FMDB *Transactions on Sustainable Environmental Sciences*

Prediction of Northern Ireland's Air Pollution Using Kookaburra Optimization Algorithm-Based Dual-Stream Self-Attention Fusion Mechanism

Rakesh Chandrashekar[1,*], Jayasheel Kumar[2], M. Arunadevi Thirumalraj[3], S. Sharan Jeev[4]

[1]Department of Mechanical Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[2]Department of Automobile Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[3]Department of Computer Science and Engineering, Karunya Institute of Technology and Science, Coimbatore, Tamil Nadu, India.
[3]Department of Computer Science and Business Management, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India.
[4]Department of Cybersecurity, University of Texas at Dallas, Richardson, Texas, United States of America.
rakesh2687@gmail.com[1], jayasheel.81088@gmail.com[2], aruna.devi96@gmail.com[3], dal905518@utdallas.edu[4]

Abstract: Worldwide, but notably in developed and emerging countries, urban pollution is a huge issue. Modern cities' outward appearance is a major source of visual pollution, which, in turn, causes a variety of problems, including physical and mental health issues, distracted driving, environmental dangers, and emotional discomfort among residents. The need for easy, accurate air quality monitoring has arisen because air pollution poses a serious risk to human health. A station in Belfast city centre provided the data used in the study, which is open to the public. The study was conducted in Northern Ireland. It is a tool for measuring air pollution. An innovative method for converting text to images is suggested, which can take the input data and produce miniature images. To enable deep mining of global information via a self-attention mechanism, this study proposes a DSSAM that integrates spectral and spatial information. This study presents the KOA, a novel bio-inspired metaheuristic for hyper-parameter optimisation that mimics the behaviour of kookaburras. Part I presents the mathematical modelling based on simulating prey hunting, and Part II presents the modelling based on simulating kookaburra behaviour, that is, ensuring their prey is slain. Predictions of five air pollutants—Nitrogen Dioxide (NO2), Ozone (O3), Particulate Matter (PM2.5, and PM10)—achieved 94.67% accuracy, 88.42% precision, 88.18% recall, and 88.25% F1-score.

Keywords: Urban Air Pollution; Environmental Dangers; Global Information; Distracted Driving; Kookaburra Optimisation Algorithm (KOA); Nitrogen Dioxide; Air Pollutants Concentration; Dual-Stream Self-Attention Fusion Mechanism (DSSAM).

Received on: 23/10/2024, **Revised on:** 27/12/2024, **Accepted on:** 13/02/2025, **Published on:** 14/09/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSESS

DOI: https://doi.org/10.69888/FTSESS.2025.000540

Cite as: R. Chandrashekar, J. Kumar, M. A. Thirumalraj, and S. S. Jeev, "Prediction of Northern Ireland's Air Pollution Using Kookaburra Optimization Algorithm-Based Dual-Stream Self-Attention Fusion Mechanism," *FMDB Transactions on Sustainable Environmental Sciences*, vol. 2, no. 3, pp. 137–150, 2025.

1. Introduction

Every year, millions of deaths are caused by air pollution [1]. Air pollution is also a major roadblock to the world's future sustainable expansion efforts. A big public health problem in recent years, this is especially bad in certain emerging nations

*Corresponding author.

like China, which ranks first worldwide for the population-weighted yearly average concentration of PM2.5 [2]. Because of the widespread use of motor vehicles in China, traffic emissions have become a major contributor to air quality [3]. Secondary pollutants, such as ozone, sulphur dioxide, and particulate matter 2.5, can vary greatly throughout time and space [4]. The causes of local air pollution and the major drivers in a specific area can be difficult to trace and analyse when various changes or conditions are involved; for example, high relative humidity promotes PM2.5 formation, higher temperature enhances the photochemical reaction in the atmosphere, and wind contributes greatly to diffuse particulate matter [5]; [6]. As a general rule, chemical processes in the atmosphere are crucial nonlinear connections between traffic emissions and the makeup of the atmosphere [7]. On the other hand, photochemical production of ozone and PM is also strongly influenced by regional climatic variables, including air temperature, wind fields, humidity, and aerosols [8]. Researchers use deep learning to deconstruct the complex interplay among emissions inventories, transportation emissions, and weather conditions to assess their influence on urban air quality [9].

An image-based deep learning model for air quality prediction, an interpretable convolutional neural network prediction, an approach for concentration, and other similar applications have led some researchers to conclude that deep learning is an appropriate tool for studying air pollution [10]; [11]. Several domains, such as intelligent driving and intelligent medicine, have leveraged deep learning technologies, which have proven their effectiveness [12]. To mitigate the effects of air pollution and pinpoint problem locations, accurate air quality prediction is essential. Nevertheless, the accuracy of the results depends on the data available and the modelling methods used [13]. A network for spectral and spatial information extraction is stacked with attention mechanisms, as used in classification algorithms, to improve interpretability [14]. To further deepen the CNN's layers and enlarge the perceptual field, the multiscale extraction module is also employed. On the other hand, performance might suffer as a result of this [15]. With this in mind, the research recommends a DS network that leverages the self-attentive process to record spatial feature correlations between the centre pixel and its neighbourhood. Moreover, to fully utilise the long-term employees, a residual pyramid structure is required [26]. At last, the network can classify images by using weighted fusion [27]. The chief points of this study are as follows: The goal is to suggest a method that combines weather factors like temperature, wind speed, and wind direction with a lagged air quality feature that uses the concentration value from the previous hour to anticipate the upcoming hour's pollutant levels. To increase forecast accuracy and reduce errors, the study also uses the datetime index, which is divided into hour, day, and month for additional information:

- A two-stream classification network is suggested. Our new joint spectral-spatial network outperforms its predecessor in two key respects.
- First, the self-attentive mechanism effectively leverages text-converted data; second, the pyramidal residual structure achieves multi-scale feature extraction without increasing network depth, leading to superior feature discrimination.
- To fine-tune the hyperparameters, KOA models its implementation in two stages and mimics the behaviour of wild kookaburras.
- The research outlined the architecture and parameters required by statistical and deep learning models to forecast each pollutant; these models have potential utility across a range of smart city applications and can serve as a baseline for future improvements.

2. Related Works

The ambient assisted living air quality measurement, warning, and forecasting system proposed by Sonawani and Patil [16] is based on the Internet of Things (IoT). An ESP32 controller with next-gen embedded system architecture and an inexpensive suggested ambient-assisted living system. It can detect indoor air pollutants such as CO, PM2.5, NO2, O3, NH3, as well as humidity, temperature, pressure, and more. To improve performance, machine learning techniques are used to calibrate data from inexpensive sensors. One innovative component of the system is a gated recurrent unit that uses multi-headed convolutional neural networks to forecast air pollution levels over the next hour. When the system is young and there is less data available for prediction, the model uses transfer learning (TL) to improve performance. To make early forecasts for the new system, data from nearby sites may be used to share expertise. It may be equipped with a mobile app that alerts users when Indoor Air Quality metrics exceed specified limits. Everything needed to combat poor air quality can be found here. The study found that the IoT-based system, which used the TL framework, improved performance by 55.42% in RMSE scores for prediction at the new target system with inadequate data, demonstrating that the system operates effectively. To forecast regional NO2 and O3 attentions in the near future, Wu et al. [17] presented a unique deep learning-based hybrid model called Res-GCN-BiLSTM. This model combines ResNet, a graph convolutional network, and BiLSTM. The underlying spatial and temporal features were initially revealed using autocorrelation and cluster analysis, respectively. They showed both the geographical similarity and the temporal daily periodicity of AQMN. The next step was to effectively leverage the discovered spatiotemporal features by adaptively incorporating meteorological data into the model. For this evaluation, researchers used meteorological data from Shanghai and hourly data collected from 51 air quality monitoring sites. Compared to the top-performing baseline model, the Res-GCN-BiLSTM model enhanced prediction accuracy and was better suited to the features of the pollutants. The mean absolute error for NO2 was 11% lower, and for O3, 17% lower. A hybrid deep-learning model

combining an LSTM with an AE for IAQ anomaly detection is proposed by Wei et al. [18] to address this issue. Researchers use a long short-term memory (LSTM) network architecture in which several LSTM cells collaborate to learn the interdependencies between time-series data. By analysing reconstruction loss rates across all data and time-series sequences, the AE finds the best threshold.

Based on the Dunedin CO2 time-series dataset, collected during a real-world placement in New Zealand schools, our experimental findings show that our model outperforms others, achieving a very high and robust accuracy of 99.50%.To predict the concentrations of NO2, SO2, O3, and CO on an hourly (1 h to 24 h) basis, Rakholia et al. [19] set out to create a multi-step multi-output multivariate model (a global model) that takes into consideration a sum of areas, information about urban spaces, and the passage of time. By examining historical covariate concentrations, the global forecasting model can simultaneously predict multiple air pollutant concentrations. From February 2021 through August 2022, data on stations in HCMC. Dark Sky Weather reported hourly weather conditions for the same period. To our knowledge, this is the first model to predict levels of NO2, SO2, CO, and O3 in HCM City using real-time, up-to-date air quality data. The suggested model was tested using actual data collected from Healthy Air stations, and its performance was measured using correlation indices. Compared to previous models industrialised for HCMC air quality forecasting of individual pollutants, the global air quality forecasting model performs better. Researchers Esager and Ünlü [20] set out to investigate how well the Libyan city of Tripoli could predict hourly surface PM2.5 mass concentrations. Using a univariate time-series technique, the study forecasted PM2.5 levels using convolutional neural networks with gated recurrent units. Based on our findings, the convolutional model outperformed the others with a mean error of 0.04 and a coefficient of variation of 99%.

Decisions on Tripoli's air quality management may be informed by these results, which offer important insights into the application of deep learning algorithms for PM2.5 forecasting. A deep learning regression method based on fully convolutional networks (FCNs) was developed by Shin et al. [21] to overcome computational limitations and the limitations of DNN designs. The mean age of air (MAA) could be quickly predicted using a data-driven image-to-image training model that preserved spatial information. Data preprocessing to generate 2D images enabled strong MAA prediction regardless of the altered interior geometry, and no further model training or structural modifications were required. As a result, the prediction error utilising the FCN-based model was approximately 43.14% lower in terms of mean absolute error and 34.77% lower in terms of root mean squared error compared to the DNN regression. Also, based on the split zones, quantitative and qualitative comparisons were made with the prediction presentations under untrained conditions using additional test datasets. To improve the accuracy of AQI predictions, Van et al. [22] suggested a new approach that integrates (i) methods for analysing data on air pollution with (ii) algorithms to determine the optimal model for AQI prediction, evaluated using three metrics: MAE, RMSE, and R2. To ensure our suggested strategy was accurate, researchers used two public datasets collected from various parts of India. When it came to AQI value prediction, XGBoost was the clear winner. As a result, XGBoost is chosen as the affordable AQI forecasting device for permanently installed measurement stations.

3. Materials

The air quality dataset utilised in this investigation is derived from publicly available sources in Northern Ireland [23]. The air quality metrics in this dataset are nitrogen dioxide (NO2), sulphur dioxide (SO2), particulate matter (PM2.5), and ozone (O3), with hourly concentrations. It also includes weather information. Between 2015 and 2020, this data was collected at an air quality monitoring station located in the heart of Belfast. This dataset comprises more than 50,000 samples. The weather data are shown statistically in Table 1.

Table 1: Statistics of meteorological data

Statistic	Temperature (C°)	Wind Speed (m/s)	Wind Direction (°)
Count	52567	525623	52582
Mean	8.27	5.64	213.19
Std	4.43	2.77	84.87
Min	0	0	0
25%	4.9	3.5	156
50%	8	5.2	236.64
75%	11.5	7.3	273.5
Max	24	20.3	360

The sum of the samples, the average, the standard deviation, the minimum, and the maximum for each variable are part of the statistical dataset. Researchers observed that there were more than 50,000 samples in all. The standard deviation of all values

is 2.77, and the mean is 5.63, while the range of all parameters is 213.19. Parameters can take values between 0 and 365. All statistical information for the air quality metrics is summarised in Table 2.

Table 2: Data of air pollutants in µg/m3

Statistic	O3	SO2	No2	PM 2.5	PM10
Count	52608	52562	52567	52591	52557
25%	29	1	13	4	8
50%	44	1	22	7	12
75%	58	2	35	11	17
Mean	43.24	1.55	26.11	9	14.14
Std	20.65	1.6	17.87	7.94	10.6
Min	0	0	1	0	0
Max	150	20	203	104	143

The NO2 concentration data have a standard deviation of 17.87, a mean of 26.11, and a range of 1 to 203. However, SO2 has the lowest mean and the smallest standard deviation at 1.6.

3.1. Data Pre-Processing

Data normalisation may be necessary to support forecasting models, as datasets are susceptible to outliers and erroneous values. A dataset's general distribution might be affected by outliers, which are values that are extremely unusual or out of the ordinary. To make the forecasting model more accurate, though, outliers need to be eliminated. Similarly, before modelling, it may be necessary to eliminate or replace certain estimated values with missing or periodic sequences of values, called invalid values, in the dataset. Lastly, the dataset is normalised by rescaling the data to fall within the predetermined range. As a result of normalisation, forecasting models can converge faster and work better with smaller-scale data. This paper's dataset was preprocessed to exclude incorrect values and outliers using the interquartile range (IQR) method. To prepare for filling in the missing numbers, researchers are categorising the data by hour, day of the week, and month. After that, researchers fill in the blanks by averaging the concentration values for the same month, day, and hour over all years in the dataset. The missing data might have a wider range of values when using this method. In the forecasting models, weather data is just another input characteristic. In addition to the features already present, researchers developed a lag feature that uses the concentration value from the previous hour and the datetime index, split by hour, day, and month, to generate new features. Consequently, researchers have considered a variety of features for pollutant prediction, including past-hour concentrations, weather, and time (day, month, hour) data for next-hour forecasting. Following the definition in (1), the input Max normalisation. where $x_{\min}$ and $x_{\max}$ Are the standards of data x:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

3.2. Text2img Converter

When converting from text to IMG, all of the input data characteristics are taken into account. Table 2 contains 2,62,885 data points for the counts of five gases; the highest count, converted to a picture format, is 620. Below is an explanation of the suggested text2IMG conversion method:

$$I_{inp} \leftarrow f(x, y) \quad (2)$$

The input case, characterised as f (x, y), is represented by I_{inp}, Which input instance? The transformation is performed in Equation (3):

$$I_T \leftarrow T\{I_{inp}\} \quad (3)$$

The transformation of the text example is applied using T.$\{I_{inp}\}$, which results in the transformed instance, represented by I_T. The case is characterised in Equation (4):

$$I_T = (\prod(W, M)) \quad (4)$$

Equation (4) is defined in Equation (5). This is a repetitive window with ones in it. The additional multiplicative mask M characterises the conversion of i-j-dimensional data:

$$W = \sum_{i,j}^{n}(1,1) \quad (5)$$

In this case, n is the selected input iteration, respectively. In this case, W denotes the entire iterative window:

$$M = \sum_{i,j}^{n} m_{x(i,j)} \quad (6)$$

A specific point x for a matching location i, ∗ j in the matrix is translated into a matrix format m_(x(i,j)) from the multiplicative instance, which M. represents. Once the instance has been changed, it is a compression technique. This method applies lossy compression to convert the matrix into a picture format.

4. Proposed Methodology

The self-attentive mechanism (DSSAM-KOA) and the dual-stream convolutional neural network (DSCNN) are explained in depth in this section.

4.1. Dual-Stream Convolutional Neural Network

A dual-stream neural network is used in the study to improve spectral-feature merging in the input data. A spectral stream processes a one-dimensional spectral curve in this network, whereas an image stream extracts spatial features from a reduced-dimensional image representation. With the dual-stream network, you can use different structures and parameters in each branch without sharing them, which is a great benefit because backpropagation won't affect them. This enables significant improvements in recognition rates by extracting features from multiple datasets and fusing them to create complementary features. Also, it may use a variety of inputs for feature extraction, which makes it flexible, leverages the network's architecture, and helps the network better grasp the features. To create spatial feature representations from source images, a convolutional neural network processes pixels and their surrounding areas as input. The paper introduces a pyramidal-structured residual network, in which each pyramidal residual block is built from three cells of different sizes. Instead of rapidly expanding the feature-mapping dimension via downsampling, the basic idea is to progressively increase it by adding more channels to each cell. The ability to learn a more representative representation from the picture blocks is enhanced as the depth of the remaining cells increases, allowing more mappings to be obtained. To further enhance the network's capacity for generalisation and prevent overfitting, a batch normalisation (BN) layer is incorporated into the pyramid residual network. Directly translated hyperspectral data does not allow for good classification of the cascaded pyramidal residual network. Band selection is followed by further modifications to the pyramid residual block structure to improve the accuracy of air pollutant categorisation in the research. In the study, one of its component layers has been replaced with a self-attention mechanism, which greatly enhances the categorisation process. The following section explains the self-attention module in more detail. Each of the three pyramid residual modules that make up the extraction is labelled 'SA', which stands for a self-attention module. To differentiate between batch normalisation layers and convolutional layers of different sizes, labelled numbers are utilised. A pyramid's total residual network module output is expressed as:

$$\Upsilon = \mathrm{Relu}\{\mathrm{zero}(P_i) + [BN(P_i) * W_i + b_i]\} \quad (7)$$

The constant jump padding is denoted as zero (), the weight matrix is W_i, and the layer bias is b_i. In contrast to the spatial feature extraction network, which uses 2D-CNN convolution kernel dimensions, the spectrum-based classification method uses 1D-CNN dimensions for the spectral stream. To complete the classification job, it is prepared for the final down-sampling operation after the last pyramid residual module. Then, it is provided to the fully connected layer.

4.2. Self-Attention Mechanism

Every operation can cover only a tiny area around a single feature point, since the kernel size in the network is often constrained. Consequently, it is a difficult task to capture characteristics that are far away. The aforementioned shortcomings are addressed by using Self-Attention, which allows for faster retrieval of global features by directly calculating the association between any two pixels in the map. The self-attention apparatus outlined in this article is illustrated in Figure 1, which shows its execution process.

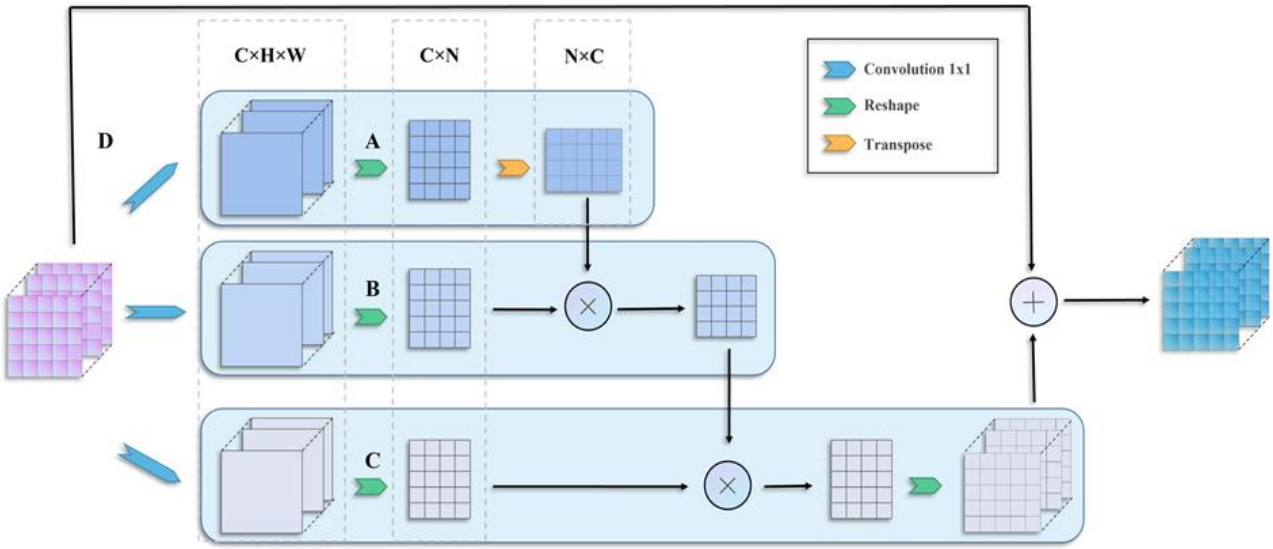

Figure 1: Self-Attention mechanism process

Consider the input source information, then for each element of the target info, determine the degree to which the Query matches each Key value. After assigning weights to the Values that correspond to each Key, the final Attention values are calculated by adding all of the weighted Values together. As a specialised attention mechanism, Self-Attention computes attention over the sequence and assigns different weights to different elements to capture their relationships. Representing global aspects more accurately with fewer parameters is possible with self-attention. Combining this feature with residual pyramid networks allows for the creation of spatial and spectral self-attention modules. Two 1×1 layers are used to generate two new feature maps, B and C, for the spatial self-attention module. {B, C} ∈ $R^{C\times H\times W}$, respectively, and then researchers resize them to RC×N, N = H × W, based on the overall number of input pixels. The spatial self-attention weight is then obtained by normalising the result of a matrix. distribution S ∈ $R^{N\times N}$:

$$s_{ji} = \frac{\exp(B_i.C_j)}{\sum_{i=1}^{N} ecp(B_i.C_j)} \tag{8}$$

Where s_{ji} position; a larger correlation exists when the two positions share comparable characteristics. Researchers also create a new feature map while feeding feature map A into the layer. D ∈ $R^{C\times H\times W}$, to $R^{C\times N}$. Then, the matrix result of size $R^{C\times H\times W}$, at the module's conclusion, is input into a weight parameter ʊ. After the multiplication operation, the result is added to An after being multiplied by ʊ. Let me give you the formula:

$$E = \omega \sum_{i=1}^{N} (s_{ji} D_i) + A_j \tag{9}$$

Where ʊ is involved in the network's learning and updating, and is initially set to 0, e is the weighted total of the input features plus the Features Figured for all sites; this is stated in the formula. As a result, it utilises a spatial self-matrix to accumulate contextual information and selectively establish global contextual linkages. The input of the unit is A ∈ $R^{C\times H\times W}$. Differentiating the spatial self- intended straight from the unique features. This includes altering the size of A into $R^{C\times N}$ and distribution X ∈ $R^{C\times C}$ is obtained by standardisation. The formula is as follows:

$$X_{ji} = \frac{\exp(A_i.A_j)}{\sum_{i=1}^{c} \exp(A_i.A_j)} \tag{10}$$

Where X_{ji} Computes the consequence of the spectrum. A matrix is accomplished among the transpose of X and A, followed by a matrix of size. $R^{C\times H\times W}$, to produce the final matrix depicting the weight distribution, as shown in Equation (11). This is followed by multiplying by adding it pixel by pixel to A:

$$E_j = \rho \sum_{i=1}^{C} (x_{ji} A_i) + A_j \tag{11}$$

Where ρ slowly learns that the network is efficient. The final total of all features across the spectral channels, weighted by the original features; this is obtained by modelling the long-range relationships between the feature maps. The discriminative power

of the traits can be improved with its help. Hence, the spectral self-attention module considers channel information before using the convolutional layer to embed features.

4.3. Fusion Weighted Mechanism

To train the network, researchers first split the entire image into small pieces, which they then use for feature extraction. Due to constraints in the neural network's fully connected layer, which does not integrate positional information, it is necessary to extract each pixel slice separately. It is not possible to classify the entire image at the pixel level because the network produces single-label outputs. Therefore, it is assumed that each pixel slice must be extracted independently. Current research on fusing dual-stream convolutional neural networks shows that combining recognition results to achieve the right rate is better than fusing only by 1.7% when using multiple fusions with complete layers, which is one of the best approaches. This work draws on the principles of integration learning in machine learning, training multiple learners on separate datasets before combining their findings to achieve better recognition accuracy. Separate learning of the spectral and image channels is followed by an increase in identification rate through a weighted fusion approach that accounts for their respective attributes. This work introduces an adaptive feature combination approach that efficiently blends channel and pixel characteristics. The feature vector that emerges from the fully connected layer's two branches is proportional to the number of data categories, with each branch corresponding to a different category. Researchers employ score weighting as their fusion strategy in two-stream networks, drawing inspiration from [24]. Separately, the weight matrices F_{se} and W_{sa} For vector S. This process can be specified as:

$$S_{se} = F_{se} * W_{se} \tag{12}$$

$$S_{sa} = F_{sa} * W_{sa} \tag{13}$$

If S_{se} and S_{sa} When they are directly summed, it fails to highlight the distinct significance of the two components. This would be equivalent to equally weighted feature splicing, which would not yield excellent performance and may have negative consequences. The network's training involves collecting several parameters to adaptively determine the space-to-spectrum ratio. The administered perfect will provide stronger data representation than channel weighting, and this may be seen as a better version of feature stitching. The following is a description of the adaptation process:

$$S = \alpha * S_{se} + \beta * S_{sa} \tag{14}$$

a and b. These are depicted as the inverse of the network's loss values. The loss value measures how far the new value is from the old label. A smaller loss value indicates that the two variables are more closely distributed, which, in turn, indicates better model performance. An appropriate objective function is required to optimise the model presented above. One popular loss function for classification is cross-entropy. As a function for air pollution categorisation, the study also used cross-entropy. Here is the cross-entropy loss function:

$$\text{Loss} = -\frac{1}{M}\sum_{m=1}^{M}\sum_{c=1}^{C} y_c^m \log(\bar{y}_c^m) \tag{15}$$

The truth label is y, and the predicted label is y. M samples make up a minibatch, while C is the number of classes. Model parameters are updated via stochastic gradient descent and backpropagation. As a random seed, the spatial flow is used to select a pixel from the same training set and the same loss, and the spectral dimension learns from the spectra of all pixels.

4.4. Hyper - Parameter Tuning Using KOA

The suggested model's hyperparameters are fine-tuned using the new optimisation technique, as detailed here in an iterative process to find viable solutions for optimisation issues. In the KOA population, kookaburras are distributed across the problem-solving space [25]. Each kookaburra uses its position in the space to determine the values of the decision variables; consequently, each kookaburra is a vector-based candidate key to the matrix, which includes kookaburras, may be represented by a matrix in accordance with Equation (16). When KOA is first implemented, Equation (17) is used to initialise the kookaburras' positions randomly:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \tag{16}$$

$$x_{i,d} = lb_d + r.(ub_d - lb_d) \quad (17)$$

X_i represents the ith kookaburra (possible answer) in this context, whereas x_(i,d) represents its dth dimension in the search space (the choice variable). N is the count of kookaburras, m is the count of variable quantity, and r is a random integer between 0 and 1. lb_d and ub_d Variable, correspondingly. One way to evaluate the function of a problem is to look at the issue-solving space as a whole, with each kookaburra representing a potential solution. A vector can be used to represent the problem's objective, as in Equation (18):

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \quad (18)$$

In this case, F is the evaluated objective function according to the ith kookaburra, and F_i is the evaluated function vector. To assess potential solutions and members of the population, it is appropriate to use the objective function values obtained from the assessment. In the objective function, the best member is the one with the highest evaluated value, and the worst member is the one with the lowest evaluated value. If researchers remember that, with each iteration, they update the kookaburras' locations in the space, reevaluate the problem's objective function, and compare the new values to determine which member of the population is the best. The suggested KOA method iteratively improves candidate solutions by updating kookaburra positions in two phases, based on simulations of kookaburra behaviours in the wild. Following this, researchers will show you how to update the KOA population in the space.

4.4.1. Phase 1: Hunting Policy (Exploration)

A carnivore, the kookaburra preys on insects, frogs, mice, reptiles, and other small birds. Despite its feeble legs, this bird's formidable neck helps it hunt. Kookaburras undergo massive positional shifts as a result of their strategy of preying on certain animals. This procedure exemplifies the global search through exploration, which is defined as a thorough examination of the local ideal to find the primary optimal area. The KOA design accounts for the locations of other kookaburras, whose positions have functional value, to mimic kookaburras' hunting approach. Hence, Equation (19) is used to identify the available prey set for each value:

$$CP_i = \{X_k : F_k < F_i \text{ and } k \neq i\}, \text{where } i = 1,2,\ldots,N \text{ and } k \in \{1,2,\ldots N\} \quad (19)$$

Here, CP_i is the set, X_k is a kookaburra, and F_k It is a function charge. The premise of the KOA design is that kookaburras attack their victims at random. An updated location for the kookaburra is determined by plugging its movements into the hunting strategy's simulation into Equation (20). According to Equation (21), if the goal function's value is enhanced at the new position, then the corresponding kookaburra will move to this new position:

$$x_{i,d}^{P1} = x_{i,d} + r\left(SCP_{i,d} - I.x_{i,d}\right), i = 1,2,\ldots,N \text{ and } d = 1,2,..,m \quad (20)$$

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} < F_i \\ X_i, \text{ else} \end{cases} \quad (21)$$

Here, X_i^{P1} Is the novel the recommended site for the initial phase of KOA? $x_{i,d}^{P1}$ is its dth dimension, FiP1 is its charge, r is a random sum range of (0, 1), $SCP_{i,d}$ is the dth dimension of designated prey for the ith kookaburra, I is an accidental sum from the set $\{1,2\}$N is the sum of kookaburra, and m is the sum of the decision variable quantity.

4.4.2. Phase 2: Ensuring that the Prey is Killed (Exploitation)

The additional distinctive behaviour is that, following an assault, they would bring their victim close to the tree and repeatedly strike it until it dies. The kookaburra then clamps its beak around its victim, stomps on it, and devours it. When kookaburras engage in this behaviour near their hunting field, they slightly shift their stance. This procedure, which embodies the idea of abuse, refers to the algorithm's capacity to find better solutions near existing regions. The KOA design uses Equation (22) to determine a random position, mimicking the behaviour of kookaburras when they travel towards the hunting site. It is believed that this movement occurs at random within a radius equal to the distance from the centre of each kookaburra. $\frac{(ub_d - lb_d)}{t}$. The radius is initially set to the extreme charge; then, in iterations, it is reduced to improve the accuracy of the local search, which

converges towards better solutions. If the goal function's value is improved by the newly computed location for each kookaburra, as per Equation (23), then the newly computed position will replace the prior one:

$$x_{i,d}^{P2} = x_{i,d} + (1 - 2r).\frac{(ub_d - lb_d)}{t}, i = 1,2,\ldots,N, d = 1,2,..,m \text{ and } t = 1,2,\ldots,T \quad (22)$$

$$X_i = \begin{cases} X_i^{P2}, F_i^{P2} < F_i \\ X_i, \text{ else} \end{cases} \quad (23)$$

Here, X_i^{P2} Is the novel recommended position of the ith phase of KOA? $x_{i,d}^{P2}$ is its third dimension, F_i^{P2} is its price, t is the repetition pawn of the procedure, besides T is the extreme sum of procedure repetitions.

4.4.3. Repetition Procedure of KOA

After all kookaburra locations have been updated in accordance with the first two phases of KOA, the first iteration is complete. Every iteration ends with saving the most recent and optimal result. The algorithm then proceeds to the next iteration using the revised placements and the newly evaluated goal function values. Until the last repetition of the procedure based on Equations (19)-(23), the process of updating positions continues. At the end of the algorithm's iterations, KOA proposes the solution to the issue based on the best candidate solution it found.

5. Results and Discussion

A model is trained and evaluated on a GeForce RTX 2080 Ti GPU, and all experiments presented in this paper are built using the PyTorch framework. A starting learning rate of 0.001 is used. The accuracy and loss for the projected model's data are shown in Figures 2 and 3, respectively.

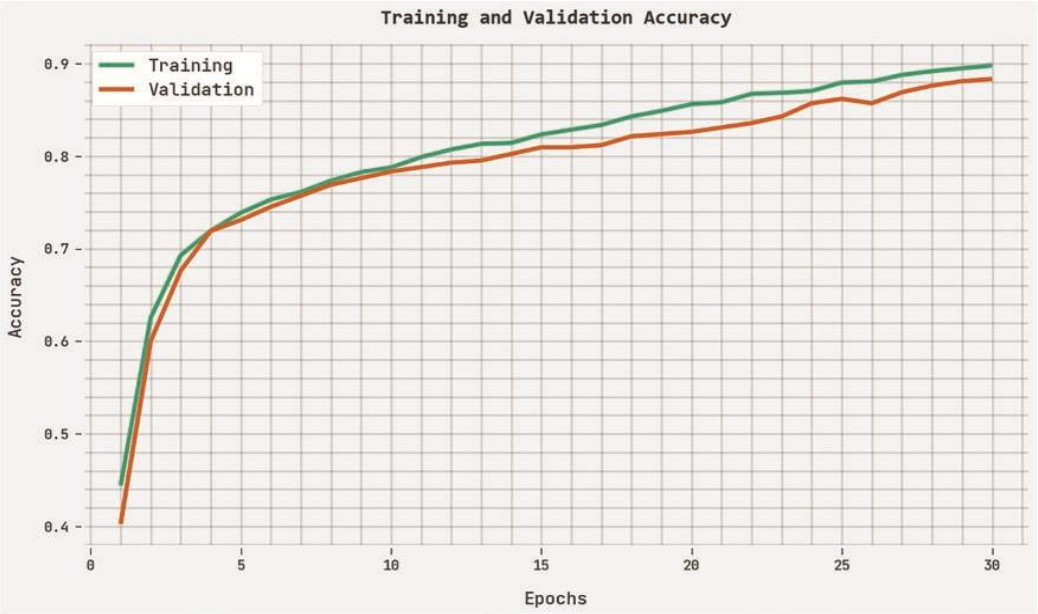

Figure 2: Accuracy investigation of the proposed approach

Figure 3 shows the model's full training behaviour, plotting the training and validation losses over time. At first, both curves show quite large loss values. This is because the model has not yet found meaningful patterns in the data. A steady, gradual drop in both losses is observed as training continues. This shows that the optimisation process is working and that the model parameters are being improved in a meaningful way. The curves' consistent downward slope, with no unexpected spikes or instability, shows that the learning rate was chosen correctly and the convergence process is stable. The training loss remains somewhat lower than the validation loss throughout training. This is what researchers would expect because the model is directly optimised on the training dataset. The tiny, persistent gap between the two curves shows that the model works well on data it hasn't seen before and doesn't have overfitting. Also, the fact that the validation loss continues to decrease in later epochs suggests that more training is beneficial for performance rather than harmful. Overall, the tight alignment and convergence of

the two curves toward lower loss values indicate that the model produces balanced, reliable learning outputs, making it well-suited for accurate predictions and real-world use.

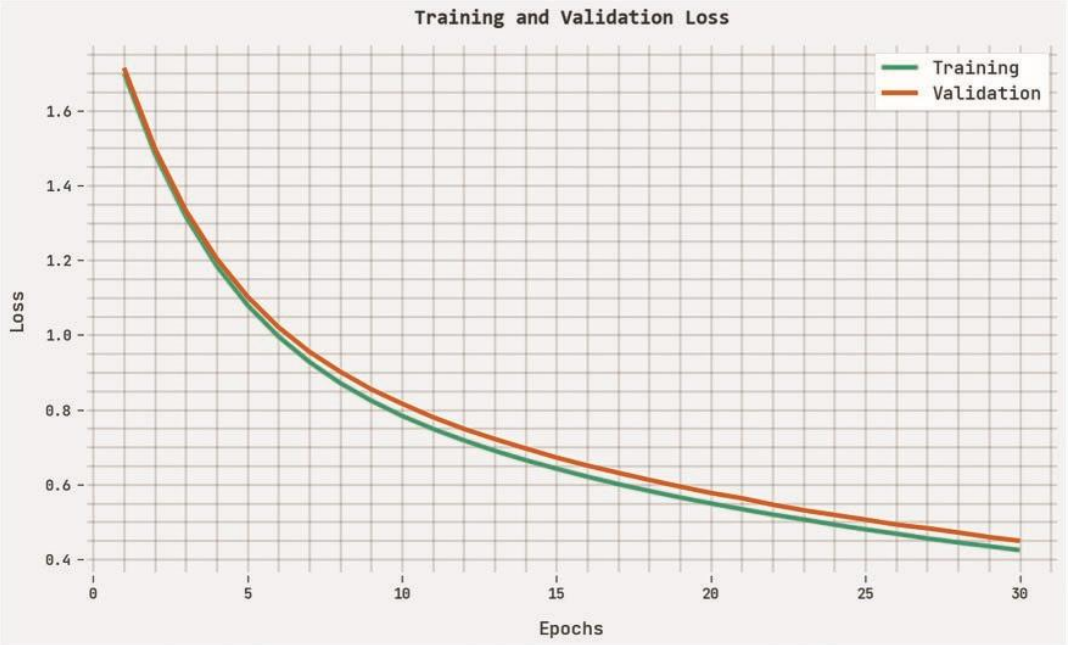

Figure 3: Loss analysis of the projected approach

5.1. Validation Analysis of the Projected Model

Table 3 presents classification accuracy for existing models across various metrics.

Table 3: Comparative analysis of the proposed model on 80%-20%

Approaches	Training acc.	Testing acc.	Sens.	Spec.	Prec.	F-score
MLP	0.886	0.849	0.792	0.850	0.887	0.837
DBN	0.902	0.871	0.825	0.870	0.904	0.863
AE	0.915	0.893	0.840	0.890	0.935	0.885
RNN	0.924	0.897	0.864	0.900	0.921	0.892
LSTM	0.907	0.879	0.812	0.880	0.931	0.868
CNN	0.911	0.885	0.833	0.880	0.926	0.877
DSCNN	0.913	0.881	0.849	0.880	0.906	0.874
Self-Attention	0.905	0.875	0.832	0.870	0.906	0.867
DSSAM-KOA	0.962	0.947	0.824	0.974	0.830	0.822

In Table 3, characterise the Comparative analysis of the Proposed Model at 80%-20%. In the investigation of the MLP model, the training accuracy was 0.886, the test accuracy was 0.849, and the sensitivity, specificity, precision, and F-score were 0.792, 0.850, 0.887, and 0.837, respectively, all of which were consistent. Then the DBN achieved training accuracy of 0.902, testing accuracy of 0.871, sensitivity of 0.825, specificity of 0.870, precision of 0.904, and F-score of 0.863, all of which were consistent. Previously, the AE model achieved training accuracy of 0.915, testing accuracy of 0.893, sensitivity of 0.840, specificity of 0.890, precision of 0.935, and F-score of 0.885, all of which were consistent. Then the RNN model achieved training accuracy of 0.924, testing accuracy of 0.897, sensitivity of 0.864, and precision of 0.900, all of which were consistent. Then the 0.921 and F-score were 0.892, congruently (Figure 4).

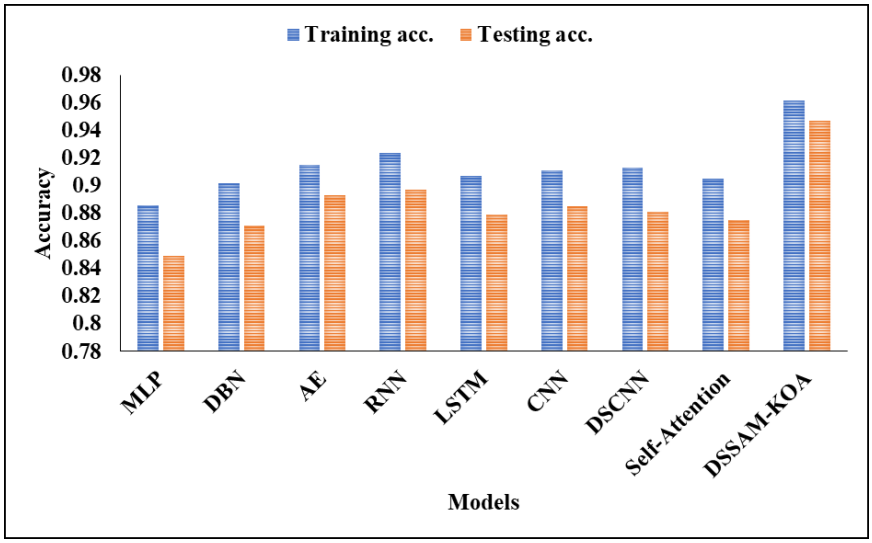

Figure 4: Accuracy analysis of various models

Then the LSTM model achieved training accuracy of 0.907 and testing accuracy of 0.879, along with previously reported sensitivity of 0.812, specificity of 0.880, precision of 0.931, and F-score of 0.868. Then the CNN achieved training accuracy of 0.911, testing accuracy of 0.885, sensitivity of 0.833, F-score of 0.880, and precision of 0.926, all of which were consistent. Then the 0.877 DSCNN model attained training accuracy of 0.913, testing accuracy of 0.881, sensitivity of 0.849, specificity of 0.880, precision of 0.906, and F-score of 0.874, all of which were congruent (Figure 5).

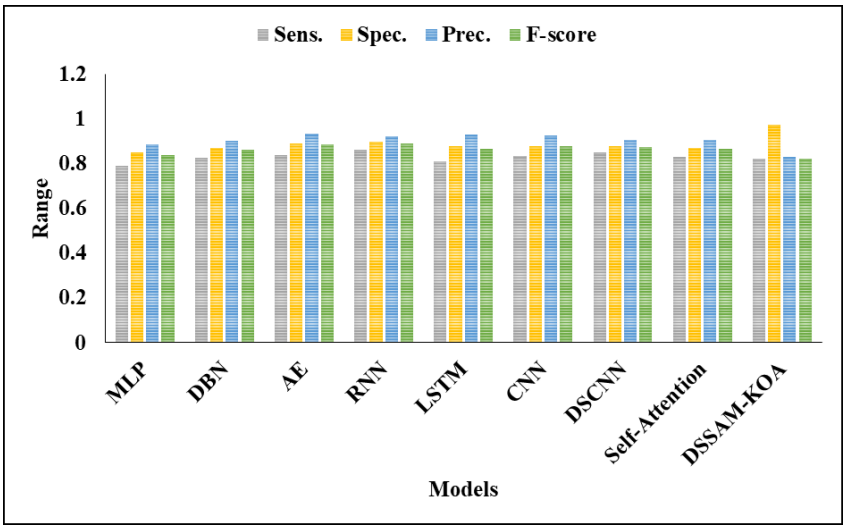

Figure 5: Graphical description of the proposed model

Then the Self-Attention model achieved training accuracy of 0.905, testing accuracy of 0.875, sensitivity of 0.832/0.870, precision of 0.906, and F-score of 0.867, all of which were congruent. At that time, the DSSAM-KOA model achieved training accuracy of 0.962, test accuracy of 0.947, sensitivity of 0.824, specificity of 0.974, precision of 0.830, and F-score of 0.822.

Table 4: Validation analysis of innumerable representations on 60%-40%

Approaches	Accuracy	Precision	Recall	F-Score
DSSAM-KOA	94.67	88.42	88.18	88.25
Self-Attention	91.60	83.35	82.35	85.30
DSCNN	87.30	82.69	83.81	83.81
LSTM	89.42	84.61	82.71	84.86
RNN	83.53	80.60	80.91	81.31

In Table 4 above, characterise the Validation analysis of numerous representations at 60%-40%. In the analysis of the DSSAM-KOA model, the accuracy was 94.67%, the precision was 88.42%, the recall was 88.18%, and the F-score was 88.25%. Then, the Self-Attention model achieved an accuracy of 91.60, a precision of 83.35, a recall of 82.35, and a final F1 score of 85.30. Then, the DSCNN model achieved an accuracy of 87.30, a precision of 82.69, a recall of 83.81, and a final F1 score of 83.81. Then the LSTM model achieved an accuracy of 89.42, a precision of 84.61, a recall of 82.71, and a final f-score of 84.86. The RNN model achieved an accuracy of 83.53%, a precision of 80.60%, a recall of 80.91%, and a final f-score of 81.31% (Figure 6).

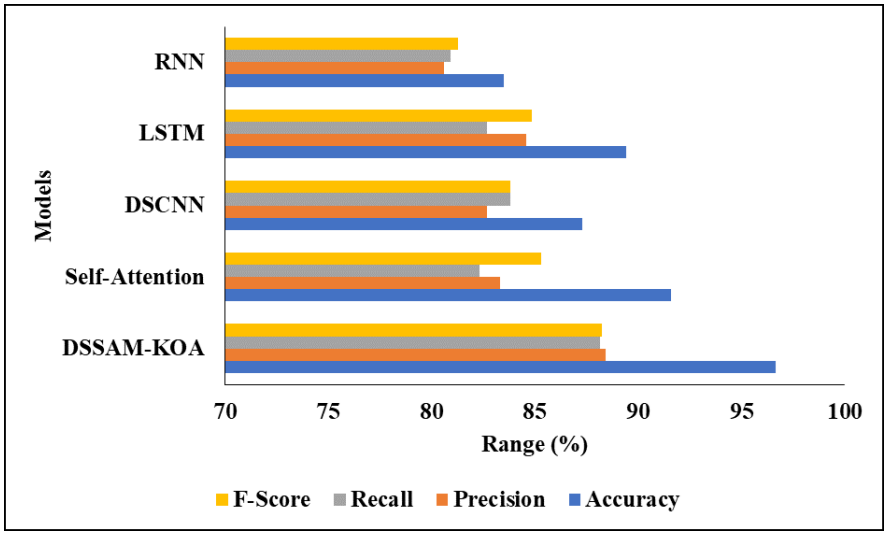

Figure 6: Graphical description of various models for 60%-40%

6. Conclusion and Future Work

Reducing health risks and environmental problems requires a precise forecast of air pollution, a worldwide health challenge. Using the suggested deep learning model, this study aims to achieve one-step air pollution prediction for a range of pollutants (e.g., NO2, O3, SO2, PM2.5, PM10). To effectively forecast air quality, this study proposes a DSSFN that fuses spatial and spectral data. The data is preprocessed into an image format to enhance categorisation accuracy. The system's capability to smear the extracted info with the data is enhanced by the dual-stream network's ability to extract and perform weighted fusion simultaneously. By increasing the intensity of analysing the association data, the self-attention mechanism effectively captures distant image details. The residual structure improves the capacity to mitigate gradient vanishing. Therefore, the model's classification accuracy improves while its complexity decreases. To increase classification accuracy, the KOA model optimises the network's hyperparameter tuning. The testing results reveal that the suggested method achieved an accuracy of 96.67%, precision of 88.42%, recall of 88.18%, and an F-score of 88.25%. There will be many expansions of the research. The primary goal is to expand our current work to incorporate additional sensor data types for multivariate analysis, such as specific matter, temperature, and relative humidity. To test the solution's viability and generalizability, I intend to expand the project's scope to include additional application areas.

Acknowledgement: The authors sincerely acknowledge the academic support and facilities provided by New Horizon College of Engineering, Karunya Institute of Technology and Science, Saranathan College of Engineering, and the University of Texas at Dallas.

Data Availability Statement: The data supporting the findings of this study can be obtained from the corresponding authors upon reasonable and justified request, ensuring openness and reproducibility of the research.

Funding Statement: This research was conducted without the support of any external funding agencies.

Conflicts of Interest Statement: The authors collectively declare that there are no conflicts of interest related to this work.

Ethics and Consent Statement: All authors confirm that the study was conducted in accordance with established ethical standards, with informed consent obtained and participant confidentiality maintained.

References

1. R. Janarthanan, P. Partheeban, K. Somasundaram, and P. N. Elamparithi, “A deep learning approach for prediction of air quality index in a metropolitan city,” *Sustainable Cities and Society*, vol. 67, no. 4, p. 102720, 2021.
2. W. Mao, W. Wang, L. Jiao, S. Zhao, and A. Liu, “Modelling air quality prediction using a deep learning approach: Method optimization and evaluation,” *Sustainable Cities and Society*, vol. 65, no. 2, p. 102567, 2021.
3. Z. Zhang, Y. Zeng, and K. Yan, “A hybrid deep learning technology for PM2.5 air quality forecasting,” *Environmental Science and Pollution Research*, vol. 28, no. 3, pp. 39409–39422, 2021.
4. W. Zhang, Y. Wu, and J. K. Calautit, “A review on occupancy prediction through machine learning for enhancing energy efficiency, air quality, and thermal comfort in the built environment,” *Renewable and Sustainable Energy Reviews*, vol. 167, no. 10, p. 112704, 2022.
5. Y. B. Kim, S. B. Park, S. Lee, and Y. K. Park, “Comparison of PM2.5 prediction performance of the three deep learning models: A case study of Seoul, Daejeon, and Busan,” *Journal of Industrial and Engineering Chemistry*, vol. 120, no. 4, pp. 159–169, 2023.
6. R. K. Inapakurthi, S. S. Miriyala, and K. Mitra, “Deep learning based dynamic behavior modelling and prediction of particulate matter in air,” *Chemical Engineering Journal*, vol. 426, no. 12, p. 131221, 2021.
7. S. Sahoo, S. Kumar, M. Z. Abedin, W. M. Lim, and S. K. Jakhar, “Deep learning applications in manufacturing operations: a review of trends and ways forward,” *Journal of Enterprise Information Management*, vol. 36, no. 1, pp. 221–251, 2023.
8. C. Huang, T. Hu, Y. Duan, Q. Li, N. Chen, Q. Wang, M. Zhou, and, P. Rao, “Effect of urban morphology on air pollution distribution in high-density urban blocks based on mobile monitoring and machine learning,” *Building and Environment*, vol. 219, no. 7, p. 109173, 2022.
9. A. Heydari, M. M. Nezhad, D. A. Garcia, F. Keynia, and L. D. Santoli, “Air pollution forecasting application based on deep learning model and optimization algorithm,” *Clean Technologies and Environmental Policy*, vol. 24, no. 4, pp. 1–15, 2022.
10. N. A. Zaini, L. W. Ean, A. N. Ahmed, and M. A. Malek, “A systematic literature review of deep learning neural network for time series air quality forecasting,” *Environmental Science and Pollution Research*, vol. 8, no. 11, pp. 4958–4990, 2021.
11. M. Niu, Y. Zhang, and Z. Ren, “Deep learning-based PM2.5 long time-series prediction by fusing multisource data—a case study of Beijing,” *Atmosphere*, vol. 14, no. 2, p. 340, 2023.
12. S. Sim, J. H. Park, and H. Bae, “Deep collaborative learning model for port-air pollutants prediction using automatic identification system,” *Transportation Research Part D: Transport and Environment*, vol. 111, no. 10, p. 103431, 2022.
13. P. Y. Kow, I. W. Hsia, L. C. Chang, and F. J. Chang, “Real-time image-based air quality estimation by deep learning neural networks,” *Journal of Environmental Management*, vol. 307, no. 1, p. 114560, 2022.
14. G. Y. Lin, Y. M. Lee, C. J. Tsai, and C. Y. Lin, “Spatial-temporal characterization of air pollutants using a hybrid deep learning/Kriging model incorporated with a weather normalization technique,” *Atmospheric Environment*, vol. 289, no. 11, p. 119304, 2022.
15. W. I. Lai, Y. Y. Chen, and J. H. Sun, “Ensemble machine learning model for accurate air pollution detection using commercial gas sensors,” *Sensors*, vol. 22, no. 12, p. 4393, 2022.
16. S. Sonawani and K. Patil, “Air quality measurement, prediction, and warning using transfer learning based IoT system for ambient assisted living,” *International Journal of Pervasive Computing and Communications*, vol. 20, no. 1, pp. 38–55, 2024.
17. C. L. Wu, H. D. He, R. Song, X. H. Zhu, Z. R. Peng, Q. Y. Fu, and J. Pan, “A hybrid deep learning model for regional O3 and NO2 concentrations prediction based on spatiotemporal dependencies in air quality monitoring network,” *Environmental Pollution*, vol. 320, no. 12, p. 121075, 2023.
18. Y. Wei, J. Jang-Jaccard, W. Xu, F. Sabrina, S. Camtepe, and M. Boulic, “LSTM-autoencoder-based anomaly detection for indoor air quality time-series data,” *IEEE Sensors Journal*, vol. 23, no. 4, pp. 3787–3800, 2023.
19. R. Rakholia, Q. Le, B. Q. Ho, K. Vu, and R. S. Carbajo, “Multi-output machine learning model for regional air pollution forecasting in Ho Chi Minh City, Vietnam,” *Environment International*, vol. 173, no. 3, p. 107848, 2023.
20. M. W. M. Esager and K. D. Ünlü, “Forecasting air quality in Tripoli: An evaluation of deep learning models for hourly PM2.5 surface mass concentrations,” *Atmosphere*, vol. 14, no. 3, p. 478, 2023.
21. S. Shin, K. Baek, and H. So, “Rapid monitoring of indoor air quality for efficient HVAC systems using fully convolutional network deep learning model,” *Building and Environment*, vol. 234, no. 3, p. 110191, 2023.
22. N. H. Van, P. Van Thanh, D. N. Tran, and D. T. Tran, “A new model of air quality prediction using lightweight machine learning,” *International Journal of Environmental Science and Technology*, vol. 20, no. 3, pp. 2983–2994, 2023.

23. A. A. Donnelly, B. M. Broderick, and B. D. Misstear, “The effect of long-range air mass transport pathways on PM10 and NO2 concentrations at urban and rural background sites in Ireland: Quantification using clustering techniques,” *Journal of Environmental Science and Health*, vol. 50, no. 7, pp. 647-658, 2015.
24. Y. Xu, B. Du, and L. Zhang, “Beyond the Patchwise Classification: Spectral-Spatial Fully Convolutional Networks for Hyperspectral Image Classification,” *IEEE Transactions on Big Data*, vol. 6, no. 3, pp. 492–506, 2020.
25. Anand, P. P., Jayanth, G., Rao, K. S., Deepika, P., Faisal, M., & Mokdad, M. (2024). Utilising Hybrid Machine Learning to Identify Anomalous Multivariate Time-Series in Geotechnical Engineering. AVE Trends In Intelligent Computing Systems, 1(1), 32–41.
26. M. Dehghani, Z. Montazeri, G. Bektemyssova, O. P. Malik, G. Dhiman, and A. E. M. Ahmed, “Kookaburra Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems,” *Biomimetics*, vol. 8, no. 6, p. 470, 2023.
27. K. A. Babu, K. Arulvendhan, P. Srinivasan, M. A. Ahmad, A. Prabha, M. M. Thariq, M. M. S. Ali, and C. C. Angelin, “Economical infotainment solution augmented with advanced telematics for collision detection, vehicle localization, and real-time health status monitoring,” *AVE Trends in Intelligent Health Letters*, vol. 1, no. 4, pp. 206–216, 2024.