**Bulletin of Electrical Engineering and Informatics**
Vol. 14, No. 2, April 2025, pp. 1320∼1329
ISSN: 2302-9285, DOI: 10.11591/eei.v14i2.8958

# POA-DT: a novel method for predicting air quality in major Indian cities

**Gayathri Megavarnan, Kavitha Venkatachalam**
Department of Data Science and Business Systems, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai, India

**Article Info**

***Article history:***

Received Jul 3, 2024
Revised Oct 21, 2024
Accepted Nov 19, 2024

***Keywords:***

Air pollution
Data analysis
Decision making
Machine learning
Pelican optimization algorithm
Prediction

## ABSTRACT

Air pollution is a critical environmental and public health concern, exacerbated by urbanization, industrial growth, and increased transportation. The air quality index (AQI) in major cities is significantly elevated due to rapid industrial expansion, fossil fuel consumption, and vehicular emissions. This study aims to predict AQIs using machine learning techniques, specifically integrating the Pelican optimization algorithm (POA) with the decision tree (DT) method to enhance accuracy. Data from prominent Indian cities—Mumbai, Delhi, Bangalore, Kolkata, and Chennai—was analyzed due to their high pollution levels. The model's performance was validated against traditional machine learning methods such as k-nearest neighbors (KNN), random forest (RF) regression, and support vector regression (SVR). Results showed the highest prediction accuracies for Kolkata at 96.68%, followed by Bangalore at 95.66%, Chennai at 93.10%, Mumbai at 92.48%, and Delhi at 86.61%. These findings demonstrate that the proposed model outperforms conventional techniques in predicting AQI, providing a foundation for effective policy-making to mitigate air pollution impacts.

***Corresponding Author:***

Gayathri Megavarnan
Department of Data Science and Business Systems, School of Computing
College of Engineering and Technology, SRM Institute of Science and Technology
Kattankulathur, Chennai-603203, Tamil Nadu, India
Email: gm4462@srmist.edu.in

## 1. INTRODUCTION

Pollution in the air is one of the most important problems that exist all over the world. There are around seven million individuals who have been impacted by a variety of ailments as a result of air pollution, according to the World Health Organization (WHO) [1]. Air pollution enhances the possibility of developing asthma, cardiovascular problems, dermatological infections, ocular ailments, throat infections, lung cancer, respiratory tract infections, and other related conditions. Long-term exposure to air pollutants could raise the probability of early mortality [2]. During the first stages of the industrial era, it was widely recognized that oil and gas, including both gasoline and coal, were the primary energy sources [3]. According to the WHO, air pollution is accountable for the deaths of 7 million individuals each year. Out of this number, 4.2 million fatalities are caused by outdoor air pollution, while 3.8 million deaths are attributed to interior air pollution resulting from the combustion of wood and charcoal [4]. This number highlights the crucial need for reliable and accurate techniques for analyzing and forecast air quality, especially in heavily populated metropolitan regions. This study is motivated by the need to address the increasing issue of air pollution in large cities, which is more serious by rapid urbanization, industrial expansion, and rising emissions from vehicles. Earlier research often used traditional machine learning methods such as k-nearest neighbors (KNN), random forest (RF) regressors, and support vector regressors [5]. In additon these traditional models lacked advanced optimization algorithms that would have enabled for the proper fine-tuning of hyperparameters, which resulted in difficult predictions of AQI. To summarize, our primary contributions may be outlined as follows:

- Novel approach is introduced that combines the Pelican optimization algorithm (POA) with decision trees (DT) to improve the prediction accuracy of the air quality index (AQI).
- The model shows notable improvements in predicting AQI for major Indian cities, outperforming traditional models like KNN, RF, and support vector regression (SVR).
- This methodology can assist governments and policymakers in taking timely actions to reduce air pollution and improve public health.

## 2. RELATED WORKS

A machine learning-based AQI prediction algorithm uses environmental monitoring and metrological measures to forecast AQI [6]. In contrast to the conventional neural network model, the Gaussian plume neural network that was presented achieved superior prediction performance while simultaneously reducing the amount of power that was used in comparison to the various monitoring methodologies that were employed [7]. A predictive model for air quality that incorporates an algorithmic approach to continuously forecast the concentration of PM 2.5 fine particulate matter, a major factor in air pollution [8]. An air pollution monitoring model is developed that utilizes internet of things (IoT) monitors and algorithms for machine learning to provide real-time data. By using multiple regression models, the prediction of air pollution may be achieved with enhanced accuracy. The experiments have shown that the regression model proposed achieved the lowest root mean square error (RMSE) and mean absolute error (MAE) [9]. The proposed methodology employs a four-layer fuzzy artificial neural network to predict the AQI using past period data. The initial fuzzy rules are derived using the past data of a particular time series to enhance the precision of forecasting [10]. A hybrid model that integrates various machine learning techniques has been developed to enhance prediction accuracy. This ensemble approach leverages support vector machines (SVM), KNN, linear regression, and logistic regression. Research indicates that this ensemble forecasting method outperforms traditional machine learning models in terms of prediction accuracy [11]. Variational mode decomposition and sample entropy are used to create a strong model for predicting the AQI. The bat method is used to improve SVM variables, which makes predictions more accurate. In terms of reliability, the optimization method is always better than hybrid models [12]. Air pollution levels in major cities were better predicted by the Gaussian Naïve Bayes approach than by other neural network methods, according to the data [13]. The suggested model is a forecasting model for the AQI that combines variational model decomposition with whale optimization techniques. The original AQI sequence can be broken down using variational mode decomposition. The VAD parameters are tuned utilizing the whale optimization technique to get enhanced performances. Utilizing a bidirectional LSTM model allows for the analysis of the dynamic characteristics of features, resulting in better prediction performance compared to traditional methods [14]. Furthermore, research has investigated the effects of various sources of air pollution, such as emissions from traffic and industrial operations, on levels of AQI [15]. A risk evaluation is performed to identify significant emission sources and analyze pollutants in the air levels utilizing multivariate approaches. This emphasizes the need of conducting complete assessments in metropolitan conditions [16].

## 3. METHOD

### 3.1. Data balancing with synthetic minority over-sampling

An imbalanced problem may be defined as a situation where the amount of instances in the majority class significantly surpasses the amount of occurrences in the minority class. This issue is frequently encountered while dealing with everyday scenarios [16]. It is essential to have data that is balanced since machine learning algorithms may have an ability to lean towards the majority class, which may result in difficulties with underfitting or overfitting. Synthetic minority over-sampling (SMOTE) uses preexisting data to generate new instances of the minority class, whereas the basic over-sampling approach duplicates the existing data, and under-sampling eliminates the majority category in the data. SMOTE is a commonly used technique for addressing imbalance problems because it has the ability to outperform basic sampling approaches by reducing issues related to overfitting and underfitting. SMOTE have been suggested to address the issue of imbalanced datasets that involve continuous features [17]. SMOTE-NC, an improved version of the SMOTE method that generates synthetic data samples with a combination of continuous and categorical attributes [18]. The SMOTE-NC procedure is executed in the following manner. Compute the KNN of the sample $z_n$ in the given class. Select $N$ samples randomly and designate each of them as $z_n$. Finally, the following is the output of the interpolation procedure for the new sample $z_{\text{new}}$ in (1):

$$z_{\text{new}} = z_n + (\hat{z} - z_n) \times \delta \quad \text{for} \quad n = 1, \ldots, N \tag{1}$$

where $\delta$ is a random integer that is distributed according to a uniform distribution within the range of 0 to 1. Figure 1 illustrates the architecture of the proposed model.

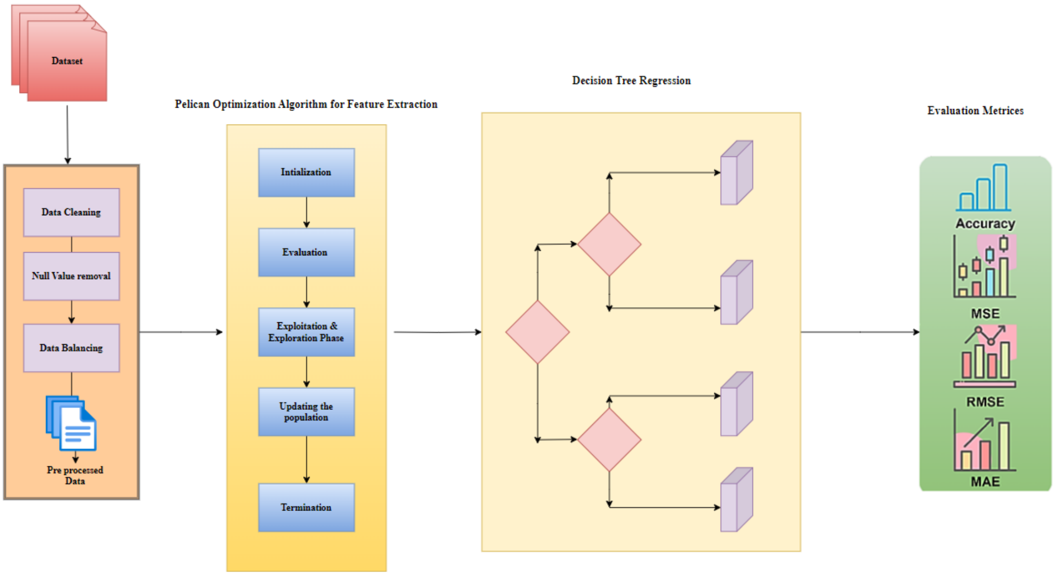


Figure 1. Proposed model POA-DT

### 3.2. Pelican optimization algorithm

The proposed technique utilizes nature-inspired Fractional Pelican optimization algorithm (FPOA) to choose the optimum features from the dataset after data balancing. The pelican is a big bird with a lengthy beak that possesses a capacious pouch in its throat, which it uses to capture and ingest prey. This species of bird exhibits a strong affinity for communal and sociable behavior, residing in flocks consisting of numerous pelicans, often numbering in the hundreds [19]. A simulation is used to update the candidate solution by mimicking the hunting technique of pelicans when they attack their food source. The simulation of this process is divided into two phases, which are as follows:

#### 3.2.1. Phase 1 (exploration phase)

On the first phase, the pelicans determine the precise position of their objective, and then they move in the direction of this well-defined region. This pelican's technique will be simulated, its examining in the search area will be analyzed, and the effectiveness of the suggested POA will be evaluated in terms of its ability to investigate various regions within the search area. An unpredictable evolution of the prey's position inside the searched region is a crucial element of the POA. Furthermore, this enhances the capacity of POA to precisely investigate the problem-solving area. In (2) provides a quantitative model for the pelican's strategy for approaching the place of its prey.

$$w_{m,n}^{\text{Pr}_1} = \begin{cases} w_{m,n} + \text{rand} \cdot (\text{pr}_n - \text{RI} \cdot w_{m,n}), & \text{if } OF_p < OF_i \\ w_{m,n} + \text{rand} \cdot (w_{m,n} - \text{pr}_n), & \text{else} \end{cases} \tag{2}$$

where:

- $w^{\mathrm{Pr}_1}_{(m,n)}$ represents the revised state of the $m$th pelican within the $n$th dimension, as determined by phase 1.
- RI denotes the random integer, which can be either 1 or 2.
- $\mathrm{Pr}_n$ specifies the spatial location of the prey in the $n$th dimension.
- $\mathrm{OF}_p$ is the objective function value.
- The parameter $I$ can be randomly assigned an integer value between 1 and 2. At the start of each loop, this value is randomly selected for all members.
- Setting this parameter to 2 results in greater displacement of a member, potentially moving them to different areas within the searching area. This parameter $I$ influences the POA's capabilty to explore the searching area thoroughly.

In the suggested POA, a pelican's revised position is accepted if it yields a better result for the objective function, indicating that the position is suitable. This particular kind of updating, which is referred to as effective updating, assists in preventing the algorithm from moving to regions that are less than optimal.

$$w_m = \begin{cases} w_m^{P_1}, & \text{if } OF_m^{P_1} < OF_m \\ w_m, & \text{else} \end{cases} \tag{3}$$

$w_m^{P_1}$ denotes the revised state from the $m$-th of the pelican. In phase 1, the value of its objective function is denoted by the symbol $OF_m^{P_1}$.

### 3.2.2. Phase 2 (exploitation phase)

Phase 2 involves the pelicans reaching the water's surface, when they will spread their wings to lift the fish into the air. They will then catch the fish in their neck pouch. Using this tactic, pelicans are able to capture a greater number of fish in the region that is being physically attacked. By modeling the behavior of pelicans, the proposed POA is able to converge to more advantageous locations within the hunting region. When this procedure is carried out, the strength of local search and the ability of POA to exploit are both increased. In order to attain a more perfect result, the approach must quantitatively assess the points that surround the pelican's location. In (4) represents the quantitative modeling of pelican hunting behaviour.

$$w_{m,n}^{P_2} = w_{m,n} + C \cdot \left(1 - \frac{it}{T}\right) \cdot (2 \cdot \text{rand} - 1) \cdot w_{m,n} \tag{4}$$

Based on the results of phase 2, the revised state of the m-th pelican in the n-th dimension is represented as $w_{m,n}^{P_2}$. The constant $C$ has the value 0.2. With $t$ representing the duration of the iteration timer and $T$ representing a maximum amount of iterations, the neighborhood radius of $w_{m,n}$ may be calculated as $C \cdot \left(1 - \frac{it}{T}\right)$. Within the neighborhood of each member of the population, $C \cdot \left(1 - \frac{it}{T}\right)$ represents the radius that a local search is performed to converge into an improved result.

This value has a substantial impact on the effectiveness of exploiting the POA in order to get the optimal global solution. During the first iterations, the coefficient has a high value, leading to a greater region being taken into account around each member. As the method replicates, the coefficient $C \cdot \left(1 - \frac{it}{T}\right)$ declines, leading to decreasing radii of neighbourhoods for each member. This enables us to thoroughly examine the vicinity of each individual in the population using smaller and more precise increments, hence allowing the POA to approach solutions which are more similar to the global optimal solution. During this step, the method of updating has been used to either accept or reject the updated location given by the pelican, as shown by (5).

$$w_m = \begin{cases} w_m^{P_2}, & \text{if } OF_1^{P_2} < OF_m \\ w_m, & \text{else} \end{cases} \tag{5}$$

The new state of the m-th pelican is shown by $w_m^{P_2}$, whereas the objective function value that was calculated during phase 2 is represented by $OF_1^{P_2}$.

## 3.3. Decision tree

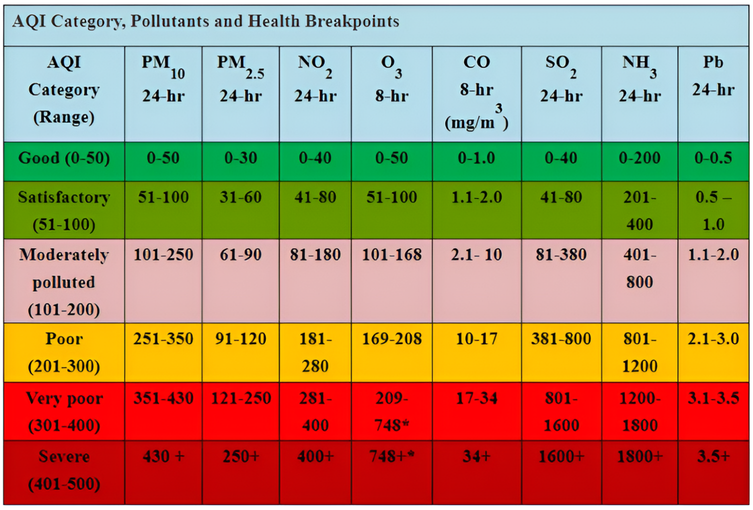
Both qualitative and quantitative variables may be predicted with the help of the DT method, which is a nonparametric approach [20]. By utilizing DT, it is possible to forecast the reactions of the data [21]. The DT has a hierarchical structure that facilitates the classification and regression of data [22]. When it comes to the tree structure, the branches that are higher up have prediction factors that are more accurate for the linked class. While the regression produces numerical responses, the categorization in the DT delivers replies that are nominal in nature [23]. In order to make a forecast, data must be collected from the root node all the way to the leaf node. It is generally the case that the replies are contained within the leaf node. Utilizing a regressor $R_l$, the model of regression is used to map the connection that exists among the feature vectors and the displacement vectors. The regressor is then utilized to make predictions regarding the displacements that occur inside the feature vectors. The final step is to determine whether or not the numbers that were predicted are the best possible answer for the problem that was presented. In order to achieve more accurate results when assessing the region of interest, the suggested model makes use of regression rather than categorization.

### 3.4. The final model for prediction

In this prediction model, which utilizes POA and DT regression. Data cleansing and removal of null values are carried out at the initial preprocessing stage. The POA technique is employed to identify the best features from the pre-processed data. A DT regression method is utilized to evaluate the selected attributes for predicting the accuracy of the AQI. The suggested model undergoes initial training using the dataset and subsequent testing utilizing samples test data. In this procedure, the complete dataset is partitioned into an 80:20 ratio for the purposes of training and testing. Performance indicators including accuracy metrics, RMSE, MAE, and MSE are used to evaluate the final testing performances [24]. The DT design has a hierarchical structure like that of a tree, with a root, branches, and leaves. The pseudocode (Algorithm 1, in Appendix) illustrates the utilization of the POA to enhance the hyperparameters of a DT model, hence improving the accuracy of predicting AQI.

## 4. RESULT AND DISCUSSION

The initial level of Air Quality Data by India (2015-2020) that was received from the Kaggle database repository [25] is used in the implementation of the approach that has been provided for the simulation evaluations. Six large cities—Mumbai, Delhi, Bangalore, Kolkata, and Chennai—were analysed for the experiment. The cities are selected according to the AQI classifications given in the dataset, which cover a categories from good to severe its shows in the Figure 2 [26].



AQI Category, Pollutants and Health Breakpoints

| AQI Category (Range) | $PM_{10}$ 24-hr | $PM_{2.5}$ 24-hr | $NO_2$ 24-hr | $O_3$ 8-hr | CO 8-hr ($mg/m^3$) | $SO_2$ 24-hr | $NH_3$ 24-hr | Pb 24-hr |
|---|---|---|---|---|---|---|---|---|
| Good (0-50) | 0-50 | 0-30 | 0-40 | 0-50 | 0-1.0 | 0-40 | 0-200 | 0-0.5 |
| Satisfactory (51-100) | 51-100 | 31-60 | 41-80 | 51-100 | 1.1-2.0 | 41-80 | 201-400 | 0.5 – 1.0 |
| Moderately polluted (101-200) | 101-250 | 61-90 | 81-180 | 101-168 | 2.1- 10 | 81-380 | 401-800 | 1.1-2.0 |
| Poor (201-300) | 251-350 | 91-120 | 181-280 | 169-208 | 10-17 | 381-800 | 801-1200 | 2.1-3.0 |
| Very poor (301-400) | 351-430 | 121-250 | 281-400 | 209-748* | 17-34 | 801-1600 | 1200-1800 | 3.1-3.5 |
| Severe (401-500) | 430 + | 250+ | 400+ | 748+* | 34+ | 1600+ | 1800+ | 3.5+ |

Figure 2. AQI categorization for multiple pollutants

The pollution levels of these large cities are determined using data that has been converted into a comma-separated values (CSV) file. First, the optimization model is used to get the desired characteristics. Then, a DT classifier is employed to classify these features. The tests are carried out with the Python tool, employing the requisite library functions to optimize both the model as well as classifier. The hyperparameters used for experiments are listed in Table 1.

Table 1. Hyperparameters for the POA-DT algorithm

| Algorithm | Parameter | Value |
|---|---|---|
| POA | Population size | 30 |
| | Number of iterations | 100 |
| | Exploration rate | 0.3 |
| | Exploitation rate | 0.7 |
| DT | Maximum depth | 10 |
| | Minimum leaf size | 5 |
| | Number of interval bins | 10 |

### 4.1. Model configuration and performance metrics analysis

In Delhi, the POA-DT model resulted in significant improvements in several metrics. MSE decreased from 0.0732 (imbalanced) to 0.0702 (balanced), MAE decreased from 0.1724 to 0.1302, RMSE decreased from 0.2084 to 0.1863, and the accuracy increased from 82.56% to 86.10%. In Chennai, MSE reduced from 0.1465 to 0.1104, MAE declined from 0.1751 to 0.0638, RMSE decreased from 0.1248 to 0.1160, and the accuracy climbed from 81.52% to 93.10%. In Mumbai, MSE had a modest rise from 0.0162 to 0.0164. MAE improved from 0.1104 to 0.1017. RMSE decreased from 0.0635 to 0.0452. Additionally, the accuracy climbed from 90.65% to 92.48%. In Bangalore, MSE rose from 0.1148 to 0.1212, but the MAE greatly improved from 0.0632 to 0.0254. RMSE reduced from 0.4247 to 0.4126, and the accuracy improved from 91.68% to 95.66%. In Kolkata, MSE reduced slightly from 0.1786 to 0.1721. MAE stayed almost same. RMSE improved from 0.4148 to 0.4050. Additionally, the accuracy climbed from 93.84% to 96.60%. Tables 2 to 6 shows the specific performance measures of the model POA-DT for the cities of Mumbai, Delhi, Bangalore, Kolkata, and Chennai, respectively. Figure 3 illustrated the results clearly demonstrate that the balanced dataset achieved higher accuracy compared to the imbalanced dataset.

Table 2. Evaluation metrics for the proposed technique in Delhi City

| Performance metrics | Imbalanced data | Balanced data |
|---|---|---|
| MSE | 0.0732 | 0.0702 |
| MAE | 0.1724 | 0.1302 |
| RMSE | 0.2084 | 0.1863 |
| Accuracy(%) | 82.558 | 86.104 |

Table 3. Evaluation metrics for the proposed technique in Chennai City

| Performance metrics | Imbalanced data | Balanced data |
|---|---|---|
| MSE | 0.1465 | 0.1104 |
| MAE | 0.1751 | 0.0638 |
| RMSE | 0.1248 | 0.1160 |
| Accuracy (%) | 81.523 | 93.101 |

Table 4. Evaluation metrics for the proposed technique in Mumbai City

| Performance metrics | Imbalanced data | Balanced data |
|---|---|---|
| MSE | 0.0162 | 0.0164 |
| MAE | 0.1104 | 0.1017 |
| RMSE | 0.0635 | 0.0452 |
| Accuracy (%) | 90.65 | 92.48 |

Table 5. Evaluation metrics for the proposed technique in Bangalore City

| Performance metrics | Imbalanced data | Balanced data |
|---|---|---|
| MSE | 0.1148 | 0.1212 |
| MAE | 0.0632 | 0.0254 |
| RMSE | 0.4247 | 0.4126 |
| Accuracy (%) | 91.684 | 95.662 |

Table 6. Evaluation metrics for the proposed technique in Kolkata City

| Performance metrics | Imbalanced data | Balanced data |
|---|---|---|
| MSE | 0.1786 | 0.1721 |
| MAE | 0.0616 | 0.0632 |
| RMSE | 0.4148 | 0.4058 |
| Accuracy (%) | 93.84 | 96.68 |

POA was used to improve the hyperparameters of the DT in order to attain a high level of accuracy in predicting AQI. A depth of 10 has been selected as the optimal value to achieve a compromise between the model's capacity for understanding complex data patterns and the need to avoid overfitting, therefore improve accuracy. By choosing a minimum leaf size of 5, provided the each leaf node included an adequate number of samples, which improved the reliability and generality of predictions. In addition, the use of 10 interval bins resulted in an appropriate amount of discretization for continuous features. This allowed for the capture of important fluctuations in the data without introducing unnecessary complexity. The combined selection of these parameters significantly contributed to the model's outstanding results, as seen by the better accuracy obtained in forecasting AQI across the analyzed cities.

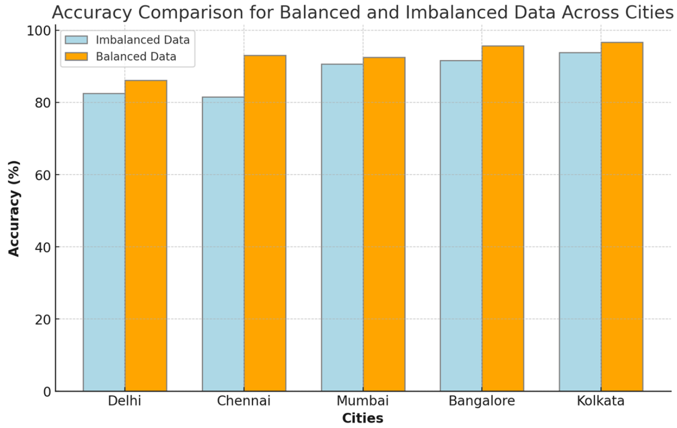


Figure 3. Accuracy comparison

### 4.2. Comparative analysis

The comparative analysis demonstrates that the combination of the POA and the DT model surpasses traditional machine learning techniques like K-NN, RF, and SVR in accurately predicting the AQI for prominent Indian cities. At Delhi, the POA-DT model demonstrated an accuracy of 86.10%, which outperformed the KNN (83.68%), RF (84.73%), and SVR (84.83%) models, while also surpassing the accuracy of the regular DT model (85.74%). Similarly, in Chennai, the accuracy of POA-DT was 93.10%, which was higher than the accuracy of K-NN (89.43%), RF (90.45%), SVR (92.48%), and ordinary DT (92.79%). In Mumbai, the accuracy of POA-DT is 92.48%, which surpasses the accuracy of K-NN (90.03%), RF (90.93%), SVR (91.03%), and DT (91.76%). Bangalore and Kolkata shown substantial improvements with POA-DT, attaining the greatest levels of accuracy at 95.66% and 96.68% respectively. The improved performance may be defined to the effective fine-tuning of the DT's hyperparameters by POA, which mitigates problems such as overfitting and enhances the model's generalization. The proposed model shows the higher accuracy when compared to the other models it shows in the Table 7.

Table 7. Comparison of AQI prediction accuracy across models and cities

| Cities | K-NN | RF | SVR | DT | POA-DT |
|---|---|---|---|---|---|
| Delhi | 83.68 | 84.73 | 84.83 | 85.74 | 86.61 |
| Chennai | 89.43 | 90.45 | 92.48 | 92.79 | 93.10 |
| Mumbai | 90.03 | 90.93 | 91.03 | 91.76 | 92.48 |
| Bangalore | 87.18 | 89.47 | 90.31 | 92.99 | 95.66 |
| Kolkata | 92.24 | 92.11 | 93.56 | 95.12 | 96.68 |

## 5. CONCLUSION

Integration of the POA with DT approaches has shown substantial improvements in the precise forecasting of the AQI in key urban areas of India. The proposed hybrid methodology has attained prediction accuracies that exceed those of conventional machine learning techniques. The findings highlight the capacity of hybrid models to improve the precision of environmental predictions, which is essential for rapidly and efficiently implementing strategies to decrease air pollution in urban regions. The use of these models has the potential to establish new benchmarks in the analysis of environmental data. This would allow the scientific community to attain more accuracy in predicting outcomes, thereby enhancing public health and providing valuable insights for the development of more efficient environmental policies for the wider population. Further studies will prioritize improving the quality of the dataset by include other cities and a wider range of climatic circumstances. Additionally, it will investigate the incorporation of complementary optimization methods and machine learning approaches to improve the prediction capabilities. The research is constrained by its dependence on historical data and the possibility of local optima in the optimization process; further studies should prioritize the incorporation of real-time data and the application of the model to more metropolitan regions to enhance forecast precision.

## APPENDIX

**Algorithm 1 .** Pseudocode for POA-DT

```
1. Data Collection and Preprocessing
    • Clean $D_{\text{cities}}$ by handling missing values, removing outliers, and normalizing features and balancing the data.
2. Initialize Pelican Population
    • Initialize a population $P = \{p_1, p_2, \ldots, p_n\}$ of $n$ pelicans.
    • For each pelican $p_i$, randomly assign a candidate solution vector $v_i = [d_i, l_i, b_i]$, where:
        – $d_i \in d_{\text{range}}$: Maximum depth of the Decision Tree.
        – $l_i \in l_{\text{range}}$: Minimum leaf size of the Decision Tree.
        – $b_i \in b_{\text{range}}$: Number of interval bins for the Decision Tree.
3. Evaluate Fitness of Initial Population
    • For each pelican $p_i$:
        – Configure a Decision Tree $DT_i$ with hyperparameters $v_i = [d_i, l_i, b_i]$.
        – Train $DT_i$ on $D_{\text{cities}}$.
        – Predict using $DT_i$ on the validation set.
        – Compute performance metrics:
          $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2$, $\text{MAE} = \frac{1}{m}\sum_{i=1}^{n}|z_i - \hat{z}_i|$, $\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(z_i - \hat{z}_i)^2}{m}}$
          Accuracy: $\text{Accuracy}_i = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$
        – Evaluate fitness $F_i$ using a combination of performance metrics (e.g., $F_i = \text{MSE}_i$ or weighted sum of MSE, MAE, RMSE, Accuracy).
4. Update Best Solution
    • Identify the pelican $p_{\text{best}}$ with best fitness $F_{\text{best}}$.
    • Store $v_{\text{best}} = v_{p_{\text{best}}}$ as the best solution.
5. Optimization Loop
    • Repeat until convergence or max iterations $T$ is reached:
    (a) Exploration Phase
            – For each pelican $p_i$:
            – Update position $v_i$ using the exploration formula.
            – Ensure updated values $v_i$ within valid hyperparameter ranges.
            – Evaluate new fitness $F_i$ for updated $v_i$.
            – If new fitness is better, accept new position.
    (b) Exploitation Phase
            – For each pelican $p_i$:
            – Update position $v_i$ using the exploitation formula.
            – Ensure updated values $v_i$ within valid hyperparameter ranges.
            – Evaluate new fitness $F_i$ for updated $v_i$.
            – If new fitness is better, accept new position.
    (c) Update Best Solution
            – Update $p_{\text{best}}$ and $v_{\text{best}}$ if better solution is found.
    (d) Check Convergence
            – If $|F_{\text{best new}} - F_{\text{best old}}| < \epsilon$ or max iterations $T$ is reached, stop.
6. Train Final Decision Tree Model
    • Set the hyperparameters of final Decision Tree $DT_{\text{opt}}$ to $v_{\text{best}}$.
    • Train $DT_{\text{opt}}$ on $D_{\text{cities}}$.
```

## REFERENCES

[1] WHO, *Air Quality Guidelines for Europe*, Regional Office for Europe, no. 91, 2020.

[2] B. Brunekreef and S. T. Holgate, ”Air pollution and health,” *The Lancet*, vol. 360, no. 9341, pp. 1233-1242, 2002.

[3] H. Tian, Y. Zhao, M. Luo, Q. He, Y. Han, and Z. Zeng, ”Estimating PM2.5 from multisource data: A comparison of different machine learning models in the Pearl River Delta of China,” *Urban Climate*, vol. 35, p. 100740, 2021, doi: 10.1016/j.uclim.2020.100740.

[4] Clean Air Fund. ”Clean Air Fund.” [Online]. Available: https://www.cleanairfund.org/. (Accessed: Oct. 10, 2023).

[5] I. H. Sarker, ”Machine Learning: Algorithms, Real-World Applications and Research Directions,” *SN Computer Science*, vol. 2, no. 3, p. 160, 2021, doi: 10.1007/s42979-021-00592-x.

[6] E. Gladkova and L. Saychenko, ”Applying machine learning techniques in air quality prediction,” *Transportation Research Procedia*, vol. 63, pp. 1999-2006, 2022, doi: 10.1016/j.trpro.2022.06.222.

[7] Y. Yang, Z. Zheng, K. Bian, L. Song, and Z. Han, ”Real-Time Profiling of Fine-Grained Air Quality Index Distribution Using UAV Sensing,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 186-198, Feb. 2018, doi: 10.1109/JIOT.2017.2777820.

[8] K. Gu, J. Qiao, and W. Lin, ”Recurrent air quality predictor based on meteorology-and pollution-related factors,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3946-3955, 2018, doi: 10.1109/TII.2018.2793950.

[9] S. Ameer *et al.*, ”Comparative analysis of machine learning techniques for predicting air quality in smart cities,” *IEEE Access*, vol. 7, pp. 128325-128338, 2019, doi: 10.1109/ACCESS.2019.2925082.

[10] Y.-C. Lin, S.-J. Lee, C.-S. Ouyang, and C.-H. Wu, ”Air quality prediction by neuro-fuzzy modeling approach,” *Applied Soft Computing*, vol. 86, p. 105898, 2020, doi: 10.1016/j.asoc.2019.105898.

[11] N. Phruksahiran, ”Improvement of air quality index prediction using geographically weighted predictor methodology,” *Urban Climate*, vol. 38, p. 100890, 2021, doi: 10.1016/j.uclim.2021.100890.

[12] C. C. Liu, T. C. Lin, K. Y. Yuan, and P. T. Chiueh, ”Spatio-temporal prediction and factor identification of urban air quality using support vector machine,” *Urban Climate*, vol. 41, p. 101055, 2022.

[13] T. Kumar and A. Doss, ”AIRO: Development of an intelligent IoT-based air quality monitoring solution for urban areas,” *Procedia Computer Science*, vol. 218, pp. 262-273, 2023, doi: 10.1016/j.procs.2023.01.008.

[14] Y. Li and R. Li, ”A hybrid model for daily air quality index prediction and its performance in the face of impact effect of COVID-19 lockdown,” *Process Safety and Environmental Protection*, vol. 176, pp. 673-684, 2023, doi: 10.1016/j.psep.2023.06.021.

[15] L. Fu, J. Li, and Y. Chen, ”An innovative decision making method for air quality monitoring based on big data-assisted artificial intelligence technique,” *Journal of Innovation & Knowledge*, vol. 8, no. 2, p. 100294, 2023, doi: 10.1016/j.jik.2022.100294.

[16] M.-J. Chen, Y. L. Guo, P. Lin, H.-C. Chiang, P.-C. Chen, and Y.-C. Chen, ”Air quality health index (AQHI) based on multiple air pollutants and mortality risks in Taiwan: Construction and validation,” *Environmental Research*, vol. 231, part 2, p. 116214, 2023, doi: 10.1016/j.envres.2023.116214.

[17] A. Koivu, M. Sairanen, A. Airola, and T. Pahikkala, ”Synthetic minority oversampling of vital statistics data with generative adversarial networks,” *Journal of the American Medical Informatics Association*, vol. 27, no. 11, pp. 1667-1674, 2020, doi: 10.1093/jamia/ocaa127.

[18] F. Shakeel, A. Sabhitha, and S. Sharma, ”Exploratory review on class imbalance problem: An overview,” in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Delhi, India, 2017, pp. 1-8, doi: 10.1109/ICCCNT.2017.8204150.

[19] A. Louchart, N. Tourment, and J. Carrier, ”The earliest known pelican reveals 30 million years of evolutionary stasis in beak morphology,” *Journal of Ornithology*, vol. 152, no. 1, pp. 15-20, 2011, doi: 10.1007/s10336-010-0537-5.

[20] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, ”Decision trees: From efficient prediction to responsible AI,” *Frontiers in Artificial Intelligence*, vol. 6, pp. 01-17, Jul. 2023, doi: 10.3389/frai.2023.1124553.

[21] V. G. Costa and C. E. Pedreira, ”Recent advances in decision trees: An updated survey,” *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4765-4800, 2023, doi: 10.1007/s10462-022-10275-5.

[22] H. Tanveer, M. A. Adam, M. A. Khan, M. A. Ali, and A. Shakoor, ”Analyzing the Performance and Efficiency of Machine Learning Algorithms, such as Deep Learning, Decision Trees, or Support Vector Machines, on Various Datasets and Applications,” *The Asian Bulletin of Big Data Management*, vol. 3, no. 2, pp. 126-136, 2023, doi: 10.62019/abbdm.v3i2.83.

[23] J. M. Klusowski and P. M. Tian, ”Large scale prediction with decision trees,” *Journal of the American Statistical Association*, vol. 119, no. 545, pp. 525-537, 2024, doi: 10.1080/01621459.2022.2126782.

[24] S. Ameer *et al.*, ”Comparative Analysis of Machine Learning Techniques for Predicting Air Quality in Smart Cities,” *IEEE Access*, vol. 7, pp. 128325-128338, 2019, doi: 10.1109/ACCESS.2019.2925082.

[25] R. Rao, ”Calculating AQI (Air Quality Index) Tutorial,” Kaggle, [Online]. Available: https://www.kaggle.com/code/rohanrao/calculating-aqi-air-quality-index-tutorial. (Accessed: Aug. 28, 2024).

[26] ”Air Quality Index (AQI) Basics,” AirNow, [Online]. Available: https://www.airnow.gov/aqi/aqi-basics. [Accessed: Aug. 12, 2024].

## BIOGRAPHIES OF AUTHORS

**Gayathri Megavarnan** received the engineer degree in computer science from Anna University Chennai in 2010. She received the master degree in computer science and engineering from Anna University Chennai in 2012. Her research interests include data analysis, machine learning, and deep learning. She can be contacted at email: gm4462@srmist.edu.in.

**Dr. Kavitha Venkatachalam** has held various academic positions over the years, contributing significantly to the field of computer science and engineering. Currently, since June 2022, she serves as a Professor in the Department of Data Science and Business Systems at SRM Institute of Science and Technology. Prior to this role, from June 2018 to May 2022, she was an Associate Professor in the same department at SRM Institute of Science and Technology. From June 2009 to June 2018, she held the position of Assistant Professor in the Department of Information Technology at Vellalar College of Engineering and Technology. Earlier in their career, from June 2005 to June 2007, she also served as an Assistant Professor in the Department of Information Technology at the same institution. She earned a bachelor of engineering (B.E.) in computer science and engineering from Madras University, India, in April 2001, followed by a master of engineering (M.E.) in computer science and engineering from Anna University, India, in April 2009. she completed their Ph.D. in information and communication engineering at Anna University, India, in July 2018. She can be contacted at email: kavithav2@srmist.edu.in.