Environmental Science

https://www.aimspress.com/journal/environmental

AIMS Environmental Science, 12(4): 576–593.
DOI: 10.3934/environsci.2025026
Received: 21 February 2025
Revised: 15 May 2025
Accepted: 04 June 2025
Published: 13 June 2025

Research article

An embedded machine learning system method for air pollution monitoring and control

Xiaobo Li[1], Xinggang Ye[2,*] and Meirong Qian[3]

[1] Department of Science and Technology, Hubei Industrial Polytechnic, ShiYan, HuBei, 442000, China. boxone451@163.com
[2] Asset Management Division, Hubei Industrial Polytechnic, ShiYan, HuBei, 442000, China. yexingang@126.com
[3] School of Environmental Engineering, Hubei Industrial Polytechnic, ShiYan, HuBei, 442000, China. qmr791192@sina.com

* **Correspondence:** Email: yexingang@126.com.

Abstract: Air pollution has very serious health impacts as well as detrimental effects on ecosystems. Addressing this issue will require operational real-time monitoring and management, specifically in urban environments. Potential sensors that could provide the necessary high accuracy are too costly for mass deployment in the Internet of Things (IoT) environment. On the other hand, low-cost sensors usually have noisy, unreliable, and less accurate outputs, preventing them from being effective tools in real-time applications. Therefore, there is a lack of cost-effective, accurate, and scalable solutions to enhance the reliability of low-cost sensors such that they can improve air quality monitoring and control in real-time. Consequently, this research proposes a novel approach that advances the use of embedded machine learning with edge computing to enhance the accuracy and real-time actions at the local level. In the system, the Raspberry Pi was utilized for processing sensor information as the embedded edge device to monitor multiple real-time air pollutants, including SO_2, NO_2, and $PM_{2.5}$. In this research, the one-rank cuckoo search-driven adaptive support vector machine (ORCS-ASVM) was proposed to improve the prediction accuracy of low-cost sensors with noise. The ORCS-ASVM machine learning model predicts air quality and reduces the amount of computing power required while providing immediate knowledge about the amount of air pollution present. This model strives to enhance the accuracy of low-cost sensor readings, thus accommodating erroneous sensor readings in pollution prediction. Data on air pollution that is continuously collected from the sensors will be monitored at regular intervals to provide real-time monitoring. Sensor data often contains noise, requiring filtering and preprocessing techniques to correct sensor errors. Smoothing and normalization are the most common preprocessing methods used to correct sensor errors. Compared with traditional prediction algorithms, the proposed model outperformed traditional methods in terms of error reduction. Air pollutants such as $PM_{2.5}$, SO_2, and NO_2 achieved detection accuracies of 94.2%, 95.2%, and 94.8%, respectively. An embedded machine learning system approach for air quality monitoring and in situ pollution mitigation is a low-cost method that can be extended to other industries, making it suitable for smart city infrastructure, smart agriculture, and other important areas. An embedded machine learning approach is applicable to many areas as it has great scalability and flexibility and is therefore very suitable for widespread adoption by IoT-based pollution monitoring networks.

Keywords: air pollution; machine learning for atmospheric forecasting; monitoring and control; pollutants; air quality; one rank cuckoo search-driven adaptive support vector machine (ORCS-ASVM)

1. Introduction

Air pollution is one of the biggest threats to the environment and public health in the modern world. In addition to the degradation of air quality, rapid urbanization, industrial growth, and vehicle emissions all negatively affect the environment and human health [1]. The major reasons for illness and death caused by air pollution have been globally identified, including respiratory and cardiovascular disorders associated with exposure to particulate matter ($PM_{2.5}$ and PM_{10}), NO_x, CO, and SO_2 [2]. Further, environmental problems like acid rain, climate change, and lower agricultural production are also consequences of air pollution [3]. This problem requires an efficient control and monitoring system. Most traditional air pollutant monitoring techniques involve manual sampling and laborious analyses in a laboratory [4]. Although these methods are straightforward, highly accurate, and reliable, they incur costly time requirements, making them unsuitable for real-time or large-scale applications [5]. Air quality control could be addressed optimistically using compact, low-power-consuming embedded systems that integrate various sensors and communication technologies [6]. An embedded system is a microcontroller-based system that performs certain actions. Embedded systems offer multiple advantages such as real-time sensing, scalability, and automation options for control modes [7]. For instance, an embedded system can raise an alarm and switch on air purifiers whenever the pollution level exceeds the tolerance level [8]. The device can also facilitate the application in central dispensation of information and remote tracking by IoT connectivity [9]. Furthermore, embedded systems are energy efficient, enabling them to run continuously for monitoring in cities where energy saving is important [10]. Embedded systems may also assist communities in learning about the means to minimize pollutants and engage neighbors regarding air quality [11]. However, some challenges remain regarding the utility of embedded systems for air pollution monitoring and management [12]. Sensor accuracy needs to be ensured, costs need to be effective, and some of the issues relating to power efficiency and environmental robustness need to be resolved prior to having systems like this in place [13]. Advanced technologies, such as AI and ML, may prove particularly valuable to prevent and predict some of the sources of deteriorated air quality, assisting systems like these [14]. Some of these problems include sensor calibration problems, costly deployments, scalability limitations to large areas, environmental factors, and power efficiency issues for real-time tracking of air pollution.

Aim: To create a cost-effective system for an embedded system utilizing cheap sensors with the ORCS-ASVM model, such that air quality predictions are made with higher accuracy to track air pollution in real-time.

Key contributions

Creating an inexpensive embedded system for monitoring air quality: The system consisted of a low-cost, real-time air pollution monitoring system by combining inexpensive electrochemical sensors with Internet of Things technologies and a Raspberry Pi board. The setup provided a scalable system that could be used in densely populated metropolitan areas and allowed for real-time monitoring of the main pollutants (SO_2, NO_2, and PM2.5).

ORCS-ASVM forecasting model proposal: A new hybrid machine learning approach was proposed, which integrates ASVM with one rank cuckoo search (ORCS) optimization. The proposed model performed better than other algorithms in predicting air quality, successfully enhancing noisy, low-cost sensor measurement prediction accuracy.

Enhanced real-time decision making with edge computing: With the ability to process data locally on the embedded device using edge computing, the system minimized reliance on central servers and realized response time decreases that allowed immediate pollution control actions that could now apply to time-sensitive real-world applications.

2. Literature review

Recent research applied IoT-based environmental toxicology–based air pollution monitoring system (ETAPM-AIT) with artificial intelligence (AI). They used an artificial algae algorithm-tuned Elman neural network (AAA-ENN) for contaminant classification and air quality forecasting. According to their research, performance was optimized through a series of intervals. The work addresses some of the biggest concerns in the field: sensors are inaccurate, life is limited, and the environment interferes significantly and cannot be easily scaled.

Rahardja et al. [15] used PM2.5 and PM_{10} sensors to measure nitrogen dioxide, sulfur dioxide, and carbon dioxide, utilizing methods such as support vector regression (SVR), linear regression (LR), and ensemble gradient boosted decision tree (EGBDT). The algorithm was assessed based on root mean squared error, mean squared error, and mean absolute error. The weakness of the algorithm lay in its low scalability and inaccurate sensors. Such problems highlight crucial issues in measuring and predicting air quality.

Taheri et al. [16] proposed a dynamic carbon dioxide (CO_2) model using ML, in which a multilayer perceptron was validated using data collected from classrooms. The model demonstrated the potential for 51.4% energy savings without compromising the requirements of heating, ventilation, and air-conditioning systems. Due to generalizability problems and its reliance on occupancy fluctuation, it had several disadvantages. Despite these drawbacks, it helped address gaps in adaptive Indoor Air Quality (IAQ) regulation and showed significant promise in energy conservation.

Kaginalkar et al. [17] targeted the Urban Air Quality Monitoring project based on big data, IoT, and AI. The system integrated a model combining sensors and satellites, employing statistical analysis for counteractions in real time. For successfully implementing the concept of UAQM in smart cities, a multiple-technology framework was used, which addressed governance issues along with environmental ones.

Kalajdjieski et al. [18] presented adversarial networks for data augmentation and encoder-decoder architectures, with the main focus being PM-level forecasting. The model increased predictiveness but also significantly increased computational costs and sensor data reliance. In real-world applications, forecasting models have to be robust, and dealing with missing data proved to be challenging.

Ramadan et al. [19] developed a system for the chrome plating industry to track and predict air pollution in real time. The system monitors pollutants, namely ammonia (NH_3), carbon monoxide (CO), nitrogen dioxide (NO_2), and $PM_{2.5}$, through IoT sensors and AI models. The results were impressive in terms of predictive precision: an R-squared (R^2) of 84% for $PM_{2.5}$ based on RF and an R^2 of 99%

for LSTM temperature prediction. Disadvantages included costly deployment and sensor calibration, even though it had the potential to enhance air quality control and reduce health risks in industry.

Babak et al. [20] presented a mathematical model with measurement support and an unmanned aerial system (UAS)-based air pollution monitoring system. Wiener's field model takes into account pollution movement, while a vector random field model applies correlation theory to calculate local pollution characteristics. Even though certain experimental studies on radioactive monitoring validated the applicability of UASs, some issues remain concerning sensor accuracy, climatic variations, and system size for larger areas. Optimizing remote pollution monitoring techniques for improved environmental management was the main goal.

Han et al. [21] assessed the efficacy of Beijing's air pollution control laws using a Bayesian deep learning model based on proxy data of aerosol optical depth (AOD) and meteorological variables. It highlighted notable progress after 2013, particularly following the action plan for clean air, and revealed that $PM_{2.5}$ had dropped by 11%. However, levels continue to be higher than WHO standards. To guarantee long-term gains, it suggested more alignment and noted a two-year lag in policy impact.

Kalaivani et al. [22] examined several machine learning (ML) techniques for using IoT sensor data to forecast and track air pollution. Various analyses concentrated on the methodology, outcomes, and shortcomings of earlier models, such as their high cost and poor accuracy, as well as any published numerical findings. Accordingly, an effective, real-time system was required for accurate air quality prediction to close the gap between current approaches and improved urban air quality management.

One of the main causes of early mortality and financial losses in emerging nations is air pollution, which is mostly caused by urbanization and vehicle emissions. Several approaches to solve this problem have been proposed, such as IoT-based air pollution monitoring [23]. Predicting air pollution levels is essential for reducing their effects. However, scalability and sensor accuracy are limited. Efficient prediction methods need to be developed for resource sustainability and environmental management.

3. Problem statement

Air pollution is one of the major issues facing the world today, endangering human health and damaging ecology beyond repair. Although regulating pollution requires fine-grained real-time monitoring, especially in densely populated cities, most existing solutions rely on expensive, long-range sensors and cloud infrastructure that cannot be deployed in large quantities in Internet of Things networks. Although inexpensive sensors offer a good substitute, results are typically noisy and inaccurate, making them unsuitable for situations where monitoring is essential. Furthermore, response and control latency results from conventional systems' inability to interpret input data in real-time. A sophisticated, scalable, and affordable system that ensures precise air quality forecasting and timely management measures is desperately needed. To overcome these limitations, this research outlines an integrated system of machine learning that utilizes the ORCS-ASVM with edge computing and the Internet of Things in efforts to advance forecast accuracy and real-time air pollution reduction. The proposed model is more effective than other methods, such as random forest [24] and TSTM [25], by eliminating existing systems' inaccuracies and noise through low-cost sensors, edge computing, and machine learning, providing a low-cost monitoring system with a high degree of scalability that allows real-time air pollution control.

4. Methodology

An embedded machine learning platform for real-time, dynamic monitoring and controlling of air pollution was developed. It combines low-cost sensors with IoT connectivity and edge computing with a novel ORCS-ASVM model for accurate and efficient real-time pollution detection. The two primary components are min-max normalization for data preprocessing and ORCS-optimized SVM. The main advantage is that the device (e.g., Raspberry Pi) processes its own sensor data. This should minimize latency, the distance travelled of computed data, and the computational load on the service. This architecture enables intelligent, scalable, and cheap air quality management in smart city spaces by being adaptive. A general methodology for controlling and monitoring air quality in real-time in IoT applications is shown in Figure 1.

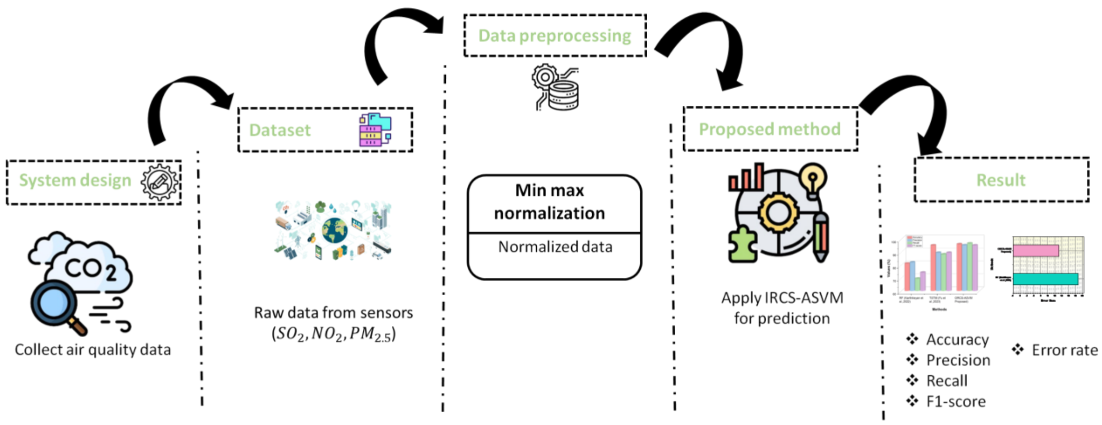

Figure 1. Overall flow for air pollution monitoring and control.

4.1. System design

An embedded system architecture that provides real-time, scalable, and cost-effective pollution management by seamlessly combining low-cost sensors, edge computing, and IoT technologies was developed. The originality of this research stems from the astute connection to the one-rank cuckoo search-based adaptive support vector machine (ORCS-ASVM) model (although it is true that many other researchers are using devices such as the Raspberry Pi and electrochemical sensors). This approach compensates for sensor noise, which is frequently present in inexpensive IoT devices, and significantly increases prediction accuracy. The technology allows for real-time data processing and decision-making on the edge device, reducing the need for central servers and facilitating timely and accurate pollution control actions. IoT connection also enables quick data transfer and continuous monitoring. This design transforms generic hardware into a robust, flexible, and scalable urban air quality monitoring and control system that could be widely used in environmental networks and smart cities.

4.2. Data set

Data on air quality gathered from a number of urban monitoring stations between 2007 and 2023 were included in this dataset. Nitrogen dioxide, sulfur dioxide, ozone, particulate matter, ammonia, and carbon dioxide are among the many pollutants that it measures. This offers important new perspectives on urban sustainability and environmental trends, making it possible to examine how

variations in air pollution levels occur in various cities and over time. This dataset provides an extensive resource for comprehending air quality and how smart cities are affected by it.
(Source: https://www.kaggle.com/datasets/zoya77/urban-air-pollution-trends-and-environmental-data)

4.3. Using min-max normalization for preprocessing

Min-max normalization is the technique used at the edge for monitoring and managing air pollution. The pollutant data is balanced between 0 and 1 using this approach. In addition to improving air quality forecast accuracy for pollutants including SO_2, NO_2, and $PM_{2.5}$, it enables fair data dissemination. In min-max normalization, the normalization values are linearly changed from the original data. The values are normalized within the given range. Eq 1 is used to change the value of property B's value from $[min_B, max_B]$ to $[new_{min_B}, new_{max_B}]$

$$\frac{u - min_B}{max_B - min_B}\left(new_{min_B}, new_{max_B}\right) + new_{min_B} \tag{1}$$

Where the new value in the specified range is denoted by u. By boosting the precision and effectiveness of algorithms for real-time air pollution monitoring and control, as well as economical pollution prediction and mitigation, the min-max normalization approach ensures that all data fall within a particular range. The suggested embedded system technique for monitoring pollutants, including SO_2, NO_2, and $PM_{2.5}$, facilitates faster and more precise decision-making in air quality management systems.

4.4. One-rank cuckoo search-driven adaptive support vector machine (ORCS-ASVM)

To increase monitoring precision and control strategies, ORCS-ASVM was included in a machine learning-based prediction of an embedded system approach for managing and monitoring air pollution. ORCSA optimizes sensor position and calibration using Lévy flights for exploration and a one-rank parameter for exploitation to improve convergence rates and solution quality. Each cuckoo nest corresponds to sensor settings and pollutant calibration to guarantee efficient data collection. Variations in pollutant concentrations, sensor calibration, and real-time data accuracy set off optimization to provide dependable monitoring. When the different levels of air quality are classified, ASVMs are employed. The soft margin is employed for balancing the correctly classified and misclassified samples, and overfitting happens in that case. In the Lagrange dual optimization method, the best classification of the pollution levels can be obtained through the use of the hyperplanes. The hybrid method integrates the predictive capability of ASVM with the optimization feature of ORCSA and is thus applicable for adaptive control and real-time monitoring of air quality. The embedded system utilizes intelligent algorithms for problem-solving regarding sensor calibration issues, data variability, and predictive capability under changed environmental conditions, making it scalable and effective for pollution control. The methodology promotes improved ecosystems and city environments since it ensures a prompt reaction to the trends of pollution. Algorithm 1 illustrates ORCS-ASVM, which removes sensor errors and improves predictive accuracy.

Algorithm 1: ORCS-ASVM

$Input$:
$Training\ dataset\ D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
$Population\ size\ n = 10$

$Maximum\ number\ of\ iterations\ MaxIter\ =\ 100$
$Search\ space\ for\ hyperparameters:$
$C\ \in\ [0.1, 100], \gamma\ \in\ [0.0001, 10]$
$Discovery\ probability\ p^{\mathrm{m}}\ =\ 0.25$
$Levy\ flight\ scaling\ factor\ \alpha\ =\ 0.01$
$Output:$
$Optimized\ SVM\ hyperparameters\ (C\_opt, \gamma\_opt)$
$Trained\ Adaptive\ SVM\ model$
$Step\ 1: Initialization$
$Randomly\ initialize\ n\ =\ 10\ solutions:$
$Example:$
$Population\ =\ [(C_1, \gamma_1), (C_2, \gamma_2), \ldots, (C_{10}, \gamma_{10})]$
$e.g., (C_1, \gamma_1)\ =\ (10, 0.01), (C_2, \gamma_2)\ =\ (5, 0.001), etc.$
$Evaluate\ the\ fitness\ of\ each\ solution\ using\ 5 - fold\ cross - validation:$
$Fitness\ =\ SVM\_accuracy(C, \gamma)$
$Example\ fitness: [85\%, 80\%, 87\%, \ldots]$
$Step\ 2: One\ Rank\ Cuckoo\ Search$
$Sort\ population\ by\ fitness\ in\ descending\ order:$
$Example\ sorted\ population: [(C_3, \gamma_3), (C_1, \gamma_1), \ldots, (C_{10}, \gamma_{10})]$
$Example\ ranks: C_3\ >\ C_1\ >\ C_4\ > \ldots$
$Step\ 3: Levy\ Flight - based\ Exploration$
$For\ each\ solution\ (C_{\mathrm{i}}, \gamma_{\mathrm{i}}):$
$-\ Generate\ a\ new\ solution\ (C\_new, \gamma\_new)\ using\ Levy\ flight:$
$(C\_new, \gamma\_new)\ =\ (C_{\mathrm{i}}, \gamma_{\mathrm{i}})\ +\ \alpha\ *\ Levy(\lambda)$
$Example: (C_3, \gamma_3)\ =\ (15, 0.005)$
$Levy(\lambda) \sim random\ value\ (e.g., [0.5, -0.2])$
$New\ solution\ =\ (15\ +\ 0.010.5, 0.005\ -\ 0.010.2)$
$=\ (15.005, 0.0048)$
$-\ Ensure\ bounds: C\ \in\ [0.1, 100], \gamma\ \in\ [0.0001, 10]$
$Example: Adjust\ C\_new\ =\ min(max(C\_new, 0.1), 100).$
$Evaluate\ the\ fitness\ of\ the\ new\ solution:$
$Example: SVM\_accuracy(C\_new, \gamma\_new)\ =\ 89\%$
$Replace\ the\ current\ solution\ if\ the\ new\ fitness\ is\ better:$
$If\ fitness(C\_new, \gamma\_new)\ >\ fitness(C_3, \gamma_3), replace\ (C_3, \gamma_3).$
$Step\ 4: Cuckoo\ Discovery\ Probability$
$With\ a\ probability\ p^{\mathrm{m}}\ =\ 0.25:$
$-\ Replace\ the\ worst - ranked\ solution\ with\ a\ new\ random\ solution:$
$Example: Replace\ (C_{10}, \gamma_{10})\ =\ (50, 0.01)\ with\ random\ (C\_new, \gamma\_new).$
$Step\ 5: Adaptation\ of\ Search\ Strategy$
$Adaptively\ adjust\ \alpha\ and\ p^{\mathrm{m}}\ as:$
$\alpha\ =\ \alpha\ *\ exp(-t\ /\ MaxIter),$
$p^{\mathrm{m}}\ =\ 0.25\ +\ 0.5\ *\ (t\ /\ MaxIter)$
$Example: At\ iteration\ t\ =\ 50, \alpha\ =\ 0.01\ *\ exp(-50/100)\ =\ 0.0061$
$p^{\mathrm{m}}\ =\ 0.25\ +\ 0.5\ *\ (50\ /\ 100)\ =\ 0.5$
$Step\ 6: Convergence\ Check$
$If\ the\ termination\ condition\ is\ met\ (e.g., t\ =\ MaxIter\ or\ no\ improvement\ for\ 10\ iterations):$
$Exit\ the\ loop.$
$Otherwise, go\ back\ to\ Step\ 2.$
$Step\ 7: Train\ Final\ Adaptive\ SVM$
$Use\ the\ best\ solution\ found\ (e.g., C\_opt\ =\ 15, \gamma\_opt$
$=\ 0.005)\ to\ train\ the\ final\ SVM\ on\ the\ full\ dataset.$
$Step\ 8: Return\ Results$
$Output\ C\_opt\ =\ 15, \gamma\_opt\ =\ 0.005, and\ the\ trained\ Adaptive\ SVM\ model.$

4.5. Adaptive support vector machine

By improving the monitoring and management of air pollution, the SVM approach is used to consider a classification issue for the application, which is composed of m sample-label pairs: $T = (w_j, z_j), (j = 1,2, \dots, m), w_j \in Q$ is a training set, and $z_j \in \{-1, +1\}$ is a class label. To optimize the margin between the hyperplane and the support vectors (nearest data points), a hyperplane is created by using the equation $\omega^S w + a = 0$, where ω is the vector of hyperplane coefficients and a is a bias factor in Eq 2.

$$\begin{gathered} min_{\omega,a} \frac{1}{2} \|\omega\|^2 \\ s.t. z_j\big(\langle \omega, w_j \rangle + a\big) \geq 1 \end{gathered} \tag{2}$$

Complex environmental patterns make it challenging to identify a hyperplane that can accurately and fully separate the data points. The prediction model's generalizability may be diminished by overfitting caused by such categorization complexity. A soft margin strategy is used to address this, rephrasing the optimization issue as follows in Eq 3.

$$\begin{gathered} min_{\omega,a,t} \frac{1}{2} \|\omega\|^2 + D \sum_{j=1}^{m} \varepsilon_j \\ s.t. z_j\big(\langle \omega, w_j \rangle + a\big) \geq 1 - \varepsilon_j, j = 1,2, \dots, m \end{gathered} \tag{3}$$

Here, ε_j is the penalty coefficient that trades between misclassification tolerance, and ε denotes the slack variable. A nonlinear mapping Φ_w onto the higher dimension of the data set is used to enable linear separation since data are frequently not separable in a linear way. This is achieved by solving Lagrange's dual form optimization problem, as shown in Eq 4.

$$\begin{gathered} \underset{\alpha}{min} \sum_j \alpha_j - \frac{1}{2} \sum_j \sum_i \alpha_j \alpha_i z_j z_i \langle \Phi(w_j) \Phi(w_i) \rangle_E \\ s.t.\ 0 \leq \alpha_j \leq D, \sum_i \alpha_j z_j = 0 \end{gathered} \tag{4}$$

The Lagrange multiplier is represented by α_j. The following Eq 5 is the decision function used to categorize the pollution levels:

$$Class(w) = sign\left(\sum_i \alpha_j z_j \langle \Phi(w_j) \Phi(w) \rangle_E + a \right) \tag{5}$$

SVM construction uses the kernel function $L(w, z) = \langle \Phi(w) \Phi(z) \rangle_E$, which determines the distance or similarity between points. Linear, polynomial, and radial basis functions are some of the most common kernel functions. The selection of an appropriate kernel is a significant first step in solving any classification task included in air pollution monitoring, and this can be done by cross-validation or through expertise. Here, four kernel functions are thoroughly investigated to identify which one is suitable enough to predict the outcome of air quality. The approach achieves precise and timely forecasts of air pollution through the capability of the embedded system to manage complicated and nonlinear data relationships.

4.6. One-rank cuckoo search (ORCS)

The ORCS algorithm was chosen as our optimization method for tracking and controlling air pollution based on its superior solution quality and convergence when compared to simpler optimizers like PSO or DE. In the context of air pollution monitoring systems, ORCS is better suited for solving sensor calibration and position since it provides better exploration and exploitation. Also, its one-rank update makes it better suited for edge computing, with fewer resources, and it has a lower computational overhead. In terms of responding to pollution forecasting and intervention, ORCS will give the fastest response times and the most accurate estimates for real-time action. ORCS is based on the brood-raising behavior of several species of cuckoos, which are parasites, creating an optimization algorithm to optimize ML predictions by optimizing sensor locations and calibration. Lévy flights are used in the CS to produce novel solutions. The process of looking for novel solutions is divided into two phases. To identify the optimal answer, the new solution is assessed and rated according to its fitness following both phases.

The ORCS was developed to improve the traditional CS's convergence rate and solution quality. The exploration and exploitation stages of ORCS are combined using the one-rank (q_{pq}) parameter, which conducts M (population size) evaluations every iteration as opposed to $2M$ evaluations in the original method in Eq 6.

$$q_{pq}^{Iter+1} = q_{pq}^{Iter} - 0.5/C \tag{6}$$

Where C is the number of dimensions of the objective function. Furthermore, a limit determined by the best solutions approach is suggested as a substitute for the invalid dimension. This enhances the rate of convergence of the conventional approach. The mechanism of being the best solution is presented. ORCS is implemented for the best possible air pollution monitoring and management.

Initialization: Enhancing the embedded system used for monitoring and controlling air pollution, a solution vector with $3m$ components are represented by each cuckoo nest $W_c(c = 1, \ldots, M)$, where m is the number of sensor elements. The sensor locations selected for air pollution monitoring are the first m components. This section's elements are all natural numbers. The sensor setup or calibration settings make up the remaining components. As a result, the following Eq 7 represents the structure of a solution vector for the air pollution monitoring problem.

$$W_c = [w_1, \ldots, w_m, O_{CH1}, \ldots, O_{CHm}, R_{CH1}, \ldots, R_{CHm}] \tag{7}$$

In the subsequent ORCS, each layer of the population was started at random. The following Eqs 8–10 show how the solution was initialized for the air pollution parameters and sensor calibration:

$$w_j = round\left[w_{min,j} + rand_1 \times \left(w_{max,j} - w_{min,j}\right)\right]; j = 1, \ldots, m \tag{8}$$

$$O_{CH,j} = O_{CHmin,j} + rand_1 \times \left(O_{CHmax,j} - O_{CHmin,j}\right); j = 1, \ldots, m \tag{9}$$

$$R_{CH,j} = R_{CHmin,j} + rand_1 \times (R_{CHmax,j} - R_{CHmin,j}); j = 1, \dots, m \tag{10}$$

Where $rand_1$ is an integer chosen at random and extracted from the uniform distribution [0 1]. The concentration levels of the pollutant are calculated as follows in Eq 11 when sensor calibration is involved with air pollution monitoring.

$$R_{CH,j} = \sqrt{(O_{CG,j}/oe_{CH,j})^2 - (O_{CH,j})^2}; \ j = 1, .., m \tag{11}$$

Each nest has to have its fitness function calculated after setup. The calculation of the fitness function is done as in Eq 12.

$$E_S = OE + L_o \sum_{j=1}^{a} (U_j - U_j^{lim})^2 + L_r \sum_{h=1}^{aq} (J_j - J_j^{lim})^2 + L_u \sum_{j=1}^{m} (oe_{CH,j} - oe_{CH,j}^{lim})^2 \tag{12}$$

The penalty factors for pollutant concentration levels, sensor calibration, and real-time data prediction accuracy are denoted by $L_{o,} \, L_r, and \, L_u$, respectively.

Generation of new cuckoo nests via Lévy flights: To improve the embedded system for monitoring and controlling air pollution, Mantegna's algorithm was used for Lévy flights. The new nests were created using the following Eq 13.

$$W_c^{new} = Wbest_c + \alpha \times rand_2 \times \Delta W_c^{new} \tag{13}$$

Where $Wbest_c$ is the previous best nest > 0, and the step size factor and ΔW_{cj}^{new} are the enhanced values computed as follows in Eq 14.

$$\Delta W_c^{new} = u \times \frac{\sigma_w(\beta)}{\sigma_z(\beta)} \times (Wbest_c - Hbest) \tag{14}$$

Eqs 15–17are explained as follows:

$$U = \frac{rand_w}{|rand_z|^{1/\beta}} \tag{15}$$

$$\sigma_w(\beta) = \left[\frac{\Gamma(1+\beta) \times sin\frac{\pi\beta}{2}}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right]^{1/\beta} \tag{16}$$

$$\sigma_z(\beta) = 1 \tag{17}$$

Here, $\Gamma(\cdot)$ is the gamma distribution function, and β is the distribution factor, which ranges from 0.3 to 1.699.

Randomizing and Alien Eggi's discovering: There is a chance (o_b) that a host bird would discover foreign eggs, that is, cuckoo eggs, and discard or leave them when a cuckoo bird deposits its egg in its nest. Like the Lévy flights, the process of a host bird finding alien eggs also results in novel solutions, which are explained in Eq 18.

$$W_c^{dis} = Wbest_c + L \times \Delta W_c^{dis} \tag{18}$$

Where L is established in the following Eq 19:

$$L = \begin{cases} 1, & if\ rand_3 < o_b \\ 0, & otherwise \end{cases} \tag{19}$$

The increased value, ΔW_c^{dis}, is determined by Eq 20.

$$\Delta W_c^{dis} = rand_4 \times [rando_1(Wbest_c) - rando_2(Wbest_c)] \tag{20}$$

Where $rando_1(Wbest_c)$ and $rando_2(Wbest_c)$ indicate random perturbations in the nest places in $Wbest_c$. Air pollution monitoring and control systems are optimized through the ORCS algorithm, improving sensor placement, calibration, and real-time data prediction accuracy by fusing an embedded system approach with intelligent optimization approaches. The approach is appropriate for edge computing environments because of its one-rank updating mechanism, which lowers computational costs. ORCS achieves the aim of improving air pollution monitoring and control accuracy by combining intelligent optimization approaches with an embedded system approach.

5. Results

The ORCS-ASVM paradigm improves machine learning-based prediction on embedded systems through the use of IoT and machine learning to guarantee precise and effective air pollution monitoring and management. This model was trained on a Windows 10 platform using Python 3.11, an Intel i7 processor, and 32 GB of RAM. The results presented in this section correspond to 75% of the training set.

To monitor air pollution using inexpensive sensors combined with the ORCS-ASVM model, the proposed embedded system's performance is evaluated in Figure 2. The solution is very practical for dynamic settings due to its outstanding scalability (97%), flexibility (95%), and IoT applicability (98%). It offers accurate and reliable air quality predictions with a 96.1% real-time accuracy rate. Additionally, the 97% cost-effectiveness of the model guarantees that the system can be deployed economically, making it suitable for many applications of air pollution monitoring, such as low-resource settings and smart cities. This validates the real-time monitoring stability and effectiveness of the system.

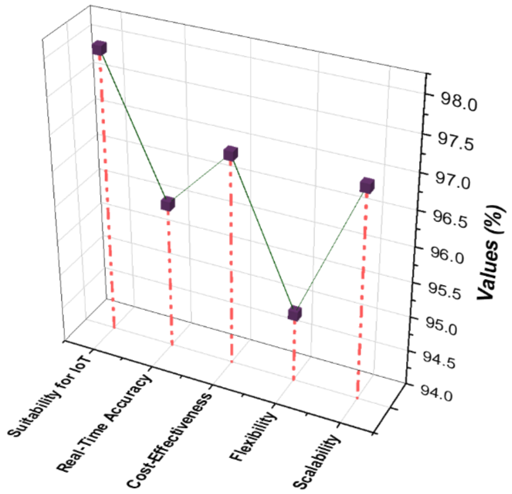

Figure 2. Performance evaluation of the ORCS-ASVM model for real-time air quality monitoring.

The dataset contains sensor readings taken from two Captor nodes measuring air pollutants [SOx, NOx, and particulate matter (PM2.5)] and environmental parameters (humidity and temperature) on November 25, 2020, organized in an hourly time series. As summarized in Table 1 (before normalization), the uncleaned sensor data indicated a range of inconsistencies and noise, which is standard for lower-cost sensors. In order to increase data reliability and make appropriate predictions, the dataset was preprocessed for smoothing and normalization. The cleaned and normalized dataset in Table 2 (after normalization) was essential to train the proposed ORCS-ASVM model for the embedded machine learning system. In pollution detection and monitoring involving an IoT-based environment, this process is crucial to allow for accurate decisions and real-time air quality forecasting.

Table 1. Air pollutant data before normalization from sensor readings.

Index	Date	NO_2	O_3	SO_2	PM2.5	PM10	CO_2	Air Quality Index
0	2020-11-25 01:00	0.090234	-0.581091	-0.567857	0.558341	0.417868	1309.433428	425.041933
1	2020-11-25 02:00	0.471677	-0.745888	-0.249477	0.870987	0.696653	1815.518471	567.328733
2	2020-11-25 03:00	0.697741	-0.736194	0.039977	1.056266	0.965293	2271.040377	679.204103
3	2020-11-25 04:00	0.923598	-0.669952	0.232744	0.965627	0.875536	2272.791060	662.476377
4	2020-11-25 05:00	1.064965	-0.536599	0.425713	0.927082	0.857532	2189.166693	646.037103
5	2020-11-25 06:00	1.064965	-0.252244	0.695749	0.870909	0.792402	1948.841726	610.055534
6	2020-11-25 07:00	1.018609	0.092119	1.393223	0.529141	1.183812	977.813426	358.873642
7	2020-11-25 08:00	0.951838	0.343435	3.049208	0.165160	0.012641	755.889869	297.996894
8	2020-11-25 09:00	0.895366	0.441422	1.971939	0.120038	-0.038303	679.986051	305.940974
9	2020-11-25 10:00	0.556374	0.547878	1.737461	0.146206	0.008149	802.170120	300.281724

Table 2. Air pollutant data after normalization for model training.

Index	Date	NO_2	O_3	SO_2	PM2.5	PM10	CO_2	Air quality index
0	2020-11-25 01:00	70.60	13.59	38.62	364.61	411.73	1309.433428	425.041933
1	2020-11-25 02:00	89.11	0.33	54.36	420.96	486.21	1815.518471	567.328733
2	2020-11-25 03:00	100.08	1.11	68.67	463.68	541.95	2271.040377	679.204103
3	2020-11-25 04:00	111.04	6.44	78.20	454.81	534.00	2272.791060	662.476377
4	2020-11-25 05:00	117.90	17.17	87.74	448.14	529.19	2189.166693	646.037103
5	2020-11-25 06:00	117.90	40.05	101.09	437.25	511.79	1948.841726	610.055534
6	2020-11-25 07:00	109.05	63.09	185.41	312.76	349.20	977.813426	358.873642
7	2020-11-25 08:00	112.41	87.98	217.44	275.53	303.47	755.889869	297.996894
8	2020-11-25 09:00	109.67	95.84	213.62	263.51	289.86	679.986051	305.940974
9	2020-11-25 10:00	93.22	104.43	152.59	271.25	302.27	802.170120	300.281724

The performance of the suggested ORCS-ASVM system for pollution control and air quality monitoring is shown in Figure 3. With strong pollutant detection capabilities of 94.2% for $PM_{2.5}$, 95.2% for SO_2, and 94.8% for NO_2, the system guarantees effective, dependable, and assured air quality monitoring. It is appropriate for large-scale applications because of its quick sensor calibration time (5 minutes) and wide system coverage area (10 km²). Such implications make the incorporation of the proposed system acceptable and viable to treat issues of air pollution in intelligent cities and other related contexts, highlighting its efficacy and capacity to scale up.

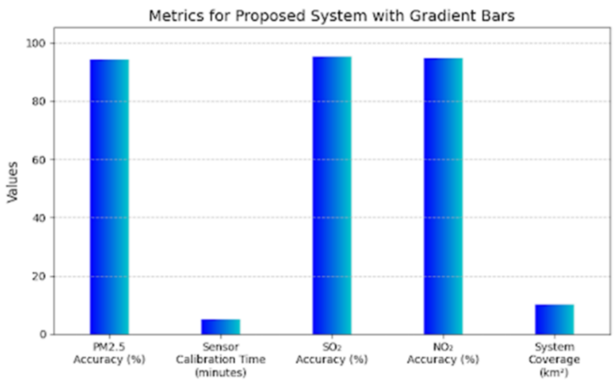

Figure 3. Evaluation of the ORCS-ASVM model's performance for air pollution monitoring and control.

The ORCS-ASVM model performs better compared to more traditional models such as random Forest [24] and time-space-type meteorology (TSTM) [25] in major performance indicators. The proposed model enhances air pollution detection and control through improved prediction accuracy, reduced error rates, and increased reliability.

Accuracy: The ratio of the number of instances predicted correctly to the number of predictions determines the accuracy of the model. ORCS-ASVM shows the highest accuracy of 96.1%, compared to random forest at 81.18% and TSTM at 95.2%, as shown in Figure 4 and Table 3. This level of precision in IoT applications guarantees accurate pollution identification with improved real-time monitoring of air quality.

Precision: The ratio of correctly forecast positive instances to all predictable positives. This is essential for the embedded system used in air pollution monitoring and control because it can detect high pollution occurrences without wrongly identifying low pollution as high. In this case, the ORCS-ASVM model achieved a precision of 95.3%, higher than the random forest model (82.13%) and the TSTM model (89.5%), as shown in Figure 4 and Table 3. This indicates its high dependability in forecasting air quality occurrences.

Recall: Estimates the ability of a model to find all relevant instances, especially when determining pollution levels within air quality monitoring. This metric is essential; it ensures that the system that was embedded can select high-pollution events with precision, as shown in Figure 4 and Table 3. The other models have recall values of 69.54% (random forest) and 88.3% (TSTM), while the proposed model ORCS-ASVM shows an excellent result of 96.7%, thus showing better detection.

F1 score: Evaluates if the classifier's balance between real positives, incorrect positives, and negative results is optimal by combining accuracy and recall. It is used to compare the air pollution monitoring system's performance in Figure 4 and Table 3. The ORCS-ASVM model achieves the highest F1 score of 95.2% over other existing models (random forest at 74.2% and TSTM at 89.5%), showing its great accuracy on air quality predictions.

Table 3. Evaluation of the random forest, TSTM, and ORCS-ASVM models' effectiveness in monitoring and controlling air pollution.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Random forest [24]	81.18	82.13	69.54	74.2
TSTM [25]	95.2	89.5	88.3	89.5
ORCS-ASVM (proposed)	96.1	95.3	96.7	95.2

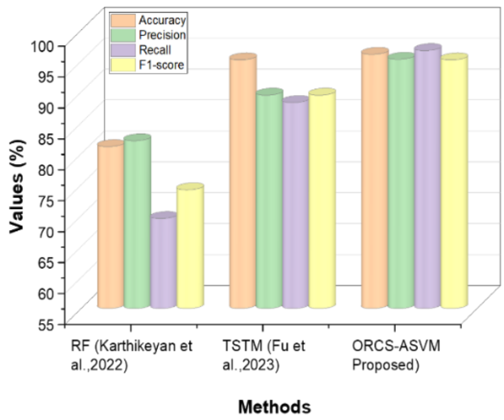

Figure 4. Comparison of performance metrics for air pollution prediction models.

Error rate: The error rate in monitoring and controlling air pollution refers to the percentage of failed predictions made by the model. The error rate of the current random forest model is 18.82% (Figure 5 and Table 4). However, the proposed ORCS-ASVM model presents an error rate of 13.21%, which is significantly inferior. This further proves the accuracy and reliability of this model for applications in real-time pollution management and air quality monitoring in embedded systems.

Table 4. Evaluation and comparison of existing models and the proposed model for monitoring and controlling air pollution.

Methods	Error rate (%)
Random forest [24]	18.82
ORCS-ASVM (proposed)	13.21

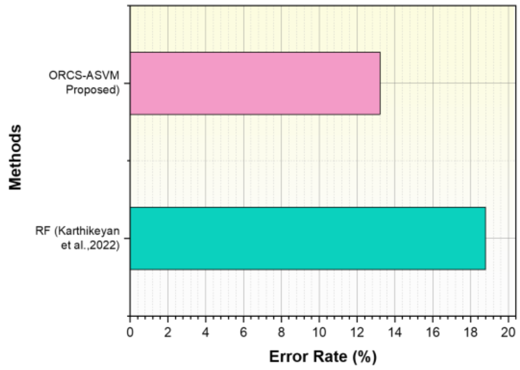

Figure 5. Analysis of error rate efficiency in air pollution monitoring.

6. Discussion

The ORCS-ASVM is an embedded ML system with considerable efficiency in real-time air pollution monitoring and control. The system incorporates low-cost sensors, edge computing, and an optimized hybrid machine learning model, and deals with key limitations of classical air quality monitoring systems like sensor data noise, larger delays, and high costs [25]. The ORCS-ASVM model showed considerable advantages over random forest and TSTM across all important evaluation results. The ORCS-ASVM model showed better predictive power (96.1%) than TSTM (95.2%) and random forest (81.18%). The ORCS-ASVM was also superior in precision (95.3%), recall (96.7%), and F1-score (95.2%), improving the system's ability to accurately classify pollution. The model displayed an error rate of 13.21%, indicating effective generalization under changing real-time environmental conditions.

A significant strength of this system is its internal edge computing architecture, which uses a Raspberry Pi [25]. This means that sensor data can be processed locally with minimal latency. As a result, in a dynamically changing environment like an urban context, real-time monitoring can be done without latency. The use of min-max normalization increases the quality of the data, allowing the SVM model to learn more efficiently, especially when the sensor readings are raw data that are noise-prone, depending on hardware quality. The ORCS function improves the convergence of the SVM by learning the sensor calibration and data classification with the SVM error [16]. The market-based solution used the balancing of exploration and exploitation function of Lévy flights, improving the accuracy of predictions with minimal overhead while still being able to be run on an edge device with limited computing resources.

Although the system has its advantages, it can only detect pollutants, namely $PM_{2.5}$, SO_2, and NO_2. As a result, future studies should expand the limited detection capabilities of the pollutant sensor system, improve the durability of the sensors, and deploy to a wider variety of geographical areas [19]. In addition, incorporating satellite data and developing privacy-respecting design features would increase the accuracy and security of the proposed model, providing an improved real-time air quality monitoring system. The ORCS-ASVM model is a scalable, flexible, and efficient real-time air quality monitoring system. It includes low-cost sensors and event-driven computing; the optimized machine learning model is adaptable to meet the needs of IoT-based smart environmental monitoring systems.

7. Conclusion

Low-cost sensors, edge computing, and the IoT were used to create a low-cost, real-time air pollution monitoring and control embedded system. The embedded edge device utilized by the system was a Raspberry Pi to monitor air pollutants such as SO_2, NO_2, and $PM_{2.5}$. With the one-rank cuckoo search-driven method, the model was able to increase sensor accuracy and decrease reliance on computing, allowing pollution to be managed on time. Smoothing methods of noise reduction and normalizing showed more reliability of sensor data. In comparison to traditional models, the proposed system demonstrated effective capacity in reducing error. Scalable and cost-effective maintenance of air quality was achieved, with a high $PM_{2.5}$ detection accuracy of 94.2%, SO_2 detection accuracy of 95.2%, NO_2 detection accuracy of 94.8%, a calibration time of 5 minutes, and a coverage area of 10 km^2. This scalable, cost-effective solution proved ideal for smart cities, agriculture, and any environment requiring continuous monitoring. Overall, this method provides a solid foundation for widespread adoption in IoT-based pollution monitoring networks. The system requires low-cost sensors, though their lifetime and precision may be limited.

Future scope: Future work can focus on expanding the system to support multi-pollutant integration across wider geographic regions using a network of edge devices. Incorporating advanced deep learning models could enhance predictive accuracy and adaptiveness to complex environmental patterns. Integration with satellite data and weather forecasting systems may further improve real-time response capabilities. The system could also benefit from energy-efficient hardware enhancements to support long-term deployment. Additionally, deploying Blockchain for secure data sharing and transparency in environmental reporting is worth exploring. Future validation through large-scale field tests in diverse urban environments will strengthen the model's generalizability and reliability.

Use of AI tools declaration

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

Conflict of interest

The authors declare no conflict of interest.

References

1. Bălă GP, Râjnoveanu RM, Tudorache E, et al. (2021) Air pollution exposure—the (in) visible risk factor for respiratory diseases. *Environ Sci Pollut Res* 28: 19615–19628. https://doi.org/10.1007/s11356-021-13208-x
2. Anjum MS, Ali SM, Subhani MA, et al. (2021) An emerged challenge of air pollution and ever-increasing particulate matter in Pakistan: A critical review. *J Hazard Mater* 402: 123943. https://doi.org/10.1016/j.jhazmat.2020.123943

3. Ahmed N, Hamid Z, Mahboob F, et al. (2022) Causal linkage among agricultural insurance, air pollution, and agricultural green total factor productivity in the United States: Pairwise Granger causality approach. *Agriculture* 12: 1320. https://doi.org/10.3390/agriculture12091320
4. Villanueva F, Ródenas M, Ruus A, et al. (2022) Sampling and analysis techniques for inorganic air pollutants in indoor air. *Appl Spectrosc Rev* 57: 531–579. https://doi.org/10.1080/05704928.2021.2020807
5. Seesaard T, Kamjornkittikoon K, Wongchoosuk C (2024) A comprehensive review of advancements in sensors for air pollution applications. *Sci Total Environ* 175696. https://doi.org/10.1016/j.scitotenv.2024.175696
6. Jin H, Kong F, Li X, et al. (2024) Artificial intelligence in microplastic detection and pollution control. *Environ Res* 119812. https://doi.org/10.1016/j.envres.2024.119812
7. Singh D, Dahiya M, Kumar R, et al. (2021) Sensors and systems for air quality assessment, monitoring, and management: A review. *J Environ Manag* 289: 112510. https://doi.org/10.1016/j.jenvman.2021.112510
8. Malla S, Sahu PK, Patnaik S, et al. (2023) IoT-enabled smart anti-smog towers: A novel approach to urban air pollution control. *Ingén Syst Inf* 28. https://doi.org/10.18280/isi.280605
9. Natrayan PC, LBT JRRG, Geetha BT, et al. (2022) IoT-enabled environmental toxicology for air pollution monitoring using AI techniques. *Environ Res* 205: 112574. https://doi.org/10.1016/j.envres.2021.112574
10. Tomson M, Kumar P, Barwise Y, et al. (2021) Green infrastructure for air quality improvement in street canyons. *Environ Int* 146: 106288. https://doi.org/10.1016/j.envint.2020.106288
11. Harrison RM, Van Vu T, Jafar H, et al. (2021) More mileage in reducing urban air pollution from road traffic. *Environ Int* 149: 106329. https://doi.org/10.1016/j.envint.2020.106329
12. Saini J, Dutta M, Marques G (2021) Sensors for indoor air quality monitoring and assessment through the Internet of Things: A systematic review. *Environ Monit Assess* 193: 66. https://doi.org/10.1007/s10661-020-08781-6
13. Subramaniam S, Raju N, Ganesan A, et al. (2022) Artificial intelligence technologies for forecasting air pollution and human health: A narrative review. *Sustainability* 14: 9951. https://doi.org/10.3390/su14169951
14. Krupnova TG, Rakova OV, Bondarenko KA, et al. (2022) Environmental justice and the use of artificial intelligence in urban air pollution monitoring. *Big Data Cogn Comput* 6: 75. https://doi.org/10.3390/bdcc6030075
15. Rahardja U, Aini Q, Manongga D, et al. (2023) Implementation of TensorFlow in air quality monitoring based on artificial intelligence. *Int J Artif Intell Res* 6. https://doi.org/10.29099/ijair.v6i1.430
16. Taheri S, Razban A (2021) Learning-based CO_2 concentration prediction: Application to indoor air quality control using demand-controlled ventilation. *Build Environ* 205: 108164. https://doi.org/10.1016/j.buildenv.2021.108164
17. Kaginalkar A, Kumar S, Gargava P, et al. (2021) Review of urban computing in air quality management as a smart city service: An integrated IoT, AI, and cloud technology perspective. *Urban Clim* 39: 100972. https://doi.org/10.1016/j.uclim.2021.100972
18. Kalajdjieski J, Trivodaliev K, Mirceva G, et al. (2023) A complete air pollution monitoring and prediction framework. *IEEE Access* 11:88730–88744. https://doi.org/10.1109/ACCESS.2023.3251346

19. Ramadan MN, Ali MA, Khoo SY, et al. (2024) Real-time IoT-powered AI system for monitoring and forecasting air pollution in an industrial environment. *Ecotoxicol Environ Saf* 283: 116856. https://doi.org/10.1016/j.ecoenv.2024.116856
20. Babak VP, Babak SV, Eremenko VS, et al. (2021) Monitoring the air pollution with UAVs. In: *Models and Measures in Measurements and Monitoring*, Springer. 191–225. https://doi.org/10.1007/978-3-030-70783-5_7
21. Han Y, Lam JC, Li VO, et al. (2021) A Bayesian LSTM model to evaluate the effects of air pollution control regulations in Beijing, China. *Environ Sci Policy* 115: 26–34. https://doi.org/10.1016/j.envsci.2020.10.004
22. Kalaivani G, Mayilvahanan P (2021) Air quality prediction and monitoring using machine learning algorithm-based IoT sensor: A researcher's perspective. In: *2021 6th Int Conf Commun Electron Syst (ICCES)*, IEEE. 1–9. https://doi.org/10.1109/ICCES51350.2021.9489153
23. Jiyal S, Saini RK (2020) Prediction and monitoring of air pollution using the Internet of Things (IoT). In: *2020 6th Int Conf Parallel Distrib Grid Comput (PDGC)*, IEEE. 57–60. https://doi.org/10.1109/PDGC50313.2020.9315831
24. Karthikeyan S, Moses LM, Perarasi T (2022) Machine learning-based air pollution monitoring and predicting causes for pollution. In: *IOP Conf Ser Earth Environ Sci* 1125: 012020. https://doi.org/10.1088/1755-1315/1125/1/012020
25. Fu L, Li J, Chen Y (2023) An innovative decision-making method for air quality monitoring based on big data-assisted artificial intelligence technique. *J Innov Knowl* 8: 100294. https://doi.org/10.1016/j.jik.2022.100294