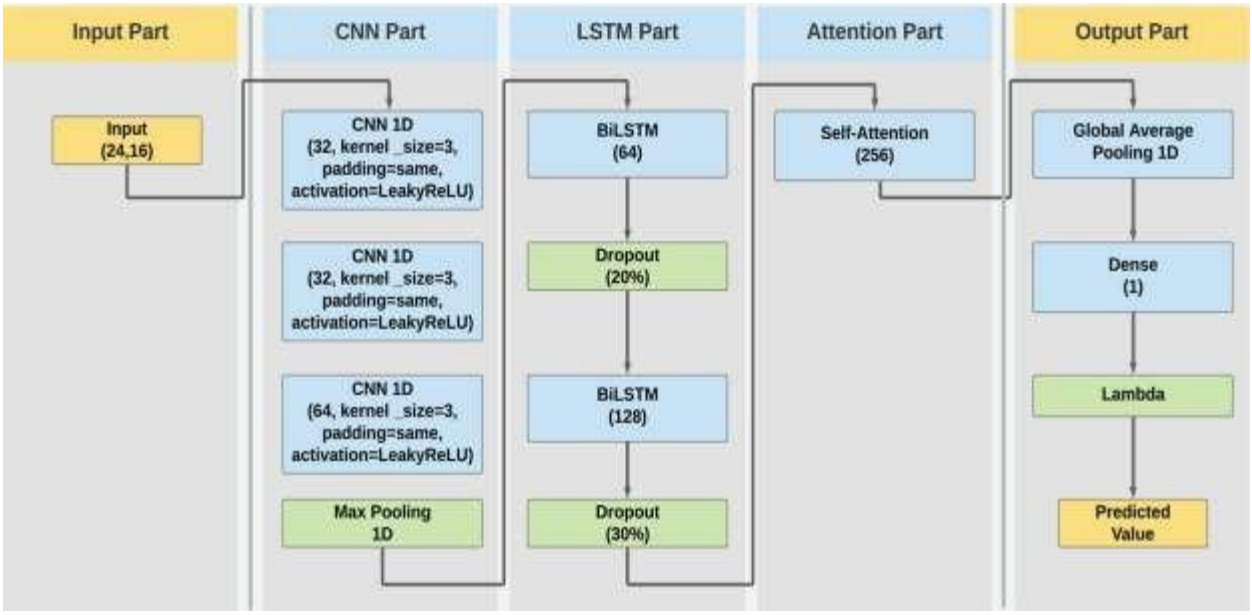


Fig. 5. Model Design.

4.4 Model Training

It is the process of feeding the model by train data (historical data) to get the prediction values (future values); this is done by learning the patterns and relationships between features. Besides that, adjust the model parameters to get the minimum error (minimum value of loss function) also, the validation loss should be less than the train loss value to overcome the overfitting problem (Lazzeri, 2021). The training process includes two steps:

1. **Compile model:** This step includes defining the optimizer, loss function, and any additional metrics to be used in the training process. For optimizer, the Adam algorithm is used because it is well suited for models that need large amount of data, such as deep learning models. For the loss function, Huber is used because it combines the MSE and MAE in a way that is useful for time series data and produces improved results, as it does not focus on outlier like MSE and does not ignore them like MAE.

2. **Fit Model:** This step defines the train data, validation data, batch size = 32 (as a standard value), number of epochs = 100 because there are no improvements after 100 epochs, and two call back checkpoints: the first to save the best model (architecture, weights, optimizer state, and training configuration) and the second save the best weights of the best model to reuse it in the implementation phase. The term (best) refers to the optimal performance metrics in the proposed model, which is characterized by the lowest validation loss observed during the training process.

The training process has been performed six times each time for one of the selected features (PM2.5, PM10, SO2, NO2, CO, O3) to save the best weights for that parameter, which means saving six weights files each file for one parameter and save just one model. Since all six models have the same input data and the same architecture, consequently, it would be markedly more efficient to employ one model with different weights according to the target parameter, as described in Fig. 6, and choosing that model based on the optimal metrics value.

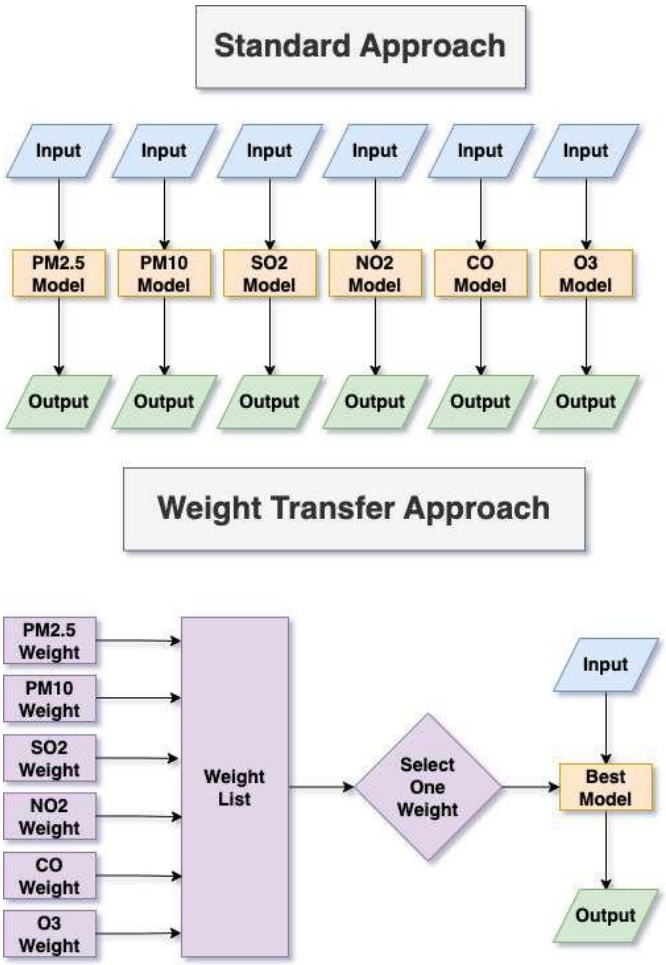


Fig. 6. Automated Approach.

4.5 Model Testing and Evaluation

At this stage, the model's generalization capability on the test data (unseen data) is assessed, and the model's performance is evaluated based on specific evaluation metrics. A detailed discussion of this process will be presented in section 5.

4.6 Model Implementation

The model is implemented by following steps:

4.6.1 User Interface

The trained model integrated with Gradio (open-source Python package) to make the user interface and get the prediction of future values for the next hours (time step), as shown in Fig 7.

Fig. 7. Model's User Interface.

4.6.2 Prediction Function

Prediction is implemented by running the created prediction function, which is executed after the user submits the requirements via the user interface. The function gets its inputs from this user interface (predicted pollutant, input data and number of predictions). Then, the same steps that were carried out at the preprocessing stage are being re-implemented here for the new input data, including filling in missing values, feature selection and normalization. Normalization is done by loading the scalar data to maintain consistency with the normalization applied previously in the preprocessing stage. Following this, the base model and all parameter weights are loaded then automatically select the weight of the required pollutant to predict. The prediction process then proceeds, capable of generating up to three predictions as per the request. Finally, the predicted concentration values of the requested pollutant are displayed as numerical values and a graphical chart with their corresponding time. The procedure of the prediction function is outlined in algorithm 1.

4.6.3 Weight Transfer

In neural networks, the model's knowledge is stored in the network's parameters. These parameters include weights and biases that are learned from the input data and adjusted throughout the training process. Once the model is trained, these parameters define how the model will make predictions on new data and can be stored in a separate file with extension (.h5).

In order to implement the automated model that predicts any pollutant of these pollutants (PM2.5, PM10, SO2, NO2, CO, O3) upon request, a new approach has been employed. This approach stands on the use of one base model for each pollutant's prediction. Alongside the base model, the weight file specific to the required pollutant is automatically transferred to that model when a prediction is requested. In each pollutant prediction, the saved weight is used to predict this pollutant on the same model. This approach makes the prediction automated, in other words, using one model instead of using six models, which provides faster prediction.

Algorithm 3.1: Prediction Function

Input:	pollutant (pred_par), input data (data), number of predictions (pred_no)
Output:	predicted pollutant concentrations value (pred_list), predicted pollutant concentrations chart (figure)
Step1:	data = select feature (data) data = fill missing value (data) load (scalar) scaled_data = normalize (data)
Step2:	load (base_model) weights_list = [PM2.5, PM10, SO2, NO2, CO, O3] pred_list = []
Step3:	For i ≤ pred_no load weight (weights_list [pred_par]) pred = predict model (scaled_data) pred_list += pred End
Step4:	figure = plot chart (pred_list)

4.7 Model Monitor and Maintain

After implementation, it is important to monitor and maintain the model to ensure the accuracy, reliability and continuity over time by checking the input data and periodically re-training the model, suggested on a weekly basis to update the model with the new data.

5. RESULTS AND DISCUSSIONS

5.1 Experimental Setup

The proposed model is implemented using the Python programming language within the Google Colab environment, a cloud-based Jupyter notebook service that facilitates writing and executing Python code via a web browser. The model variables setting are listed in Table 2.

Table 1
Variables Setting.

Variable	Value
Normalizer	Min-Max Scalar
Input window length	24
Input shape	(24,16)

CNN kernel size	3
CNN padding	Same
Activation Function	Leaky ReLU
Optimizer	Adam
Learning rate	0.001
Loss function	Huber
Train set	70%
Test set	15%
Validation set	15%
Training Batch size	32
Number of epochs	100
Early stopping monitor (patience)	20

5.2 Experimental Results of Single-Step Prediction

In this section, the results of evaluation metrics (R2, RMSE, MAE) for single-step prediction are presented, analyzed and discussed. Besides, the loss values (train loss, validation loss and test loss) are presented to check whether the model has overfit or not. When the train loss is lower than validation loss or test loss, it indicates that the model may be overfitting. First the proposed model compared with different methods to focus on what are the changes and components that increase its accuracy and performance. Then compare the proposed model with the state-of-the-art models that used the same dataset.

5.2.1 Preprocess Dataset

The Beijing dataset is hourly observation includes (35064 instances and 18 attributes), after feature selection became (35040 instances and 16 attributes), one site (Aotizhongxin) of the dataset used for experimental results.

For filling in missing values of the dataset, three different methods have been tested and compared as follows:

- Drop method: it drops the missing values that decreases the amount of data and disrupts the sequence and continuity of data, whereas forecasting time series data needs sufficient and sequential data to predict accurately.
- Mean method: it fills the missing values with the mean value of the specified feature which can be misleading, especially if the missing values are not missing at random. For time series data, filling gaps with mean values can distort the temporal patterns as this gap will be filled by one constant value for several time series values, which conflicts with the nature of time series data.
- Linear interpolation method: this method uses the closest known values to predict the value between the two closest known values (before and after the missing value) and fills the gap based on the trend between these points. This preserves the trend and can often lead to more accurate prediction for sequential data, making it more suitable for time series data.

Table 3 represents the evaluation of these methods and indicates that the linear interpolation outperforms other methods in evaluation metrics. Table 4 represents the loss values that shows there is no overfit for the interpolation method while the other methods have.

Table 3 Evaluation Results of Different Methods for Filling Missing Values.

Evaluation	Fill Missing Value	PM2.5	PM10	SO2	NO2	CO	O3
R^2	**Drop**	0.927	0.866	0.920	0.905	0.926	0.948
	Mean	0.938	0.866	0.931	0.901	0.878	0.937
	Interpolate	**0.951**	**0.890**	**0.941**	**0.918**	**0.935**	**0.959**
RMSE	**Drop**	22.111	35.595	6.454	11.579	334.462	13.152
	Mean	19.947	33.798	5.762	11.553	406.947	14.144
	Interpolate	**17.878**	**30.645**	**5.338**	**10.606**	**312.649**	**11.538**
MAE	**Drop**	10.468	18.373	2.909	7.278	177.750	7.743
	Mean	10.031	17.497	2.913	7.378	182.268	8.286
	Interpolate	**9.321**	**16.664**	**2.714**	**6.91**	**158.71**	**7.048**

Table 4 Loss Values of Different Methods for Filling Missing Values.

Structure	Loss	PM2.5	PM10	SO2	NO2	CO	O3
Drop	**Train Loss**	9.706	17.138	2.492	6.606	176.892	7.072
	Val Loss	9.451	17.040	2.355	6.675	169.346	7.072
	Test Loss	9.988	17.882	2.504	6.799	177.251	7.278
Mean	**Train Loss**	10.250	17.661	2.668	6.967	184.705	7.682
	Val Loss	9.979	17.696	2.559	6.761	183.301	7.915
	Test Loss	9.550	17.006	2.488	6.900	181.769	7.819
Interpolate	**Train Loss**	9.566	17.256	2.469	6.462	164.961	6.994
	Val Loss	9.206	17.122	2.362	6.351	164.066	6.806
	Test Loss	8.842	16.174	2.310	6.432	158.211	6.583

5.2.2 Loss Function

The proposed model uses Huber as a loss function instead of the MSE which is the commonly used one. Table 5 shows that Huber is slightly better than MSE, for fitting the model and rate of errors, when compared with Fig. 8 of dataset outliers (after preprocessing) found that PM10 which has the highest outlier number is less improved by Huber and SO2 is the most improved with the lowest number of outliers, also for PM2.5 there is an overfit in MSE while it is removed in Huber as described in Table 6. That indicates the Huber can balance between regular and outlier values while MSE is sensitive to outlier values, despite the improvement rate being low but it is crucial for time series data and air quality data type. It is worth mentioning that outlier values should not be processed because this would conflict with the nature of air quality data that is affected by multiple factors and events.

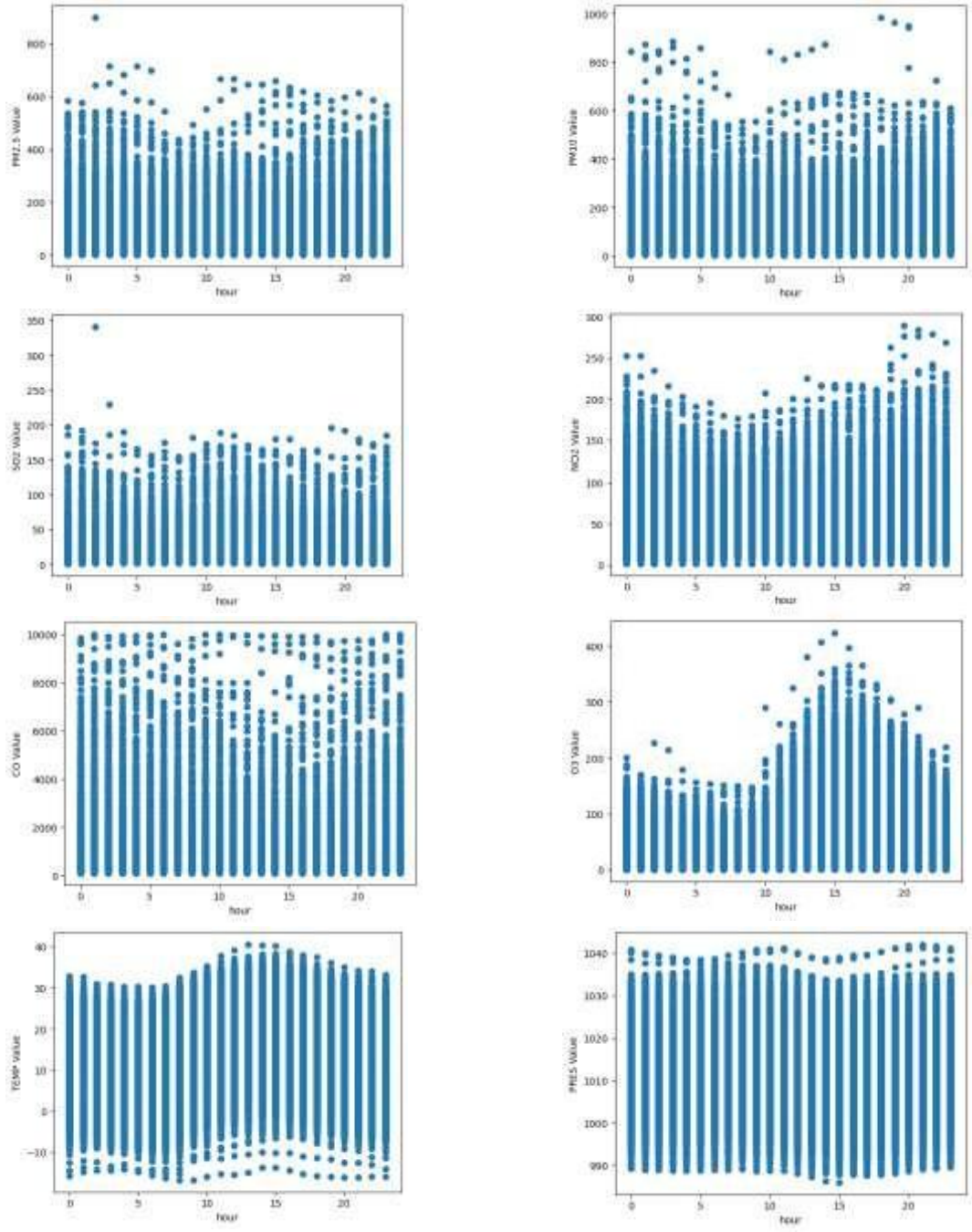
Table 5 Evaluation Results of Different Loss Functions.

Evaluation	Loss Function	PM2.5	PM10	SO2	NO2	CO	O3
R^2	MSE	0.947	0.886	0.936	0.916	0.934	0.958
	Huber	**0.951**	**0.890**	**0.941**	**0.918**	**0.935**	**0.959**
RMSE	MSE	18.725	31.196	5.559	10.754	315.191	11.705
	Huber	**17.878**	**30.645**	**5.338**	**10.606**	**312.649**	**11.538**
MAE	MSE	10.137	17.700	2.962	7.241	166.833	7.658
	Huber	**9.321**	**16.664**	**2.714**	**6.91**	**158.71**	**7.048**

Table 6 Loss Values of Different Loss Functions.

Structure	Loss	PM2.5	PM10	SO2	NO2	CO	O3
MSE	Train Loss	296.945	995.039	32.190	108.630	99941.351	129.199
	Val Loss	345.627	1109.295	34.609	117.346	116568.687	142.519
	Test Loss	350.641	973.220	30.912	115.653	99345.945	137.024
Huber	Train Loss	9.566	17.256	2.469	6.462	164.961	6.994
	Val Loss	9.206	17.122	2.362	6.351	164.066	6.806
	Test Loss	8.842	16.174	2.310	6.432	158.211	6.583

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://www.theaspd.com/ijes.php



International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://www.theaspd.com/ijes.php

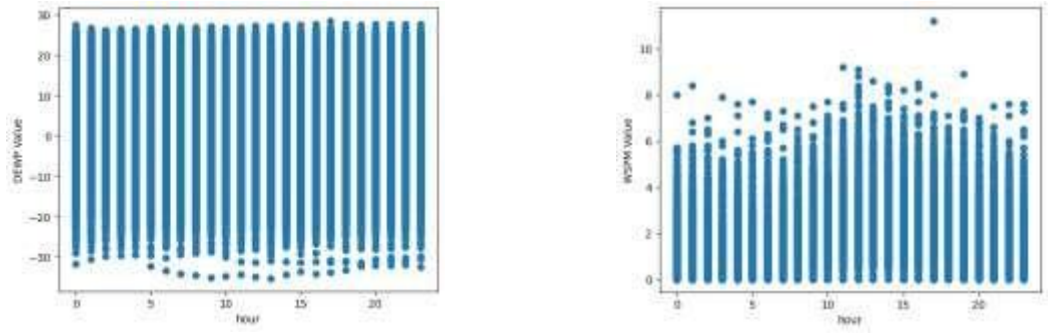


Fig. 8. Outliers of Dataset

5.2.3 Weight Transfer

The efficacy of this approach has been demonstrated through its application. By utilizing the saved weights for each parameter in prediction on the same model, that makes the prediction process automated. This implies that instead of deploying six different models, one model is deployed, which enhances the speed of prediction and simplifies development and management of the model. The results of this approach are the same as the result when using six model files, each file for a different pollutant, whereas the upload time decreased in this approach because the size of the weight file (2.5 MB) is considerably smaller than that of the model file (7.4 MB) for each pollutant.

5.2.4 Proposed Model

The proposed model predicts the concentrations of the six pollutants (PM2.5, PM10, SO2, NO2, CO, O3), the prediction results are varied from pollutant to another and all of them are typically considered high accuracy.

Table 7 shows the results of the best epoch that contains the best result of validation loss. This indicates that none of the pollutants are overfitted, that is because of shuffling data, using the dropout in the model structure and using early stopping monitor in the training phase.

Table 7 Loss values of Proposed Model.

Pollutant	Train Loss	Val Loss	Test Loss	Training Time (mm)	Best Training Epoch	Overfit
PM2.5	9.566	9.206	8.842	13	43	No
PM10	17.256	17.122	16.174	10	32	No
SO2	2.469	2.362	2.310	11	37	No
NO2	6.462	6.351	6.432	11	37	No
CO	164.961	164.066	158.211	13	48	No
O3	6.994	6.806	6.583	11	31	No

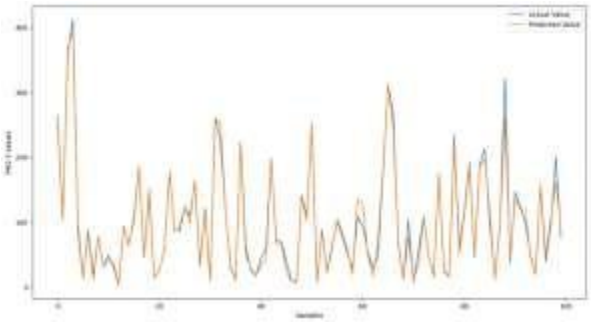
The evaluation metrics results are represented in Table 8. For R2 the result ranging from 0.890 to 0.959 that indicates the proposed model has a strong fit to the data for each pollutant, even though 0.890 to 0.959 is a narrow range, some pollutants may be easier to predict than others due to various influencing factors affecting their concentrations. For other metrics, the model performs well for each pollutant, which means they have low prediction errors. It is worth noting that the high results of CO for (RMSE and MAE) because of its data values in the thousands. For instance, O3 has 0.959 while PM10 has 0.890, which may suggest several reasons:

Table 8 Evaluation Metrics Results of Proposed Model.

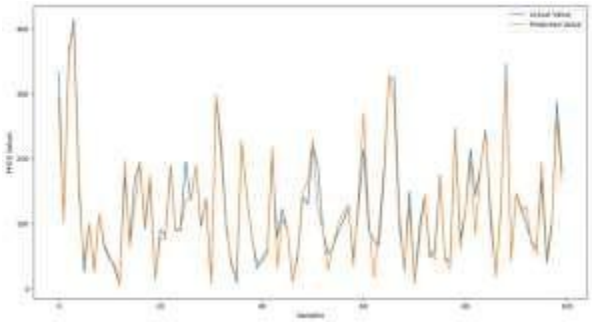
Pollutant	R^2	RMSE	MAE
PM2.5	0.951	17.878	9.321
PM10	0.890	30.645	16.664
SO2	0.941	5.338	2.714
NO2	0.918	10.606	6.910
CO	0.935	312.649	158.710
O3	0.959	11.538	7.048

The prediction results for all pollutants for 100 samples randomly selected from the test data described in Fig. 9.

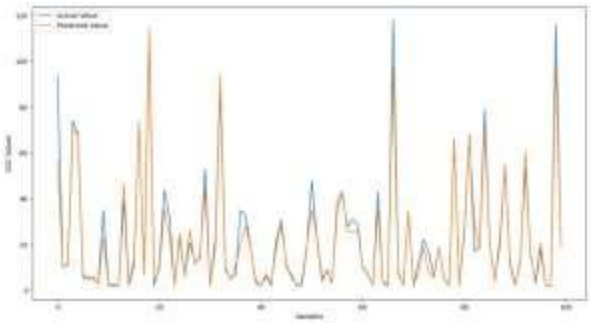
Moreover, the Cross-Correlation between the actual value and the predicted value of the model is close to 1 at lag = 0, as described in Fig. 10 (a) which indicates that the model's predicts the next value and the predictions are aligned with the actual output, that is because of shuffling the data batches. In contrast Fig. 10 (b) illustrates the Cross-Correlation at lag = -1 that indicates the model predicts the current value and does not align with the actual output so it fails to predict the next value, that because not shuffling the data batches. This satiation cannot be detected using evaluation metrics because the difference between the current value and the next value is very small for time series data. However, it can be detected using this function Cross-Correlation.



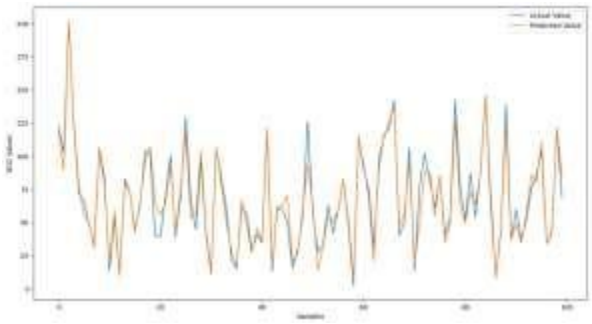
(a) PM2.5 Concentration



(b) PM10 Concentration

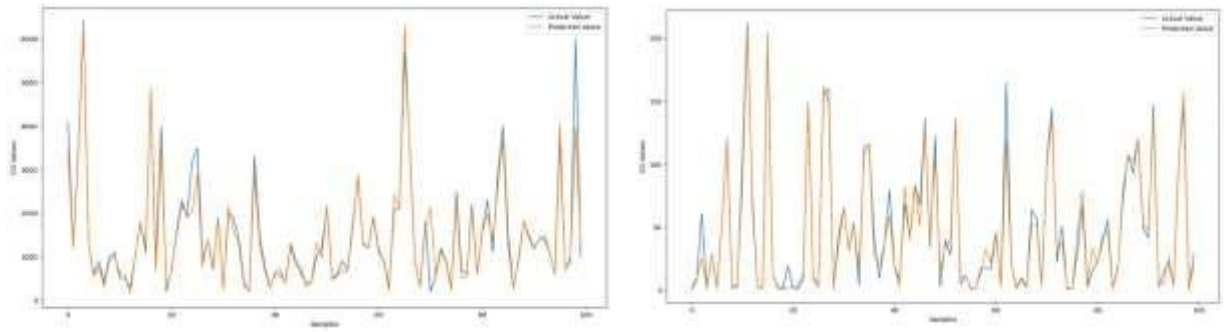


(c) SO2 Concentration



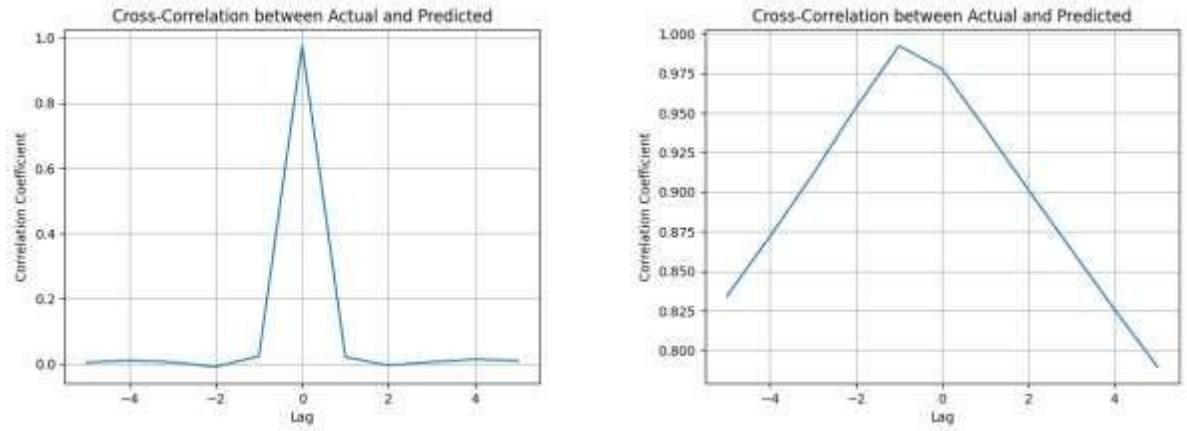
(d) NO2 Concentration

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://www.theaspd.com/ijes.php



(e) CO Concentration (f) O3 Concentration

Fig. 9. Prediction Results of 100 Samples



(a) Shuffling Data (b) Without Shuffiling Data

Fig. 10. Cross-Correlation of the Model

5.2.5 Iraqi Dataset

It was difficult to obtain Iraqi data as sequential and integrated data. Only one pollutant (PM.2.5) was able to be obtained from this authority IQAir ("IQAir | First in Air Quality,") by contacting them. They provided just the concentration of PM2.5 (this is the only air quality and meteorology data they have) for Baghdad from 01/Jan/2020 to 28/Feb/2023 as hourly data.

The proposed model trained on this dataset after customizing input features to predict the PM2.5 concentration, the description of these datasets after preprocessing data is highlighted in Table 9. The results obtained from evaluation metrics and loss values are in Table 10 and Table 11, respectively.

Table 9 Description of Datasets.

Dataset	Parameters	Instances	Attributes	Period	Location	Type
Beijing	PM2.5, PM10, SO2, NO2, CO, O3, TEMP,	35064	16	01/3/2013 to 28/2/2017	Beijing	Hourly

IQAir	PRES, DEWP, RAIN, WD, WSPM PM2.5	16433	5	01/3/2020 to 28/2/2023	Baghdad	Hourly

Table 10 Evaluation Metrics Results of Datasets.

Dataset	R^2	RMSE	MAE
Beijing	0.951	17.878	9.321
IQAir	0.820	20.550	9.575

Table 11 Loss Values of Different Datasets.

Pollutant	Train Loss	Val Loss	Test Loss	Training Time (mm)	Best Training Epoch
Beijing	9.566	9.206	8.842	13	43
IQAir	10.068	9.865	9.094	5	41

Obviously, the Beijing results are outperformed because it has a good amount of data (35064 instances) and besides that it provides data of air quality and meteorological parameters that affect PM2.5 concentration. On the other hand, IQAir has less amount of data (16433 instances) and it lacks other parameters that affect PM2.5 concentration. So the amount of data is a fundamental and important factor in the accuracy and efficiency of air quality prediction, along with the parameters that affected the predicted pollutant. Furthermore, the proposed model can work on different datasets.

5.2.6 State-of-the-art Models Comparison

The proposed model CNN-BiLSTM-Attention is compared with the state-of-the-art models that used the same dataset (Beijing Multi-Site Air-Quality Data). The prediction of PM2.5 concentration and O3 concentration are pointed out in Table 12 and Table 13. The comparison of the proposed model with other models is as follows one after another:

Table 12 Beijing Dataset Comparison for PM2.5 Concentration.

Model	R^2	RMSE	MAE
CNN-BiLSTM-Attention	**0.95**	17.87	**9.32**
TPA-CNN-LSTM (Zhou et al., 2023)	-	20.65	11.09
GCN-BiLSTM (Gao et al., 2023)	-	20.67	9.42
CNN-BiLSTM-GRU (Hu et al., 2023)	**0.95**	**17.20**	9.80

Table 13 Beijing Dataset Comparison for O3 Concentration.

Model	R^2	RMSE	MAE
CNN-BiLSTM-Attention	**0.95**	**11.53**	**7.04**

CNN-BiLSTM-GRU (Hu et al., 2023)	0.95	13.43	8.99

TPA-CNN-LSTM model:
The results for both RMSE and MAE evaluation metrics indicate that the outperform of the proposed model that presents RMSE of 17.802 is significantly better than the 20.65 of the TPA-CNN-LSTM. This implies that the proposed model predictions are generally closer to the observed values. And for MAE of 9.423, the proposed model surpasses the TPA-CNN-LSTM model which has MAE of 11.09. This indicates that on average, the proposed model tends to deviate less from the true values compared to TPA-CNN-LSTM.

The main difference in the architecture of these two models is the type of attention mechanism. The proposed model used Self-Attention while the TPA-CNN-LSTM used Temporal Attention. The use of Self-Attention allows the proposed model to weigh different time intervals or features based on their relevance throughout the entire dataset, which is not considered in Temporal Attention. Thus, it might provide flexibility and broader context to be the key factors behind its enhanced performance. Also, TPA-CNN-LSTM used the standard LSTM, whereas the proposed model used bidirectional approach via BiLSTM. This likely offers a more comprehensive view of the temporal patterns in the data, which, combined with self-attention, provides a detailed and broad understanding of the input sequence.

GCN-BiLSTM:
This model does not include in related work that is use GCN instead of CNN, the GCN-BiLSTM model is proposed by Gao et al. (2023) [20]. Graphical Convolutional Network (GCN) is used to capture the spatial correlation between monitoring stations and assign weights based on distance, whereas the BiLSTM is utilized to extract temporal features. This model is employed to predict the hourly PM2.5 concentration in Beijing - China. According to evaluation metrics, the proposed model appears to outperform the GCN-BiLSTM in terms of RMSE, that indicates it makes fewer large errors in prediction. Even so, in terms of MAE, the proposed mode has slightly better result. For spatial extraction, the proposed model uses CNN as a grid structure that each layer has parameters learned during training, while GCN-BiLSTM uses GCN as a graph structure that each layer updates node representations based on their neighbors to provide deeper insights, but this increases the complexity of the model. On the other hand, CNN has more flexibility, making it more suitable for air quality data. However, GCN-BiLSTM does not use the attention mechanism whereas the proposed model uses it to enhance the model's capability by prioritizing information.

CNN-BiLSTM-GRU:
Regarding PM2.5 prediction, the proposed model and CNN-BiLSTM-GRU have slight differences in the evaluation metrics, while for O3 prediction, the proposed model outperformed the CNN-BiLSTM-GRU model in RMSE and MAE metrics. The proposed model demonstrated superior predictive accuracy, with a lower RMSE of 11.538 compared to 13.43, and a lower MAE of 7.048 compared to 8.99. These results suggest that the attention mechanism may have provided a critical edge by effectively capturing temporal dependencies that the GRU layers did not encapsulate as effectively.

It is evident that the proposed model provides more accurate and reliable predictions for the given dataset. Besides that, it has automated prediction for the six pollutants (PM2.5, PM10, SO2, NO2, CO, O3), whereas other models even do not have the capability to predict all of them.

5.3 Experimental Results of Multi-Step Prediction

In this section, the results of the testing phase using evaluation metrics (R2, RMSE, MAE) for multi-step prediction are analyzed and discussed. There are several strategies for multi-step prediction, these three strategies recursive, direct and MIMO strategy, are compared predicting PM2.5 concentration as follows:

Recursive strategy, uses the same model of the single-step prediction and returns back the output prediction as input for the next time step prediction. Direct strategy, is used to predict one specific time step, so three models have created to predict three time steps, each model is for one time step and the evaluation results are shown in Table 14. MIMO strategy, one model has been created with the number of neuron output as same as the number of time step required and the evaluation results as shown in Table 15. Moreover, the loss values are in Table 16. The direct strategy achieved better results than the rest and could be train for more epochs than MIMO strategy, therefore, it is applied in the proposed model.

Table 14 Direct Strategy Evaluation for PM2.5 Prediction.

Step	R^2	RMSE	MAE
Step 1	0.951	17.878	9.321
Step 2	0.879	28.405	14.816
Step 3	0.860	30.426	17.538

Table 15 MIMO Strategy Evaluation for PM2.5 Prediction.

Step	R2	RMSE	MAE
Step 1	0.954	17.217	9.744
Step 2	0.883	27.968	14.376
Step 3	0.866	29.750	16.931
Average	0.901	25.585	13.684

Table 16 Loss Values of Strategies.

Pollutant	Train Loss	Val Loss	Test Loss	Training Time (mm)	Best Training Epoch
Direct Step 1	9.566	9.206	8.842	13	43
Direct Step 2	14.982	14.839	14.330	18	33
Direct Step 3	20.911	19.633	17.049	19	37
MIMO	14.405	14.350	13.197	19	28

6. CONCLUSIONS

This paper proposed a hybrid CNN-BiLSTM-Attention model to automatically predict the concentration of six pollutants (PM2.5, PM10, SO2, NO2, CO, O3). The weight transfer approach makes the prediction process automated by transferring the weights of each pollutant to the single base model instead of multiple models, thereby providing flexibility in model management, faster prediction and less memory size. Huber loss function effectively balances between the normal and outlier values, that is important for the type of air quality data. Shuffling data leads to loss of temporal dependencies, whereas keeping it without shuffle leads to overfitting and not aligning prediction on the actual output value. Therefore, only shuffling the batches of data after data windowing solves both these issues.

The amount of the dataset plays a significant role in prediction accuracy, also air quality and meteorology data that affected pollutants have an important role in the model's performance. Therefore, the accuracy and performance of the model require a sufficient dataset containing the required predicted pollutants, all parameters that affected these pollutants and rich temporal granularity, that allows the model to extract complex patterns of air quality data. The proposed model used Beijing dataset and obtained good results as it is a regression forecasting type for all evaluation metrics and surpasses some of the state-of-the-art models of the same dataset.

Future endeavors in this domain would benefit from building upon the findings of this study. However, there are some limitations to this work, which are: there is a need for more data resources that impact the concentrations of specified pollutants and multi-step prediction needs to improve by using a dataset for a longer period.

The future exploration is using additional data resources alongside air quality and meteorological data, such as data from satellite imagery, industrial emissions, traffic patterns and unforeseen events like wildfires, that might enhance the model's predictive capabilities. Also aim to implement long-term prediction, more than three time steps with favorable results.

REFERENCES

1. Air pollution [WWW Document], n.d. URL https://www.who.int/health-topics/air-pollution#tab=tab_1 (accessed 7.7.23).
2. Barrow, D., Kourentzes, N., Sandberg, R., Niklewski, J., 2020. Automatic robust estimation for exponential smoothing: Perspectives from statistics and machine learning. Expert Syst Appl 160. https://doi.org/10.1016/j.eswa.2020.113637
3. Beijing Multi-Site Air-Quality Data Set [WWW Document], n.d. URL https://www.kaggle.com/datasets/sid321axn/beijing-multisite-airquality-data-set (accessed 11.20.23).
4. Bekkar, A., Hssina, B., Douzi, S., Douzi, K., 2021. Air-pollution prediction in smart city, deep learning approach. J Big Data 8. https://doi.org/10.1186/s40537-021-00548-1
5. Bhanja, S., Das, A., 2021. A hybrid deep learning model for air quality time series prediction. Indonesian Journal of Electrical Engineering and Computer Science 22, 1611–1618. https://doi.org/10.11591/ijeecs.v22.i3.pp1611-1618
6. Ciampiconi, L., Elwood, A., Leonardi, M., Mohamed, A., Rozza, A., 2023. A survey and taxonomy of loss functions in machine learning 1. https://doi.org/10.48550/arXiv.2301.05579
7. Dai, H., Huang, G., Wang, J., Zeng, H., Zhou, F., 2021. Prediction of air pollutant concentration based on one-dimensional multi-scale cnn-lstm considering spatial-temporal characteristics: A case study of Xi'an, China. Atmosphere (Basel) 12. https://doi.org/10.3390/atmos12121626
8. Dairi, A., Harrou, F., Khadraoui, S., Sun, Y., 2021. Integrated Multiple Directed Attention-Based Deep Learning for Improved Air Pollution Forecasting. IEEE Trans Instrum Meas 70. https://doi.org/10.1109/TIM.2021.3091511
9. Das, B., Dursun, Ö.O., Toraman, S., 2022. Prediction of air pollutants for air quality using deep learning methods in a metropolitan city. Urban Clim 46. https://doi.org/10.1016/j.uclim.2022.101291
10. Deng, T., Manders, A., Segers, A., Bai, Y., Lin, H.X., 2021. Temporal transfer learning for ozone prediction based on CNN-LSTM model, in: ICAART 2021 - Proceedings of the 13th International Conference on Agents and Artificial Intelligence. SciTePress, pp. 1005–1012. https://doi.org/10.5220/0010301710051012

11. Dey, P., Chaulya, S.K., Kumar, S., 2021. Hybrid CNN-LSTM and IoT-based coal mine hazards monitoring and prediction system. Process Safety and Environmental Protection 152, 249–263. https://doi.org/10.1016/j.psep.2021.06.005
12. Du, S., Li, T., Yang, Y., Horng, S.J., 2021. Deep Air Quality Forecasting Using Hybrid Deep Learning Framework. IEEE Trans Knowl Data Eng 33, 2412–2424. https://doi.org/10.1109/TKDE.2019.2954510
13. Gao, L., Liao, M., Zhang, D., 2023. Multi-site air quality prediction based on graph convolutional neural network-bi-directional LSTM model, in: Fifth International Conference on Computer Information Science and Artificial Intelligence (CISAI 2022). Proc. SPIE 12566. https://doi.org/10.1117/12.2667705
14. Gayakwad, M., Patil, S., Kadam, A., Joshi, S., Kotecha, K., Joshi, R., Pandya, S., Gonge, S., Rathod, S., Kadam, K., Shelke, M., 2022. Credibility Analysis of User-Designed Content Using Machine Learning Techniques. Applied System Innovation 5. https://doi.org/10.3390/asi5020043
15. Gonzalez, S., 2020. Improving Deep Learning Through Loss-Function Evolution (Ph.D. Dissertation). THE UNIVERSITY OF TEXAS, AUSTIN.
16. Hu, J., Chen, Y., Wang, W., Zhang, S., Cui, C., Ding, W., Fang, Y., 2023. An optimized hybrid deep learning model for PM2.5 and O3 concentration prediction. Air Qual Atmos Health 16, 857–871. https://doi.org/10.1007/s11869-023-01317-0
17. IQAir | First in Air Quality [WWW Document], n.d. URL https://www.iqair.com/ (accessed 8.23.23).
18. Jiang, P., Bychkov, I., Liu, J., Hmelnov, A., 2021. Predicting of air pollutant concentrations based on spatio-temporal attention convolutional LSTM networks, in: Advanced Information and Computation Technologies and Systems. CEUR Workshop Proceedings, pp. 83–90.
19. Keren, G., 2019. Neural Network Supervision: Notes on Loss Functions, Labels and Confidence Estimation (Ph.D. Dissertation). University of Passau.
20. Kerkhof, M., Wu, L., Perin, G., Picek, S., 2023. No (good) loss no gain: systematic evaluation of loss functions in deep learning-based side-channel analysis. J Cryptogr Eng. https://doi.org/10.1007/s13389-023-00320-6
21. Kim, J., Wang, X., Kang, C., Yu, J., Li, P., 2021. Forecasting air pollutant concentration using a novel spatiotemporal deep learning model based on clustering, feature selection and empirical wavelet transform. Science of the Total Environment 801. https://doi.org/10.1016/j.scitotenv.2021.149654
22. Lazzeri, Francesca., 2021. Machine learning for time series forecasting with Python, 1st ed. John Wiley & Sons, Inc.
23. Li, R., Ye, X., Yang, F., Du, K.L., 2023. ConvLSTM-Att: An Attention-Based Composite Deep Neural Network for Tool Wear Prediction. Machines 11. https://doi.org/10.3390/machines11020297
24. 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 (Switzerland) 10. https://doi.org/10.3390/app10061953
25. Mahadik, S., 2023. Air Quality Forecasting Using Deep Learning Framework. Int J Res Appl Sci Eng Technol 11, 6578–6583. https://doi.org/10.22214/ijraset.2023.53176
26. Masood, A., Ahmad, K., 2021. A review on emerging artificial intelligence (AI) techniques for air pollution forecasting: Fundamentals, application and performance. J Clean Prod 322. https://doi.org/10.1016/j.jclepro.2021.129072

27. Méndez, M., Merayo, M.G., Núñez, M., 2023. Machine learning algorithms to forecast air quality: a survey. Artif Intell Rev. https://doi.org/10.1007/s10462-023-10424-4
28. Mengara, A.G.M., Kim, Y., Yoo, Y., Ahn, J., 2020. Distributed deep features extraction model for air quality forecasting. Sustainability (Switzerland) 12, 1–19. https://doi.org/10.3390/su12198014
29. Mengara Mengara, A.G., Park, E., Jang, J., Yoo, Y., 2022. Attention-Based Distributed Deep Learning Model for Air Quality Forecasting. Sustainability (Switzerland) 14. https://doi.org/10.3390/su14063269
30. Niu, Z., Zhong, G., Yu, H., 2021. A review on the attention mechanism of deep learning. Neurocomputing 452, 48–62. https://doi.org/10.1016/j.neucom.2021.03.091
31. Peixeiro, M., 2022. Time Series Forecasting in Python. Manning Publications Co., Shelter Island.
32. Saini, T., Chaturvedi, P., Dutt, V., 2021. Modelling Particulate Matter Using Multivariate and Multistep Recurrent Neural Networks. Front Environ Sci 9. https://doi.org/10.3389/fenvs.2021.752318
33. Samal, K.K.R., Panda, A.K., Babu, K.S., Das, S.K., 2021. An improved pollution forecasting model with meteorological impact using multiple imputation and fine-tuning approach. Sustain Cities Soc 70. https://doi.org/10.1016/j.scs.2021.102923
34. Shah, S.R. Bin, Chadha, G.S., Schwung, A., Ding, S.X., 2021. A Sequence-to-Sequence Approach for Remaining Useful Lifetime Estimation Using Attention-augmented Bidirectional LSTM. Intelligent Systems with Applications 10–11. https://doi.org/10.1016/j.iswa.2021.200049
35. Subramaniam, S., Raju, N., Ganesan, A., Rajavel, N., Chenniappan, M., Prakash, C., Pramanik, A., Basak, A.K., Dixit, S., 2022. Artificial Intelligence Technologies for Forecasting Air Pollution and Human Health: A Narrative Review. Sustainability (Switzerland) 14. https://doi.org/10.3390/su14169951
36. Tsokov, S., Lazarova, M., Aleksieva-Petrova, A., 2022. A Hybrid Spatiotemporal Deep Model Based on CNN and LSTM for Air Pollution Prediction. Sustainability (Switzerland) 14. https://doi.org/10.3390/su14095104
37. Wang, J., Li, J., Wang, X., Wang, T., Sun, Q., 2022. An air quality prediction model based on CNN-BiNLSTM-attention. Environ Dev Sustain. https://doi.org/10.1007/s10668-021-02102-8
38. Wang, Q., Ma, Y., Zhao, K., Tian, Y., 2022. A Comprehensive Survey of Loss Functions in Machine Learning. Annals of Data Science 9, 187–212. https://doi.org/10.1007/s40745-020-00253-5
39. Wu, Z., Zhao, W., Lv, Y., 2022. An ensemble LSTM-based AQI forecasting model with decomposition-reconstruction technique via CEEMDAN and fuzzy entropy. Air Qual Atmos Health 15, 2299–2311. https://doi.org/10.1007/s11869-022-01252-6
40. Zaini, N., Ean, L.W., Ahmed, A.N., Malek, M.A., 2022. A systematic literature review of deep learning neural network for time series air quality forecasting. Environmental Science and Pollution Research 29, 4958–4990. https://doi.org/10.1007/s11356-021-17442-1
41. Zhang, B., Rong, Y., Yong, R., Qin, D., Li, M., Zou, G., Pan, J., 2022. Deep learning for air pollutant concentration prediction: A review. Atmos Environ 290. https://doi.org/10.1016/j.atmosenv.2022.119347
42. Zhang, J., Peng, Y., Ren, B., Li, T., 2021. Pm2.5 concentration prediction based on cnn-bilstm and attention mechanism. Algorithms 14. https://doi.org/10.3390/A14070208
43. Zhou, Z., Liu, X., Yang, H., 2023. PM 2.5 Concentration Prediction Method Based on Temporal Attention Mechanism and CNN-LSTM. Academic Journal of Science and Technology 5. https://doi.org/10.54097/ajst.v5i3.8009