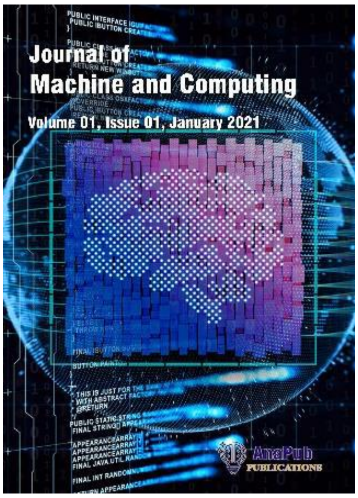
Journal Pre-proof

Mitigating Air Pollution Risks with Deep Learning: A Quantum-Optimized Approach for Nitrogen Dioxide Prediction in Los Angeles



Sivakumaran AR, Cuddapah Anitha, Manjula Arunraj,

Ebinezer M.D.J, Venkatesh Babu S and Gokila S

DOI: 10.53759/7669/jmc202505056

Reference: JMC202505056

Journal: Journal of Machine and Computing.

Received 10 March 2024

Revised form 29 September 2024

Accepted 20 January 2025

Please cite this article as: Sivakumaran AR, Cuddapah Anitha, Manjula Arunraj, Ebinezer M.D.J, Venkatesh Babu S and Gokila S, "Mitigating Air Pollution Risks with Deep Learning: A Quantum-Optimized Approach for Nitrogen Dioxide Prediction in Los Angeles", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505056

Mitigating Air Pollution Risks with Deep Learning: A Quantum-Optimized Approach for Nitrogen Dioxide Prediction in Los Angeles

[1]AR.Sivakumaran
Associate Professor
Department of Information Technology
Malla Reddy Engineering College for Women
Secunderbad-500100
Telangana, India
sivakumaransns@gmail.com

[2]Dr. Cuddapah Anitha,
Associate Professor,
Department of Computer Science and Engineering, School of Computing, Mohan Babu University, (Erstwhile Sree Vidyanikethan Engineering College), Tirupati-517102, Andhra Pradesh.
cuddapah.anitha@mbu.asia

[3]Dr. Manjula Arunraj,
Assistant professor,
Department of Health Information Management and Technology,
College of Applied Medical Sciences,
King Faizal University, Al-Ahsa,
Saudi Arabia
maronraj@kfu.edu

[4]Mr. M. J. D. Ebinezer
Assistant Professor,
Department of CSE,
Koneru Lakshmaiah Education Foundation,
Vaddeswaram, AP, INDIA.
ebinezer.mjd@gmail.com

[5]S. Venkatesh Babu ,
Associate Professor,
Department of Information Technology,
KGISL Institute of Technology,
Coimbatore.
Tamil Nadu, India
venkateshbabu.s@kgkite.ac.in

[6]GOKILA S
Assistant Professor
Department of Information Technology
Karpagam Institute of Technology
Coimbatore, Tamil Nadu
gokilasadhasivan@gmail.com

Abstract

Air pollution causes about seven million premature deaths globally every year, making it a critical issue that requires urgent attention. The key to mitigating its devastating effects lies in understanding its nature, identifying sources and trends, and predicting its. Accurate Real-time air pollution forecasting is a challenging task due to its spatiotemporal dynamics, requiring sophisticated modeling approaches. In our study, employed the Sequential Array-based Convolutional LSTM (SACLSTM) framework, which captures spatial and temporal correlations by integrating deep CNNs for spatial analysis with deep LSTM models for temporal prediction. To further enhance the model's accuracy, optimized the SACLSTM parameters using the Quantum-based Draft Mongoose Optimization Algorithm (QDMOA). Using ten days of nitrogen dioxide (NO_2) data from Los Angeles County, developed a sequential encoder-decoder network capable of predicting air pollution levels ten days into the future. By reformatting satellite air quality images into a 5D tensor, achieved precise predictions of nitrogen dioxide concentrations across various locations and time periods in Los Angeles. Our results are thoroughly documented with metrics and visualizations, clearly demonstrating the factors behind the improved accuracy. The comparison of results highlights the effectiveness of our approach in providing reliable air pollution forecasts.

Keywords: Air pollution; Convolutional Neural Networks; Quantum based Draf Mongoose Optimization Algorithm; Long Short-Term Memory; Los Angeles.

Introduction

Air is vital for human survival, making it imperative to monitor and understand its quality for health preservation. Each year, including approximately 600,000 children [1]. Globally, one person dies prematurely due to air pollution every five seconds [2, 3]. With urban populations expected to rise from 54% in 2015 to 68% by 2050, and up to 89% in the United States, it is essential to develop comprehensive mitigation strategies and forecasting systems to limit exposure to harmful urban air and reduce deaths caused by air pollution [4].

Scientific research identifies air pollution as the greatest environmental risk, with rapid industrialization releasing harmful gases that significantly degrade air quality and threaten public health [5]. Air pollution levels are quantified using the Air Quality Index (AQI), a numerical measure based on pollutants such as NO_2, SO_2, CO, O_3, PM10, PM2.5, NH_3, and benzene [6]. In some applications, the AQI is calculated using six key pollutants: PM10, PM2.5, SO_2, NO_2, CO, and O_3 [7]. Elevated AQI levels indicate severe pollution with detrimental health effects [8]. Real-time AQI data is recorded hourly and daily by meteorological stations, providing valuable input for air quality monitoring [9, 10].

This study utilizes AQI data from Indian cities, which has been mined and analyzed. Three regression analysis methods are employed to identify the most accurate predictive approach [11, 12]. Addressing the spatiotemporal complexity of air quality forecasting is challenging, as prior research primarily focuses on either spatial or temporal correlations [13, 14]. Using ConvLSTM, enables the simultaneous analysis of incorporating both time-based and location-based factors to enhance the prediction and accuracy respectively [15, 16]. GCNs learn feature embeddings from graph structures, while ConvLSTM models process spatial and temporal data, making them suitable for complex air quality predictions [17].

The study introduces the Sequential Array-based Convolutional LSTM (SACLSTM) framework, which combines temporal predictive models (deep LSTM) and spatial predictive models (deep Convolutional Neural Networks). To further enhance classification accuracy, the Quantum-based Draft Mongoose Optimization Algorithm (QDMOA) is implemented to fine-tune SACLSTM parameters. Comprehensive visualizations, metrics, and data analyses demonstrate the model's effectiveness, offering practical insights for mitigating air pollution.

2. Related works

Here's a concise and reformulated version of the provided text:

Yu et al. [18] proposed the Multi-Granularity Transformer, which includes the residual de-redundant block mitigates redundant information that could mislead the model, while the spatiotemporal attention block captures air quality data correlations. The dynamic fusion block combines predictions and assesses the importance of data at different levels of granularity. Experimental results on three datasets demonstrated that the model outperformed 11 baselines by 5%.

Chen et al. [21] introduced analyzing the relationships between monitoring stations. The model constructs multi-scale spatial-temporal graphs and uses a temporal fusion module to capture correlations in both spatial and temporal data. Experiments with datasets from Beijing and Tianjin established the model's superior performance in both single-step and multi-step predictions. Ablation studies validated the importance of the graph and attention mechanisms in improving the model's effectiveness.

Sundaramoorthy et al. [22] developed an advanced air quality prediction system using real-world data from three public sources. After data cleaning, they introduced the Fused Eurasian Oystercatcher-Pathfinder Algorithm (FEO-PFA) for dual optimization, improving feature selection and weight optimization. The refined features were input into the Multiscale Depth-wise Separable Adaptive Temporal prediction. Empirical analyses showed significant improvements, with the proposed model reducing the average cost function by 5.5%, MAE by 28%, and RMSE by 14%, outperforming traditional methods.

3. Proposed Methodology

The suggested methodology for air quality prediction is briefly explained in this part. Figure 1 illustrates it, and the following subsections provide descriptions of each of its blocks.

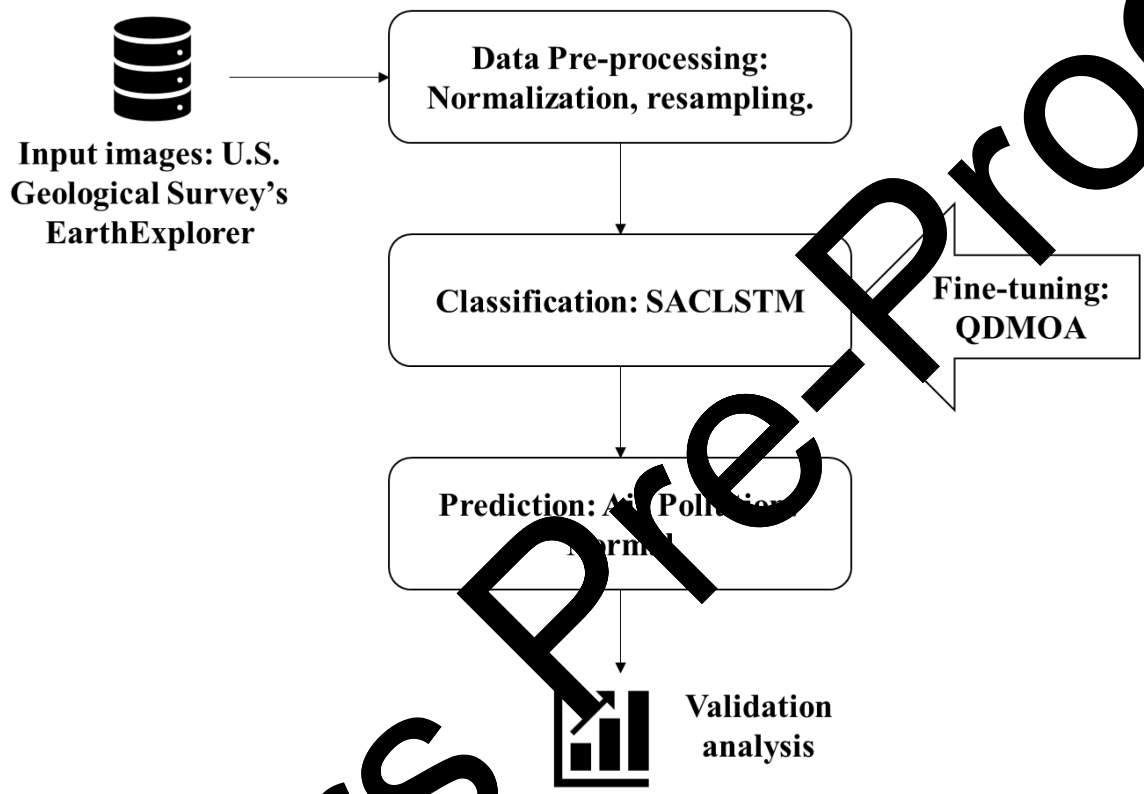


Figure 1: Workflow of the Proposed model

3.1. Dataset

The input data taken from the U.S. Earth Explorer database, utilizing records from the Sentinel-2 satellite, which was launched on 23rd June 2015 [23]. Operated by the European Space Agency since March 2015, Sentinel-2 captures atmospheric and land information using 13 spectral bands, with an orbital swath of 290 km [24].

For this study, two spectral bands were chosen that are pertinent to air pollution. The initial band, centered at a wavelength of 442.7 nm, was used to measure coastal aerosol levels, enabling the observation of fine particulate matter such as dust, smoke, and general particulate matter. The second, a narrower band with a center wavelength of 945.1 nm, was used to measure atmospheric nitrogen dioxide concentrations. An example input is illustrated in Figure 2 [24].

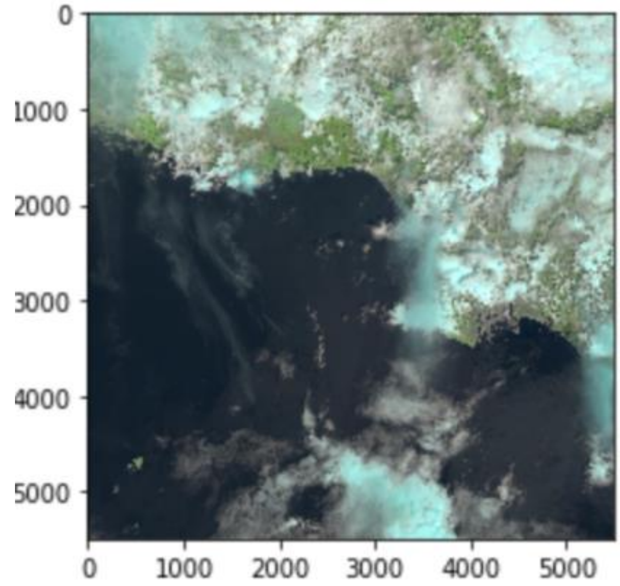


Figure 2. The original data comes from the satellite image of Los Angeles captured on 29th April 2019, by European Space Agency (ESA).

Whereas the clouded white like formations represent particulate matter, the blue structures represent air pollution that is exclusively caused by nitrogen dioxide.

3.2. Data Preprocessing

To prepare data for the proposed model, converted images of 225 highest resolution GeoTIFF into a 5D tensor. These images covered the T11SLT tile, a 100 km × 100 km area that represents approximately 75% of western Los Angeles County, spanning datas of 1642 days. Each image was captured two days apart, corresponding to the Sentinel-2 satellite's orbital period.

For model input, focused exclusively on the blueish, like a cloud structures indicating Nitrogen Dioxide. Using the OpenCV Python library, the GeoTIFF dataset was first converted into JPEG format. Due to the high volume of data, resampled it into two smaller-resolution datasets: one with 400×400 pixel JPEG images and another with 40×40 pixel JPEG images, both comprising all 225 images.

To isolate relevant features, applied a mask targeting light blue hues in the (0,60,60) to (225,255,255) RGB range. All non-light blue regions were masked to black, or the (0,0,0) RGB color. Figure 3 provides an example of a masked image from this process.

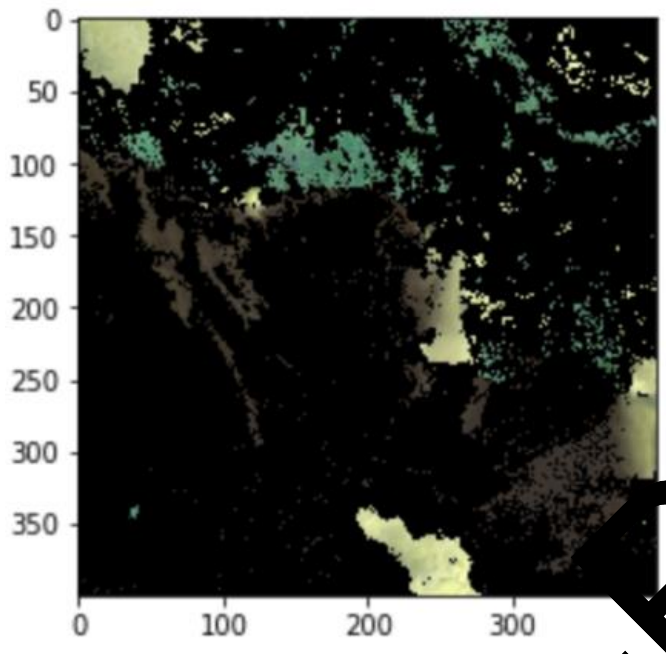


Figure 3. Masked Image.

Created four datasets by resampling the original image collection into two resolutions (40×40 pixels and 400×400 pixels) and two formats (binary and masked RGB images). For the binary datasets, all bright blue pixels were converted to 1, and all black pixels were converted to 0. This resulted in two binary datasets: one at 40×40 pixels and the other at 400×400 pixels. For the masked RGB datasets, the bright blue and black color scheme was retained, resulting in two additional datasets at the same resolutions.

Once the resampling and formatting were complete, the datasets were organized for use in the proposed models. The dataset dimensions were as follows: (225, 400, 400, 3) for 400×400 masked RGB images, (225, 40, 40, 3) for 40×40 masked RGB images, (225, 400, 400, 1) for 400×400 binary images, and (225, 40, 40, 1) for 40×40 binary images. Finally, all datasets were batched by grouping every five image frames into a single sample for further processing.

3.3. Background

This part has the review of CNN and LSTM, which are the key components of the suggested algorithm's framework, before presenting the approach suggested in this article.

3.3.1. CNN

CNNs, a prominent advancement in deep neural networks (DNNs), have become vastly acknowledged for their effectiveness across many like segmentation and detection [25]. CNNs have consistently outperformed traditional machine learning methods in these areas. The structure of CNNs generally consists of several essential elements.

3.3.1.1 Convolutional layer

This is equipped with multiple convolution kernels, is tasked with identifying and extracting important features from the input data. Every element of convolution kernel functions similarly to a neuron in a feedforward neural network, representing a and a bias term. In the convolutional layer, each neuron is connected to a specific region of the previous layer, with the region’s size determined by the convolution kernel, often referred to as the "receptive field." As the convolution kernel processes the input, it systematically scans the features, performing matrix multiplication within the receptive field and summing the results, with the deviations superimposed at each step:

$$Y^{l+1}(c,d) = [Y^l \otimes w^{l+1}](c,d) + b$$
$$= \sum_{K_l}^{f=1} \sum_{f}^{e=1} \sum_{f}^{y=1} \left[Y_l^k (s_0 c + x, s_0 d + y) w_{l+1}^k (e,f) \right] + 1 \quad (1)$$

$(c,d) \in \{0,1,\dots,Z_{l+1}\}, Z_{l+1} = \frac{Z_l + 2q - f}{s_0} + 1$ (2)

The convolution layer, which uses multiple convolution kernels, is tasked with extracting the input data features. Every element of the convolution kernel acts similarly to a neuron in a feedforward neural network, where it represents a weight vector. Neurons in the convolutional layer are connected to a local region of the preceding layer, with the size of this region determined by the convolution kernel, known as the "receptive field". The output at layer l + 1 can be expressed using Eq. (3).

$Y^{l+1} = \sum_{K_l}^{f=1} \sum_{f}^{e=1} \sum_{f}^{y=1} \left(Y_{c,d,k}^l w_{l+1}^k \right) + b = w_T^{l+1} Y_{l+1} + b, Y^{l+1} = Y$ (3)

Equation (4) describes the activation function in the convolutional layer, which helps to capture and represent more complex features in the data.

$A_{c.d.k}^l = f(Y_{c,d,k}^l)$ (4)

Relu, It is defined as Eq. (5):

$f(e) = max(0, e)$ (5)

3.3.1.2 Pooling layer

After the features are extracted, the feature map is forwarded to the pooling layer to select key features and filter relevant information. The pooling layer substitutes a point's value with statistical values derived from its surrounding region from the surrounding area in the feature map. Like the convolutional layer, the pooling layer's region selection depends on factors such as the pooling size, step size, and padding, which dictate how the pooling function scans through the feature map. This process is usually represented by Equation (6).

$A_k^l(c,d) = \left[\sum_{x=1}^{f} \sum_{y=1}^{f} A_k^l (s_0 c + e, s_0 d + f)^p \right]^{\frac{1}{P}}$ (6)

In Equation (6), the step size (s_0) and pixel coordinates (c, d) carry the same meaning as in the convolutional layer, with the step size (s) being a predefined parameter. When p=1p = 1p=1, the process is called average pooling, where the average value within the pooling region is used. Max pooling, on the other hand, selects the maximum value in the pooling area when p→∞p \to \inftyp→∞. These two techniques—mean pooling and max pooling—are commonly used in CNN design, both of which help preserve texture information and the image’s background while reducing feature map size. Typically, strides are set to two, and the pooling filters are 2x2.

In convolutional layers, as the layers are stacked, the feature map size shrinks. For example, if a 5x5 convolution kernel with unit steps and no padding is applied to a 16x16 input image, it will produce a 12x12 feature map. To counteract this size reduction, padding techniques are used to enlarge the feature map. The two most commonly used padding techniques are replication padding and zero padding.

3.3.1.3 Fully connected layer

The CNN layer serves a similar function, once the data undergoes the excitation function, its spatial structure is lost as it is flattened into a vector. The earlier layers, including convolutional, pooling, and activation functions, are responsible for extracting features from the input data.

In essence, the fully connected layer syndicates these extracted features in a nonlinear way to produce the final output. It functions as the "classifier" within the CNN. The previous layers are transformed into a feature space the input data. This layer does not focus on extracting new features but instead utilizes the high-level, already learned features to achieve the final learning goal.

3.3.2 Long short term memory networks

LSTM is one type of time RNN. It was created especially to solve the long-term need problem with the general RNN. It has been successfully used in a variety of fields, including financial time series, video tagging, visual description creation, machine translation, and speech recognition [26]. In all RNNs, the repeating neural network module is accessible in chain form. Its main components are forgetting, input, and output gates.

3.3.2.1 Forgetting Gate

$z_t = \delta\left(E_f.[h_{t-1}, x_t] + b_f\right)$ (7)

The sigmoid function yields zero, some of the data must be forgotten; otherwise, it will continue to be sent within the United States. Eq. (7) includes the current output value x_f current output weight E_f current output bias E_f, and output value of the previous layer h_(t-1).

3.3.2.2 Input gate

$j_t = \delta(E_i.[h_{t-1}, x_t] + b_t)$ (8)

$\tilde{B}_t = tanh(E_B.[h_{t-1}, x_t] + b_B)$ (9)

The gate function is responsible for updating the status of the previous unit in a neural network. Specifically, the forget gate layer determines which information should be retained or discarded. This gate is composed of a combination of a sigmoid function and a tanh function, which work together to regulate what data gets updated in the system's state.

Equations (8) and (9) define this process more precisely. The current output values jtj_tjt and BtB_tBt represent the two components of the input gate.

3.3.2.3 Output gate

$p_t = \delta\left(W_p[h_{t-1}, x_t] + b_p\right)$ (10)

$h_t = p_t * \tanh\,(B_t)$ (11)

The first two gates are primarily responsible for updating the state of the system, while the third gate performs calculations using the information from the previous state. The gate control mechanism determines how much of the state value p_t is exposed to the outside world at any given time p_t. Essentially, it decides what information should be added or removed from the state based on the updated data.

Equations (10) and (11) describe the process more clearly. At p_t, the previous hidden state $ht -$ $1h\_\{t - 1\}$ and the current input xtx_t undergo another transformation using a sigmoid function (referred to as the output gate). This generates an output, p_t, which is then multiplied by the updated state p_t, after being activated.

3.4. The process of the proposed SACLSTM

The SACLSTM model operates with a two-dimensional matrix as its input, where the matrix is determined by the variables generate a historical prediction. If predictions are based on ggg days, the input matrix will have a size of g×fg \times fg×f, where fff represents the number of variables for each day.

To capture the 30-day changes, the SACLSTM applies an initial variable filter for continuous feature extraction. This approach, similar to how CNNs apply a 3×3 filter to images, helps in combining features into a more complex matrix. The layer can also filter out irrelevant variables by setting certain filter weights to zero, acting as a feature selection step. Pooling and convolution operations combine lower-level features from inputs into higher-level features, aggregating data over specific time periods.

The second layer of the network employs 64 filters. The final prediction, by the last pooling layer are reshaped into a feature vector and passed into the LSTM unit for deeper feature extraction the features are generated. The model's output predicts air quality changes for the next day, with the result discretized into one of three values: 0, 1, or -1.

In the experimental setup, the SACLSTM takes a 30-day input with multiple variables for each day. The input matrix for the 2D-CNN part is 30×variables30 \times \text{variables}30×variables.

3.4.1. Fine-tuning using QDMO based optimization

To enhance the presentation of the hybrid LSTM model, QDMO is used to get the optimal hyperparameters for the model. DMO is a stochastic metaheuristic technique inspired by the social and feeding behaviors of the dwarf mongoose (Helogale). These animals tend to forage in groups but may also forage alone. They adopt a seminomadic lifestyle, creating sleeping mounds near food sources. The DMO method utilizes a statistical model based on these behaviors to determine the best course of action for optimization.

Like other population-based optimization techniques, DMO begins with a random initialization phase, where a population of potential within defined bounds. This is followed by intensification and diversification steps that guide the solutions toward the optimal global solution. The DMO method starts by establishing a candidate pool of solutions, randomly generated between the smallest and largest allowed values for the problem. As described in [27], the DMO method simulates the natural foraging habits of dwarf mongooses. The process transitions to the scout group phase, where the latest scout group uses information from previous searches to locate new food sources.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & x_{i,j} & \vdots \\ x_{m,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \quad (12)$$

The value is often strongminded by Eq. (13) as a randomly distributed integer with a consistent distribution that is between (UB) and (LB).

$$\chi_j = unifrnd\ (LB, UB, D) \quad (13)$$

$$a = \frac{fit_i}{\sum_{i=1}^{n} fit_i} \quad (14)$$

In Eq. (4), n is hierarchized by Eq. (15):

$$n = n - bs \quad (15)$$

to revert the X_i solution value, the DMO uses Eq. (16).

$$X_{i+1} = X_i + phi \times peep \quad (16)$$

Eq. (17) the SM to be improved upon.

$$sm_i = \frac{fit_{i+1} - fit_i}{max\{|fit_{i+1}, fit_i|\}} \quad (17)$$

To compute the regular $SM(\varphi)$, assumed as

$$\phi = \frac{\sum_{i=1}^{n} sm_i}{n} \quad (18)$$

In the scouting phase, according to the nomadic behavior of dwarf mongooses, the current position of the SM (sleeping mound) is disregarded in favor of exploring new potential food sources or locations. This process involves simultaneously searching and foraging for food. The "sitters," which are individuals not actively foraging, fulfill their needs by trading information, akin to sharing resources.

$$X_{i+1} = f(x) \begin{cases} X_i - CF \times phi \times rand \times [X_i - M] \\ \qquad \phi_{i+1} > \phi_i \\ X_i + CF \times phi \times rand \times [X_i - M] \\ \qquad otherwise \end{cases} \quad (19)$$

The group's approach suggests that the success or failure of the exploration phase should guide the simulation of future actions and the newly discovered scouting location (X). In this context, CF represents the distance the mongoose (M) can travel, as shown in Eq. (20). As the algorithm progresses through its iterations, the focus shifts from discovering new locations to optimizing the use of a profitable one. This shift is aided by a parameter that accelerates the exploration phase, ensuring efficient searching and decision-making during the early stages of the process.

$$M = \sum_{i=1}^{n} \frac{X_i \times sm_i}{X_i} \quad (20)$$

$$CF = \left(1 - \frac{iter}{Max_{iier}}\right)^{\left(\frac{2\times iter}{Max_{iter}}\right)} \quad (21)$$

$$phase = \begin{cases} Scout, & C < L \\ Babysitting, & C \geq L \end{cases} \quad (22)$$

The QDMO algorithm is built using a quantum-based optimization (QBO) method, where binary integers (0 or 1) represent the potential inclusion or exclusion of features. Each aspect of QBO is modeled by quantum bits (Qbit(q)), where q is a superposition of both binary standards, 0 and 1. The algorithm resets the data from previous forage collections when the counter (C) exceeds the exchange threshold. To ensure that the alpha-group weights decrease over time, a specific number of iterations are used, and the caretaker's weight is made to [illegible], facilitating improved results as the DMO process progresses.

$$q = a + i\beta = e^{i\theta}, |a|^2 + |\beta|^2 \quad (23)$$

QBO's major objective,

$$q(t+1) = q(t) \times R(\Delta\theta) = [a(t)\beta(t)] \times R(\Delta\theta) \quad (24)$$

$$R(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) - sin(\Delta\theta) \\ sin(\Delta\theta) - \cos(\Delta\theta) \end{bmatrix} \quad (25)$$

Equation (25) describes the angular velocity at which the [illegible] bit of rotates. By incorporating QBO, the DMO technique's ability to find the best solution is enhanced, striking a balance between exploration and exploitation. QDMO is a novel feature selection (FS) method, using 30% of the total data for testing and 70% for training on subsets. The data is used to evaluate fitness for each agent in the population. Agents with the lowest fitness are assigned to attain optimal performance. During the exploitation phase, the DMO operator refines the solution. This process continues until the termination criterion is met. Once QDMO is applied for feature selection, the process proceeds.

The following illustrates that x_i is the solution formula for Eq. (16):

$$X_i = [q_{i1}|q_{i2}] \dots |q_{iD}] = [\theta_{i1}|\theta_{i2}|\dots\theta_{iD}],\ i = 1,2,\dots,N \quad (26)$$

$$BX_{i,j} = \begin{cases} 1 & if\ rand < |\beta|^2 \\ 0 & otherwise \end{cases} \quad (27)$$

From Eq. (25) values for rand is [0, 1].

from $|BX_{i,j}|$:

$$Fit = \rho \times \gamma + (1-\rho) \times \left(\frac{|BX_{i,j}|}{D}\right) \quad (28)$$

4. Results and Discussion

The study involves various assessments and a comprehensive assessment of the model's performance in comparison to other learning methods. The research is conducted on a Windows 10 computer equipped with a seventh-generation Intel Core i7 processor. The proposed technique and additional learning models are implemented using Python programming, utilizing libraries such as TensorFlow, Scikit-learn, and Keras. The system configuration for the research model is provided in Table 2.

Table 2: System Configuration

Details	Component
Scikitlearn, TensorFlow	Libraries
Python 3.8	Language
8 GB	RAM
64-bit window 10	OS
Core i7, 7th Gen with 2.8 GHz processor	CPU
Nvidia, 1060, 8 GB	GPU

4.1. Validation analysis of Proposed model with Existing techniques

The performance validation of proposed model with existing procedures are tested on different metrics and it is visually shown in Figure 4 and 5.

Table 3: Proposed model with Existing techniques

Classifier	F1-Score	Precision	Recall	MCC	Accuracy (%)
SAC-LSTM-QDMO	0.97	0.96	[illegible]	0.95	0.97
BiLSTM	0.95	[illegible]	0.96	0.93	0.96
LSTM	0.94	0.93	0.95	0.91	0.95
RNN	0.93	0.92	0.94	0.9	0.94
CNN	0.96	0.95	0.97	0.94	0.96
DBN	0.92	[illegible]	0.93	0.89	0.93
ELM	[illegible]	0.9	0.92	0.87	0.92
XGBoost	[illegible]	0.89	0.91	0.86	0.91

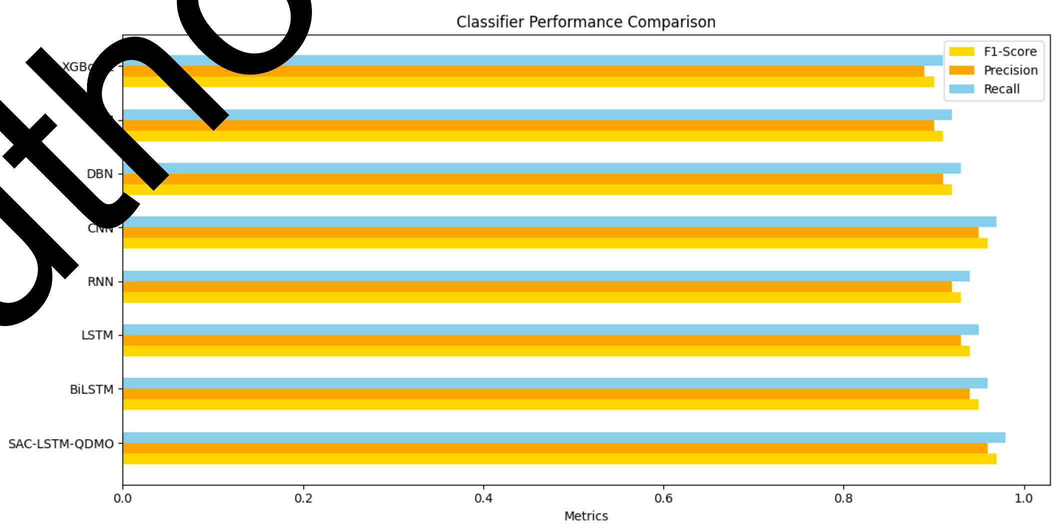


Figure 4 : (a) Graphical Description of proposed classical

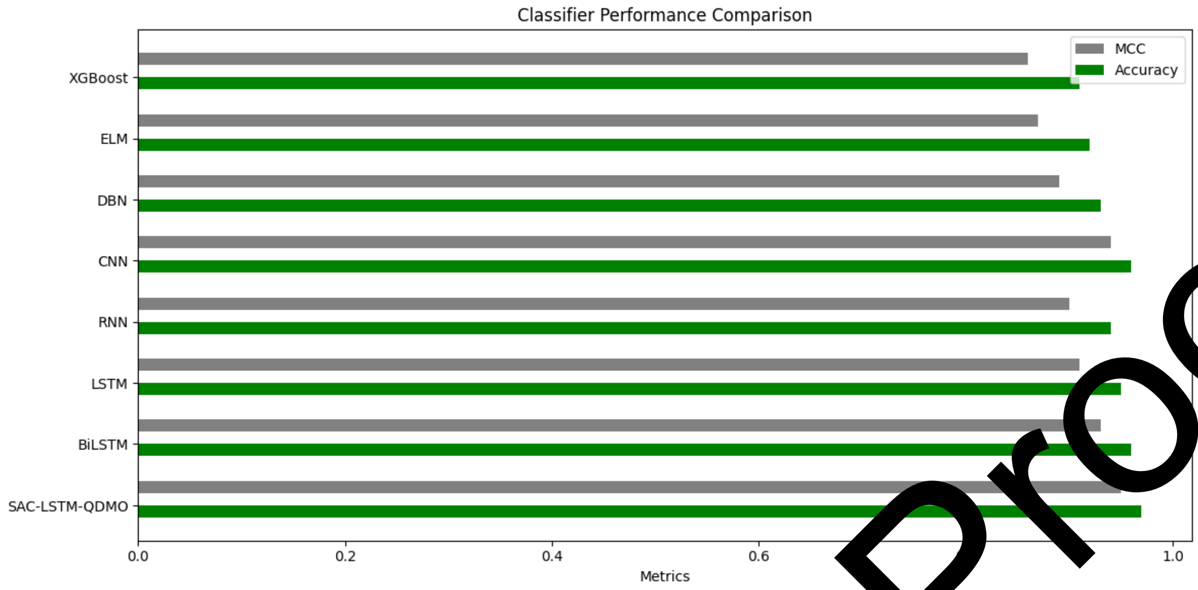


Figure 4 : (b) Graphical Description of proposed classical

The performance of various machine learning methods—XGBoost, ELM (Extreme Learning Machine), DBN (Deep Belief Network), CNN (Convolutional Neural Network), RNN, LSTM, BiLSTM (Bidirectional LSTM), and SACLSTM-QDMO (Self-Attention Convolutional LSTM with Quadratic Dimensionality Reduction and Multi-Objective Optimization)—across MCC (Matthews Correlation Coefficient), Accuracy (%), Recall, Precision, besides F1-Score metrics. A variety of indicators are used to evaluate the performance validation of the proposed model in comparison to the methods that are already in place, and the results are clearly displayed in Tables 3 and 4.

Table 4: Proposed model with Existing techniques

Classifier	**Accuracy (%)**	**F1-Score**	**Precision**	**Recall**	**MCC**
SAC-LSTM-QDMO	98	0.97	0.96	0.98	0.95
BiLSTM	97.5	0.95	0.94	0.96	0.93
LSTM	97	0.94	0.93	0.95	0.91
RNN	96.5	0.93	0.92	0.94	0.9
CNN	97.8	0.96	0.95	0.97	0.94
DBN	96	0.92	0.91	0.93	0.89
ELM	95	0.91	0.9	0.92	0.87
XGBoost	94	0.9	0.89	0.91	0.86

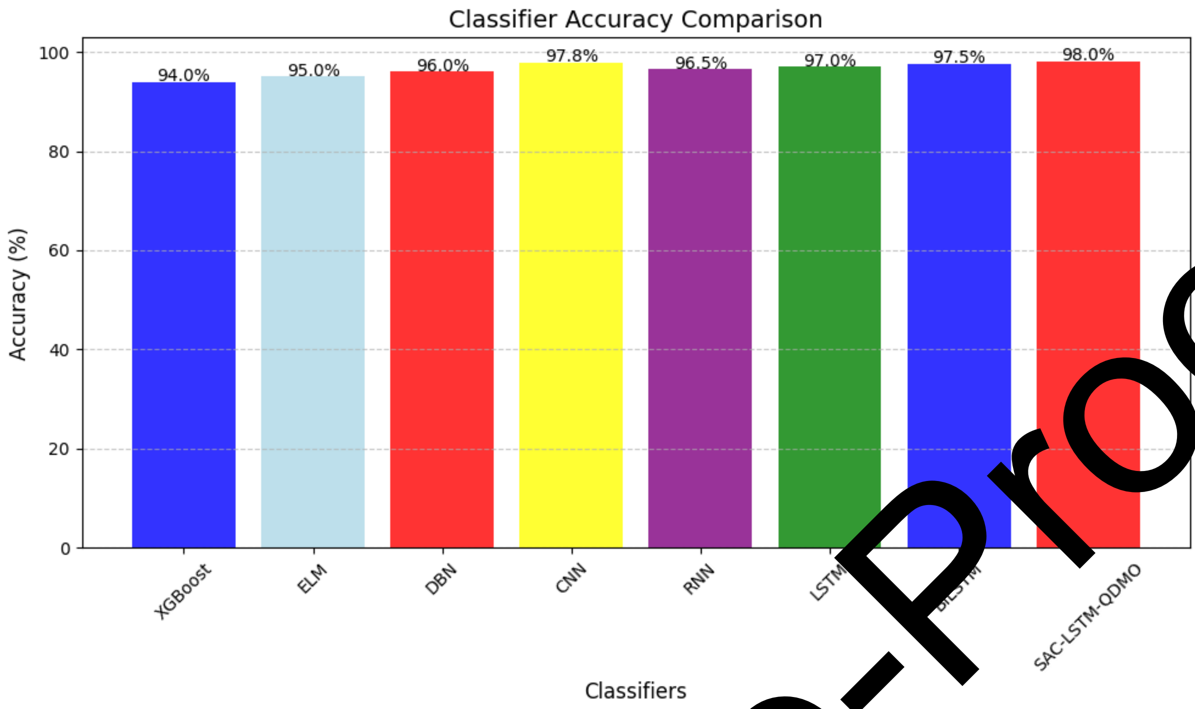


Figure 5: Visual Study of proposed classical with existing techniques

The SACLSTM-QDMO model demonstrates the best performance with the highest MCC (0.9820), accuracy (98.17%), recall (0.993[illegible]), precision (0.9932), and F1-Score (0.9882). Other models like BiLSTM and LSTM show strong results, with MCC values of 0.9731 and 0.9744, and F1-Scores of 0.9788 and 0.9756, respectively. While RNN and CNN also perform well with F1-Scores above 0.97, methods like XGBoost and ELM achieve lower accuracy and F1-Scores, making them less optimal for the task. This analysis highlights SACLSTM-QDMO's superiority in delivering robust and precise predictions.

5. Conclusion

This study develops predictive models using the advanced SACLSTM model, which incorporates convolutional layers to extract air quality data. The goal is to categorize and forecast urban air pollutants, particularly nitrogen dioxide, in the greater area. The model considers both patterns, accounting for the relationships between air quality, surrounding areas, and past and future data. It has been shown that the combination of convolutional and LSTM units outperforms traditional CNN and LSTM models in both statistical analysis and predictions. By optimizing the parameters of the SACLSTM with QDMOA, classification accuracy is improved. The algorithm is capable of forecasting nitrogen dioxide levels in Los Angeles over a ten-day period. This work provides valuable insights into nitrogen dioxide flow patterns for the next five years. Future studies will incorporate ground-based sensors to monitor numerous atmospheric and pollutant characteristics, such as temperature, wind speed, ozone, PM2.5, and carbon monoxide. Additionally, this approach may be expanded to other locations to enhance air pollution prediction models.

.

References

[1] Zhang, Z., Zhang, S., Chen, C., & Yuan, J. (2024). A systematic survey of air quality prediction based on deep learning. Alexandria Engineering Journal, 93, 128-141.

[2] Prado-Rujas, I. I., García-Dopico, A., Serrano, E., Córdoba, M. L., & Pérez, M. S. (2024). A multivariable sensor-agnostic framework for spatio-temporal air quality forecasting based on Deep Learning. Engineering Applications of Artificial Intelligence, 127, 107271.

[3] Mishra, A., & Gupta, Y. (2024). Comparative analysis of Air Quality Index prediction using deep learning algorithms. Spatial Information Research, 32(1), 63-72.

[4] Hettige, K. H., Ji, J., Xiang, S., Long, C., Cong, G., & Wang, J. (2024). AirPhyNet: Harnessing Physics-Guided Neural Networks for Air Quality Prediction. arXiv preprint arXiv:2402.03784.

[5] Sundaramoorthy, R. A., Ananth, A. D., Seerangan, K., Nandagopal, R., Balusamy, B., & Selvarajan, S. (2024). Implementing heuristic-based multiscale depth-wise separable adaptive temporal convolutional network for ambient air quality prediction using real time data. Scientific Reports, 14(1), 18437.

[6] Suthar, G., Kaul, N., Khandelwal, S., & Singh, S. (2024). Predicting land surface temperature and examining its relationship with air pollution and urban parameters in Bengaluru: A machine learning approach. Urban Climate, 53, 101830.

[7] Abbas, Z., & Raina, P. (2024). A wavelet enhanced approach with ensemble based deep learning approach to detect air pollution. Multimedia Tools and Applications, 83(6), 17531-17555.

[8] Nandi, B. P., Singh, G., Jain, A., & Tayal, D. K. (2024). Evolution of neural network to deep learning in prediction of air, water pollution and its Indian context. International Journal of Environmental Science and Technology, 21(1), 1021-1036.

[9] Kek, H. Y., Bazgir, A., Tan, H., Lee, C. T., Hong, T., Othman, M. H. D., ... & Wong, K. Y. (2024). Particle dispersion for indoor air quality control considering air change approach: a novel accelerated CFD-DNN prediction. Energy and Buildings, 306, 113938.

[10] Aravinda, K., Kumar, B. S., Kavin, B. P., & Thirumalraj, A. (2024). Traffic Sign Detection for Real-World Application Using Hybrid Deep Belief Network Classification. In Advanced Geospatial Practices in Natural Environment Resource Management (pp. 214-233). IGI Global.

[11] Aram, S. A., Nketiah, E. A., Saalidong, B. M., Wang, H., Afitiri, A. R., Akoto, A. B., & Lartey, P. O. (2024). Machine learning-based prediction of air quality index and air quality grade: a comparative analysis. International Journal of Environmental Science and Technology, 21(2), 1345-1360.

[12] Wang, S., McGibbon, J., & Zhang, Y. (2024). Predicting high-resolution air quality using machine learning: Integration of large eddy simulation and urban morphology data. Environmental Pollution, 344, 123371.

[13] Tang, D., Zhan, Y., & Yang, F. (2024). A review of machine learning for modeling air quality: Overlooked but important issues. Atmospheric Research, 107261.

[14] Thirumalraj, A., Chandrashekar, R., Gunapriya, B., & kavin Balasubramanian, P. (2024). NMRA-Facilitated Optimized Deep Learning Framework: A Case Study on IoT-Enabled Waste Management in Smart Cities. In Developments Towards Next Generation Intelligent Systems for Sustainable Development (pp. 247-268). IGI Global.

[15] Liu, Q., Cui, B., & Liu, Z. (2024). Air Quality Class Prediction Using Machine Learning Methods Based on Monitoring Data and Secondary Modeling. Atmosphere, 15(5), 553.

[16] Ahmed, A. A. M., Jui, S. J. J., Sharma, E., Ahmed, M. H., Raj, N., & Bose, A. (2024). An advanced deep learning predictive model for air quality index forecasting with remote satellite-derived hydro-climatological variables. Science of The Total Environment, 906, 167234.

[17] Ansari, M., & Alam, M. (2024). An intelligent IoT-cloud-based air pollution forecasting model using univariate time-series analysis. Arabian Journal for Science and Engineering, 49(3), 3135-3162.

[18] Yu, C., Wang, F., Wang, Y., Shao, Z., Sun, T., Yao, D., & Xu, Y. (2025). Mgsfformer: A multi-granularity spatiotemporal fusion transformer for air quality prediction. Information Fusion, 113, 102607.

[19] Huang, R., Hu, R., & Chen, H. (2025). A novel hybrid model for air quality prediction via dimension reduction and error correction techniques. Environmental Monitoring and Assessment, 197(1), 1-22.

[20] Zhang, Z., Zhang, S., Chen, C., & Yuan, J. (2024). A systematic survey of air quality prediction based on deep learning. Alexandria Engineering Journal, 93, 128-141.

[21] Chen, X., Hu, Y., Dong, F., Chen, K., & Xia, H. (2024). A multi-graph spatial-temporal attention network for air-quality prediction. Process Safety and Environmental Protection, 181, 442-451.

[22] Sundaramoorthy, R. A., Anandan, A. [illegible]., Seerangan, K., Nandagopal, M., Balusamy, B., & Selvarajan, S. (2024). Implementing heuristic-based multiscale depth-wise separable adaptive temporal convolutional network for ambient air quality prediction using real time data. Scientific Reports, 14(1), 18437.

[23] USGS. Usgs earthexplorer satellite imagery database. Website. https://earthexplorer.usgs.gov/.

[24] Muthukumar, P., Cocom, E., Holm, J., Comer, D., Lyons, A., Burga, I., ... & Pourhomayoun, M. (2021). Real-time spatiotemporal air pollution prediction with deep convolutional lstm through satellite image analysis. In Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020 (pp. 315-326). Springer International Publishing.

[25] Aruna, T. M., Kumar, P., Naresh, E., Divyaraj, G. N., Asha, K., Thirumalraj, A., ... & Yadav, A. (2024). Geospatial data for peer-to-peer communication among autonomous vehicles using optimized machine learning algorithm. Scientific Reports, 14(1), 20245.

[26] Thirumalraj, A., Aravinda, K., El-Kenawy, E. S. M., & Khodadadi, N. ScatterNet-based IPOA for predicting violent individuals using real-time drone surveillance system. In Industry 6.0 (pp. 182-204). CRC Press.

[27] Elaziz, M. A., Ewees, A. A., Al-qaness, M. A., Alshathri, S., & Ibrahim, R. A. (2022). Feature selection for high dimensional datasets based on quantum-based dwarf mongoose optimization. Mathematics, 10(23), 4565.