



---

*Research article*

## Hybrid Deep Learning-Based Air Pollution Prediction and Index Classification Using an Optimization Algorithm

Sreenivasulu Kutala<sup>1</sup>, Harshavardhan Awari<sup>2,\*</sup>, Sangeetha Velu<sup>3</sup>, Arun Anthonisamy<sup>4</sup>, Naga Jyothi Bathula<sup>5</sup> and Syed Inthiyaz<sup>6</sup>

<sup>1</sup> Department of Computer Science and Engineering, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India; sreenu.kutala@gmail.com

<sup>2</sup> Department of CSE (AIML & IoT), VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India; harshavgse@gmail.com

<sup>3</sup> Department of Computer Science and Engineering, Ramaiah Institute of Technology, India; drsangeethav@msrit.edu

<sup>4</sup> Department of Computer Science and Business Systems, Panimalar Engineering College, Chennai, India; drarun.srm@gmail.com

<sup>5</sup> Department of ECE, DMS SVH College of Engineering, Machilipatnam, Andhra Pradesh, India; jyothibathula30@gmail.com

<sup>6</sup> Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India; syedinthiyaz@kluniversity.in

\* **Correspondence:** Email: harshavgse@gmail.com.

**Abstract:** Lockdowns were implemented in nearly all countries in the world in order to reduce the spread of COVID-19. The majority of the production activities like industries, transportation, and construction were restricted completely. This unprecedented stagnation of resident's consumption and industrial production has efficiently reduced air pollution emissions, providing typical and natural test sites to estimate the effects of human activity controlling on air pollution control and reduction. Air pollutants impose higher risks on the health of human beings and also damage the ecosystem. Previous research has used machine learning (ML) and statistical modeling to categorize and predict air pollution. This study developed a binary spring search optimization with hybrid deep learning (BSSO-HDL) for air pollution prediction and an air quality index (AQI) classification process during the pandemic. At the initial stage, the BSSO-HDL model pre-processes the actual air quality data and makes it compatible for further processing. In the presented BSSO-HDL model, an HDL-based air quality prediction and AQI classification model was applied in which the HDL was derived by the use

of a convolutional neural network with an extreme learning machine (CNN-ELM) algorithm. To optimally modify the hyperparameter values of the BSSO-HDL model, the BSSO algorithm-based hyperparameter tuning procedure gets executed. The experimental outcome demonstrates the promising prediction classification performance of the BSSO-HDL model. This model, developed on the Python platform, was evaluated using the coefficient of determination  $R^2$ , the mean absolute error (MAE), and the root mean squared error (RMSE) error measures. With an  $R^2$  of 0.922, RMSE of 15.422, and MAE of 10.029, the suggested BSSO-HDL technique outperforms established models such as XGBoost, support vector machines (SVM), random forest (RF), and the ensemble model (EM). This demonstrates its ability in providing precise and reliable AQI predictions.

**Keywords:** air pollution monitoring; air quality index; deep learning; parameter optimization

## 1. Introduction

Air pollution is a major concern on a global scale. According to estimates from the World Health Organization (WHO), air pollution has caused illnesses in around 7 million individuals. Lung cancer, bronchitis, asthma, heart disease, skin infections, eye disorders, throat infections, and other ailments are among the conditions that are made more likely by air pollution. The risk of dying young may increase with prolonged exposure to air pollution. Children may face developmental challenges include impaired cognitive development and lung function. Early deliveries, low birth weights, and other difficulties are among the complications that expectant mothers may face. Apart from illnesses, a major danger to plants is air pollution. The high volume of emissions from vehicles and businesses contributes significantly to greenhouse effects. Air pollution will have a significant economic impact by raising healthcare costs for both individuals and the government. Productivity will suffer as a result of health problems caused by air pollution, resulting in economic costs for organizations and the government.

According to a statistic, India's air pollution ranking will be eighth out of 131 countries worldwide in 2022. Chad, a nation in central Africa, has an AQI of 169, which is the highest amount of pollution. With an AQI of 164, Iraq is ranked second, and Pakistan is ranked third with an AQI of 159. With an AQI of 156, Bangladesh is ranked fifth, Burkina Faso is in sixth place, and Bahrain is in fourth place with an AQI of 157. With an AQI of 151, Kuwait comes in at number eight. The average index of air quality in India is 144. Twenty-one major Indian cities are expected to have the highest population in 2019 according to another report. According to the statistical data, air pollution in India must be tackled adequately in order to protect the environment and human lives. Originating in Wuhan, China, Covid-19 was a highly contagious disease that quickly swept around the world. On January 20, 2021, coronavirus infection caused more than 2 million deaths across the globe, having a death rate of 3.4% worldwide [1]. In response to the coronavirus infection, a nationwide lockdown of cities was suggested by the government of China after January 2020, i.e., its 1.3 billion people were staying inside their houses. Nearly all production activities, namely industries, transportation, and construction, were restricted [2]. This unexpected stagnation of trade and consumption has minimized air pollution emission, offering natural and typical test sites to predict the effects of human action controlling on air pollution reduction and control [3].

Observing the air pollution stages indicates the presence of air quality (AQ) that is measured by using sensor technologies. Measuring the amounts of carbon dioxide (CO), nitrogen dioxide (NO<sub>2</sub>),

sulfur dioxide (SO<sub>2</sub>), and ozone (O<sub>3</sub>) in the atmosphere produces the air quality index (AQI) [4]. Particulate matter is the main contaminant used to calculate the AQI (PM<sub>10</sub> and PM<sub>2.5</sub>). Air quality levels are categorized into bad, good, moderate, severe, extremely poor, and satisfactory using the AQI, which has a range of 0 to 500. The environment and public health are affected differently by each AQI level [5]. Every polluting factor has its effects and source; therefore, one can get an idea about the air-polluting sources in a region based on the maximum-level polluting particles, for instance, a higher level of NO<sub>2</sub> indicates the burning of fossil fuel in that area, as well as denoting high traffic in that place, and so on [6]. The pollutant standard index (PSI), also called the air pollution index (API) or air quality index (AQI), is an illustration that shows the concentration of different pollutants within a given range. A simple technique for evaluating the effect of the lockdowns on AQ was making comparison between the average concentration of pollutants during and before lockdown. For determining the precise value of the AQI and to identify which of the air-polluting factors were accountable for this tragedy, several sensors from various groups presently available are utilized, such as electrochemical sensors related to a chemical reaction between the electrode in gases in the air and liquid inside a sensor, a photoionization detector, and even optical sensors or optical particle counters [7].

Many studies utilized ML and artificial neural network (ANN) techniques for predicting the AQ. But owing to the difficulty of the data attributable to seasonality and trend, many methods lack effective forecasting and classification of air pollution [8]. Provided the complicated data handling capacity and learning ability of ML, the usage of ML methods has quickly amplified. But critical problems like hyperparameter tuning, data pre-processing, data splitting, and class imbalance issues were poorly addressed for optimizing the model's performance [9]. Specifically, many research works displayed low performance for the class with less observation and higher accurateness for the class with more observation; evidently, illusory accuracy was attained due to all of these problems. ML methods could offer output to nearly any given input related to training, but data pre-processing and proper hyperparameter tuning could foster the method with regard to stability, accuracy, and sensitivity [10]. There occurs a gap in the collective findings of the prevailing ML-related air pollution research because of improper optimization and data management. In recent times, DL methods have revealed a superior performance than ML on various predictive problems.

Machine learning models are being employed in statistical linear approaches to reduce their computational complexity. Support vector and random forest regression are used in nonlinear regression forecasting. However, the regression model's performance lags because of the amount of data. Therefore, by choosing the best features from the dataset for the prediction process, the complexity can be decreased. Back propagation neural networks are used in some techniques to analyze predictions. Nevertheless, those prediction models have a local minima and need long-term learning. In comparison to other approaches, the convergence pace is relatively slow. Effective hyperparameter optimization is ensured by the BSSO component, resulting in optimal model performance and eliminating the risks associated with local minima. Based on empirical data, BSSO-HDL performs better than earlier models in important performance metrics, offering more accurate and reliable predictions that are necessary for real-time air quality monitoring and decision-making.

This study presents a binary spring search optimization with hybrid DL (BSSO-HDL) for air pollution prediction and the AQI classification process during a pandemic. To transform the input data into a format that is useful, the BSSO-HDL model initially requires data pre-processing. The HDL model, which combines the development of a convolutional neural network with an extreme learning machine (CNN-ELM) technique, is used for air quality prediction and AQI classification. The BSSO

algorithm-based hyperparameter tuning procedure is approved in order to modify the BSSO-HDL model's hyperparameter standards as optimally as possible. To ensure the presentation improvements of the BSSO-HDL procedure, a detailed experimentation analysis is undertaken.

## 2. Existing air pollution monitoring schemes

Stephan et al. [11] used machine learning approaches to investigate the impact of COVID-19 on India's weather and renewable energy (RE) transitions. In this present COVID-19 crisis, the RE part helps in their low price and the Indian government has to perform procedures for running generators dependent upon renewable energy sources (RES). Unlike a fossil fuel-based power plant, RES could not be exposed to a similar supply chain disruption during this present epidemic condition.

For the goal of AQI prediction, Li et al. [12] provided multiscale entropy and a thorough ensemble empirical model decomposition. Using empirical model decomposition, the AQI data has to be broken down. The components of the intrinsic model function are also produced by employing the intrinsic mode function of the bald eagle search method. Finally, in order to achieve higher prediction performance, rat swarm optimized kernel ELM is employed. Even if this model performs better, the presented method has a rather high computational complexity.

Yang et al. [13] suggested an AQI prediction model that used a regression model to assess Beijing's and Taiyuan City's quality. For data decomposition, the variational decomposition model was originally included in the approach. Another step of decomposition was carried out for the remaining decomposed components. Ultimately, the components were recreated with greater correlation using enhanced support vector regression. Superior MSE and RMSE values were obtained with the presented technique, leading to superior prediction performances.

Sassi and Fourati [14] proposed an IoT model for AQ monitoring and forecast employing augmented reality (AR) for data visualization and DL for data analysis. Utilizing recurrent neural network (RNN) and long short-term memory (LSTM) units as a framework to use data in the AQ time-series dataset was made possible by the way the framework was constructed. Moreover, the integration of AR visualization with projected IoT techniques facilitates natural interaction between people and IoT devices, enhancing the comprehension of an AQ dataset through effective control of a more thorough analysis of data and quick decision-making processes.

Shahne et al. [15] performed a study in Mashhad, Iran, to evaluate the potential links between Covid-19 instances and deaths and AQ environments. The LSTM-based hybrid DL structure has been applied to the traffic index, influence count of mortality, active COVID-19 cases, meteorological datasets, and AQI. Tsan et al. [16] proposed using DL to investigate the relationship between confirmed COVID-19 cases and air pollution. The author used LSTM-DL to train on established COVID-19 instances and AQI limits over four different lag periods: one, three, seven, and fourteen days.

During the COVID-19 lockdown, Lovric et al. [17] used a machine learning technique to look at isolated improvements in the quality of the air in Graz, Austria. Simple historical measurement comparisons to multiple different pollutants have been effectively replaced by the machine learning approach. Indicators of true pollution during the shutdown were forecasted using the true versus predictable variance. In order to anticipate the concentration, the machine learning techniques showed a higher degree of generalization. So, this technique is appropriate to analyze decreases in pollution concentration. Tyagi et al. [18] proposed, for predicting the AQI of the Delhi area during COVID-19, utilizing time series modeling (the ML technique). Time series modeling contains methods for appropriating a gathered dataset and making use of it to forecast the future values. The investigation

was dependent upon main pollutants such as particulate matter, ozone, SO, CO, NH<sub>3</sub>, and NO.

SVM, seasonal autoregressive integrated moving average (SARIMA), and LSTM models were among the several machine learning models that Maltare et al. [19] compared and examined for Ahmedabad AQI prediction. The proposed methodology eliminated unused information and blank cells from the dataset in the preparation phase. Moreover, different classifiers were fed the preprocessed data, and their effectiveness was assessed. It is evident from experimental data that the support vector machine model outperforms other models.

In order to improve prediction accuracy, Jing et al. [20] introduced a dynamic graph neural network-based predictive model for the AQI that includes configurable edge attributes. The model parameters and edge attributes were used in the stated approach to generate a bidirected dynamic graph. As a result, during the prediction procedure, adaptive edge information was collected, improving prediction performance above traditional methods.

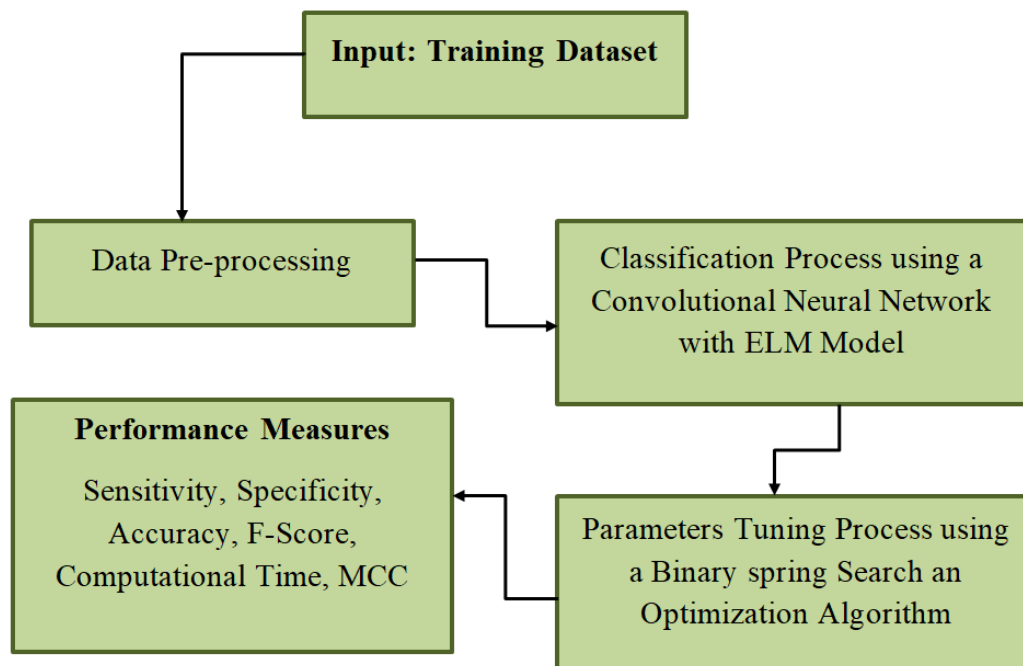
Conventional methods frequently struggle to capture the intricate temporal and spatial patterns originating in the data on air quality. This hybrid method provides a comprehensive description that leverages the capabilities of many DL models, such as CNNs for spatial analysis and LSTMs for temporal dependencies. The accumulation of an optimization technique progresses this framework by fine-tuning parameters to reach optimal performance, growing accuracy and reliability. This sophisticated predictive capability is important for providing rapid and precise air quality predictions, which are important for public health advisories, regulatory compliance, and proactive environmental management. Consequently, the hybrid tactic not only fills gaps left by traditional approaches, but it also creates a new benchmark for predictive analytics in air pollution predicting.

### 3. The proposed BSSO-HDL model

This work introduces a novel BSSO-HDL approach that is perfect for AQI classification and air pollution prediction during the widespread COVID-19 pandemic. Primarily, the BSSO-HDL model performs data pre-processing to transform raw data into a useful format. Additionally, it uses the CNN-ELM method to efficiently perform AQI prediction and classification tasks. Finally, the CNN-ELM model's hyperparameters are optimally tuned using the BSSO algorithm. Figure 1 displays the whole BSSO-HDL method procedure.

#### 3.1. Data pre-processing

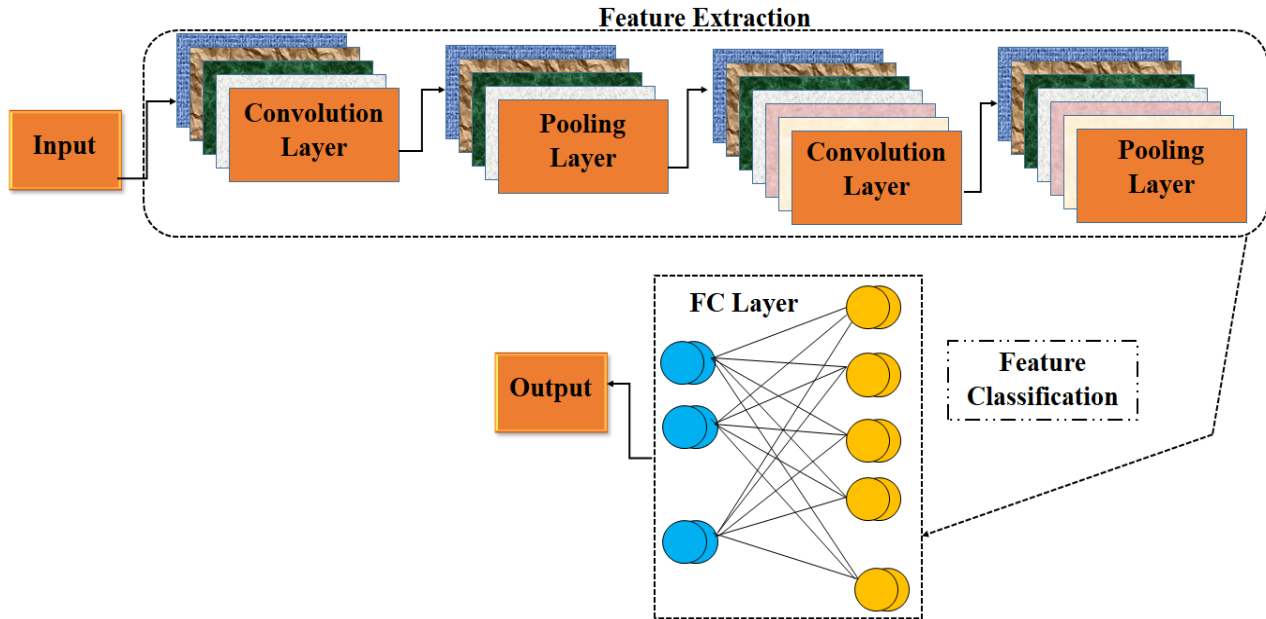
The raw information underwent the data cleaning process in order to make data ready for modeling, increase the data understanding, and handle missing values. The initial step was to understand the missing values in the datasets. In contrast, CO, NO, NO<sub>2</sub>, SO<sub>2</sub>, O<sub>3</sub>, NO<sub>x</sub>, PM 2.5, and AQI are the more prominent values of pollutants. Pandas "dropna" function is used for removing missing values, where any NA value is existing in columns or rows. Based on the substitute field in the data for avoiding redundancy, the AQI\_Bucket, field's city, date, and Year\_Month have been removed.



**Figure 1.** Overall process of the BSSO-HDL approach.

### 3.2. Prediction and AQI classification

In the BSSO-HDL model, the HDL model is applied for both prediction and classification processes and is derived by the integration of the CNN-ELM model. The HDL network contains two stages, namely classification and feature extraction. The feature extraction stage encompasses the max pooling, convolution, and normalization layers [21]. Also, it gives a detailed description of the correlation parameters, namely, the stride of every sliding window, the number of every filter, the size of every feature map, and the kernel size of every filter. For instance, the initial convolution layer comprises 96 filters, its kernel size is 7 while the extent of the feature map is  $56 \times 56$ , and the sliding window has a stride of 4. A single convolutional layer is performed after the two phases, and fully connected layers convert the feature map into 1D vectors that are advantageous to the classification. Lastly, we integrate the ELM architecture with the proposed CNN models, and later utilize a hybrid mechanism for classifying the tasks of age and gender. Figure 2 illustrates the infrastructure of the CNN-ELM classifier. Then, the design of the hybrid architecture is discussed in detail.



**Figure 2.** Framework of the CNN-ELM model.

### 3.2.1. Convolution layer

In this work, convolution is implemented between the preceding layer and a sequence of filters, extracting features from the input feature map. Generally,  $\eta_{ij}^{mn}$  represents the value of the unit at location  $(m, n)$  in the  $j^{th}$  feature map in the  $i$ -th layer, and it is formulated as follows:

$$\eta_{ij}^{mn} = \sigma \left( b_{ij} + \sum_{\delta} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ij\delta}^{pq} \eta_{(i-1)\delta}^{(m+p)(n+q)} \right). \quad (1)$$

In Eq (1),  $b_{ij}$  signifies the bias of the feature maps, whereas  $\delta$  is the index over the sequence of feature maps in the  $(i-1)th$  layers that are interconnected with the convolution layer.  $w_{ij\delta}^{pq}$  means the value at location  $(p, q)$  of kernels that are interconnected with the  $k$ -th feature maps, and the width and height of the filter kernels are  $P_i$  and  $Q_i$ , respectively.

The convolution layer provides a non-linear mapping from the lower-level depiction of the image to the higher-level semantics, and it is given by the following:

$$\eta_j = \sigma \left( \sum w_{ij} \otimes \eta_{(i-1)} \right). \quad (2)$$

In Eq (2),  $\otimes$  denotes the convolution function while  $w_{ij}$ , which is arbitrarily initialized and trained with BPNN, signifies the value of the  $i$ -th layer in the  $j$ -th feature maps.  $\eta_{(i-1)}$  indicates

the output of the  $(i - 1)$  layer and  $\eta_j$  is described by the output of the  $j$ -th feature maps in the convolution layer.

### 3.2.2. Contrast normalization layer

The proposed study aims to improve the local competition among its neighbors and one neuron as well as force the features of distinct feature maps in a similar spatial position to be calculated, which is inspired by neuroscience computation [22]. To accomplish the objective, two normalization processes, normalization, are implemented. Now,  $\eta_{mn/c}$  represents the values of the unit at location  $(m, n)$  in the  $k$ -th feature maps.

$$z_{mnk} = \eta_{mnk} - \sum_{p=-P_{i-1/2}}^{P_{i-1/2}} \sum_{q=-Q_{i-1/2}}^{Q_{i-1/2}} \sum_{j=1}^{j_i} \varepsilon_{pq} \eta_{(m+p)(n+q)j}. \quad (3)$$

In Eq (3),  $\varepsilon_{pq}$  represents a normalized Gaussian filter with  $7 \times 7$  size at the initial phase and  $5 \times 5$  size at the next phase.  $z_{mnk}$  signifies the input of the divisive normalization operation and the output of the subtractive normalization operation. The operator of divisive normalization is shown below:

$$\eta_{mnk} = \frac{z_{mnk}}{\max(M, M(m, n))}, \quad (4)$$

Where

$$M(m, n) = \sqrt{\sum_{p=-P_{i-1/2}}^{P_{i-1/2}} \sum_{q=-Q_{i-1/2}}^{Q_{i-1/2}} \sum_{j=1}^{j_i} \varepsilon_{pq} \eta_{(m+p)(n+q)j}^2}, \quad (5)$$

And

$$M = \left( \sum_{m=1}^{s1} \sum_{n=1}^{s2} M(m, n) \right) / (s1 \times s2). \quad (6)$$

In the abovementioned normalized process, the Gaussian filter  $\varepsilon_{pq}$  is evaluated using the zero-padded edges that imply the output size of the normalized operation similar to input.

### 3.2.3. Max pooling layer

In general, the pooling method aims to convert the joint feature representations into suitable ones that keep important data while discarding inappropriate information [23]. Every feature map in the subsampling layer is receiving a max pooling operation that is performed on the respective feature maps in the convolution layer. Eq (7) indicates the value of the unit at location  $(m, n)$  in the  $j$ -th feature maps in the subsampling or the  $i$ -th layers after the max pooling function:



$$\eta_{ij}^{mm} = \max \left\{ \eta_{(i-1)j}^{m-}, \eta_{(i-1)j}^{(m+1)(n+1)}, \dots, \eta_{(i-1)j}^{(m+P_1)(n+Q_i)} \right\} \quad (7)$$

The max pooling function creates location invariance over large local regions and down-sampling of the input feature map.

### 3.2.4. ELM classification layer

After the subsampling and convolution functions, we used ELM to categorize the 1D vector that is transformed from the feature map. As mentioned above, it upgrades only the output weight whereas hidden-layer bias and input weight are set at random. Therefore, it arbitrarily produces the input parameter and evaluates the output weight in the training phase. The entire procedure without an iterative process enhances the neural network generalization capability. The output (comprising  $2048 \times 1$  dimensionality) of fully linked layer is the input of ELM whereas the number of hidden neurons is the parameter that is demonstrated in this work.

Connecting the convolutional network to the ELM is another crucial method. This method uses the fully connected layer's output as the input for the ELM, which comes preceding the convolutional layer [24]. Backward and forward propagation functions are the fundamental parts of the hybrid model and are thoroughly analyzed in the subsequent section.

### 3.3. Hyperparameter tuning

In the BSSO-HDL model, the BSSO algorithm-based hyperparameter tuning process gets executed for the HDL model. The BSSO is a physics-based optimization approach that is used for solving different optimization challenges [25]. The presented method has a population matrix whose member is distinct weights that are moved in the searching space for achieving an optimum solution. Each desired weight is interconnected to one another in these systems via a unique spring whose stiffness coefficient can be defined according to the objective function value. The major conception is to utilize Hooke's law among the springs and weights for accomplishment of the equilibrium opinion.

Hooke's law can be determined by Eq (8).

$$F_s = -kx \quad (8)$$

The spring force is denoted by  $F_s$  in Eq. (8), the spring constant is denoted by  $k$ , and the spring density, or stretch, is shown by  $\chi$ .

Here, based on Hooke's law, which is the same as the populace-based algorithm, the mathematical formula of the BSSO is modeled. The BSSO has a populace matrix where every row signifies a population associated as a weight. Therefore, each population member refers to a vector, whereby every vector component defines the parameter value of the optimization issue. In this study, every population member is presented as follows.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^m) \text{ for } i = 1, 2, \dots, N \quad (9)$$

In Eq (9),  $X_i$  indicates the  $i$ -th members of the population matrix,  $x_i^d$  denotes the status of the  $d$ -th dimensions of the  $i$ -th members of the population matrices,  $m$  represents the number of parameters, and  $N$  shows the amount of population members [26]. The first location of every

population member is considered at random in the searching space. Next, with the force that the spring applies on the weight, the population member moves in the search space and it is upgraded in all of the iterations as follows.

$$K_{i,j} = K_{\max} |F_n^i - F_n^j| \max(F_n^i, F_n^j) \quad (10)$$

Here,  $K_{i,j}$  indicates the spring constant that connects  $i$  to weight  $j$ ,  $K_{\max}$  shows the maximal worth of the spring constant (the worth is 1), and  $F_n$  indicates the regularized impartial purpose, where  $F_n^i$  denotes a normalized objective function for the  $i$ -th members. It is shown below:

$$F_n^i = \frac{f_{obj}^i}{\min(f_{obj})} \quad (11)$$

$$F_n^i = \frac{\min(F_n^i)}{F_n^i} \quad (12)$$

Here,  $f_{obj}$  refers to the vector of objective purpose, where  $f_{obj}^i$  indicates the objective purpose for the  $i$ -th members. An  $m$ -parameter problem has an  $m$ -dimension search space. The fixed point for a member is a member who has the best objective function when compared to others. This makes two single forces to be employed to every associate on every alliance from the right and left, which is defined in the following:

$$F_{total_R}^{j,d} = \sum_{i=1}^{n_R^d} K_{i,j} x_{i,j}^d \quad (13)$$

$$F_{total_L}^{j,d} = \sum_{l=1}^{n_L^d} K_{l,j} x_{l,j}^d \quad (14)$$

Now,  $F_{total_R}^{j,d}$  and  $F_{total_L}^{j,d}$  represent the overall forces exerted on the  $d$ -th dimensions of the  $i$ -th members of the populace from the left and right,  $n_R^d$  shows the amount of fixed points on the right in the  $d$ -th dimension or axis, and  $n_L^d$  indicates the amount of fixed points on the left in the  $d$ -th dimension or axis. It is evaluated as follows:

$$dX_R^{j,d} = \frac{F_{total_R}^{j,d}}{K_{equal_R}^j} \quad (15)$$

$$dX_L^{j,d} = \frac{F_{total_L}^{j,d}}{K_{equal_L}^j} \quad (16)$$

Let  $dX_R^{j,d}$  and  $dX_L^{j,d}$  be the displacement count of the right and left side for the  $j$ -th members

in the  $d$ -th dimension or axis. In such cases, the last displacement values are evaluated by combining Eqs (15) and (16) based on Eq (17).

$$dX^{j,d} = dX_R^{j,d} + dX_L^{j,d} \quad (17)$$

In Eq (17),  $dX^{j,d}$  indicates the last displacement for the  $j$ -th members in the  $d$ -th dimension or axis [27]. After defining the displacement count, the novel location of every member in the searching space is upgraded as follows.

$$X^{j,d} = X_0^{j,d} + r_1 \times dX^{j,d} \quad (18)$$

In Eq (18),  $X_0^{j,d}$  indicates the preceding location of the  $j$ -th members in the  $d$ -th dimension or axis, and  $r_1$  shows an arbitrary value with a standard distribution between [0–1]. The different steps of executing the BSSO are formulated below:

Begin

Step 1: Define the problem and its search space.

Step 2: Generate a random starting population.

Step 3: Normalize and evaluate the objective function.

Step 4: Upgrade the spring constant.

Step 5: Hooke's law is used to compute the amount of displacement to the left and right.

Step 6: Evaluate last displacement.

Step 7: Update population.

Step 8: Repeat Steps 3–7 until the stopping criteria are met.

Step 9: Return the optimal solution for the goal function.

End

### 3.3.1. Binary spring search algorithm (BSSA)

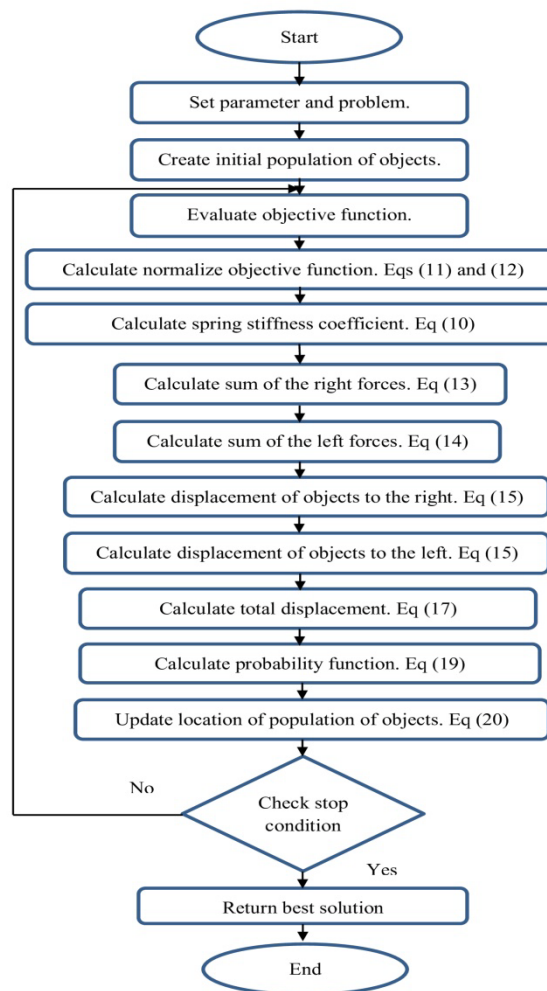
The binary version of the spring search technique is presented in this section. Real values are represented in SSA using binary digits (zero and one) in a binary format. Because the search space is discrete, there needs to be a sufficient number of binary values assigned to each variable along the axis. Since there are only two potential values for a binary representation, displacement is the process of altering a value from zero to one or from one to zero. Applications utilizing the binary form of the displacement concept depend on probability functions. Depending on the amount that this probability function changes, each member's new places in each problem dimension could either increase or decrease. In the BSSA,  $dX^{j,d}$  denotes the likelihood of  $X^{j,d}$  reaching zero or one. Both the binary and real versions follow the same methods for computing spring forces, figuring out spring constant values, displacement per population member, and update steps. The main difference is in the way they are updated in comparison to the population. Equation (19) states that for every member, the probability function, which is limited between zero and one, determines the probability of a dimensional change.

$$S(dX^{i,d}(t)) = |\tanh(dX^{i,d}(t))| \quad (19)$$

Equation (20), therefore, adjusts each member's new dimension position according to the probability function values.

$$\begin{aligned} \text{If } rand < S(dX^{i,d}(t)) \text{ Then } X^{i,d}(t+1) &= \text{complement}(X^{i,d}(t)) \\ \text{Else } X^{i,d}(t+1) &= X^{i,d}(t) \end{aligned} \quad (20)$$

Given Eq (20), there is a probability associated with each individual in a population moving. The greater the value of  $dX^{j,d}$ , the higher the likelihood of object  $j$  moving in dimension  $d$ .  $rand$  is a random number having a normal distribution in the range of [0–1]. Figure 3 depicts the multiple BSSA steps as a flowchart.



**Figure 3.** A diagram of the binary spring search algorithm.

Examine the subsequent standard function to demonstrate that the suggested approach looks for the most effective solution:

$$f(x) = \sum_{i=1}^2 x_i^2 \quad (21)$$

### 3.3.2. Features of the BSSA

The spring force law was simulated in order to develop a new optimizer inside the suggested BSSA. The BSSA approach defines population members as interconnected weights that navigate the problem space. It is made simpler for these people to share knowledge because of the spring force mechanism. Each member has a rough idea of its surroundings, impacted by the locations of other things. To improve the population member arrangement over multiple iterations, an optimization technique has been devised. This is accomplished by adjusting the spring stiffness coefficient in the intervals between algorithmic rounds. Springs with a greater stiffness coefficient attract other objects by aligning with those that perform better fitness functions. Any object receives a force according to its size. Superior circumstances call for slower, shorter steps from objects. In order to do this, bigger weights are paired with springs that have a higher stiffness coefficient. Because of this mechanism, weights with an enlarged fitness function investigate their surroundings more thoroughly. The springs' force and stiffness coefficients steadily diminish over time. As a result, objects tend to concentrate around ideal locations, necessitating faster and more precise region identification. Over time, the spring's rigidity decreases.

### 3.3.3. Exploration and exploitation of the BSSA

Exploitation power and exploration power are important considerations when determining which optimization algorithms are suitable for solving particular optimization problems. The capacity of an optimization algorithm to identify the most effective solution is measured by its exploitation power. As a matter of fact, the potential for exploitation is higher for an algorithm that might produce a solution that is more like the original one. The exploration index is useful for determining the speed that an optimization algorithm can travel a predefined search space for a particular task. This becomes more important in domains with several local optimums. Consequently, the population of an algorithm that is able to systematically search the whole search space can be directed toward the central optimal areas and away from neighboring suboptimal parts. These criteria emphasize how crucial it is that early iterations of optimization algorithms have robust exploratory capabilities in order to examine a variety of regions in the search space. To attain the intended results, the algorithm's exploitation capabilities must be modified as it moves closer to its final iterations [28–29].

Using the appropriate population members as a basis, the BSSA can precisely scan the search space. The BSSA balances the two crucial indicators of exploration and exploitation by looking at the spring constant as the primary parameter. For allowing people within the population to apply Hooke's law and spring force to explore various regions of the search space, the BSSA's spring constant equation is initially created with large values. The spring constant has decreasing values as the algorithm's iterations increase and approach the final iterations; this closer examination of potential optimum sites provides the greatest feasible result. Eq (18) uses the previously mentioned method to modify the spring constant while preserving the power split between exploration and exploitation.

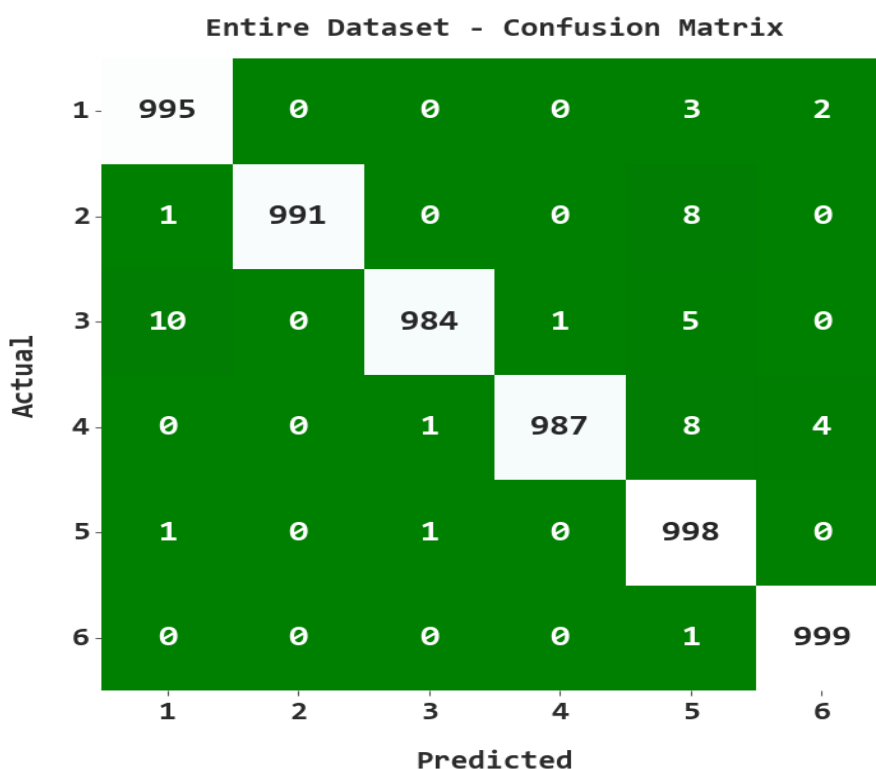
#### 4. Experimental validation

The air quality monitoring results of the BSSO-HDL model are tested using a series of simulations. We have used a dataset comprising 6000 samples under six class labels. Table 1 shows that there are 1000 samples in the dataset for each class.

**Table 1.** Dataset description.

AQI	Class	Description	No. of Samples
0–50	1	Good	1000
51–100	2	Satisfactory	1000
101–200	3	Moderate	1000
201–300	4	Poor	1000
301–400	5	Very Poor	1000
401–500	6	Severe	1000
Number of Samples			6000

Figure 4 reports the confusion matrix of BSSO-HDL generated for the whole dataset. According to the number, the BSSO-HDL model has identified 995 instances from period 1, 991 from period 2, 984 from period 3, 987 from period 4, 998 from period 5, and 999 from period 6.



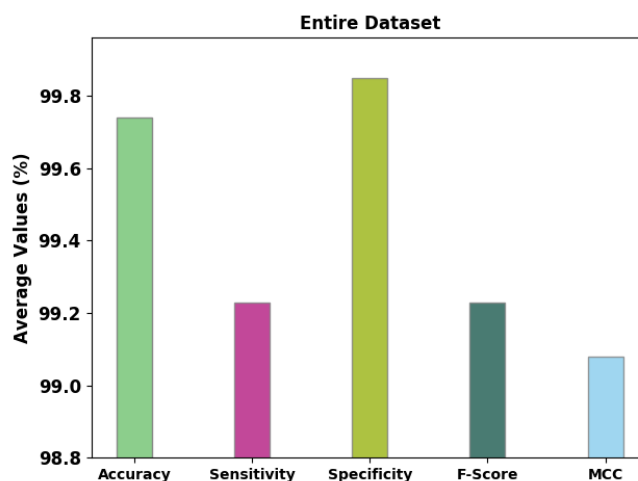
**Figure 4.** Confusion matrix of the BSSO-HDL method under the entire dataset.

The classification performance accomplished by the BSSO-HDL algorithm on the entire dataset of air quality monitoring is portrayed in Table 2 and Figure 5. The BSSO-HDL algorithm has identified

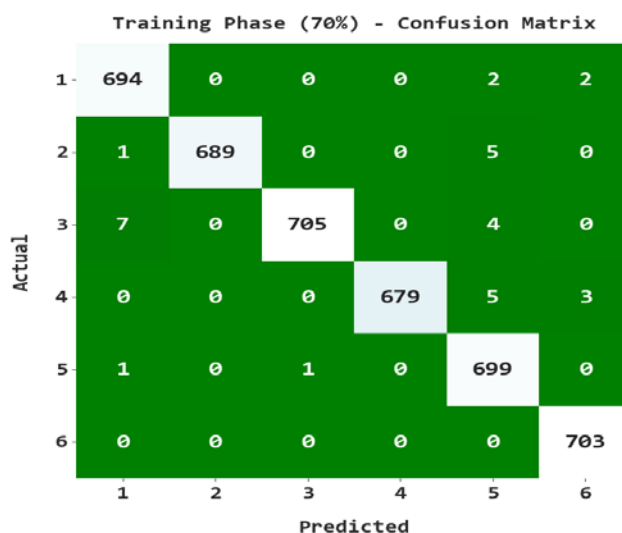
samples under class 1 with  $accu_y$  of 99.72%,  $sens_y$  of 99.50%,  $spec_y$  of 99.76%,  $F_{score}$  of 99.15%, and MCC of 98.98%. Eventually, the BSSO-HDL algorithm detected samples under class 2 with  $accu_y$  of 99.85%,  $sens_y$  of 99.10%,  $spec_y$  of 100%,  $F_{score}$  of 99.55%, and MCC of 99.46%. Next to that, the BSSO-HDL algorithm has differentiated samples under class 5 with  $accu_y$  of 99.55%,  $sens_y$  of 99.80%,  $spec_y$  of 99.50%,  $F_{score}$  of 98.67%, and MCC of 98.40%.

**Table 2.** Analysis of the BSSO-HDL method's results for each class within the whole dataset.

Entire Dataset					
Labels	Accuracy	Sensitivity	Specificity	F-Score	MCC
1	99.72	99.50	99.76	99.15	98.98
2	99.85	99.10	100.00	99.55	99.46
3	99.70	98.40	99.96	99.09	98.92
4	99.77	98.70	99.98	99.30	99.16
5	99.55	99.80	99.50	98.67	98.40
6	99.88	99.90	99.88	99.65	99.58
Average	99.74	99.23	99.85	99.23	99.08



**Figure 5.** Overall dataset average analysis using the BSSO-HDL method.



**Figure 6.** Confusion matrix utilizing 70% TR data and the BSSO-HDL technique.

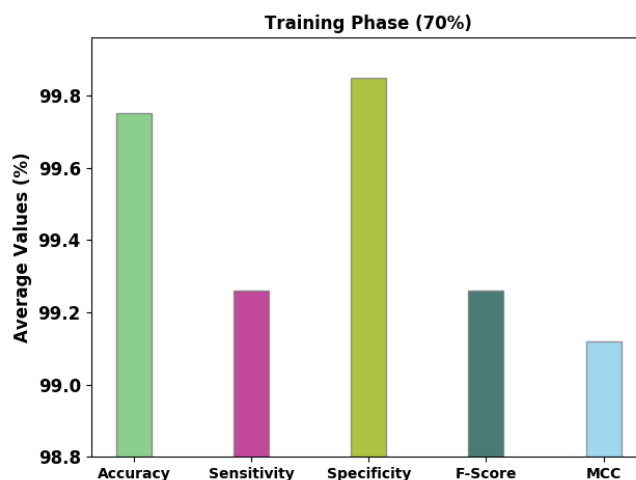
Figure 6 presents the confusion matrix that the BSSO-HDL model generated for the full 70% of the TR dataset. According to the graph, the BSSO-HDL model recognized 694 models in period 1, 689 models in period 2, 705 examples in period 3, 679 models in period 4, 699 models in period 5, and 705 models in period 6.

**Table 3.** Analysis of the BSSO-HDL technique's results using various class labels when less than 70% of the TR information are present.

Training Phase (70%)					
Labels	Accuracy	Sensitivity	Specificity	F-Score	MCC
1	99.69	99.43	99.74	99.07	98.89
2	99.86	99.14	100.00	99.57	99.48
3	99.71	98.46	99.97	99.16	98.99
4	99.81	98.84	100.00	99.41	99.30
5	99.57	99.71	99.54	98.73	98.48
6	99.88	100.00	99.86	99.65	99.58
Average	99.75	99.26	99.85	99.26	99.12

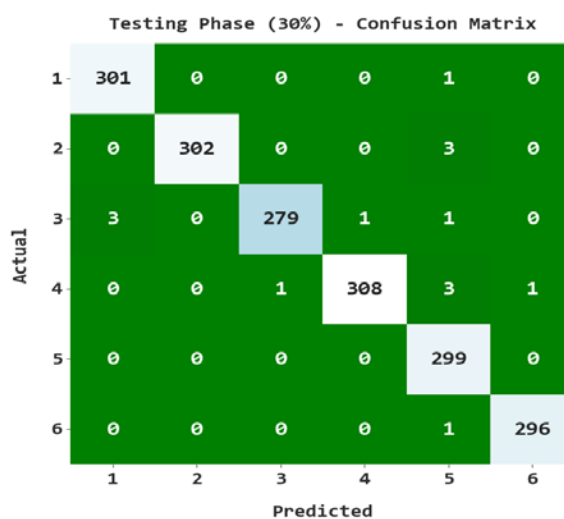
The classification performance offered by the BSSO-HDL model on 70% of the TR dataset of air quality monitoring is reported in Table 3 and Figure 7. The BSSO-HDL model has standard examples under class 1 with  $accu_y$  of 99.69%,  $sens_y$  of 99.43%,  $spec_y$  of 99.74%,  $F_{score}$  of 99.07%, and MCC of 98.89%. In time, the BSSO-HDL technique has distinguished samples under class 2 with  $accu_y$  of 99.86%,  $sens_y$  of 99.14%,  $spec_y$  of 100%,  $F_{score}$  of 99.57%, and MCC of 99.48%. Following this, the BSSO-HDL technique has differentiated examples under class 5 with  $accu_y$  of 99.57%,  $sens_y$  of 99.71%,  $spec_y$  of 99.54%,  $F_{score}$  of 98.73%, and MCC of 98.48%.





**Figure 7.** Average BSSO-HDL technique analysis with 70% of TR data.

Figure 8 displays the confusion matrix that the BSSO-HDL model generated on 30% of the TS dataset. The BSSO-HDL model determined that there were 301 examples in class 1, 302 examples in lesson 2, 279 examples in class 3, 308 examples in class 4, 299 examples in class 5, and 296 examples in class 6, as shown on the graph.

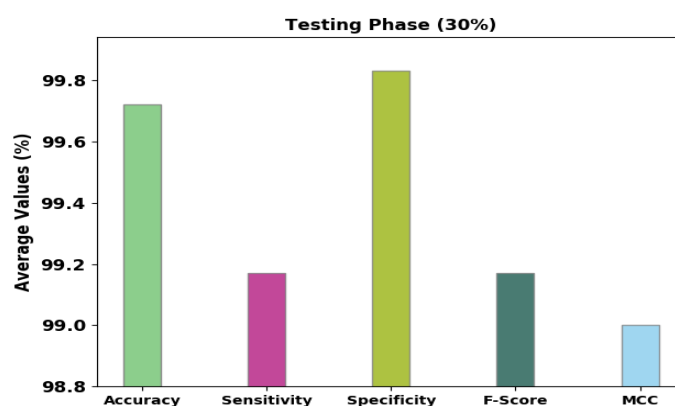


**Figure 8.** Confusion matrix of the BSSO-HDL method under 30% of TS information.

The classification performance gained by the BSSO-HDL model on 30% of the TS dataset of air quality monitoring is given in Table 4 and Figure 9. The BSSO-HDL technique has found samples under class 1 with  $accu_y$  of 99.78%,  $sens_y$  of 99.67%,  $spec_y$  of 99.80%,  $F_{score}$  of 99.34%, and MCC of 99.21%. Ultimately, the BSSO-HDL technique has detected samples under class 2 with  $accu_y$  of 99.83%,  $sens_y$  of 99.02%,  $spec_y$  of 100%,  $F_{score}$  of 99.51%, and MCC of 99.41%. At last, the BSSO-HDL method has distinguished examples under class 5 with  $accu_y$  of 99.50%,  $sens_y$  of 100%,  $spec_y$  of 99.40%,  $F_{score}$  of 98.52%, and MCC of 98.23%.

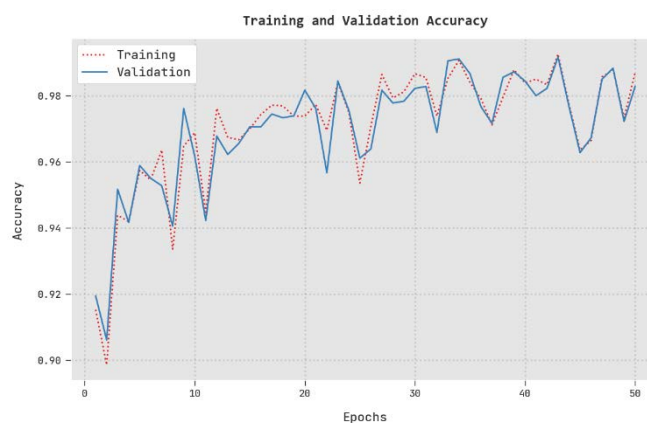
**Table 4.** Results of the BSSO-HDL method with various class labels within 30% of the TS data.

Testing Phase (30%)					
Labels	Accuracy	Sensitivity	Specificity	F-Score	MCC
1	99.78	99.67	99.80	99.34	99.21
2	99.83	99.02	100.00	99.51	99.41
3	99.67	98.24	99.93	98.94	98.74
4	99.67	98.40	99.93	99.04	98.84
5	99.50	100.00	99.40	98.52	98.23
6	99.89	99.66	99.93	99.66	99.60
Average	99.72	99.17	99.83	99.17	99.00

**Figure 9.** Average analysis of BSSO-HDL method under 30% of TS information.

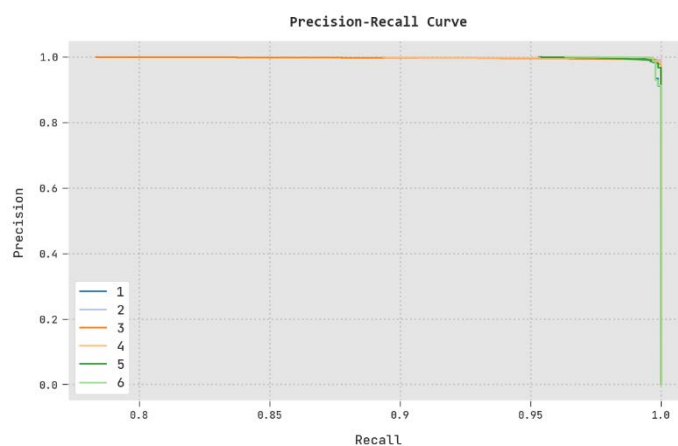
The training accuracy (TRA) and validation accuracy (VLA) on test data acquired with the BSSO-HDL process are shown in Figure 10. The experimental findings show that the maximum VLA and TRA values were obtained using the BSSO-HDL approach. The VLA, in particular, appeared to be bigger than the TRA.

The validation loss (VLL) and training loss (TRL) achieved on test data using the BSSO-HDL model are shown in Figure 11. According to the experimental data, the BSSO-HDL technique achieved the lowest TRL and VLL requirements. In particular, the VLL is lower than the TRL.

**Figure 10.** TRA and VLA analysis of the BSSO-HDL method.



**Figure 11.** TRL and VLL analysis of the BSSO-HDL method.



**Figure 12.** Precision-recall analysis of the BSSO-HDL method.

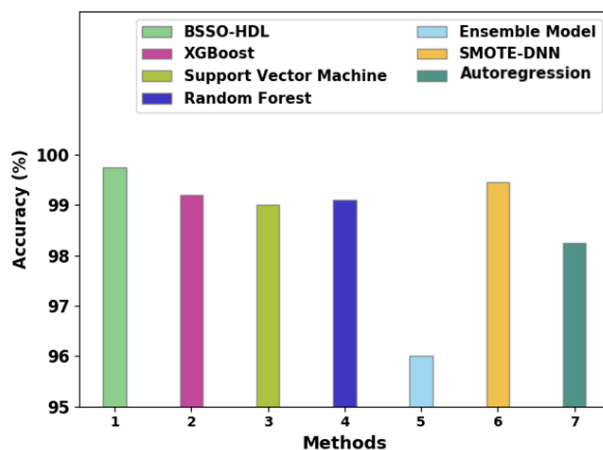
Using test data, Figure 12 illustrates an important precision-recall aspect for the BSSO-HDL technique. The results show that, in every class, the BSSO-HDL approach produced better precision-recall metrics.

**Table 5.** Comparison of the BSSO-HDL method with existing procedures.

Models	$accu_y$ (%)
BSSO-HDL	99.75
XGBoost	99.20
Support Vector Machine	99.00
Random Forest	99.10
Ensemble Model	96.00
SMOTE-DNN	99.45
Autoregression	98.25

To ensure better consequences of the BSSO-HDL model, a comparative  $accu_y$  examination is made with other existing models in Table 5 and Figure 13. The experimental values imply that the ensemble algorithm has achieved the smallest  $accu_y$  value of 96%, whereas the autoregression

algorithm has achieved a slightly raised  $accu_y$  of 98.25%. Along with that, the XGBoost, SVM, and RF algorithms have resulted in closer  $accu_y$  values of 99.20%, 99%, and 99.10%, respectively. Meanwhile, the SMOTE-DNN algorithm has accomplished a reasonable  $accu_y$  value of 99.45%. However, the presented BSSO-HDL algorithm has showcased better performance with an increased  $accu_y$  of 98.25%.

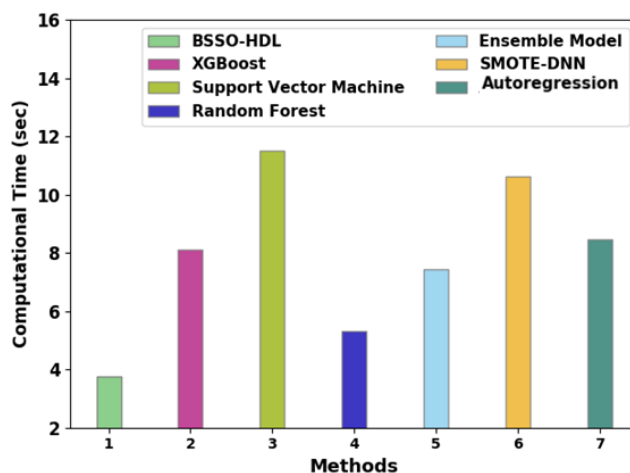


**Figure 13.** Comparative analysis of the BSSO-HDL method with existing procedures.

Finally, Table 6 and Figure 14 exhibit modern techniques together with a brief comparison of computation time (CT) utilizing the BSSO-HDL methodology. The obtained values inferred that the SVM and SMOTE-DNN algorithms have reached lower CT of 11.50s and 10.61s, respectively. Meanwhile, the XGBoost and autoregression algorithms have resulted in slightly decreased CT of 8.12s and 8.46s. Concurrently, the ensemble algorithm has gained reasonable CT of 7.44s. Although the RF algorithm has accomplished near optimal CT of 5.30s, the BSSO-HDL algorithm has provided lower CT of 3.75s. The detailed experimental results represent the supremacy of the BSSO-HDL model over other ML models.

**Table 6.** CT analysis of the BSSO-HDL method with existing methodologies.

Methods	Computational Time (sec)
BSSO-HDL	3.75
XGBoost	8.12
Support Vector Machine	11.50
Random Forest	5.30
Ensemble Model	7.44
SMOTE-DNN	10.61
Autoregression	8.46



**Figure 14.** CT analysis of the BSSO-HDL method.

The performance of the prediction model can be assessed using the MAE, RMSE, and  $R^2$  statistics. These metrics can evaluate the degree of data change and accuracy as well as the predictive power of sophisticated machine learning models. The calculation equation is

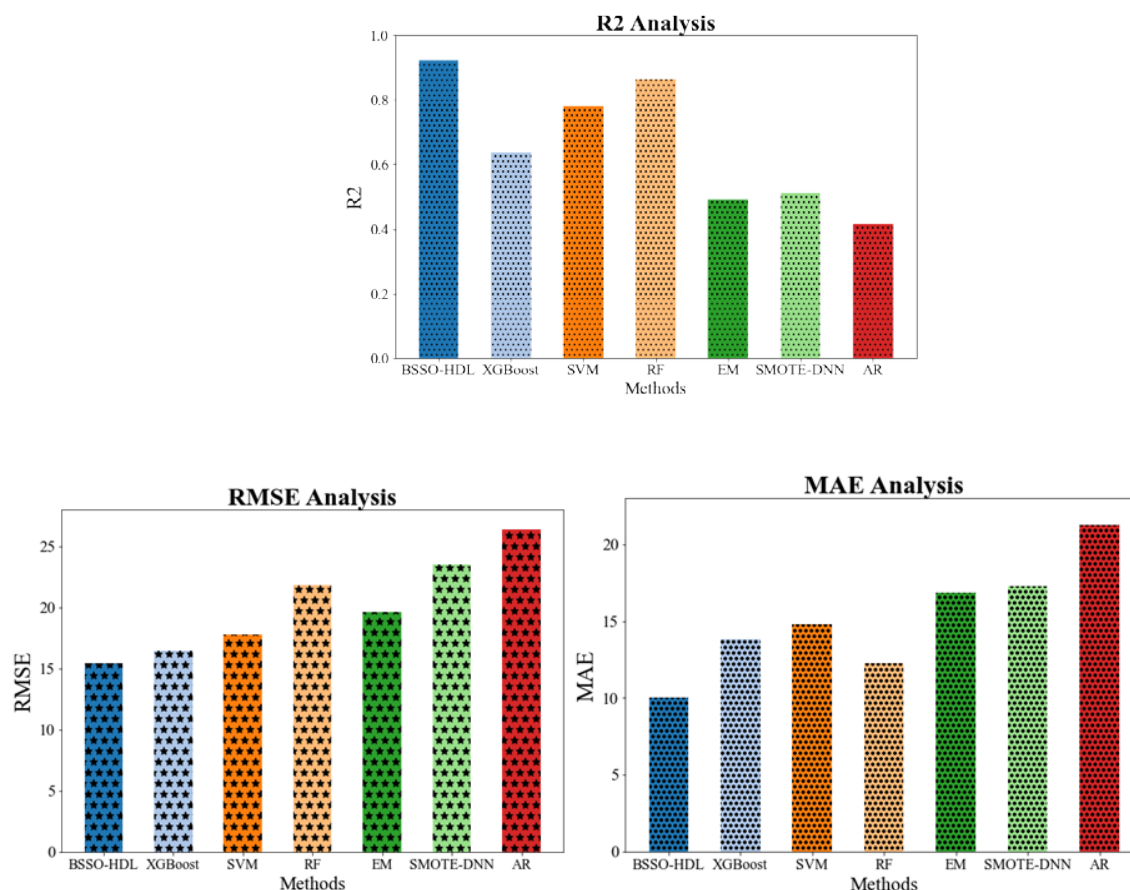
$$MAE = \frac{1}{T} \sum_{i=1}^T |x - y_i| \quad (22)$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{i=1}^T (x - y_i)^2} \quad (23)$$

$$R^2 = 1 - \frac{\sum_{i=1}^T (x - y_i)^2}{\sum_{i=1}^T (x - \bar{x})^2} \quad (24)$$

**Table 7.** Comparative analysis of the BSSO-HDL approach with existing methodologies.

Methods	$R^2$	RMSE	MAE
BSSO-HDL	0.922	15.422	10.029
XGBoost	0.635	16.439	13.823
Support Vector Machine (SVM)	0.781	17.826	14.823
Random Forest (RF)	0.865	21.826	12.285
Ensemble Model (EM)	0.4912	19.625	16.826
SMOTE-DNN	0.5122	23.527	17.273
Autoregression (AR)	0.4144	26.425	21.28



**Figure 15.** Comparative analysis of the BSSO-HDL approach.

Table 7 and Figure 15 demonstrate an extensive  $R^2$  evaluation of the BSSO-HDL technique using alternative models. With a higher  $R^2$  value of 0.922, these results show that the BSSO-HDL approach was improved. The  $R^2$  values generated by the subsequent algorithms are as follows: XGBoost (0.635), random forest (RF) (0.865), SMOTE-DNN (0.5122), autoregression (AR) (0.4144), ensemble model (EM) (0.4912), and support vector machine (SVM) (0.781).

The above figure presents a complete MAE and RMSE study of the BSSO-HDL technique with contemporary algorithms. The figure demonstrates improved performance with minimal MAE and RMSE values for the BSSO-HDL approach. The XGBoost, support vector machine (SVM), random forest (RF), ensemble model (EM), SMOTE-DNN, and autoregression (AR) techniques have produced maximum MAEs of 13.823, 14.823, 12.285, 16.826, 17.273, and 21.28, respectively, in comparison to the BSSO-HDL approach's minimal MAE of 10.029. The BSSO-HDL algorithm has also achieved the lowest RMSE of 15.422 in terms of RMSE, while the approaches of XGBoost, support vector machine (SVM), random forest (RF), ensemble model (EM), SMOTE-DNN, and autoregression (AR) have produced the highest RMSE of 16.439, 17.826, 21.826, 19.625, 23.527, and 26.425, respectively.

## 5. Conclusion

In conclusion, a major development in the domain of environmental monitoring and forecasting is the hybrid deep learning-based air pollution prediction and index classification using an optimization algorithm. This technique provides greater accuracy and robustness in air pollution level prediction

and air quality index classification by combining the features of multiple deep learning architectures with an efficient optimization algorithm. In the obtainable BSSO-HDL model, an HDL-based air quality prediction and AQI classification model is applied in which the HDL is derived by the use of a CNN-ELM model. To optimally change the hyperparameter standards of the BSSO-HDL model, the BSSO algorithm-based hyperparameter tuning procedure was executed. The results of the experiment designate that the BSSO-HDL model performs well in predictive classification. Nevertheless, one disadvantage of the BSSO-HDL method is its complexity, which results in greater computing costs and longer training durations, thus limiting its utility for real-time applications in resource-constrained contexts. Moreover, relying on huge amounts of high-quality data for training may limit the model's usefulness in areas with sparse or inaccurate data. The black-box aspect of deep learning models also complicates interpretability, making it problematic for stakeholders to understand and trust the predictions. The experimental results show that the BSSO-HDL model has good prediction and classification performance. This Python-based model is evaluated using the  $R^2$ , MAE, and RMSE error measures, with an  $R^2$  of 0.922, RMSE of 15.422, and MAE of 10.029. Future research could look into developments to the proposed BSSO-HDL model or the making of hybrid models that combine the qualities of multiple methods. Comparative studies incorporating additional datasets and numerous modeling approaches may yield more robust conclusions on the recommended model's efficiency.

### Use of AI tools declaration

The author declares they have not used Artificial Intelligence (AI) tools in the creation of this article.

### Acknowledgments

Sreenivasulu Kutala: Methodology, Validation, Investigation; Harshavardhan Awari: Conceptualization, Supervision, Project administration; Sangeetha Velu: Validation, Investigation, Writing—review and Editing; Arun Anthonisamy: Formal analysis, Data curation, Writing—review and Editing; Naga Jjyothi Bathula: Methodology, Writing—original draft preparation; Syed Inthiyaz: Conceptualization, Resources; All authors have read and approved the final version of the manuscript for publication.

### Conflict of interest

We declare that there is no conflict of interest regarding this research work.

### References

1. Rahman M M, Paul K C, Hossain M A, et al. (2021) Machine Learning on the COVID-19 Pandemic, Human Mobility and Air Quality: A Review. *IEEE Access* 9: 72420–72450. <https://doi.org/10.1109/ACCESS.2021.3079121>
2. Xing X, Xiong Y, Yang R, et al. (2021) Predicting the effect of confinement on the COVID-19 spread using machine learning enriched with satellite air pollution observations. *Proc Natl Acad Sci* 118: 33. <https://doi.org/10.1073/pnas.2109098118>

3. Sethi J K, Mittal M (2020) Monitoring the Impact of Air Quality on the COVID-19 Fatalities in Delhi, India: Using Machine Learning Techniques. *Disaster Med Public Health Prep* 6: 604-611. <https://doi.org/10.1017/dmp.2020.372>
4. Yang J, Wen Y, Wang Y, et al. (2021) From COVID-19 to future electrification: Assessing traffic impacts on air quality by a machine-learning model. *P Nati A Sci* 118: e2102705118. <https://doi.org/10.1073/pnas.2102705118>
5. Rybarczyk Y, Zalakeviciute R (2021) Assessing the COVID-19 Impact on Air Quality: A Machine Learning Approach. *Geophysl Res Lett* 48: e2020GL091202. <https://doi.org/10.1029/2020GL091202>
6. Liu H, Yue F, Xie Z (2022) Quantify the role of anthropogenic emission and meteorology on air pollution using machine learning approach: A case study of PM<sub>2.5</sub> during the COVID-19 outbreak in Hubei Province, China. *Environ Pollut* 300: 118932. <https://doi.org/10.1016/j.envpol.2022.118932>
7. Gatti R C, Velichevskaya A, Tateo A, et al. (2020) Machine learning reveals that prolonged exposure to air pollution is associated with SARS-CoV-2 mortality and infectivity in Italy. *Environ Pollut* 267: 115471. <https://doi.org/10.1016/j.envpol.2020.115471>
8. Gao M, Yang H, Xiao Q, et al. (2022) COVID-19 lockdowns and air quality: Evidence from grey spatiotemporal forecasts. *Socio-Econ Plan Sci* 83: 101228. <https://doi.org/10.1016/j.seps.2022.101228>
9. Wijnands J S, Nice K A, Seneviratne S, et al. (2022) The impact of the COVID-19 pandemic on air pollution: A global assessment using machine learning techniques. *Atmos Pollut Res* 13: 101438. <https://doi.org/10.1016/j.apr.2022.101438>
10. Wibowo F W (2021) Prediction of air quality in Jakarta during the COVID-19 outbreak using long short-term memory machine learning. *IOP Conference Series: Earth and Environmental Science* 704: 012046. <https://doi.org/10.1088/1755-1315/704/1/012046>
11. Stephan T, Al-Turjman F, Ravishankar M, et al. (2022) Machine learning analysis on the impacts of COVID-19 on India's renewable energy transitions and air quality. *Environ Sci Pollut Res* 29: 79443–79465. doi: 10.1007/s11356-022-20997-2. <https://doi.org/10.1007/s11356-022-20997-2>
12. Li G, Tang Y, Yang H (2022) A new hybrid prediction model of air quality index based on secondary decomposition and improved kernel extreme learning machine. *Chemosphere* 305: 135348. <https://doi.org/10.1016/j.chemosphere.2022.135348>
13. Yang H, Zhang Y, Li G (2023) Air quality index prediction using a new hybrid model considering multiple influencing factors: A case study in China. *Atmos Pollut Res* 14: 1016777. <https://doi.org/10.1016/j.apr.2023.101677>
14. Sassi M S H, Fourati L C (2021) Deep Learning and Augmented Reality for IoT-based Air Quality Monitoring and Prediction System. *IEEE* 2021. <https://doi.org/10.1109/ISNCC52172.2021.9615639>
15. Shahne M Z, Sezavar A, Najibi F (2022) A hybrid deep learning model to forecast air quality data based on COVID-19 outbreak in Mashhad, Iran. *Ann Civ Environ Eng* 6: 019–025. <https://doi.org/10.29328/journal.acee.1001035>
16. Tsan Y T, Kristiani E, Liu P Y, et al. (2022) In the Seeking of Association between Air Pollutant and COVID-19 Confirmed Cases Using Deep Learning. *Int J Environl Res Pub He* 19: 6373. <https://doi.org/10.3390/ijerph19116373>



17. Lovrić M, Pavlović K, Vuković M, et al. (2021) Understanding the true effects of the COVID-19 lockdown on air pollution by means of machine learning. *Environ Pollut* 274: 115900. <https://doi.org/10.1016/j.envpol.2020.115900>
18. Tyagi A, Gaur L, Singh G, et al. (2022) Air Quality Index (AQI) Using Time Series Modelling During COVID Pandemic. *Lect Notes Electr Eng* 2022: 441–452. [https://doi.org/10.1007/978-981-16-8546-0\\_36](https://doi.org/10.1007/978-981-16-8546-0_36)
19. Maltare N N, Vahora S (2023) Air quality index prediction using machine learning for Ahmedabad city. *Digital. Chemical. Engineering* 7: 100093. <https://doi.org/10.1016/j.dche.2023.100093>
20. Xu J, Wang S, Ying N, et al. (2023) Dynamic graph neural network with adaptive edge attributes for air quality prediction: A case study in China. *Heliyon* 9: 17746. <https://doi.org/10.1016/j.heliyon.2023.e17746>
21. Ghoneim A, Muhammad G, Hossain M S (2020) Cervical cancer classification using convolutional neural networks and extreme learning machines. *Future Gener Comp Sy* 102: 643–649. <https://doi.org/10.1016/j.future.2019.09.015>
22. Dehghani M, Montazeri Z, Dehghani A, et al. (2021) Binary Spring Search Algorithm for Solving Various Optimization Problems. *Appl Sci* 11: 1286. <https://doi.org/10.3390/app11031286>
23. Kamalraj R, Neelakandan S, Kumar M R, et al. (2021) Interpretable filter based convolutional neural network (IF-CNN) for glucose prediction and classification using PD-SS algorithm. *Measurement* 183: 109804. <https://doi.org/10.1016/j.measurement.2021.109804>
24. Kavitha T, Mathai P P, Karthikeyan C, et al. (2021) Deep Learning Based Capsule Neural Network Model for Breast Cancer Diagnosis Using Mammogram Images. *Interdiscip Sci* 2021: 1-17. <https://doi.org/10.1007/s12539-021-00467-y>
25. Reshma G, Al-Atroshi C, Nassa V K, et al. (2022) Deep Learning-Based Skin Lesion Diagnosis Model Using Dermoscopic Images. *Intell Autom Soft Co* 31: 621–634. <https://doi.org/10.32604/iase.2022.019117>
26. Harshavardhan A, Boyapati P, Neelakandan S, et al. (2022) LSGDM with Biogeography-Based Optimization (BBO) Model for Healthcare Applications. *J Healthc Eng* 2022: 1–11. <https://doi.org/10.1155/2022/2170839>
27. Neelakandan S, Beulah J R, Prathiba L, et al. (2022) Blockchain with deep learning-enabled secure healthcare data transmission and diagnostic model. *Int J Model Simul Sc* 13: 2241006. <https://doi.org/10.1142/S1793962322410069>
28. Mao W, Wang W, Jiao L, et al. (2020) Modeling air quality prediction using a deep learning approach: Method optimization and evaluation. *Sustain Cities Soc* 65: 102567. <https://doi.org/10.1016/j.scs.2020.102567>
29. Jurado X, Reiminger N, Benmoussa M, et al. (2022). Deep learning methods evaluation to predict air quality based on Computational Fluid Dynamics. *Expert System Applications* 203: 117294. <https://doi.org/10.1016/j.eswa.2022.117294>

