Air Quality Prediction Using a U-Net Inspired 1D-CNN with Attention Mechanisms

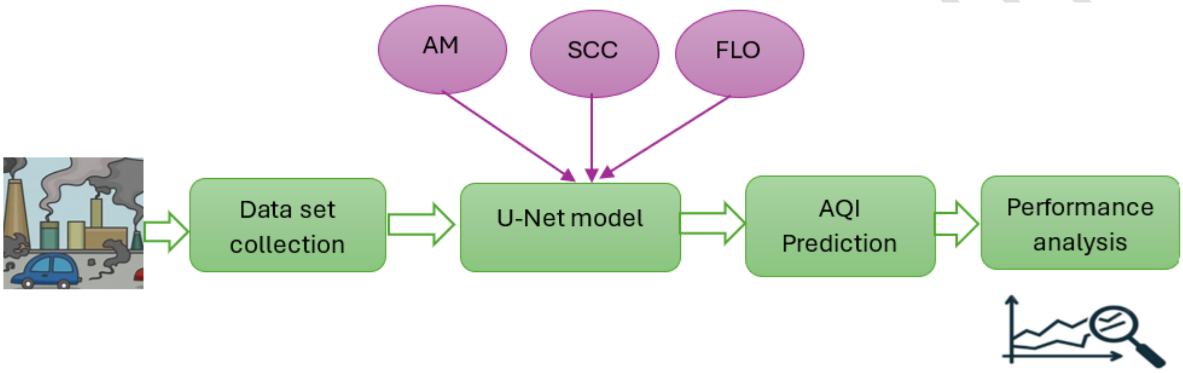
Kanimozhi M[1*]& Malathy P[2]

[1]Assistant professor, Dept of CSE, SRM Madurai College for Engineering and Technology, pottapalyam,sivagangai, kanimozhi@srmmcet.edu.in

[2]Assistant professor, Dept of EEE, PSNA College of Engineering and Technology, Dindigul, Malathyme@psnacet.edu.in

*Corresponding author: M.Kanimozhi, email: kanimozhi@srmmcet.edu.in

Graphical Abstract



Abstract

Air quality prediction is crucial for environmental monitoring and public health. This work presents a new prediction model called Attentive Self-Calibrated Optimized U-Net (ASCO-UNet) for air quality prediction using a one-dimensional convolutional neural network (1D- CNN). The proposed model resembles U-Net architecture which is used in image segmentation tasks popularly. The proposed model uses the strength of U-Net's encoder-decoder design to capture both local and global features effectively. To enhance the model's performance, we integrate attention mechanisms to focus on the most relevant features. In addition, the self-calibrated convolutions are applied to adjust the convolutional filters to improve feature representation. The parameters of the proposed model are fine-tuned using the Frilled Lizard Optimization (FLO) algorithm for optimal performance. Experimental results show that the ASCO-UNet outperforms traditional models, including LSTM, GRU, and Transformer-based models with significant improvements in validation loss, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The ASCO-UNet achieves a loss of 1.7263, MAE of 0.9390, MSE of 2.980, and RMSE of 1.7263, outperforming benchmark methods such as Transformer-based models and GRU+Attention.

Keywords: 1D- CNN, Air quality prediction, Frilled Lizard Optimization, self-calibrated convolutions, U-Net architecture.

1 Introduction

Air pollution is the mixing of any chemical, physical or biological agent in an indoor or outdoor atmosphere that changes the quality of the environment [1]. Air quality is mainly correlated to the earth's climate and ecosystems globally. The different origins of air pollution are classified into multiple areas like transport, industrial activities, waste burning, power production, building construction and agriculture. Air pollution is life life-threatening issue across all countries [2].

Air pollution is a critical health which can lead to premature death, exacerbating conditions like asthma and cardiovascular diseases. Cardiovascular diseases include heart attacks and strokes. Air pollution is also affecting the development of children. It can cause low infant birth weight, wheezing, coughing, and shortness of breath [2]. Research continues to reveal new connections between air pollution and various health issues. The figure shows the risk factors arranged based on disability-adjusted life years (DALYs). DALYs is a measure of disease burden. Figure 1 shows that air pollution is a major element for making humans' poor health.

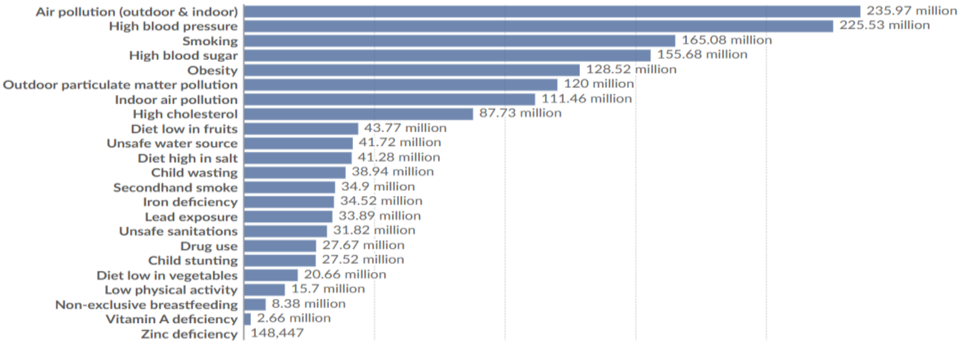


Figure 1: Risk level of air pollution by WHO [3]

To effectively control air pollution, innovative techniques are required which integrate both predictive measures and control strategies. The key elements of these techniques include regulatory measures, technological advancements, public awareness campaigns, predictive modelling, and international cooperation. Predictive modelling is an essential tool in air pollution control strategies. The data from air quality monitoring stations can forecast pollution levels and identify potential pollution events before they occur. The invention of Internet of Things (IoT) based technologies supports continuous data collection in air quality monitoring stations. These stations continuously collect real-time data on various pollutants such as PM2.5, PM10, NO2, SO2, CO, and O3. Then, the predictive models use collected and extracted relevant features for future prediction. This model is based on Machine Learning (ML)and Artificial Intelligence(AI) to analyse patterns and trends [3]. The accurate and timely predictions allow authorities to identify pollution hotspots and implement necessary actions such as traffic restrictions, temporary shutdowns of industrial facilities and public advisories[4].

In this work, an innovative model based on 1D-CNN is proposed for accurate air pollution prediction. The novelty of the ASCO-UNet model lies in the integration of attention mechanisms and self-calibrated convolutions within a U-Net-inspired architecture, enabling enhanced feature representation and adaptability. Additionally, the use of the Frilled Lizard Optimization (FLO) algorithm for parameter tuning ensures optimal performance in air quality prediction tasks. Moreover, the attention mechanism and self-calibration strategies are added to improve the performance further. The remaining sections of the paper are ordered as follows. Section 2 explores related works in air pollution prediction models. Section 3 explains the proposed air quality prediction architecture. Section 4 details the experimental results and section 5 gives the conclusion.

2 Related work

Maleki, H. et al. [5] developed a PM2.5 prediction model using an artificial neural network (ANN). The ANN model included nine factors in the input stage, thirty neurons in the hidden stage, and finally one in the output stage. Similarly, Nikpour, P. et al. [6] developed a Gelato model that combines ANN with XGBoost for air quality prediction.

Y. Han, et al [7] proposed a fusion-based model for air quality prediction. The proposed model uses an attention-layer strategy to learn the influential historical features from input data. The prediction results from different learning models are combined to produce a final prediction result. Z. Qi, et al [8] presented a new feature extraction technique to process the air quality data. The proposed approach extracts spatial-temporal features from pollution data to increase overall prediction accuracy.

In their study, X. Yi, et al [9] developed a short-term air pollution prediction model using a deep neural network. The hyperparameters of the network are tuned using a genetic algorithm. Results show that the deep learning model betters the previously proposed time series model, achieving an average error of 4.52% on the test set. An air quality prediction model based on a recurrent neural network is proposed by K. Gu, et al [10]. The recurrent model considers multiple metrological data for the accurate prediction of air quality. In deep learning, Self-Supervised Learning (SSL) has gained more attention due to the processing capability of unlabeled data to learn useful representations. The Graph Neural network-based model combined with SSL is introduced by J. Han, et al [11] for air pollution forecasting. Compared to another model, the graph-based model effectively learns contextual patterns to improve the prediction accuracy.

Autoencoder-based PM 2.5 forecast model is developed by C. -Y. Lo et al [12]. An autoencoder-based prediction model consists of an autoencoder architecture combined with predictive layers. The autoencoder learns a compressed representation of the pollution data. The experimental results show that the autoencoder-based prediction model is more accurate in forecasting air quality. A three-stage model for air quality prediction was proposed by Sun, M. et al [13]. It involves a backpropagation neural network with swarm optimization for the empirical analysis of the air quality dataset. B. Liu, et al [14] developed a gated recurrent unit (GRU) model for air pollution prediction in Beijing station. Compared to the LSTM model, the GRU model has an additional gate of RESET gate to remove the irrelevant features for further processing. Results show that the GRU model increased the prediction accuracy by 0.2% on the Beijing live dataset. Likewise, L. Wang et al [15] proposed a Bidirectional-GRU model for PM2.5 prediction. Compared to the GRU model, the Bi-GRU model processes the data in both forward and reverse directions and improves the data learning capacity. Federated Learning is a decentralized ML approach that allows multiple models to jointly train a model without sharing their raw data. It enables the aggregation of model updates from different sources to build a robust prediction model. Abimannan et al [16] proposed an air pollutant prediction model based on federated learning and compared it with other models.

The stacking-based LSTM model is suggested by N. Jin et al [17] for air quality forecast. The stacking of the LSTM layer is used to learn the temporal features of air pollution data deeply without any additional complexity. The results show that the daily average value of AQI can be accurately calculated with an error rate of 7.45 and a correlation of 89.6%. Similarly, the LSTM model combined with an autoencoder is introduced by X. Xu et al [18] for PM2.5 prediction. Experiments on datasets show that the proposed model betters other models, increasing performance by 6.2% to 8.5% compared to sole LSTM models.

X. Meng, et al [19] proposed a deep learning model based on a dual attention mechanism for air quality prediction. The attention mechanism in the learning model is used to selectively process the input data based on the priority level. The integration of the attention mechanism considerably reduces mean square error (MSE) from 0.035 to 0.027. The parameter-tuned deep learning model is constructed by J. Wang et al [20] for air quality forecast. The parameters like epochs, learning rate and filter sizes are tuned by a modified particle swarm optimizer. J. Qiao, et al [21] introduce a prediction model using graph convolution. Graph Convolution technique processes and analyses the data as graphs. Unlike CNNs, graph convolution is designed to handle data where the relationships between entities are represented as a graph.

H. A. D. Nguyen et al [22]., propose an LSTM-combined Bayesian neural network for PM2.5 prediction. To improve the spatial correlation extraction property, a new type of activation function is introduced. Periyanan, A. et al [23] proposed a Modified GRU model based on new activation functions. The new activation function includes additional parameters to control the slope for positive and negative values. Sigamani, S. et al [24] presented an air quality prediction model using a Deep Feedforward Neural Network (DFNN). The parameters of the DFNN model were tuned using Fractional Tangent Two-Stage Optimization. Ghufran Isam Drewil et al [25] introduced an LSTM model combined with a Genetic Algorithm (GA). The learning rate of the LSTM model was tuned using GA. Similarly, Srivastava, H. et al [26] developed an LSTM model called Xavier Reptile Switan-h-LSTM to increase the accuracy of air quality prediction. Likewise, Baron, Sam B. et al [27] used an LSTM model with Model-Agnostic Meta-Learning (MAML) to explicitly target air quality parameters.

Q. Shao et al [28] proposed a three-fold prediction model for air quality analysis. Initially, the data is pre-processed with Variational Mode Decomposition. Then, the CNN model is used for feature extraction. Finally, the XGBoost is applied for the final prediction. The proposed model is verified in Nanjing real-time data sets. Results show that the model achieves a 21.89% decrease in RMSE and a 20.05% decrease in MAE.

A hybrid model-based CO forecast model is presented by S. Du, et al [29]. The proposed model includes 1D-CNNs and Bi-directional Long long-term memory networks (Bi-LSTM) to learn the spatial-temporal features from air pollution data. The outcomes show that the CNN+LSTM model has at least 1.1 times lower error rate than the existing models.

Similarly, W. Zheng et al [30] proposed a CNN-combined LSTM model for air quality assessments.

3 Proposed System Architecture

The proposed air quality prediction model is constructed using a 1D-CNN similar to the U-Net architecture. This model is designed to effectively capture both local and global features through its encoder-decoder structure. The proposed model includes attention mechanisms and self-calibrated convolutions for effective feature representation and to increase prediction accuracy. In addition, the FLO is applied for best parameter tuning. The overall architecture is shown in Figure 2.

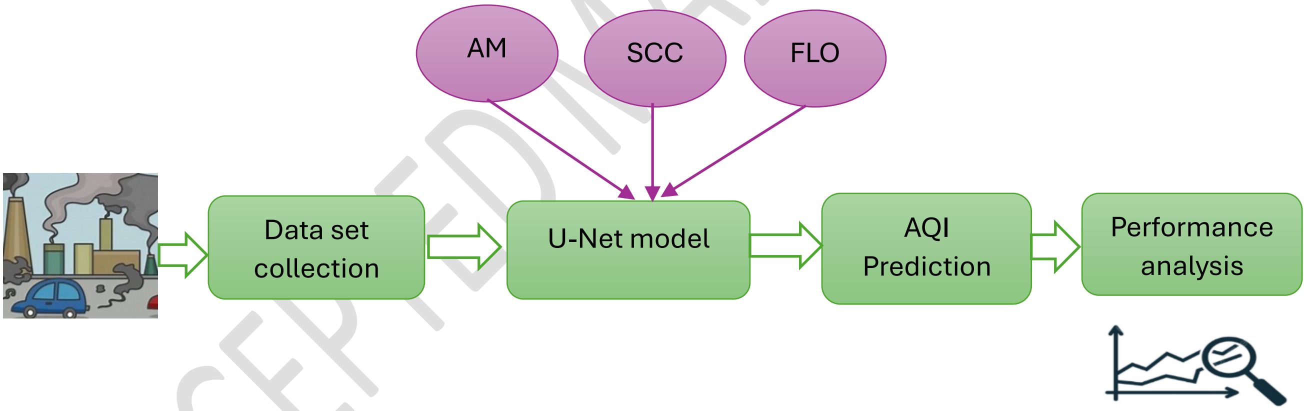


Figure 2: Overall architecture

3.1 Encoder-Decoder Architecture

. The architecture of the proposed Attentive Self-Calibrated Optimized U-Net (ASCO -UNet) is shown in Figure 3 The encoder consists of multiple 1D convolutional layers to reduce the temporal resolution and increase the number of feature maps. Each convolutional layer is followed by a ReLU activation function and a max-pooling layer to downsample the input. The layer receives the raw time-series data. The functions of 1D Convolutional Laye is defined as follows:

$$Y_i = \sigma(\sum_{k=1}^{K} W_k * X_{i+k-1} + b) \tag{1}$$

Where, Y_i is output feature map, X_i is the input feature map, W_k is the k-th convolutional filter, b is the bias term and σ is the activation function (ReLU). The functions of the Max Pooling layer are as follows:

$$Y_i = \max_{j=1\ldots p} X_{i+j-1} \tag{2}$$

Where p is the pooling window size. At the bottleneck, the model captures the most compressed representation of the input data and keeps the essential features for reconstruction.

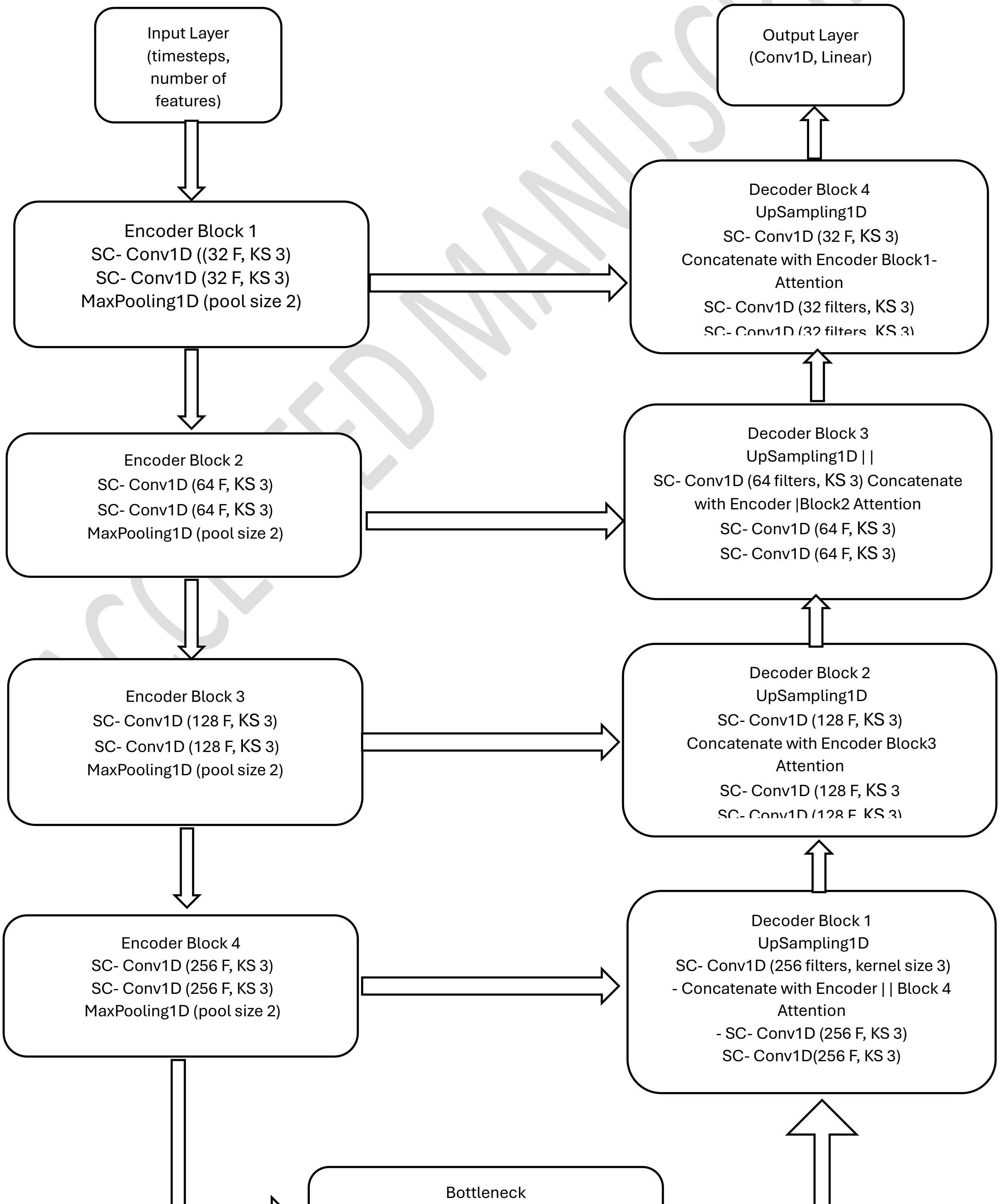


Figure 3: Proposed ASCO -UNet

The decoder mirrors the encoder using upsampling layers and 1D convolutions to reconstruct the temporal resolution. The skip connections from the encoder layers to the corresponding decoder layers are used to retain high-resolution features. The function of upsampling layers is as follows:

$$Y_i = X_{\lceil i/2 \rceil} \tag{3}$$

Where, $\lceil . \rceil$ is the ceiling operation.

3.2 Attention Mechanism(AM)

The attention mechanism used to focus on the most important features of the input sequence, It improves the prediction accuracyW_V by weighing important features more heavily. The operations of attention layers are as follows:

$$Attention\ (Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

Where, Q, K and V refer to the Query, Key, and Value matrices respectively. It can be computed as follows:

$$Q = W_Q X \tag{5}$$

$$K = W_K X \tag{6}$$

$$V = W_V X \tag{7}$$

Where, W_Q, W_K, and W_V are the weight matrices.

3.3 Self-Calibrated Convolutions (SCC)

In existing convolutions, the static filters are applied to the input features. In contrast, self-calibrated convolutions dynamically modify these filters based on the input features themselves. The process of self-calibrated convolutions involves two main steps: generating a calibration feature map and applying this calibrated map to the input features. The operations of self-calibrated convolutions are as follows:

$$F = \sigma(W_1 * X + b_1) \tag{8}$$

$$Y = \sigma(W_2 * (F \odot X) + b_2) \tag{9}$$

Where $\odot$ is the element-wise multiplication.

3.4 Frilled Lizard Optimization (FLO) Algorithm

The Frilled Lizard Optimization (FLO) algorithm is inspired by the hunting and defensive behaviours of the frilled lizard [31]. This type of lizard is found in Northern Australia and spends most of its time in trees. The unique behviour of a frilled lizard is moving bipedally when hunting or escaping predators. During movement, it aligns its head behind its tail base for balancing. This agile predator primarily feeds on insects and invertebrates after spotting prey. It finds optimal solutions based on exploration and exploitation processes. The stage of FLO is divided into three stages: Hunting Strategy, Retreat Strategy and Repetition Process.

Hunting Strategy

In the exploration phase, the algorithm mimics the hunting behavior of frilled lizards. It performs a global search to explore the solution space broadly. The position of candidate prey (CP) for each frilled lizard is expressed as follows:

$$CP_i = \{X_k : F_k < F_i \; and \;\; k \neq i\} \; where \; i = 1,2, \ldots . . N \; and \; k \in \{1,2 \ldots N\} \tag{10}$$

Where, X_k is the population member and F_k is its objective function value. i is the corresponding i th frilled lizard. The new position of frilled lizard $(x_{i,d}^{P1})$ mathematically expressed as follows:

$$x_{i,d}^{P1} = x_{i,d} + rand(PP_{i,d} - Rx_{i,d}) \tag{11}$$

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} < F_i \\ X_i,\ Else \end{cases} \tag{12}$$

Where PP is the selected prey for the ith-frilled lizard. $rand$ is the random number varying from zero to one. R is the number that varies from one to two. d is the dimension of the search space. F_i^{P1} is the objective function.

Exploitation Phase (Retreat Strategy)

In the exploitation phase, the algorithm emulates the retreat behavior of frilled lizards. It performs a local search to explore the new solutions. The new position of frilled lizard ($x_{i,d}^{P2}$) mathematically expressed as follows:

$$x_{i,d}^{P2} = x_{i,d} + (1 - 2rand)\frac{(ul_d - ll_d)}{t} \tag{13}$$

$$X_i = \begin{cases} X_i^{P2}, F_i^{P2} < F_i \\ X_i,\ Else \end{cases} \tag{14}$$

Where and is the random number that varies from zero to one. ul and ll denote the upper and lower limits of the search space. t is the number of iterations.

Repetition Process of FLO

The FLO algorithm iteratively updates the positions of frilled lizards in the solution space using equations (1) to (14). Each iteration updates the positions and maintains the best candidate solution by comparing objective function values. The process continues until the best solution is found.

3.4.1 Parameter tuning using FLO

The parameters of the UNet model are tuned using FLO. Initially, the models are trained using random hyperparameters. The MAE of the model is set as fitness functions to tune the parameters using FLO. The pseudocode of the proposed tuning is given below:

Pseudocode

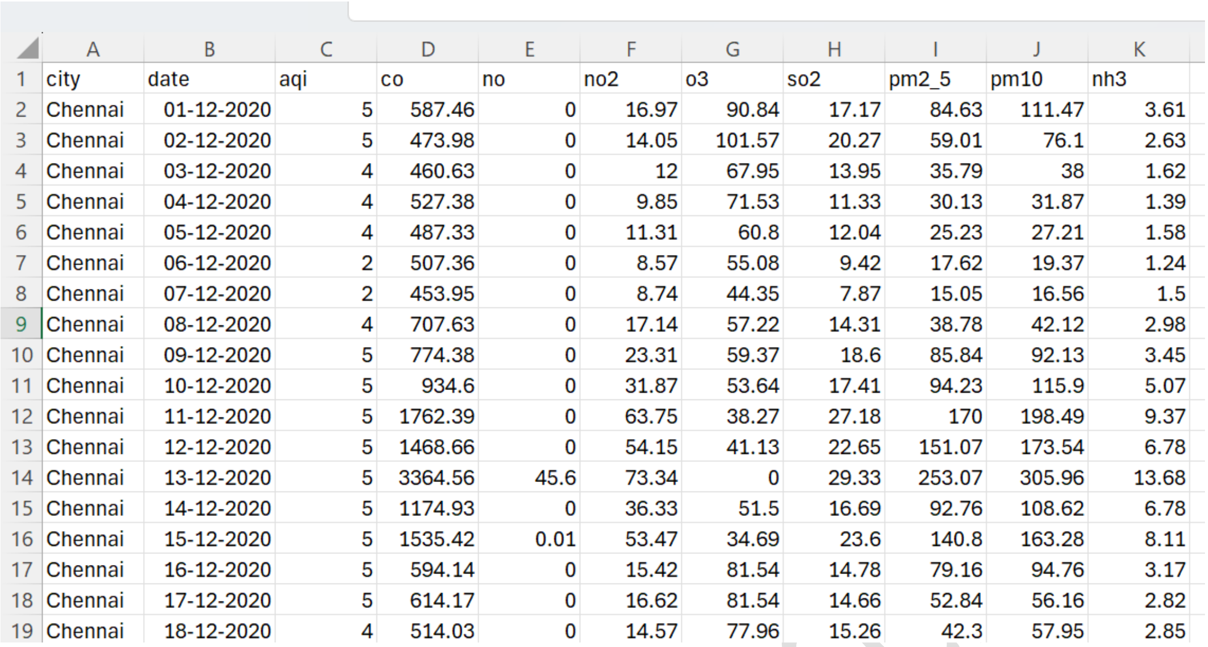
```
Initialize the population of frilled lizards (solutions) with random positions in the search space.
Initialize parameters such as maximum iterations, population size, and bounds of the search space.
Define the objective function to evaluate the fitness of each solution, which includes:
  - Extracting parameters (filters, kernel_size, dropout_rate) from the solution.
  - Building and training the neural network model with the extracted parameters.
  - Evaluating the model performance (validation accuracy) and returning it as the fitness value.
FOR each iteration DO:
  FOR each frilled lizard DO:
    Calculate the fitness of each lizard using the objective function.
    Identify candidate prey (solutions with better fitness) for each lizard.
        // Hunting Strategy (Exploration Phase)
    FOR each frilled lizard DO:
      Select a prey randomly from the candidate prey.
      Update the position of the lizard using:
      new_position = current_position + random_number * (prey_position - random_number * current_position)
      IF the new position is better than the current position THEN
        Update the position of the lizard to the new position.
      END IF
    END FOR
        // Retreat Strategy (Exploitation Phase)
    FOR each frilled lizard DO:
      Update the position of the lizard using:
            new_position = current_position + (1 - 2 * random_number) * (upper_bound - lower_bound) / current_iteration
            IF the new position is better than the current position THEN
                Update the position of the lizard to the new position.
            END IF
        END FOR
    END FOR
        Update the best solution found so far.
END FOR
Return the best solution (best_params).
// Train and evaluate the final model with the best parameters
Extract filters, kernel_size, and dropout_rate from best_params:
    filters = [int(best_params[]), int(best_params[]), int(best_params[]), int(best_params[]), int(best_params[4])]
    kernel_size = [int(best_params[]), int(best_params[]), int(best_params[]), int(best_params[]), int(best_params[])]
    dropout_rate = best_params[]
Build and train the neural network model using the best parameters:
    Define the neural network architecture with the extracted parameters (filters, kernel_size, dropout_rate).
    Train the model on the training data.
    Evaluate the model on the validation/test data.
Return the final model performance.
```

The algorithm starts by initializing a population of frilled lizards. Here, each represents a potential solution in the parameter space of the neural network. Each lizard's position corresponds to the neural network parameters such as filters, kernel sizes, and dropout rates. The algorithm also sets essential parameters like the maximum number of iterations and the size of the population to control the optimization process. The core of the optimization is the objective function which evaluates the quality of each solution (lizard). This function extracts the parameters from each lizard which builds and trains a neural network model using these parameters. Then, assesses the model's performance based on validation accuracy.

The optimization process proceeds through multiple iterations. To refine the search and exploit known good solutions, each lizard updates its position by moving towards a new position determined by a formula involving the bounds of the search space and the current iteration number. After each iteration, the algorithm updates the best solution found so far based on the fitness values of the lizards. This assures that the best-performing solution is tracked throughout the optimization process. A final model is built and trained using these parameters. The model's performance is then evaluated on validation or test data to determine its effectiveness.

4 Experimental results

The dataset consists of air pollution data collected from various cities across India from 2020 to 2023. The data set visualization is shown in Figure 4. It includes several critical columns that detail the air quality conditions in these urban areas. The dataset includes the columns of City**,** Date**,** AQI**,** CO (Carbon Monoxide, NO (Nitric Oxide), NO2 (Nitrogen Dioxide), O3 (Ozone, SO2 (Sulphur Dioxide, PM2.5 (Particulate Matter 2.5), PM10 (Particulate Matter 10) and NH3 (Ammonia)**.** The entire data set is divided into training and test data sets for evaluation purposes.



	A	B	C	D	E	F	G	H	I	J	K
1	city	date	aqi	co	no	no2	o3	so2	pm2_5	pm10	nh3
2	Chennai	01-12-2020	5	587.46	0	16.97	90.84	17.17	84.63	111.47	3.61
3	Chennai	02-12-2020	5	473.98	0	14.05	101.57	20.27	59.01	76.1	2.63
4	Chennai	03-12-2020	4	460.63	0	12	67.95	13.95	35.79	38	1.62
5	Chennai	04-12-2020	4	527.38	0	9.85	71.53	11.33	30.13	31.87	1.39
6	Chennai	05-12-2020	4	487.33	0	11.31	60.8	12.04	25.23	27.21	1.58
7	Chennai	06-12-2020	2	507.36	0	8.57	55.08	9.42	17.62	19.37	1.24
8	Chennai	07-12-2020	2	453.95	0	8.74	44.35	7.87	15.05	16.56	1.5
9	Chennai	08-12-2020	4	707.63	0	17.14	57.22	14.31	38.78	42.12	2.98
10	Chennai	09-12-2020	5	774.38	0	23.31	59.37	18.6	85.84	92.13	3.45
11	Chennai	10-12-2020	5	934.6	0	31.87	53.64	17.41	94.23	115.9	5.07
12	Chennai	11-12-2020	5	1762.39	0	63.75	38.27	27.18	170	198.49	9.37
13	Chennai	12-12-2020	5	1468.66	0	54.15	41.13	22.65	151.07	173.54	6.78
14	Chennai	13-12-2020	5	3364.56	45.6	73.34	0	29.33	253.07	305.96	13.68
15	Chennai	14-12-2020	5	1174.93	0	36.33	51.5	16.69	92.76	108.62	6.78
16	Chennai	15-12-2020	5	1535.42	0.01	53.47	34.69	23.6	140.8	163.28	8.11
17	Chennai	16-12-2020	5	594.14	0	15.42	81.54	14.78	79.16	94.76	3.17
18	Chennai	17-12-2020	5	614.17	0	16.62	81.54	14.66	52.84	56.16	2.82
19	Chennai	18-12-2020	4	514.03	0	14.57	77.96	15.26	42.3	57.95	2.85

Figure 4: Data visualization

Before optimization, the model is initialized with the following hyperparameters: convolutional layers with filter sizes of [32, 64, 64, 64, 128] and kernel sizes of [3, 3, 3, 3, 3], and a dropout rate of 0.2. After optimization using FLO, the final hyperparameters are determined as follows: convolutional layers with filter sizes of [61, 37, 57, 78, 97] and kernel sizes of [3, 6, 6, 5, 4], and a dropout rate of 0.11. These optimized parameters led to improved model performance, as reflected in the reduced mean absolute error (MAE) during validation. This paragraph provides a clear comparison of the parameter values before and after optimization, highlighting the changes and their impact on model performance. Figure 5 shows the fitness evaluation curve of optimization as a function of Mean Absolute Error (MAE) values. It is observed that the optimization process is improving the model's accuracy over time. The proposed model is analyzed using the following metrics. These metrics compare original values with predicted values:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\bar{y} - y)^2 \tag{15}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\bar{y} - y)^2} \tag{16}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\bar{y} - y| \tag{17}$$

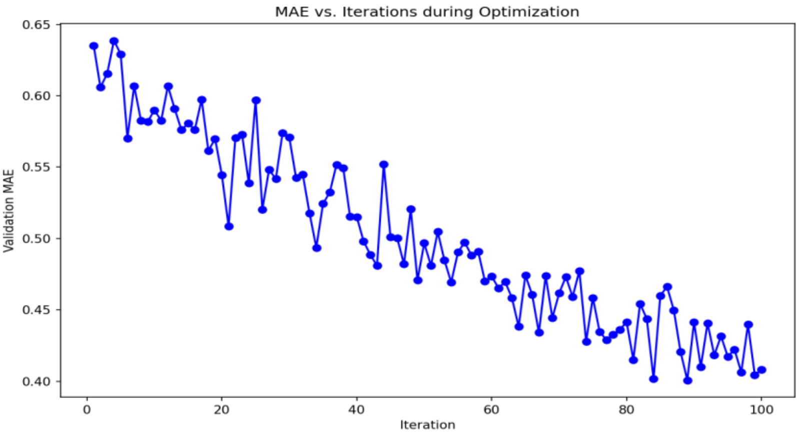


Figure 5: Fitness curve for the optimization process

Figures 6 – 11 show the Loss and MAE values of both training and validation datasets for different architectures. The proposed model is compared with well-known Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU) model and the proposed versions of Attention-based (AB-UNet), In AB-UNeT, Self Calibrated Convolution Based UNet (SCCB-UNet) and Multi-Scale Dense Networks (MSD) -UNet models. The validation loss decreases initially and then slightly higher level than the training loss which denotes some overfitting. Likewise, the model based on MSD shows higher MAE and Loss rates. In the proposed model, the graph shows that the MAE decreases as the number of iterations increases. This indicates that the optimization process is improving the model's accuracy over time. Further, the initial rapid decrease in MAE suggests that the model is quickly learning from the data. Overall, the proposed versions of the UNet model show superior performance compared to all other models. This is evident in both the lower loss and MAE values during validation, suggesting better generalization and less overfitting. The measured performance values are given in Table 1.

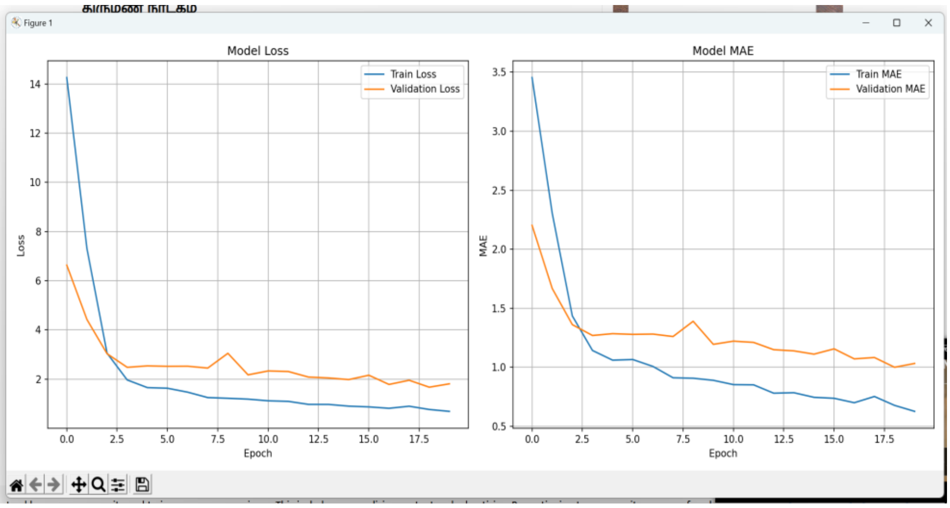


Figure 6: Loss and MAE curve of UNet model

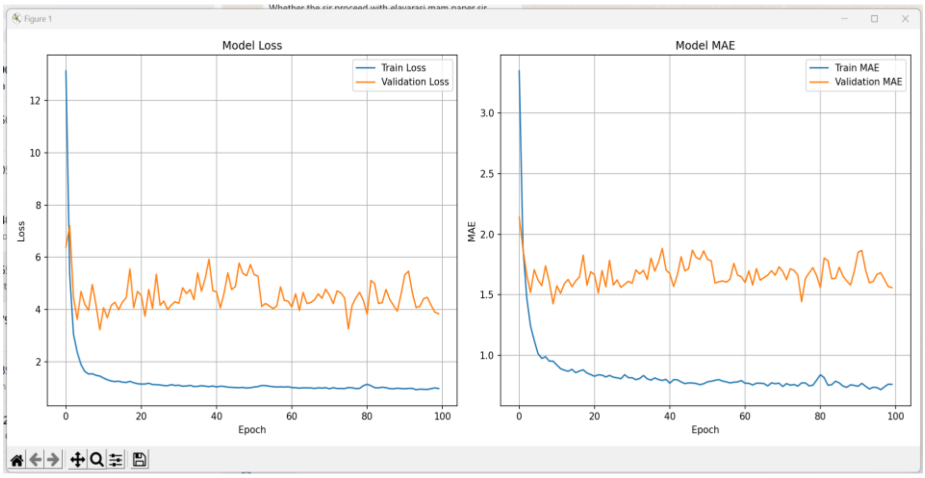


Figure 7: Loss and MAE curve of UNet model with attention

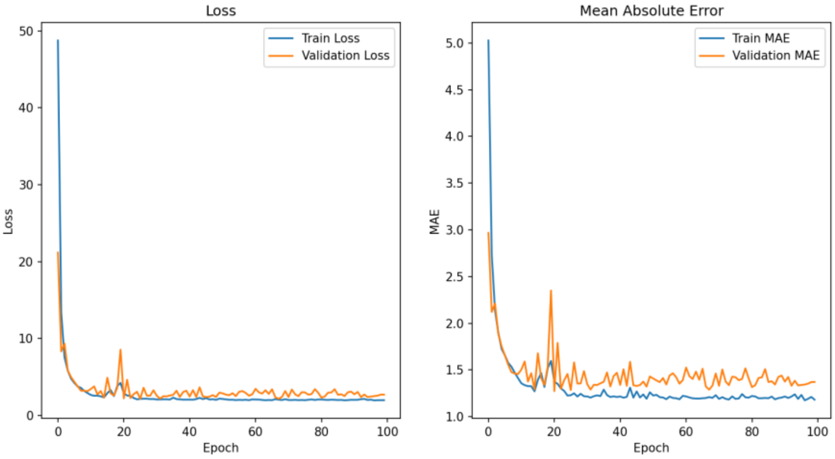


Figure 8: Loss and MAE curve of UNet model with Self calibration

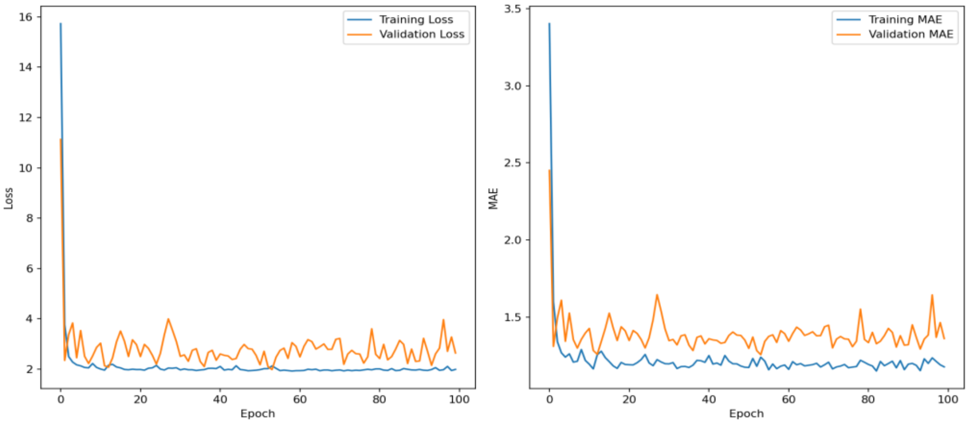


Figure 9: Loss and MAE curve of UNet model with FLO

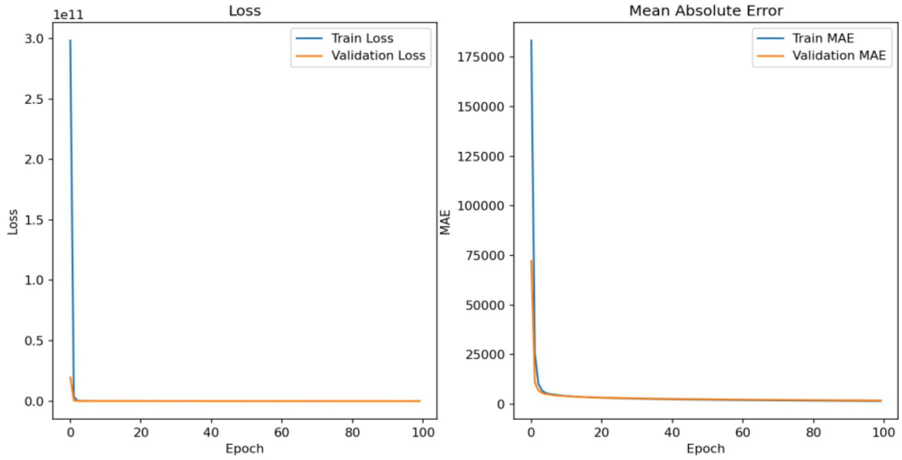


Figure 10: Loss and MAE curve of MSD -UNet

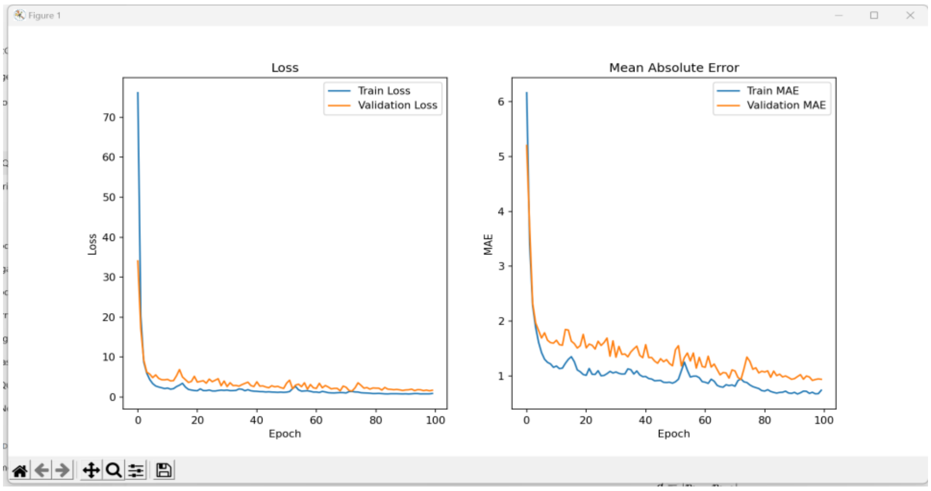


Figure 11: Loss and MAE curve of ASCO -UNet

Table 1: Performance analysis of diffetent prediciotn models.

Method	**Loss**	**MAE**	**MSE**	**RMSE**
UNET	1.8114	1.0329	3.281170	1.8114
LSTM	2.5	1.2	6.25	2.5

GRU	2.2	1.15	4.84	2.2
Attention based (AB-UNet)	3.8337	1.5586	14.69726	3.8337
Self Calibrated Convolution Based UNet (SCCB-UNet)	2.7054	1.3725	7.319189	2.7054
Optimized UNet	1.743268	1.160759	3.038985	1.743268
Multi-Scale Dense Networks (MSD) - UNet	5.3340	2.8675	26.566	6.567
Attentive Self-Calibrated Optimized U-Net (ASCO -UNet)	1.7263	0.9390	2.980112	1.7263

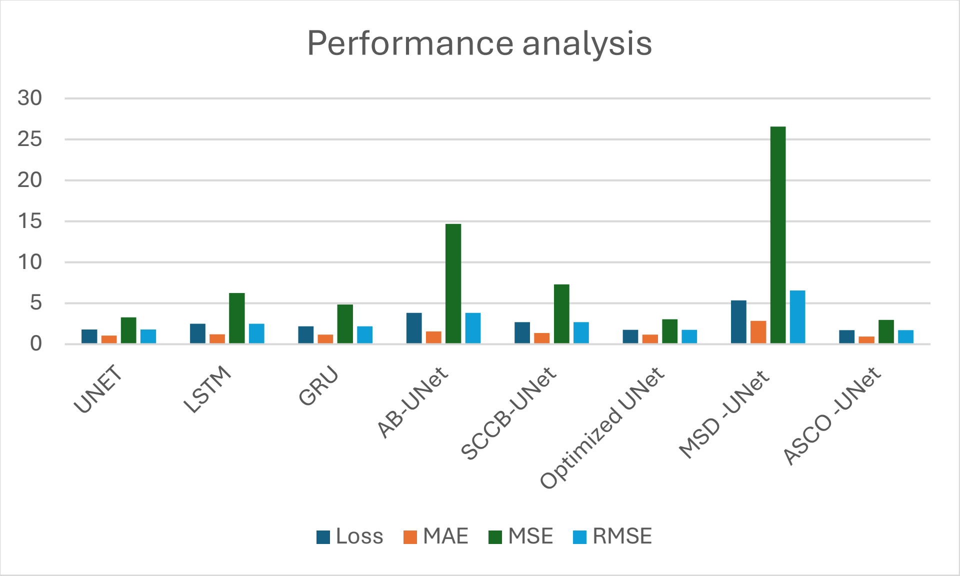


Figure 12: Performance analysis

The U-Net model achieved a loss of 1.8114 and an MAE of 1. The LSTM model showed a higher loss of 2.5 and an MAE of 1.2. The GRU model performed slightly better than LSTM with a loss of 2.2 and an MAE of 1.15. Among the U-Net variations, the AB-UNet shows a significantly higher loss of 3.8337 and an MAE of 1.5586. The SCCB-UNet showed moderate performance with a loss of 2.7054 and an MAE of 1.3725.

The Optimized UNet which includes fine-tuned hyperparameters achieved a lower loss of 1.7433 and an MAE of 1.1608. The MSD-UNet showed the highest loss of 5.3340 and an MAE of 2.8675. The ASCO-UNet integrating attention mechanisms and self-calibrating convolutions optimized through hyperparameter tuning achieved the best performance among all models with a loss of 1.7263 and an MAE of 0.9390. This model's superior results highlight the effectiveness of combining these advanced techniques for capturing complex patterns in the data. The results are graphically shown in Figure 12.

Table 2: Performance analysis of different prediction models.

Method	**Loss**	**MAE**	**MSE**	**RMSE**
ANN	3.4567	1.6754	11.9376	3.4567
ANN + XGBoost	3.1025	1.5489	9.6273	3.1025
DFNN	2.8594	1.4932	8.1749	2.8594

Modified GRU	2.4211	1.3726	6.1211	2.4211
Three-stage model	3.7124	2.5408	6.9154	3.8124
LSTM + GA	2.1019	1.2745	4.8272	2.1019
MAML-LSTM	2.0123	1.2017	4.5238	2.0823
ARIMA [32]	4.3215	2.3514	13.7123	3.7021
LSTM + Attention [33]	2.1765	1.2856	5.2143	2.2834
Transformer-based Model [34]	1.8743	1.1024	3.5829	1.8920
BiLSTM + Attention [35]	1.9123	1.1234	3.7456	1.9345
CNN-LSTM [36]	2.1546	1.3489	4.9567	2.2243
Deep-AIR ([37]	2.0114	1.1678	4.3211	2.0123
ELM (Extreme Learning Machine, [38]	2.5478	1.3245	5.6879	2.3847
GRU + Attention [39]	1.8956	1.2894	3.5987	1.8973
Hybrid Model [32]	2.3465	1.4123	6.1245	2.5124
Bayesian LSTM [40]	2.0478	1.2034	4.4598	2.1143
Self-Attention Transformer [41]	1.9845	1.8873	3.4546	1.9421
ASCO -UNet	1.7263	0.9390	2.980112	1.7263

The performance comparison of the proposed model with the previously proposed model is given in Table 2. The proposed ASCO-UNet model achieves the best performance when compared to all other models. Its effectiveness overcomes other models like ANN, Modified

GRU, and LSTM+GA by using attention mechanisms, self-calibrated convolutions, and the FLO algorithm. These combinations allow the ASCO-UNet to reduce prediction error and increase overall accuracy.

5 Conclusion

In this work, a new prediction model inspired by U-Net architecture with 1D convolutions is proposed to increase the accuracy of air quality prediction. The proposed architecture learns both local and global features. In addition, the model's performance is analyzed by the attention modules and self-calibrated convolutions with parameter tuning. The results show that the proposed model achieved a better result with MAE, MSE, and RMSE values of 0.9390, 2.980, and 1.7263, respectively. The major advantage of the proposed architecture is the proper use of self-calibrated convolutions to adaptively refine feature maps and attention mechanisms to handle critical features. However, the model performance also depends on parameter tuning which shows higher computation cost and limits scalability in resource-constrained environments. Future work can focus on reducing computational overhead by integrating lightweight attention mechanisms and searching unsupervised learning approaches to reduce the dependence on labeled data. Additionally, real-time deployment in IoT systems for air quality monitoring could further enhance its efficiency in practical applications.

References

1. Abelsohn A, Stieb D, Sanborn MD, Weir E (2002) Identifying and managing adverse environmental health effects: 2. Outdoor air pollution. CMAJ 166(9):1161–1167
2. Hossein et al,Health risk assessment of BTEX compounds (benzene, toluene, ethylbenzene and xylene) in different indoor air using Monte Carlo simulation in zahedan city, Iran Kamani, Heliyon, Volume 9, Issue 9, 2029.
3. World Health Organization, "Air quality levels: Global Health Observatory," [Online]. Available: https://www.who.int/data/gho/indicator-metadata-registry/imr-details/158. [Accessed: Dec. 30, 2024].
4. Delavari, M., Beyranvand, F., Jahangiri, M. et al. Increasing the Permeability of Carbon Dioxide and Nitrogen Gases Through a Polymer Membrane Consisting of a Modified

Polyether Block Amide and Experimental Design. J Polym Environ 32, 4822–4841 (2024).

5. Maleki, H., Sorooshian, A., Goudarzi, G. et al. Air pollution prediction by using an artificial neural network model. Clean Techn Environ Policy 21, 1341–1352 (2019).
6. Nikpour, P., Shafiei, M. & Khatibi, V. Gelato: a new hybrid deep learning-based Informer model for multivariate air pollution prediction. Environ Sci Pollut Res 31, 29870–29885 (2024).
7. Y. Han, J. C. K. Lam, V. O. K. Li and Q. Zhang, "A Domain-Specific Bayesian Deep-Learning Approach for Air Pollution Forecast," in IEEE Transactions on Big Data, vol. 8, no. 4, pp. 1034-1046, 1 Aug. 2022, doi: 10.1109/TBDATA.2020.3005368.
8. Z. Qi, T. Wang, G. Song, W. Hu, X. Li and Z. Zhang, "Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-Grained Air Quality," in IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 12, pp. 2285-2297, 1 Dec. 2018, doi: 10.1109/TKDE.2018.2823740.
9. X. Yi, Z. Duan, R. Li, J. Zhang, T. Li and Y. Zheng, "Predicting Fine-Grained Air Quality Based on Deep Neural Networks," in IEEE Transactions on Big Data, vol. 8, no. 5, pp. 1326-1339, 1 Oct. 2022, doi: 10.1109/TBDATA.2020.3047078
10. K. Gu, J. Qiao and W. Lin, "Recurrent Air Quality Predictor Based on Meteorology- and Pollution-Related Factors," in IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 3946-3955, Sept. 2018, doi: 10.1109/TII.2018.2793950.
11. J. Han, H. Liu, H. Xiong and J. Yang, "Semi-Supervised Air Quality Forecasting via Self-Supervised Hierarchical Graph Neural Network," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 5, pp. 5230-5243, 1 May 2023, doi: 10.1109/TKDE.2022.3149815.
12. C. -Y. Lo et al., "Recurrent Learning on PM2.5 Prediction Based on Clustered Airbox Dataset," in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 10, pp. 4994-5008, 1 Oct. 2022, doi: 10.1109/TKDE.2020.3047634.
13. Sun, M., Rao, C. & Hu, Z. Air quality prediction using a novel three-stage model based on time series decomposition. Environ Dev Sustain (2024). https://doi.org/10.1007/s10668-024-04955-1
14. B. Liu, S. Yan, J. Li, Y. Li, J. Lang and G. Qu, "A Spatiotemporal Recurrent Neural Network for Prediction of Atmospheric PM2.5: A Case Study of Beijing," in IEEE Transactions on Computational Social Systems, vol. 8, no. 3, pp. 578-588, June 2021, doi: 10.1109/TCSS.2021.3056410.

15. L. Wang et al., "Short-Term PM2.5 Prediction Based on Multi-Modal Meteorological Data for Consumer-Grade Meteorological Electronic Systems," in IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 3464-3474, Feb. 2024, doi: 10.1109/TCE.2024.3354073.

16. S. Abimannan, J. A, S. Shukla and D. Satheesh, "Federated Learning for Improved Air Pollution Prediction: A Combined LSTM-SVR Approach," 2023 IEEE 4th Annual Flagship India Council International Subsections Conference (INDISCON), Mysore, India, 2023, pp. 1-7, doi: 10.1109/INDISCON58499.2023.10270612.

17. N. Jin, Y. Zeng, K. Yan and Z. Ji, "Multivariate Air Quality Forecasting With Nested Long Short Term Memory Neural Network," in IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8514-8522, Dec. 2021, doi: 10.1109/TII.2021.3065425.

18. X. Xu and M. Yoneda, "Multitask Air-Quality Prediction Based on LSTM-Autoencoder Model," in IEEE Transactions on Cybernetics, vol. 51, no. 5, pp. 2577-2586, May 2021, doi: 10.1109/TCYB.2019.2945999.

19. X. Meng, Q. Liu, C. Yang, L. Zhou and Y. -M. Cheung, "A Novel Deep Learning-Based Robust Dual-Rate Dynamic Data Modeling for Quality Prediction," in IEEE Transactions on Industrial Informatics, vol. 20, no. 2, pp. 1324-1334, Feb. 2024, doi: 10.1109/TII.2023.3275700.

20. J. Wang, Z. Peng, X. Wang, C. Li and J. Wu, "Deep Fuzzy Cognitive Maps for Interpretable Multivariate Time Series Prediction," in IEEE Transactions on Fuzzy Systems, vol. 29, no. 9, pp. 2647-2660, Sept. 2021, doi: 10.1109/TFUZZ.2020.3005293.

21. J. Qiao, Y. Lin, J. Bi, H. Yuan, G. Wang and M. Zhou, "Attention-Based Spatiotemporal Graph Fusion Convolution Networks for Water Quality Prediction," in IEEE Transactions on Automation Science and Engineering, doi: 10.1109/TASE.2023.3285253.

22. H. A. D. Nguyen et al., "Long Short-Term Memory Bayesian Neural Network for Air Pollution Forecast," in IEEE Access, vol. 11, pp. 35710-35725, 2023, doi: 10.1109/ACCESS.2023.3265725.

23. Periyanan A. and Dr. S. Palanivel Rajan (2024), Deep learning-based air pollution prediction model using modified gated recurrent unit, Global NEST Journal, 26(6), 06192.

24. Sigamani, S. (2024). Air quality index prediction with optimisation enabled deep learning model in IoT application. Environmental Technology, 1–17. https://doi.org/10.1080/09593330.2024.2409993

25. Ghufran Isam Drewil and Riyadh Jabbar Al-Bahadili,Air pollution prediction using LSTM deep learning and metaheuristics algorithms,Measurement: Sensors,Volume 24,2022,100546,.

26. Srivastava, H., Kumar Das, S. Air pollution prediction system using XRSTH-LSTM algorithm. Environ Sci Pollut Res 30, 125313–125327 (2023).

27. Baron Sam B, Isaac Sajan R, and Chithra R. SMAML-Enhanced LSTM for Air Quality Time Series Forecasting. Water Air Soil Pollut 235, 783 (2024).

28. Q. Shao, J. Chen and T. Jiang, "A Novel Coupled Optimization Prediction Model for Air Quality," in IEEE Access, vol. 11, pp. 69667-69685, 2023, doi: 10.1109/ACCESS.2023.3293249.

29. S. Du, T. Li, Y. Yang and S. -J. Horng, "Deep Air Quality Forecasting Using Hybrid Deep Learning Framework," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 6, pp. 2412-2424, 1 June 2021, doi: 10.1109/TKDE.2019.2954510.

30. W. Zheng and J. Hu, "Multivariate Time Series Prediction Based on Temporal Change Information Learning Method," in IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 10, pp. 7034-7048, Oct. 2023, doi: 10.1109/TNNLS.2021.3137178.

31. Ibraheem Abu Falahah, Osama Al-Baik, Saleh Alomari, Gulnara Bektemyssova,Frilled Lizard Optimization: A Novel Bio-Inspired Optimizer for Solving Engineering Applications,Computers, Materials and Continua,Volume 79, Issue 3,2024.

32. Duan, J., Gong, Y., Luo, J. et al. Air-quality prediction based on the ARIMA-CNN-LSTM combination model optimized by dung beetle optimizer. Sci Rep 13, 12127 (2023). https://doi.org/10.1038/s41598-023-36620-4

33. Pranolo, A., Zhou, X. & Mao, Y. A novel bifold-attention-LSTM for analyzing PM2.5 concentration-based multi-station data time series. *Int J Data Sci Anal* (2024). https://doi.org/10.1007/s41060-024-00666-y

34. V. Rai, S. Kumar, T. Singh and R. Kapoor, "PM2.5 Level Forecasting using Transformer-Based Model," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2023, pp. 216-221, doi: 10.1109/ICACITE57410.2023.10182822.

35. Wang, J., Li, J., Wang, X. et al. An air quality prediction model based on CNN-BiNLSTM-attention. Environ Dev Sustain (2022). https://doi.org/10.1007/s10668-021-02102-8

36. Gilik, A., Ogrenci, A.S. & Ozmen, A. Air quality prediction using CNN+LSTM-based hybrid deep learning architecture. Environ Sci Pollut Res 29, 11920–11938 (2022).

37. Han &Yang, et al. "Deep-AIR: A hybrid CNN-LSTM framework for air quality modeling in metropolitan cities." arXiv preprint arXiv:2103.14587 (2021).

38. Md. Naseera, E Geethanjali, G Shruthi, KLS Sravya Nikitha. Using deep learning and an enhanced extreme learning machine based on genetic algorithms to forecast air quality index. Int J Cloud Comput Database Manage 2024;5(2):19-23.

39. Hengjun Huang, Chonghui Qian,Modeling PM2.5 forecast using a self-weighted ensemble GRU network: Method optimization and evaluation,Ecological Indicators,Volume 156,2023.

40. Huynh A. D. Nguyen et al., "Long Short-Term Memory Bayesian Neural Network for Air Pollution Forecast," in IEEE Access, vol. 11, pp. 35710-35725, 2023, doi: 10.1109/ACCESS.2023.3265725.

41. W. Cao, W. Qi and P. Lu, "Air Quality Prediction Based on Time Series Decomposition and Convolutional Sparse Self-Attention Mechanism Transformer Model," in IEEE Access, vol. 12, pp. 155340-155350, 2024, doi: 10.1109/ACCESS.2024.3484579.