



Hybrid Models for Forecasting Allocative Localization Error in Wireless Sensor Networks



Guo Li^a, Hongyu Sheng^{b,*}

^a State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu 610054, China

^b College of Robotics, Beijing Union University, Beijing 100101, China

ARTICLE INFO

Keywords:

Machine Learning
Allocative Localization Error
Wireless Sensor Networks
Metaheuristic Algorithms

ABSTRACT

This study presents a machine learning-based approach to forecast Allocative Localization Error (ALE) in Wireless Sensor Networks (WSNs), addressing challenges such as dynamic network topologies and resource constraints. The approach utilizes Radial Basis Function (RBF) models enhanced with advanced optimization algorithms, including Coot Optimization Algorithm (COA), Smell Agent Optimization (SAO), and Northern Goshawk Optimization (NGO) to improve ALE prediction accuracy. Hybrid models (RBCO, RBSO, and RFNG) are developed by integrating these optimization techniques, which refine critical RBF parameters, such as spread and center selection, through iterative optimization. Furthermore, an ensemble framework (RSNC) combines all three optimizers with RBF to achieve superior performance. The proposed methods are validated using R^2 and RMSE metrics, demonstrating their ability to minimize ALE, optimize resource allocation, and extend network lifespans. The study highlights the practical applicability of these models in real-world scenarios, such as environmental monitoring and industrial automation, offering enhanced efficiency and economic benefits. The RFNG model, in particular, achieved the lowest Mean Absolute Relative Error (MARE) of 0.049, demonstrating superior performance compared to other approaches in the test section. Moreover, RBNG obtained 0.069 and 0.978 values for the RMSE and R^2 , respectively, which were the most suitable values compared to other models, namely RBGO, RBSO, RSNC, and RBF. The results indicate that the proposed hybrid models significantly improve the prediction of ALE, leading to more efficient node deployment and better network management. This research provides valuable insights into leveraging machine learning for WSN optimization, benefiting researchers, network engineers, and industries relying on sensor networks for applications such as environmental monitoring, smart cities, and asset tracking.

1. Introduction

1.1. Study Background

Wireless Sensor Networks (WSNs) have surged in recent years because of the development of numerous new and cutting-edge applications and the increasing use of WSN networks in various fields. Researchers from all over the world are becoming increasingly interested in this field (Singh et al., 2023). Numerous sensor nodes with sensing and computing capabilities that are dispersed randomly make up a WSN. MOTE refers to each sensing device in a WSN network (Balasubramanian and Govindasamy, 2023). A WSN comprises several sensor nodes, one or more base stations called gateways, and an end-user (Matin and Islam, 2012; Sedaghat et al., 2023). One node's output is

wirelessly transmitted to the base station to gather, analyze, and log data. To transmit data from a source node to a sink node, every node in the WSN is a router (Kumar and Nayyar, 2014; Nayyar and Sharma, 2014; Nayyar and Singh, 2014). The sensor's data is made available to the end users through a website or an application on the console terminal. Sensor networks encounter problems typically not present in other networks, such as low communication range, limited processing and storage capacity, hardware availability, power and resource constraints, and cost (Muruganantham and Balakrishnan, 2021; Mur-alidharan et al., 2021; Imran et al., 2010; Sundani et al., 2011). Due to the high cost, time, and complexity associated with deployment and implementation in such networks, developers would prefer to obtain firsthand information on feasibility and reflectivity, which are crucial for system implementation before hardware implementation.

* Corresponding author.

E-mail address: 1314.sheng@163.com (H. Sheng).

Researchers are developing new protocols, strategies, and algorithms to overcome these obstacles and make sensor networks more dependable. This opens up new opportunities for sensor network deployment in more real-world contexts (Nayyar and Singh, 2015). It is, therefore, vitally important and desirable to assess and analyze the approaches that are being suggested. However, testing and evaluating the protocols or theories put forth through actual experiments is impossible because doing so would be more difficult, time-consuming, and expensive (Kim et al., 2020).

Thus, to solve this issue, "TESTBEDS, EMULATORS, and SIMULATORS are useful tools to test and evaluate the performance of protocols and algorithms submitted (Minakov et al., 2016). It is simple to create and evaluate a new protocol for sensor networks using simulation tools (Pandey and Kushwaha, 2019). This technique has several benefits, including minimal installation costs, ease of use, real-time data analysis, and the ability to determine side effects and other network-wide impacting aspects (Sobieh et al., 2006, Silmi et al., 2020).

The Allocative Localization Error (ALE) in WSNs is defined as the difference between the actual physical placement of the sensor nodes and that ascertained by the localization algorithms (Guo et al., 2014). A WSN comprises many sensor nodes spread across a geographical region that is expected to gather data and forward it to a base station (Frery et al., 2008). The position of the node has to be determined precisely to perform target tracking, environment monitoring, geographic routing, and other applications (Adday et al., 2022).

It seems that the error stems from several characteristics of WSN environments. Firstly, the environmental parameters such as signal strength reduction, interference, and multi-path fading affect the wireless signal strength and, thus, the distance estimation between the involved nodes (Kadel et al., 2020). Also, other hardware considerations, like imprecise sensors and small computational capabilities, are sources of localization errors. Also, since the nodes are not uniformly distributed, areas with high node density receive congested signals, while areas with low densities have poor coverage (Vuran and Akyildiz, 2009). Additionally, it can be stated that the selection of the localization algorithm affects the precision of node positioning. There is also the range-based approach, for instance, the time of flight, signal strength, and the range-free approach, including the connectivity and proximity approaches (Dardari et al., 2007). In this case, all the methods have their merits and demerits based on the balance between strictness, levels of work complexity, and resource utilization. Here, no single algorithm is preferably the best for all scenarios; therefore, variation in localization quality exists (He et al., 2013).

The implicational influence of ALE is as follows for the WSN applications. The location of the nodes may be slightly off, which will cause inaccurate data to be collected and, therefore, wrong analysis and decision-making are done (Motwakel et al., 2023, Wang et al., 2020, Gebremariam et al., 2023). Besides, it degrades the efficiency of the services sensitive to location information, such as target tracking and event detection. Therefore, reducing localization errors is a major actuality and research problem in WSNs as it fuels innovations in algorithm complexity, signal processing methods, and node placement. These challenges can be achieved, and WSNs can improve the precision and reliability of localization or many applications (Alrajeh et al., 2015).

1.2. Literature Review

This section has addressed several methods to improve node localization accuracy. Several studies (Singh et al., 2020, Sankar et al., 2022, Nidhya et al., 2023) have used ML to improve localization accuracy, providing Bayesian methodology for WSN node positioning as provided by Morelande et al. (Morelande et al., 2008). The recommended strategy is a development of past work called gradual rectification. These 2 approaches are contrasted under various circumstances, with the Cramér-Rao bound (CRB) serving as the standard. The recommended method was demonstrated to be clearer than the prior one.

Additionally, Ghargan et al. (Gharghan et al., 2016) outlined a procedure wherein an Artificial Neural Network (ANN) is separately hybridized with 3 optimization procedures: Gravitational Search Algorithm (GSA), Backtracking Search Algorithm (BSA), and Particle Swarm Optimization (PSO). The GSA-ANN hybrid outperformed the alternative techniques with an estimated inaccuracy of average distances of 0.2 meters in both indoor and outdoor scenarios. According to the latest poll, Bouallegue and Ahmadi (Ahmadi and Bouallegue, 2017) gathered several cutting-edge ML methods for WSN node location. The cumulative localization error distribution curves were analyzed using various techniques, including the ANN, Decision Tree (DT), Naive Bayes (NB), and Support Vector Machine (SVM) methods. Based on each ML algorithm's aggregated localization error distributions, this investigation discovered that NB outperformed all other methodologies. Bhatti et al. (Bhatti et al., 2020) used many supervised, unsupervised, and ensemble ML approaches to build the "if Ensemble" outlier detection system for an indoor localization context. In this instance, Random Forest (RF) classifiers, K-nearest neighbor (KNN), and SVM are the learning under supervision approaches; isolation forest (i Forest) is the unsupervised learning strategy. These techniques are applied to stacking, an ensemble learning process. Each ML method's output is compared with the model's performance, including stacking. The model whose name is stacking has a high localization accuracy of 97.8% when employing the recommended outlier identification methods. In recent, an approach to node localization known as KernelELM according to Hop-count Quantization (KELM-HQ) was presented by Wang et al. (Wang et al., 2020). The qualified KELM calculates the unidentified nodes' locations. It is demonstrated that the proposed method reduces the localization error by 34.6%, 19.2%, and 11.9%, respectively, compared to the fast-SVM, GADV-Hop, and DV-Hop-ELM algorithms. Ultimately, by applying a parameter estimation ML model, this work seeks to solve the issue of positioning accuracy in earlier studies.

Moreover, several bioinspired algorithms have been developed to create a less complex algorithm for the range-based method. Gopakumar and Jacob (Gopakumar and Jacob, 2008) first presented a node localization technique based on PSO, miming a fish swarm's foraging activity to find food. Although the implementation of this method tended to become trapped in a local optimum, leading to premature convergence, the approach produced good early results. Cheng and Xia (Cheng and Xia, 2016) introduced a modified version of Cuckoo Search (CS), which increased the traditional CS algorithm's convergence rate. They changed the mutation probability and the random walk step size to enhance the search procedure. Goyal and Patterh (Goyal and Patterh, 2014) used CS in 2014 to localize nodes in WSNs. It reduced the localization error with observable outcomes. This is primarily due to the CS algorithm's tuning settings, which facilitate the computation process.

In addition, Sharma et al. (Sharma et al., 2021) explored how ML techniques optimized low-power IoT nodes in smart city applications such as traffic monitoring, waste management, and healthcare. It highlighted challenges like network coverage, energy consumption, and bandwidth and proposed ML to enhance WSN-IoT systems. The survey, the first of its kind, revealed that supervised learning was the most widely used approach (61%), followed by reinforcement (27%) and unsupervised learning (12%), demonstrating their potential to improve smart city operations. Nayak et al. (Nayak et al., 2021) focused on energy conservation in WSNs, which is essential for sustaining battery-powered nodes in hazardous environments and IoT applications. It emphasized the need for efficient communication protocols to prolong network lifespan. Traditional methods addressed many WSN challenges but lacked accuracy in predicting system behavior. Therefore, low-complexity mathematical models were explored to tackle tasks like routing, data fusion, localization, and object tracking. The paper extensively reviewed machine learning techniques applied to address WSN issues, focusing on routing problems. Kim et al. (Kim et al., 2021) explored using WSNs in dynamic environments to monitor and collect raw data for transmission to a base station. It highlighted the technical

challenges of deploying WSNs in real-world settings and the limitations of traditional task-specific techniques in adapting to dynamic conditions. ML techniques, particularly deep learning (DL), were identified as effective solutions due to their ability to handle dynamic situations through adaptive learning processes. DL techniques, employing deep neural networks, were shown to extract high-level features from raw sensor data, offering benefits such as reduced computational complexity, enhanced energy efficiency, and improved feasibility in finding optimal solutions. However, the study also noted drawbacks, including long training times, the need for large datasets, and high energy consumption unfavorable for resource-constrained WSNs. While ML applications in WSNs have been reviewed extensively, fewer studies have focused on DL applications. This review presented recent advancements in ML techniques for WSNs, emphasizing DL methods and their corresponding neural network architectures for various applications.

1.3. Objectives of the Research

WSNs play a vital role in applications like environmental monitoring, industrial automation, and target tracking, where accurate localization is critical for reliable data acquisition and efficient network performance. However, ALE remains a significant challenge arising from uneven spatial distribution, resource constraints, and dynamic environmental changes. These challenges lead to inefficient routing, increased energy consumption, and reduced network lifespan, necessitating robust solutions to enhance localization and network efficiency.

ML models have emerged as powerful tools to address ALE by capturing non-linear relationships and extracting patterns. Despite their potential, traditional ML methods face limitations such as slow convergence and high computational demands, particularly in resource-constrained WSNs. This study aims to develop advanced ML frameworks for ALE prediction by integrating Radial Basis Function (RBF) with optimization algorithms—Coot Optimization Algorithm (COA), Smell Agent Optimization (SAO), and Northern Goshawk Optimization (NGO). Hybrid models (RFCO, RBSO, RFNG) and an ensemble framework (RSNC) are proposed to enhance ALE prediction, improve resource allocation, and prolong network lifespan.

The contributions include:

- Development of hybrid ML models for accurate ALE prediction.
- Enhanced localization accuracy and network efficiency across dynamic WSN scenarios.
- Real-world applicability in critical applications like environmental monitoring.
- Practical benefits include reduced costs and extended network lifespan.

By addressing ALE comprehensively, the study advances WSN optimization techniques, ensuring robust deployment in real-world scenarios.

The organization of this study is structured as follows:

Section 2 provides a detailed description of the dataset used in the analysis. In **Section 3**, the developed model and optimization algorithms are described, along with the relevant mathematical equations. This section also outlines the evaluation metrics employed and discusses the hybrid method utilized in the study. The findings of the generated models are displayed in Part 4, along with a thorough comparison of their respective performances. Part 5 wraps up the research by summarizing the main conclusions and ramifications.

2. Data Collection

2.1. Data Description

Various factors are related to the ALE of WSN, like trance range,

Table 1

The statistical properties of dataset variables.

Variables	Indicators				
	Category	Min	Max	Avg	St. Dev.
anchor_ratio	Input	10	30	20.52	6.708
trans_range	Input	12	25	17.88	3.093
node_density	Input	100	300	159.8	70.86
iterations	Input	14	100	47.89	24.55
sd_ale	Input	0.0033	1.092	0.266	0.182
ALE	Output	0.3964	2.268	0.981	0.396

anchor ratio, nod density, iteration, SD-ALE, and ALE. The parameters used in this study as the inputs are transmission range, the number of anchors used, the number of participating nodes, the number of iterations, and the standard deviation of ALE and ALE. Transmission range defines the maximum distance for the sensor communication that affects the signal's strength and ability to localize the event. Anchor ratio means the ratio of fixed reference nodes needed to achieve position triangulation of the sensors. Thus, node density plays a critical role in the coverage of the network as well as the efficiency in the aggregation of data. Iteration count defines the number of loops to go through in training a model. SD-ALE measures the deviation range in localizations, and ALE denotes the real error to be predicted, which is imperative for evaluating a model's accuracy.

The mapping between the input and the output is described in detail in **Fig. 2**, where the changes of the input parameters visible affect the outcome. This graphic interpretation helps identify or comprehend these variables' relationship and trend direction. It is evident in WSNs that the interdependency between anchor ratio, node density, trans-range, number of iterations, SD-ALE, and ALE is one important factor that can affect the spatial localization of the nodes thoroughly.

Firstly, there is the anchor ratio, which is the ratio of the number of anchor nodes to all nodes in the network. An anchor ratio tends to be higher to localize the target accurately because the anchors represent landmarks. Node density, however, influences connectivity and coverage; higher node density can lower the ALE due to geometry imposed on localization algorithms. The trans-range controls the ability of nodes to communicate and influences the distance of communication. It affects the handshake and the topology control in a way that determines how the nodes exchange the localization information and constraints. All iterative steps within localization algorithms entail adjustments of the nodes' placements concerning the received signal strength or distance, directly impacting the degree of ALE reduction.

This study's data collection process involved gathering various attributes related to ALE in WSNs. The dataset used here was not a benchmark but was prepared based on certain measurements and conditions relevant to the objectives. **Table 1** summarizes the statistical properties of the variables used for training and testing the ML models. It went through several preliminary steps to have a very reliable dataset regarding the analysis of ALE in WSNs.

Simulations of different WSN scenarios were realized and provided data for various conditions related to node density, transmission range, and anchor ratio. This must-have environment has a wide set of conditions to run the subsequent analysis. Then, the key attributes are measured and recorded, including factors that most greatly impact localization error, such as the nodes' spatial distribution and signal transmission quality. Such a measurement is important, as it directly affects ALE predictions.

Each of these measurements was followed by consistent data logging to ensure that the information collected would be reliable. This consistent logging of data during measurement was important for dataset integrity and accurately captured all the variables involved in the experiment. Before its analysis, the data had been pre-processed to remove inconsistencies or noise. This was an important step in data cleaning for machine learning models. Hence, this ensured that only

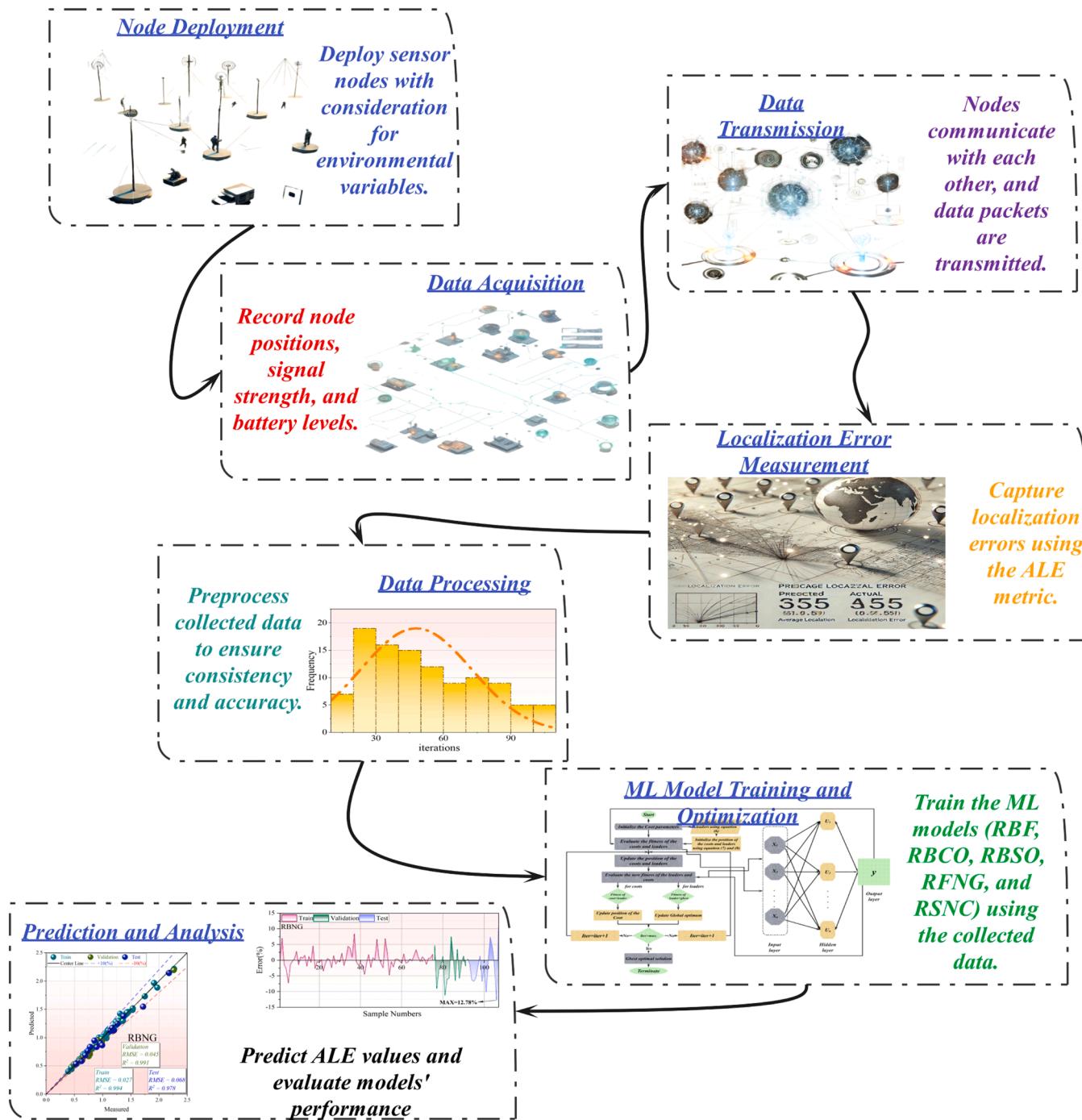


Fig. 1. Diagram of study process.

quality input that would result in quality predictions was available. Besides, the pre-processing step will remove outliers and irrelevant information to enhance the dataset's quality and its appropriateness for analysis and modeling in subsequent steps.

In addition, Fig. 1 is presented to show the data collection and modeling process.

3. Methods

3.1. Radial Basis Function (RBF)

Lowe and Broomhead (Broomhead and Lowe, 1988) describe that RBF according to popular belief, BF is a forward-propagating network

trained by a supervised approach. As seen in Fig. 3, the 3 main levels of the RBF are the entry, hidden, and final tiers. Sigmoid, Gaussian, and hardly multiquadric procedures are among the many variations of RBFs. The Gaussian Radial Basis operation with a center and a dispersion is one of the most well-known functions. This is a simple part of the network; a node does not conduct the action at the entry layer.

Moreover, there is no difference between the variable number and the node count of the input layer. The calculator layer is the second hidden layer, which uses RBF with circular type to find solutions. This layer receives data from the input layer, which applies a non-linear mapping using the RBF to the input value. The RBFNN's balanced operation calculates the difference between a given input and its midpoint. Nonetheless, there are other types of RBFs, including the

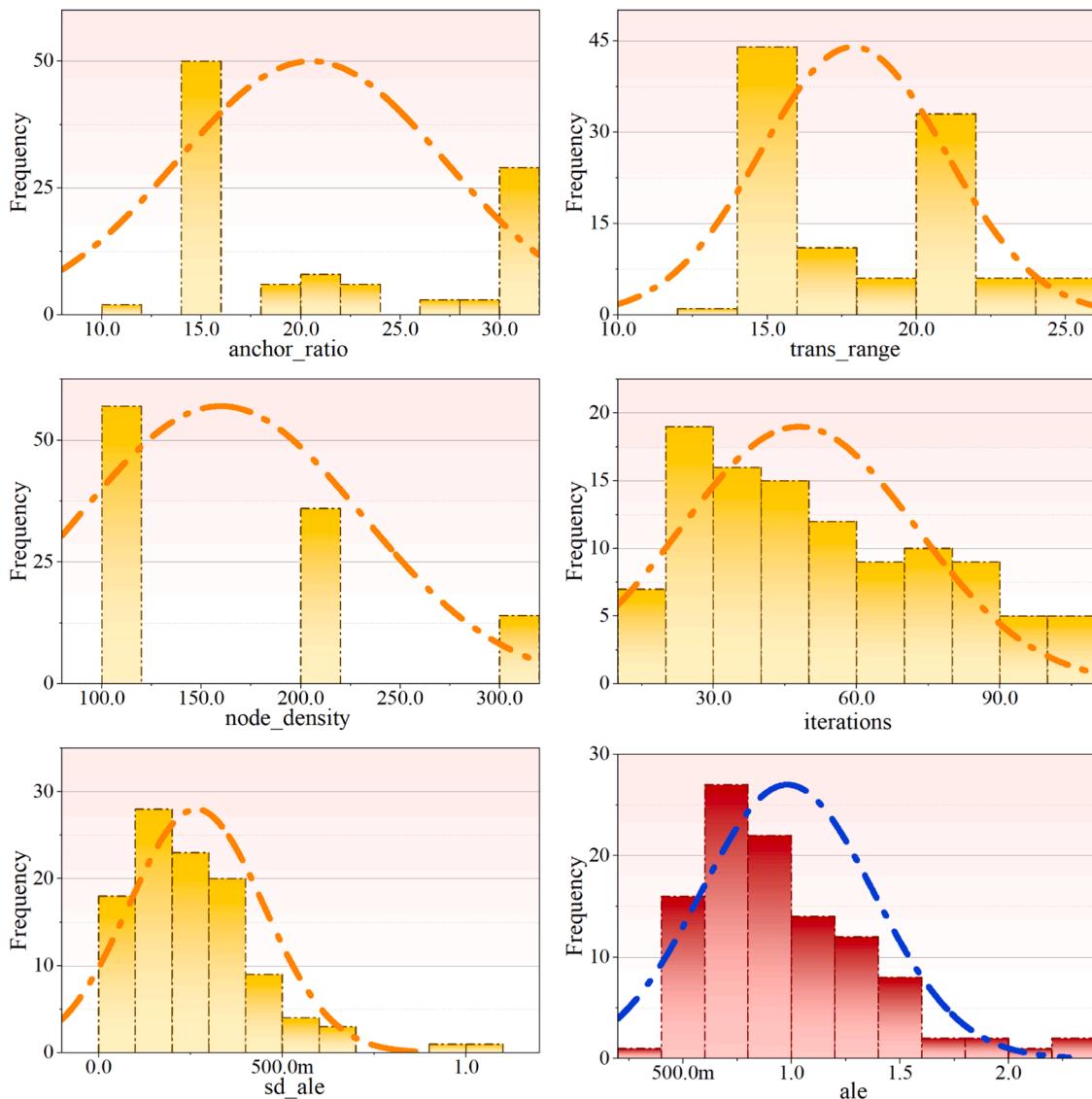


Fig. 2. Histogram distribution of input and output variables.

Sigmoid, Gaussian, inverse multi-quadratic, and hardly multiquadric functions. An established formula known as the Gauss RBF was developed by spread rate and center (Luan et al., 2006).

3.2. Coot Optimization Algorithm (COA)

Inspired by the coordinated actions of Coots, a kind of aquatic bird, the COA uses a metaheuristic optimization technique. In the water, coots exhibit a wide variety of locomotion patterns that are intended to get them to certain food sources or spots. These patterns include leader-driven, leader-guided, leader-ran, and random movements. These operations become part of the structure of the COOT algorithm. During implementation, the process starts by randomly generating a main majority, using Eq. (1) as described in the guidelines by (Naruei and Keynia, 2021):

$$\text{CootPos}(i) = \text{rand}(1, N) \times (UB - LB) + LB \quad (1)$$

$\text{CootPos}(i)$ is the positional value of a single coot; N is the number of parameters involved or the task's difficulty. LB and UB both serve as a representation of the top and lowest bounds of the exploration area that is being searched.

$$UB = [UB_1, UB_2, \dots, UB_N], LB = [LB_1, LB_2, \dots, LB_N] \quad (2)$$

Following the first majority arrangement, the placements of the coots are successively modified by applying 4 different motion patterns.

3.2.1. Random Movement

For this specific motion, the beginning position Q assignment is randomly selected using the procedure given in Eq. (3):

$$Q = \text{rand}(1, N) \times (UB - LB) + LB \quad (3)$$

In order to stay out of local best solutions, the location is changed following Eq. (4):

$$\text{CootPos}(i) = \text{CootPos}(i) + A \times R_2 \times (Q - \text{CootPos}(i)) \quad (4)$$

The value R_2 is a randomly generated integer between 0 and 1, and A is found by applying the procedure given in Eq. (5):

$$A = 1 - L \times \left(\frac{1}{\text{Iter}} \right) \quad (5)$$

L represents the current number of iterations, while "Iter" shows the maximum number allowed.

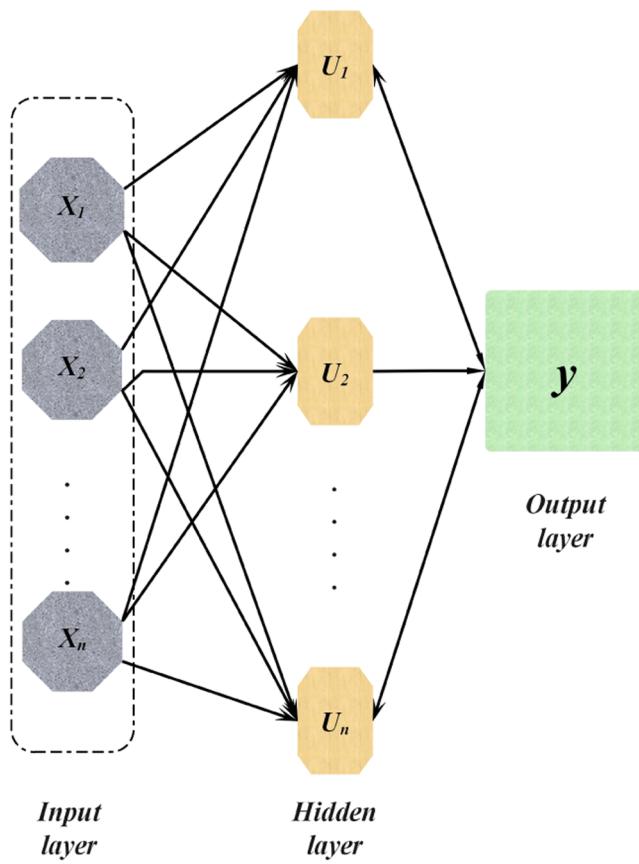


Fig. 3. RBF Flowchart.

3.2.2. Chain Movement

The formula in Eq. (6) may be used to calculate the mean position of 2 coot birds to perform the sequential motion.

$$CootPos(i) = \frac{CootPos(i-1) + CootPos(i)}{2} \quad (6)$$

The position of the next coot in the sequence is indicated by $CootPos(i-1)$.

3.2.3. Adjusting Position According to the Leader

Within each cluster, the location of a coot bird changes in response to the leader's position, causing the follower to move closer to the leader. The designation of the head is determined by the formula given in Eq. (7):

$$K = 1 + (i \bmod NL) \quad (7)$$

In this instance, i stands for the number allocated to the follower coot bird, K for the leader's index, and NL for the total number of leaders in the group. A coot's location is updated throughout this specific movement using the formula found in Eq. (8):

$$\begin{aligned} CootPos(i) &= LeaderPos(K) + 2 \times R_1 \times \cos(2R\pi) \\ &\times (LeaderPos(K) - CootPos(i)) \end{aligned} \quad (8)$$

$CootPos(i)$ represents the coot bird's current position, whereas $LeaderPosK$ represents the location of the chosen leader. R_1 While another random value, designated as R , is picked from the range [-1, 1], a

random value is selected from the interval [0, 1]. These values are part of the computation for updating the location described in Eq. (8).

3.2.4. Leader Movement

Based on the ideas presented in Eq. (9), the leader's positions change to move from localized optimal positions to global optimality.

$$LeaderPos(i) = \begin{cases} B \times B_3 \times \cos(2\pi R) \times (g_{best} - LeaderPos(i)) + g_{best} B_4 < 0.5 \\ B \times B_3 \times \cos(2\pi R) \times (g_{best} - LeaderPos(i)) - g_{best} B_4 \geq 0.5 \end{cases} \quad (9)$$

In this case, random values chosen from the interval [0, 1] are represented by B_3 and B_4 , whereas g_{best} denotes the best location that may be reached. Fig. 4 presents the flowchart of the RBF. The formula given in Eq. (10) is used to calculate the value of variable B :

$$B = 2 - L \times \left(\frac{1}{Iter} \right) \quad (10)$$

3.3. Northern Goshawk Optimization (NGO)

3.3.1. The Original Northern Goshawk Optimization

Effectively hunting and capturing prey influences the NGO algorithm's search strategy. As such, the algorithm consists of 3 stages: population building, identifying the prey, and apprehending the prey.

3.3.2. Initialization

First, as shown below, the original northern goshawk population can be represented by a matrix designated as X .

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,M} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,M} \end{bmatrix} \quad (11)$$

The i -th in the population is represented by the X_i , where $1 \leq i < N$. The population size is represented by N , while the objective function's dimension is denoted by M . The components of X_i for a single-objective optimization problem with upper bound UB and lower bound LB may be calculated as follows:

$$x_{ij} = LB + rand.(UB - LB), 1 \leq i \leq N; j \leq M. \quad (12)$$

3.3.3. Prey Identification

The northern goshawk will select its victim during the first phase and make an effort to track it down. This behavior shows the algorithm's ability to explore the whole potential space globally because the prey is chosen randomly. Eq. (14) mimics the northern goshawk attacking its prey, assuming that the prey ($prey_i$) as specified in Eq. (13), is the target selected by the individual X_i .

$$prey_i = X_{p,i} = 1, 2, \dots, N; p = 1, 2, \dots, i-1, \dots, N. \quad (13)$$

$$\begin{cases} X_i^{new} = X_i + r(prey_i - IX_i), Fit(prey_i) < Fit(X_i) \\ X_i^{new} = X_i + r(X_i - prey_i), Fit(prey_i) \geq Fit(X_i) \end{cases} \quad (14)$$

I is a vector made up of 1_s and 2_s , and r is a random vector with values ranging from 0 to 1. The purpose of adding r and I is to make the algorithm more stochastic so that the search space may be explored more thoroughly. After that, each X_i is updated by Eq. (15):

$$\begin{cases} X_i = X_i^{new}, Fit(X_i^{new}) < Fit(X_i), \\ X_i = X_i, Fit(X_i^{new}) \geq Fit(X_i). \end{cases} \quad (15)$$

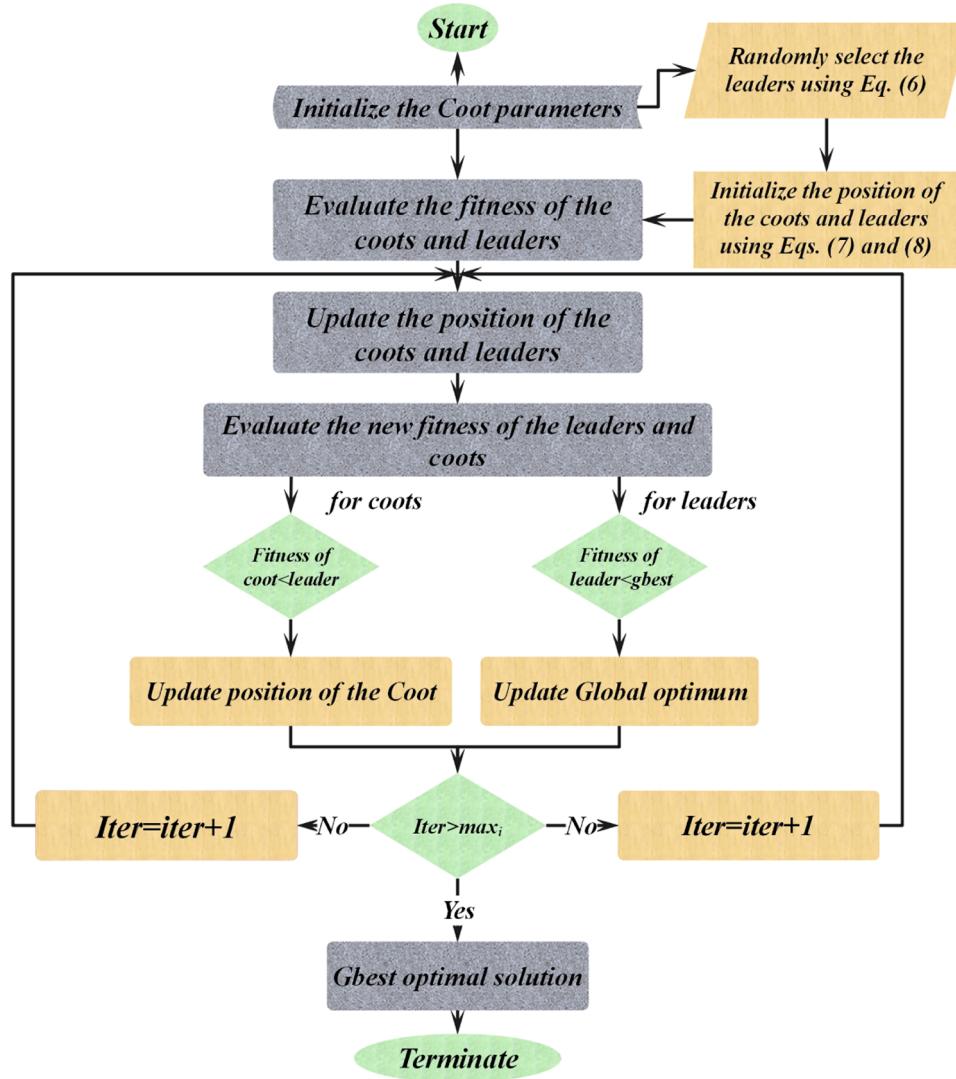


Fig. 4. COOT Flowchart.

3.3.4. Prey Capture

The victim starts to run away as the northern goshawk closes up on it and launches its assault. The northern goshawk has to keep hunting throughout this time. The northern goshawk's remarkable pursuit speed allows it to successfully pursue and eventually catch its prey in almost any situation. Concerning a circle whose radius is r , which denotes the range of the search, this stage can be repeated by applying Eq. (16):

$$X_i^{new} = X_i + R(2r - 1)X_i \quad (16)$$

R is computed as $R = 0.02(1 - t/T)$, where t is the current iteration and T is the maximum number of iterations. After this calculation, Eq. (17) is applied to each unique X_i .

$$\begin{cases} X_i = X_i^{new}, & Fit(X_i^{new}) < Fit(X_i), \\ X_i = X_i, & Fit(X_i^{new}) \geq Fit(X_i). \end{cases} \quad (17)$$

3.4. The improved Northern Goshawk Optimization

The NGO algorithm has the potential for enhancement in terms of exploration and exploitation, but it provides optimization with simplicity and precision. Its rudimentary prey-chasing paradigm might produce poor performance and sluggish convergence. To overcome this, an improved version known as ENGO incorporates multi-strategy opposite learning models and polynomial extrapolation.

3.4.1. Polynomial Interpolation Strategy

Finding the lowest point of an interpolating polynomial obtained from discrete data is possible using the technique known as polynomial extrapolation (Hu et al., 2022). Different polynomial degrees are approached using quadratic and cubic extrapolations, where the minimum is found by setting the derivative $\varphi'(t)$ to 0. The population of northern goshawks functions as discrete data inside the feasible space. To produce the quadratic interpolation function $\varphi(X)$ in Eq. (18), 3

individuals are chosen from this population: (X_i , X_{i+1} , and X_{i+2}).

$$\varphi(X) = a_0 + a_1 X + a_2 X^2. \quad (18)$$

$$\begin{cases} \varphi(X_i) = a_0 + a_1 X_i + a_2 X_i^2, \\ \varphi(X_{i+1}) = a_0 + a_1 X_{i+1} + a_2 X_{i+1}^2, \\ \varphi(X_{i+2}) = a_0 + a_1 X_{i+2} + a_2 X_{i+2}^2. \end{cases} \quad (19)$$

3 factors, a_0 , a_1 , and a_2 , may be obtained by applying Eq. (20) to Eq. (19).

$$\begin{cases} a_0 = \frac{(X_{i+1} - X_{i+2})\varphi(X_i) + (X_{i+2} - X_i)\varphi(X_{i+1}) + (X_i - X_{i+1})\varphi(X_{i+2})}{(X_i - X_{i+1})(X_{i+1} - X_{i+2})(X_{i+2} - X_i)} \\ a_1 = \frac{(X_{i+1}^2 - X_{i+2}^2)\varphi(X_i) + (X_{i+2}^2 - X_i^2)\varphi(X_{i+1}) + (X_i^2 - X_{i+1}^2)\varphi(X_{i+2})}{(X_i - X_{i+1})(X_{i+1} - X_{i+2})(X_{i+2} - X_i)} \\ a_2 = \frac{(X_{i+1} - X_{i+2})X_{i+1}X_{i+2}\varphi(X_i) + (X_{i+2} - X_i)X_{i+2}X_i\varphi(X_{i+1}) + (X_i - X_{i+1})X_iX_{i+1}\varphi(X_{i+2})}{(X_i - X_{i+1})(X_{i+1} - X_{i+2})(X_{i+2} - X_i)} \end{cases} \quad (20)$$

where the quadratic curve's minimum is located $\varphi'(X)$, equate $\varphi'(X)$ to $a_1 + 2a_2X = 0$. The quadratic curve $\varphi(X)$ hits its lowest value when X is equal to $\frac{a_1}{2a_2}$. This is then added to Eq. (20), allowing the following determination of X :

$$\begin{aligned} X^* &= \frac{1}{2} \\ &\times \frac{(X_{i+1}^2 - X_{i+2}^2)Fit(X_i) + (X_{i+2}^2 - X_i^2)Fit(X_{i+1}) + (X_i^2 - X_{i+1}^2)Fit(X_{i+2})}{(X_{i+1} - X_{i+2})Fit(X_i) + (X_{i+2} - X_i)Fit(X_{i+1}) + (X_i - X_{i+1})Fit(X_{i+2})}. \end{aligned} \quad (21)$$

At the end, the person is updated as shown in Eq. (22) following a comparison of the acquired X^* with the initial X_i :

$$\begin{cases} X_i = X^*, Fit(X^*) < Fit(X_i), \\ X_i = X_i, Fit(X^*) \geq Fit(X_i) \end{cases} \quad (22)$$

The general quality of the northern goshawk population will be improved when this surgery is applied to the whole population, which will help to get closer to the ideal solution.

3.4.2. Multi-Strategy Opposite Learning Method

Local optima are a common source of difficulty for optimization techniques, particularly in complicated situations. Using various learning techniques, the NGO algorithm overcomes this and improves its capacity to break out of local optima. These techniques are used to compute newly created individuals for every member of the northern goshawk population (Li et al., 2022, Mukherjee et al., 2021).

$$\tilde{X}_i = LB + UB - X_i \quad (23)$$

$$\bar{X}_i = rand\left[\frac{LB + UB}{2}, LB + UB - X_i\right] \quad (24)$$

$$\begin{aligned} \hat{X}_i &= LB + UB - X_i \\ \bar{X}_i &= rand\left[\frac{LB + UB}{2}, LB + UB - X_i\right] \end{aligned} \quad (25)$$

Fig. 5 shows the flowchart of the NGO.

3.5. Smell Agent Optimization (SAO)

An algorithm called SAO was presented by Salawudeen et al. (Salawudeen et al., 2021) to implement the relationship between the smell agent and the evaporated smell molecule. They modeled SAO in three phases: trailing, sniffing, and random.

3.5.1. Sniffing phase

Eq. (1) can be used to represent the position of molecules.

$$x_i^{(t)} = \begin{bmatrix} X_{(1,1)} & X_{(1,2)} & \dots & \dots & X_{(1,D)} \\ X_{(2,1)} & X_{(2,2)} & \dots & \dots & X_{(2,D)} \\ \dots & \dots & \dots & \dots & \dots \\ X_{(N,1)} & X_{(N,2)} & \dots & \dots & X_{(N,D)} \end{bmatrix} \quad (26)$$

where D is the number of dimensions, and N is the number of molecules (Salawudeen et al., 2018). The following equation was used to generate the initial positions of the molecules in the first instance:

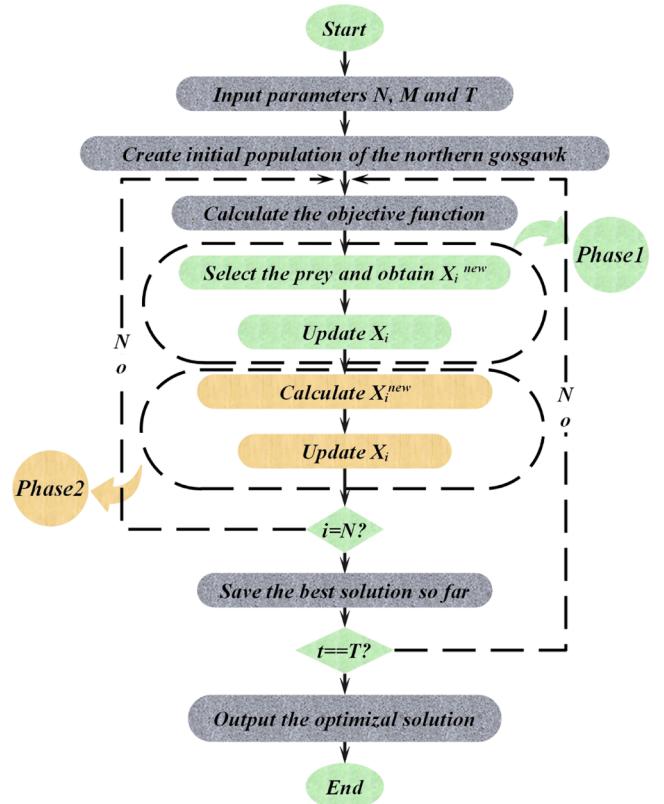


Fig. 5. Flowchart of NGO.

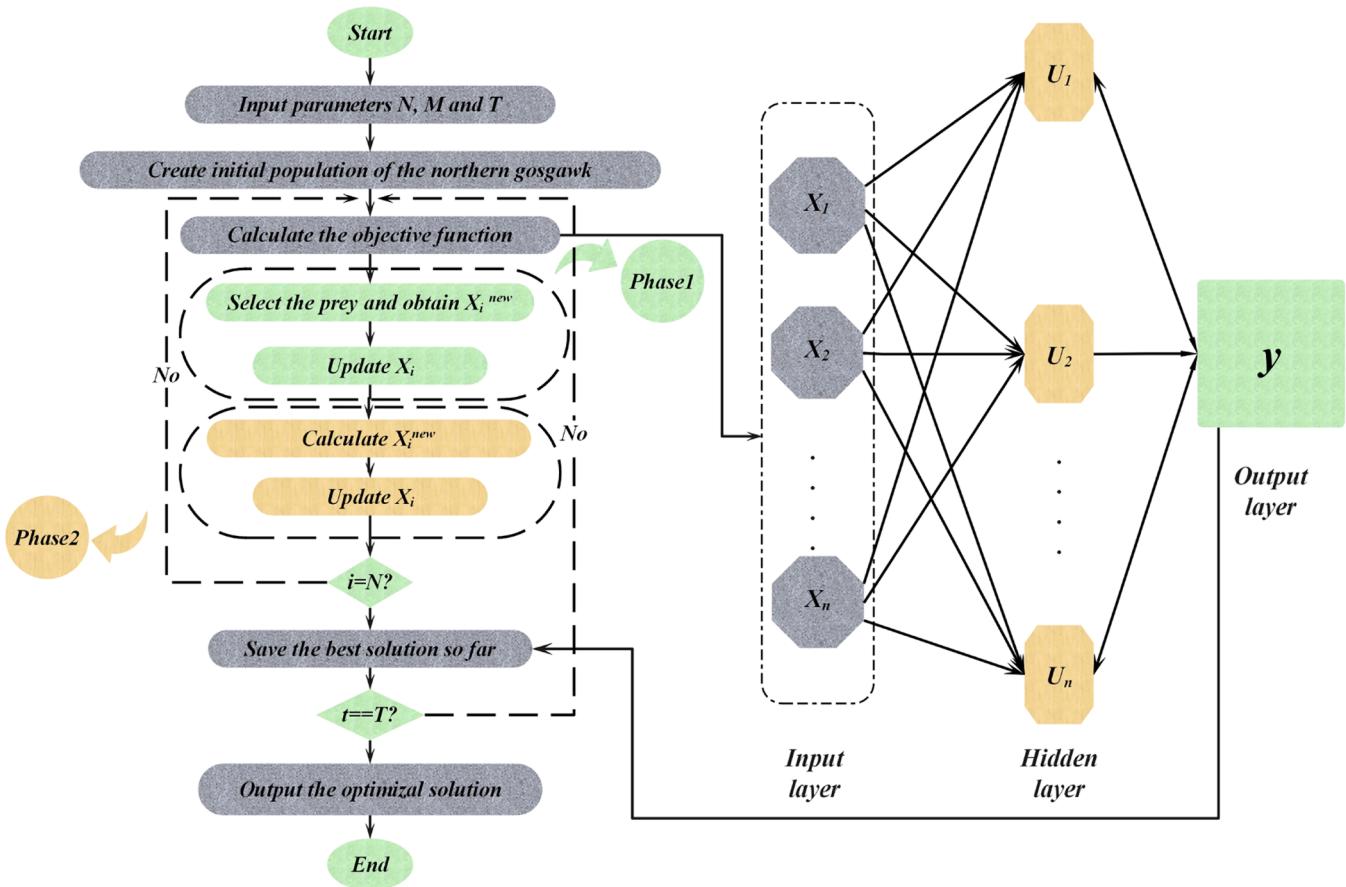


Fig. 6. Hybridization flowchart of RBF + NGO (RBNG).

$$x_i^{(t)} = lb_i + r_0 \times (ub_i - lb_i) \quad (27)$$

Where lb_i denotes the lower boundary of i th dimension and ub_i denotes the upper boundary, and r_0 is a number between 0 and 1.

Every agent has a velocity, which can be updated using the following equations and whose initial velocity can be represented by Eq. (3).

$$v_i^{(t)} = \begin{bmatrix} v_{(1,1)} & v_{(1,2)} & \dots & \dots & v_{(1,D)} \\ v_{(2,1)} & v_{(2,2)} & \dots & \dots & v_{(2,D)} \\ \dots & \dots & \dots & \dots & \dots \\ v_{(N,1)} & v_{(N,2)} & \dots & \dots & v_{(N,D)} \end{bmatrix} \quad (28)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)} \times \Delta t \quad (29)$$

where Δt is constant and equals 1; this allows the agent to perform a progressive step (Salawudeen et al., 2018). The position of molecules is updated using the following equation:

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)} \quad (30)$$

The formula below is used to update the velocity of molecules:

$$v_i^{(t+1)} = v_i^{(t)} + \nu \quad (31)$$

where the velocity update component, denoted by ν , can be computed as follows:

$$\nu = r_1 + \sqrt{\frac{3KT}{m}} \quad (32)$$

$$\frac{1}{2} nm\nu = \frac{3}{2} KnT \quad (33)$$

Where m is the mass of a molecule, T is its temperature, k is a fixed number, and r_1 is a real number that falls between 0 and 1.

The values of T and m , which are derived from the gas ideal theory, are 0.825 and 0.175, respectively.

3.5.2. Trailing mode

Agents mimic searching for the source of the scent during this phase. The mathematical equation that follows is created to mimic this behavior:

$$X_i^{(t+1)} = X_i^{(t)} + r_2 \times olf \times (X_{agent}^{(t)} - X_i^{(t)}) - r_3 \times olf \times (X_{Worst}^{(t)} - X_i^{(t)}) \quad (34)$$

where r_2 is a random number used to penalize olf influence to $X_{agent}^{(t)}$, and r_3 is a random number used to penalize olf influence to $X_{Worst}^{(t)}$, and olf denotes the capacity of the olfaction.

3.5.3. Random mode

The smell concentration may gradually vanish if the smell agents are widely spaced apart, which could result in traps in nearby areas. The random stage is simulated using the following equation, and the random phase aids agents in escaping from nearby areas:

$$X_i^{(t+1)} = X_i^{(t)} + r_4 \times SL \quad (35)$$

where r_4 is a random number that is used to penalize the SL value, and SL refers to a fixed number that represents the length of the step.

3.6. Performance Evaluation Metrics

- o RMSE (Root Mean Square Error) and R^2 (Coefficient of

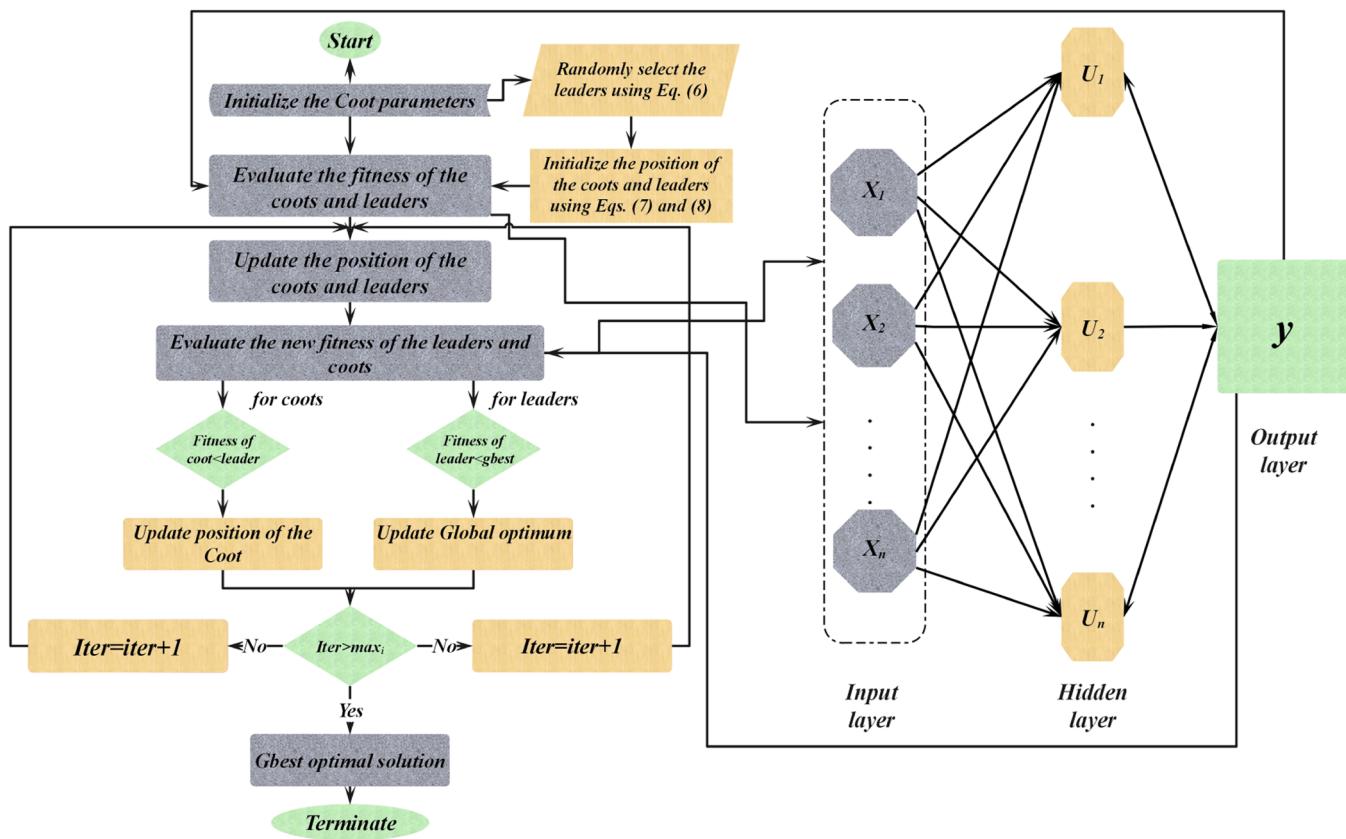


Fig. 7. Hybridization flowchart of RBF + COA (RBCO).

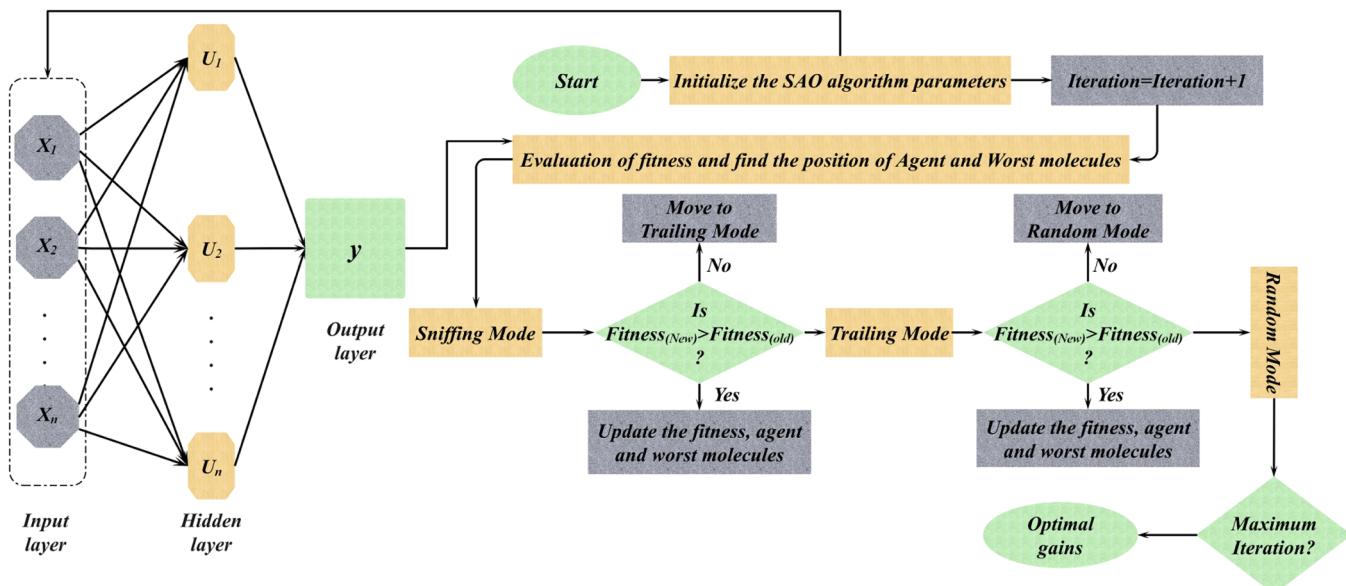


Fig. 8. Hybridization flowchart of RBF + SAO (RBSO).

Determination):

This is made feasible by utilizing the RMSE to give customers an understanding of the median size of errors between projected and actual values. The square root of the arithmetic mean of the squared discrepancies between the expected and actual numbers is used to calculate it. R^2 is a metric that demonstrates how well the independent variables can account for the variance in the dependent variable. It gauges the degree of relationship between the model and the data. An R^2 value of 1 in-

dicates the ideal match, and R^2 value units range from 0 to 1. N is the total number of data points; the observed values are represented by y_i , and the anticipated values are indicated by (\hat{y}_i) . R^2 and RMSE is assessed as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (36)$$

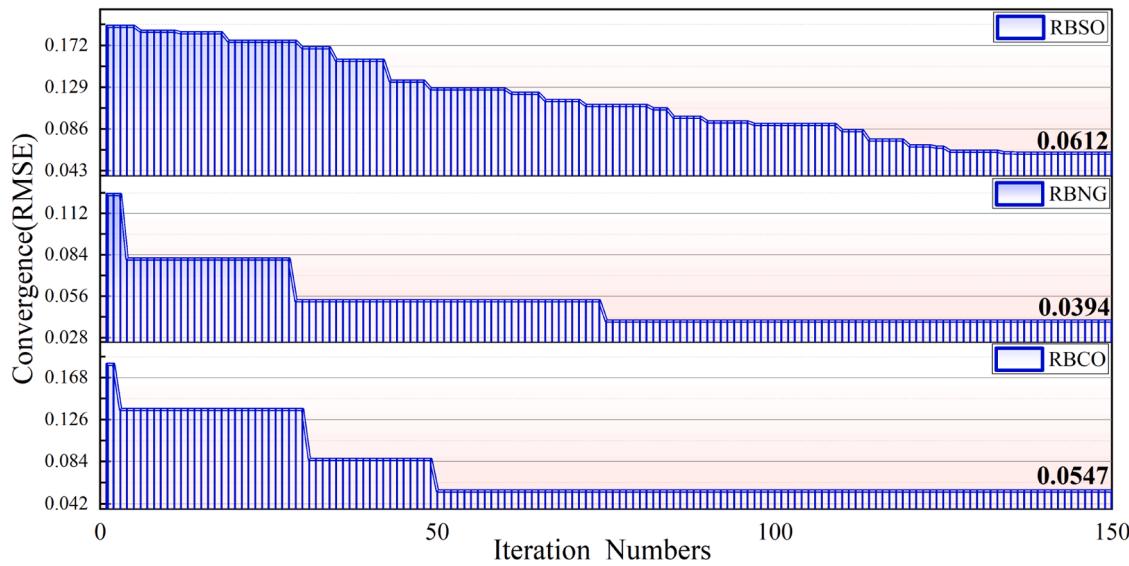


Fig. 9. Convergence graph of the hybrid models.

Table 2
Outcomes of the developed models.

Framework	Model	Phase	Index values				
			RMSE	R ²	MSE	NRMSE	MARE
Single	RBF	Train	0.055	0.979	0.003	0.001	0.057
		Validation	0.063	0.980	0.004	0.004	0.469
		Test	0.113	0.939	0.013	0.007	0.087
Hybrid (RBF+COO)	RBCO	Train	0.042	0.988	0.002	0.001	0.046
		Validation	0.050	0.986	0.002	0.003	0.059
		Test	0.095	0.952	0.009	0.006	0.069
Hybrid (RBF+NGO)	RBNG	Train	0.028	0.995	0.001	0.000	0.018
		Validation	0.045	0.991	0.002	0.003	0.045
		Test	0.069	0.978	0.005	0.004	0.049
Hybrid (RBF+SAO)	RBSO	Train	0.049	0.985	0.002	0.001	0.053
		Validation	0.063	0.955	0.004	0.006	0.058
		Test	0.119	0.946	0.014	0.011	0.082
Ensemble (RBF+ SAO +NGO+CO)	RSNC	Train	0.034	0.993	0.001	0.000	0.035
		Validation	0.046	0.973	0.002	0.005	0.043
		Test	0.096	0.966	0.009	0.009	0.067

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (37)$$

o Mean Squared Error (MSE)

The mean squared error is referred to as "MSE." The term "error" refers to the discrepancy between expected and actual numbers. Because of the squaring, bigger mistakes are penalized more severely than smaller ones.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (38)$$

where n is the number of observations, y_i is the actual value, and \hat{y}_i is the predicted value. o **Mean absolute relative error (MARE):**

$$MARE = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{y_i} \quad (39)$$

where, y_i is the measured value, and \hat{y}_i is the error value. o **Normalized Root Mean Square Error (NRMSE):**

A popular statistic for evaluating the precision of prediction models is the NRMSE. It is the RMSE normalized form, which indicates how much the observed and anticipated values deviate from one another. o

NRMSE Formula

The NRMSE is calculated as follows:

$$\frac{RMSE}{N} \quad (40)$$

N represents the number of datasets.

3.7. Hybridization

The integration of RBF Regression with the COA, SAO, and GOA involved data pre-processing, model development, and hybrid optimization. The core model, RBF, is a neural network that uses radial basis functions to handle non-linear relationships, making it suitable for predicting ALE in WSNs. RBF was initially trained with historical WSN data, but determining optimal parameters like spread and weights was challenging.

To enhance performance, COA was integrated with RBF by optimizing two key parameters: the spread, which affects function approximation smoothness, and center selection, which determines which data points act as centers. The initialized fitness evaluation, iterative parameters optimization, and model updating were performed by the proposed COA.

Similarly, GOA and SAO were combined with RBF to execute the

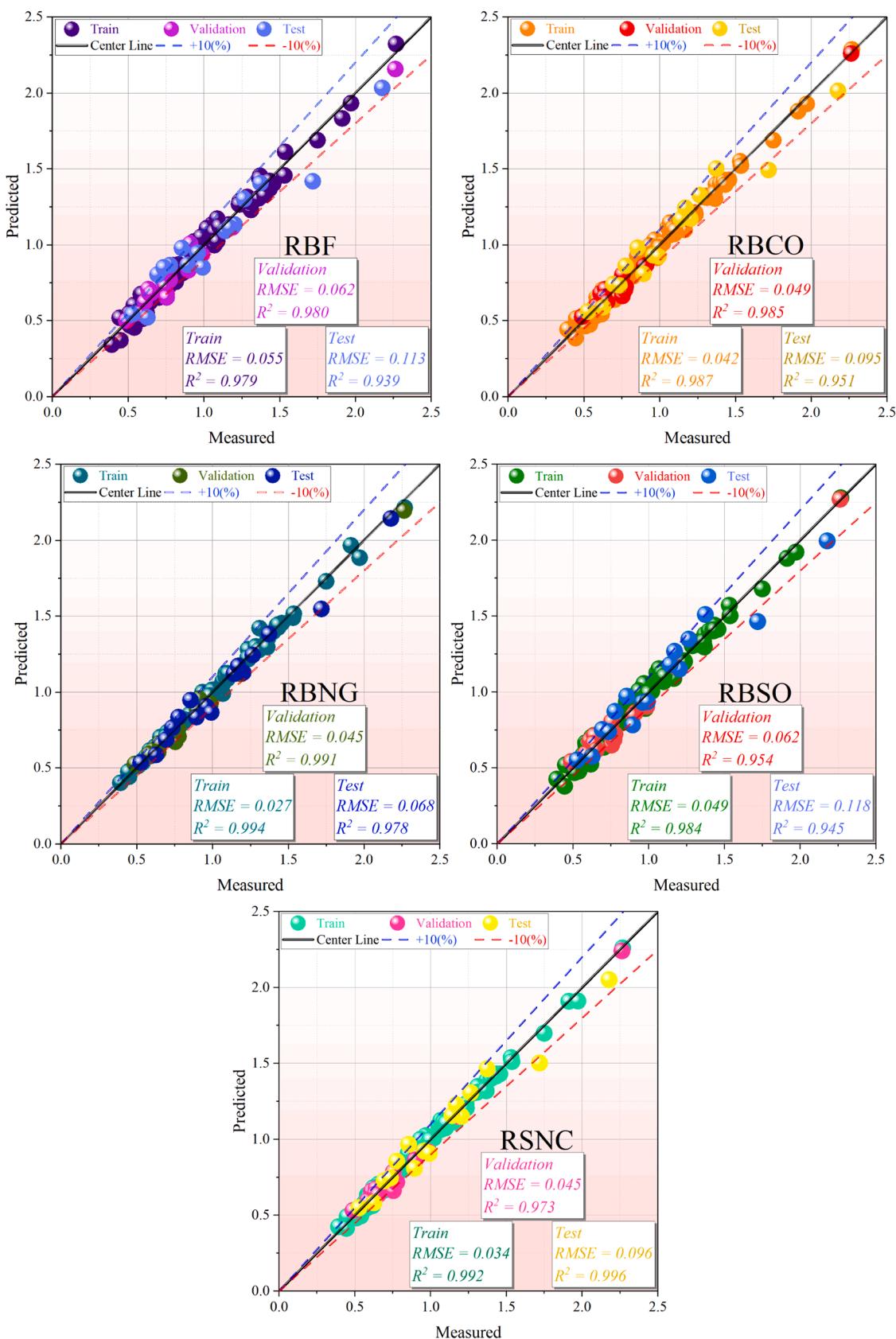


Fig. 10. Correlation between the predicted and measured values.

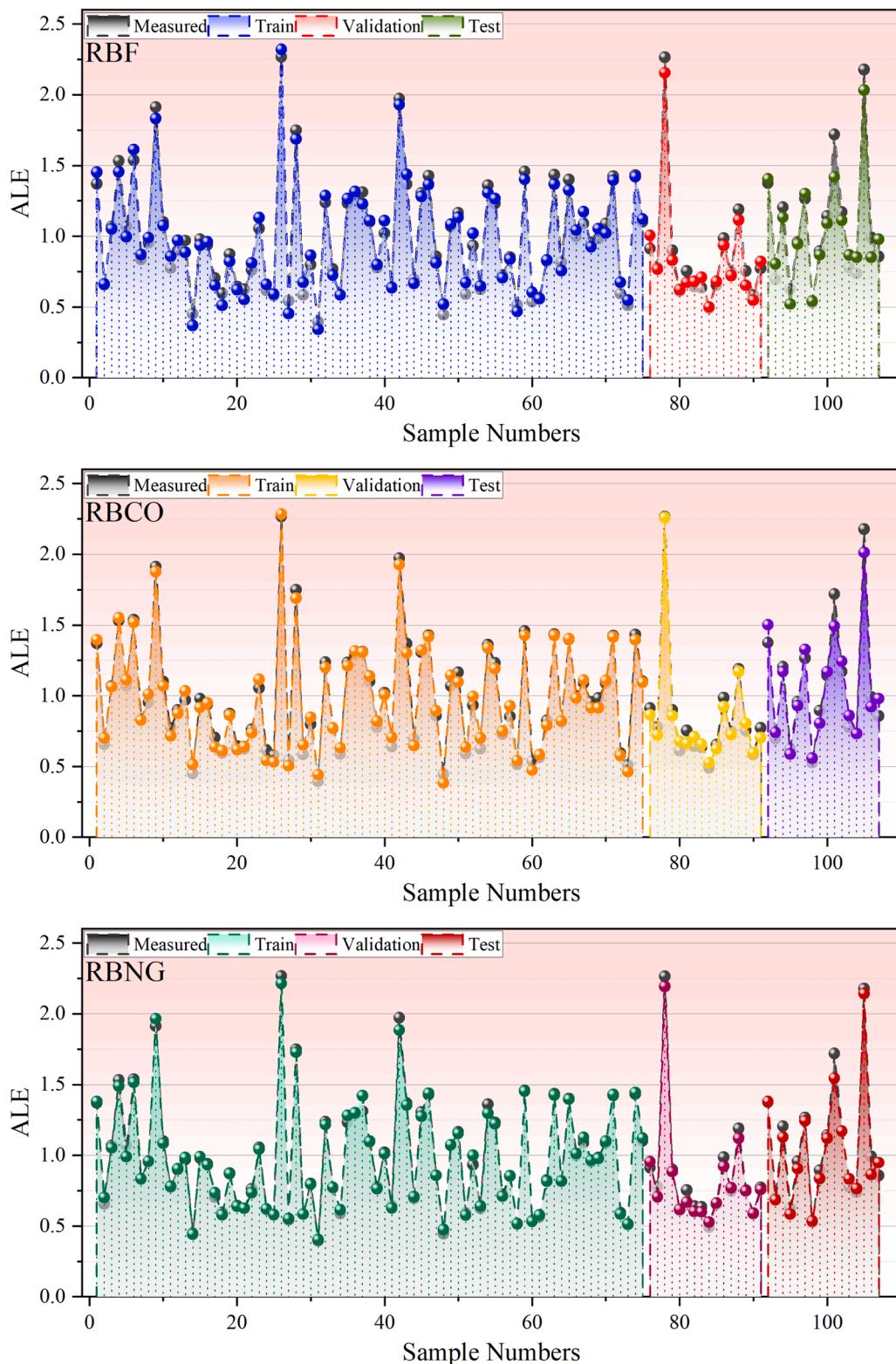


Fig. 11. The comparison between predicted and measured values.

optimization tasks in spread and center selection by balancing exploration and exploitation. GOA incorporated an adaptive search methodology that involved population initialization, evaluation of the fitness function, and updating parameters adaptively.

Figs. 6 to 8 show the flowchart of hybridization for RBNG, RBCO, and RBSO, respectively.

4. Results and Discussion

The results of the models are compared systematically using various charts and tables to identify whether the hybrid model gives better prediction accuracy. The result of this thorough examination pinpoints not only the best model but simultaneously underlines the limits beyond

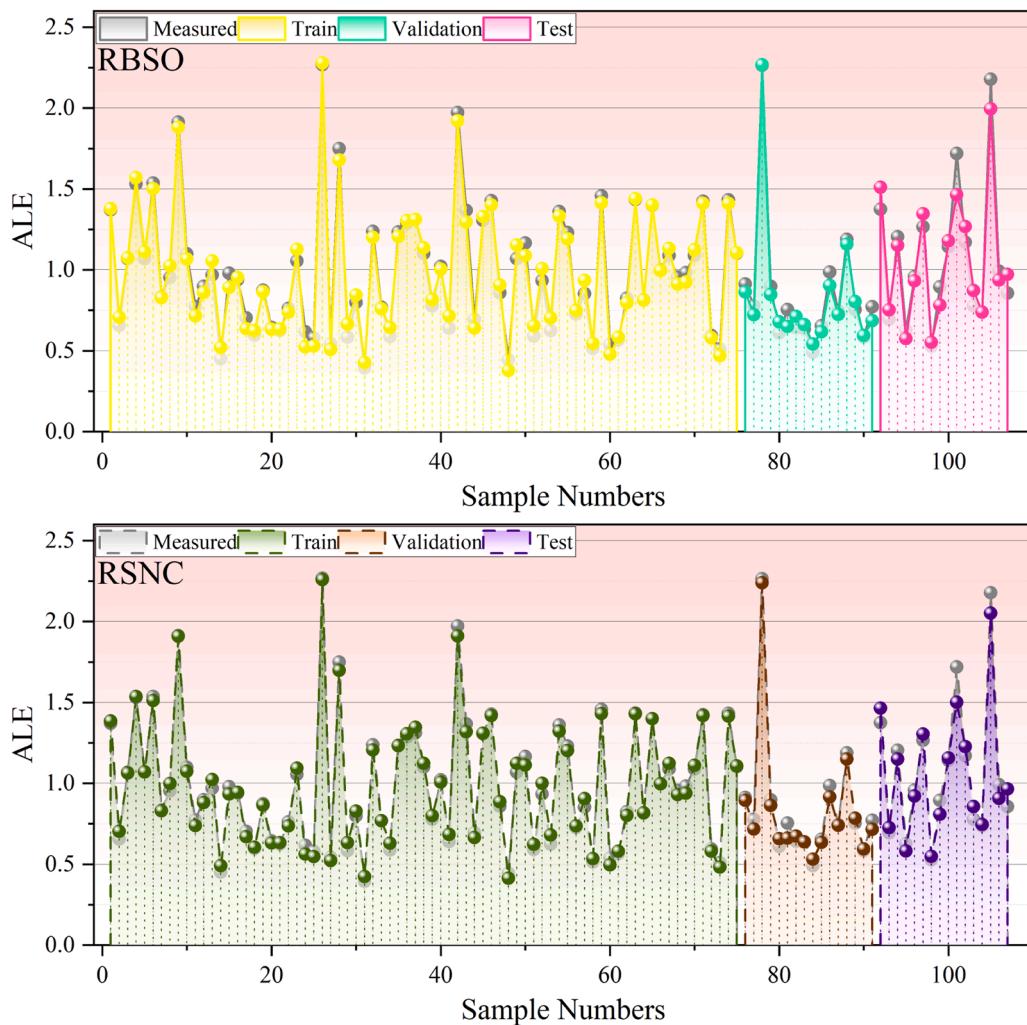


Fig. 11. (continued).

which the ML methods used in this study cannot be applied. More precisely, this comparison makes these restrictions well known, and therefore, one can get an insight into the bounds of performance of each algorithm and its real applicability. These will be beneficial insights into the discipline that will further contribute to refining the future directions of research and enhance the overall value and relevance of the work reported in this publication.

The convergence curve of Fig. 9 represents the comparison of the RMSE between RBNG, RBSO, and RBCO for the selection of the best model in the specific context. The starting RMSE for the RBNG model is 0.12, which then starts to converge quickly down to 0.08 in the initial few iterations. It further reduces to 0.06 at the 30th iteration and achieves an optimum value of RMSE of 0.03938 at around the 70th iteration. In contrast, the model of RBCO began with a higher initial RMSE value of 0.18, which was reduced to 0.08 at the 30th iteration. Its optimal performance is at the 50th iteration, with an RMSE of 0.05474. Moreover, RBSO obtained the optimal RMSE in the 125th iteration equal to 0.0612. This comparative analysis underscores the superior performance of the RBNG model in terms of convergence speed and error minimization in this section.

Table 2 presents a comparison carried out for three different phases of the model performance: Training, Validation, and Testing. The comparison is between the three frameworks of ML, which contain single, hybrid, and ensemble. It is noticed that during the training phase, the worst performance is contributed by the RBF model, as evident from its RMSE value of 0.055, which stands high in front of the RBNG model,

having an RMSE of 0.028. Hence, the RBNG model is followed as the best model in this stage, and the performance of the RBCO model with an RMSE of 0.042 is at an intermediate level.

In the validation phase, the RBCO model shows a commendable R^2 value of 0.986, outperforming the baseline RBF model, which has an R^2 value of 0.980. Nevertheless, the RBNG model outperformed them all with the highest R^2 value of 0.991, making it the most proficient model. Besides, the performance trend for the models is relatively stable during the period being tested. Because of its lowest RMSE and highest R^2 values, the RBNG model still holds the dominant position and performs very strongly with its high predictive capability. After that, the best will be the RBCO model, which has shown good correlation values and a medium error metric that ensures reliable performance. On the other hand, the RBF model continuously shows the poorest performance among them, with the highest RMSE and lowest values of R^2 during all phases. This comparison analysis highlights the RBNG model's effectiveness as the most trustworthy predictor in this experimental configuration, with the RBF model being the least successful and the RBCO model acting as a workable bridge.

In Fig. 10, the depicted variations among the developed models are discerned. Beginning with the baseline model, it is evident that its density along the central axis is inferior to that of its hybrid counterparts. This assertion is substantiated by the performance metrics of the baseline model, which yields an RMSE of 0.113 and an R^2 value of 0.939 during the testing phase. Conversely, the RBCO model exhibits greater density along the central axis than the baseline model. This observation

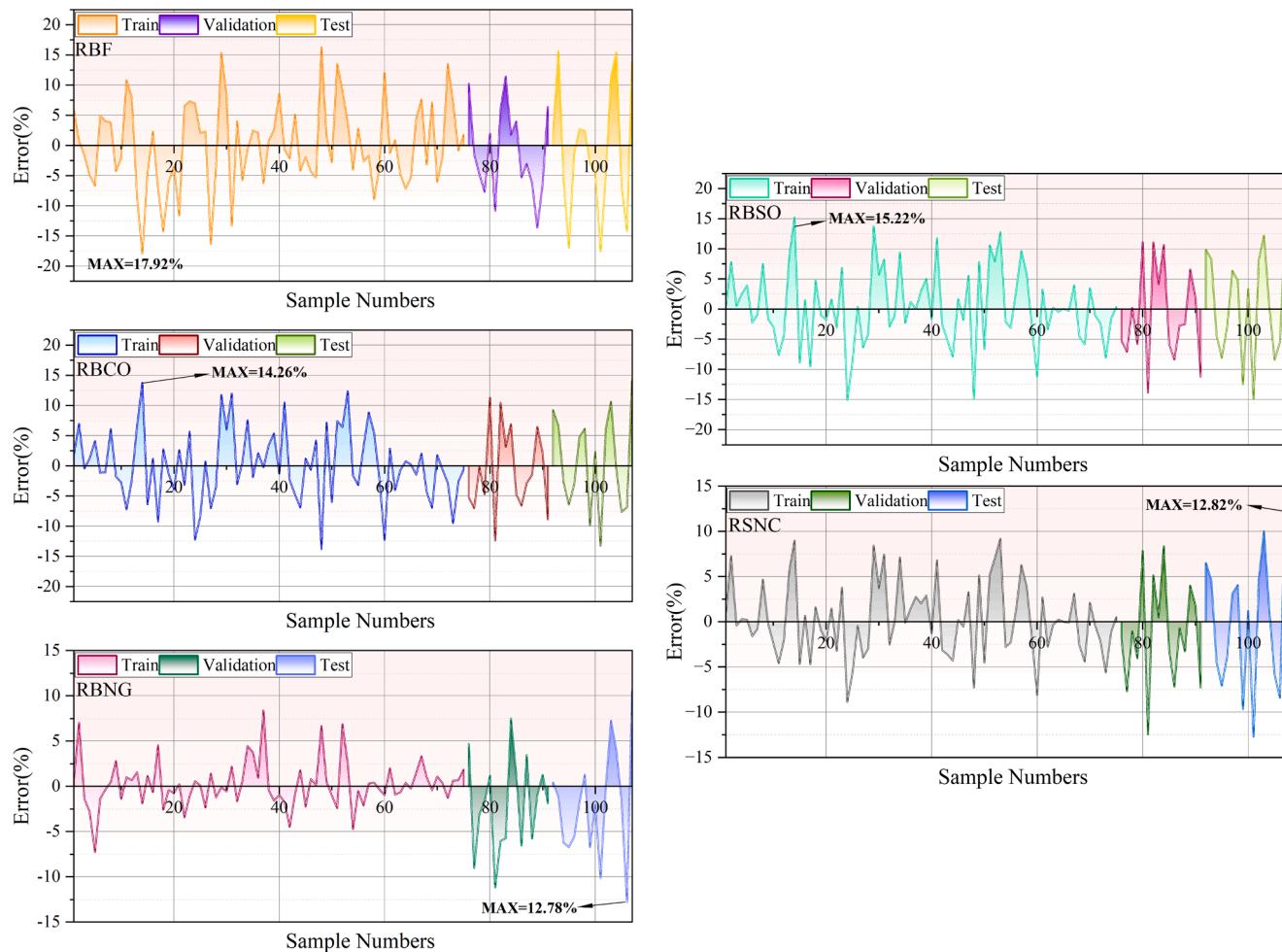


Fig. 12. The error percentage of the models in the line plot.

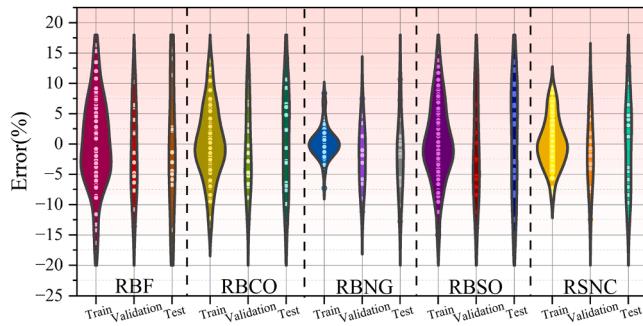


Fig. 13. Comparison of the models' error.

is supported by the testing phase metrics, where the RBCO model achieves an RMSE of 0.095 and an R^2 value of 0.951. However, the RBNG model, with an RMSE of 0.068 and an R^2 value of 0.978 in the same phase, demonstrates the highest density along the central axis, thus establishing its superiority as the best-performing model. Moreover, RSNC had the optimal performance compared to single and hybrid models except for RBNG.

In three separate stages—training, validation, and testing—Fig. 11 shows and contrasts the projected values of the models being presented with their measured values. The RBNG model's projected values during the training phase match the measured values more closely than those of the other models, according to the plot analysis. Following the RBNG

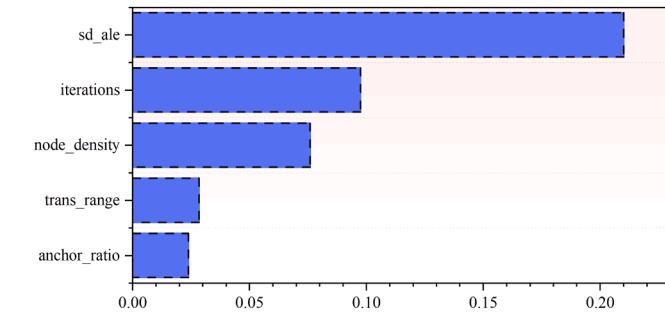


Fig. 14. A sensitivity analysis using the SHAP method.

model, the RBCO model demonstrates better alignment with the measured values in the training phase than the base model. However, in the validation phase, the RBCO model's predicted values exhibit a closer fit to the measured values than those of the RBNG model.

However, it is pointed out that the accuracy achieved by the models under consideration changes when the models are tested. As retained from the training phase, the RBNG model demonstrates improved fitness with the observed values. During the testing phase, the RBCO model surpasses the base model of the observed values but has a slightly smaller standard profile compared to the RBNG model. Such consistency in the methodology's performance through the different phases shows how effective and adaptable the RBNG methodology is.

On its part, the basic model approaches the observed values in every

Table 3Comparison of the present and published study based on R^2 and RMSE.

Author	Ref.	The best model	Evaluator	
			R^2	RMSE
Zhu and Zhu	(Zhu and Zhu, 2024)	RFGH	0.982	0.078
Gounari and Kanzilieris	(Gounari and Kanzilieris, 2024)	DTRT	0.975	0.083
Ileri	(Ileri, 2024)	LGBM-PSO	0.9043	0.349
Present study		RBNG	0.978	0.069

phase with the least extent of agreement and exhibits the lowest predictive capability of the 3 models. The comparison study shows how reliable the model is in predicting results, which is shown by the testing and training results of the RBNG model as well as the validation of the RBCO model. Taking all of this into consideration, it could be concluded that the presented RBNG model is the most reliable out of the mentioned approaches, which is especially true for applications that require high levels of predicted accuracy during the training and testing phases.

Fig. 12 presents a stacked area plot illustrating the error percentages of the utilized models. The data reveals that the RBF model exhibits its highest error percentage during the training phase, reaching 17.92%. Similarly, the RBCO model exhibits its maximum error percentage during the training phase at 14.26%. Contrarily, the RBNG model, which was highlighted as the best model in the training phase, presents the minimum error percentage of this phase when compared with all other models.

The maximum percentage error, in the case of the RBNG model, is in the test phase at 12.78%. This would depict that its performance is relatively consistent across the phases. On the other hand, there are fewer error percentages from both the validation and testing phases of the RBF model compared to the training phase, which hints at better performance after training. Similarly, the performance of the RBCO model is comparable to the base model in terms of better performance in the validation and testing phases than in the training phase.

Improvements in all RBF and RBCO models developed during the validation and testing phases do not mean that they outperform the RBNG model. The overall performance analysis may show that the RBNG model has maintained error rates lower when compared from one phase to another, confirming its stronger generalization capability and robustness compared with the other models assessed.

Fig. 13 compares the error percentages of the models employed through training, validation, and testing stages. A careful look at this figure develops different trends in the spread of error percentages within each model. The RBNG has seen the highest density of error percentages centered around zero during the training phase, which indicates the best performance and minimum deviation from actual values. Contrarily, the RBCO model has a larger spread of the percentage errors between -20 to 15 points, thus having more variability and making this model less accurate than the RBNG model. The base model supports the lowest density and widest spread at the training phase with an error percentage range from -10 to 20.

During the testing phase, all models show their weakest performance, as evidenced by the broader distribution of their error percentages. The error percentages of the RBF, RBCO, RBSO, RBNG, and RSNC models are dispersed below a range of -20 to 20. Overall, these observations underscore the RBNG model's robustness and consistency, particularly in the training phase, while highlighting the increased challenges and variability faced by all models during the testing phase.

Fig. 14 depicts the influence of each input on the prediction process and identifies which sensitivity parameter has the greatest impact on the prediction performance of the RFNG model. Findings indicate that iteration significantly affects the prediction of the RFNG model, following SD-ALE parameters. Furthermore, node density exerts a stronger influence on prediction accuracy compared to trans-range and anchor-ratio in the RFNG model.

It is noteworthy that such observations presuppose the relevance of iteration and SD-ALE parameters when implementing the RFNG model. It turns out that the key indicator is iteration, which appears to be the most significant factor influencing the results of prediction, even when SD-ALE parameters are taken into consideration. Moreover, the crucial role played by node density affected the increase in the assessment interval's prediction by 30 % and showed more impact than trans-range and anchor-ratio factors. These findings are novel and useful in enhancing the precision of the RFNG model's predictions by identifying fundamental transformed parameterities with high sensitivity and graded influence on the prediction results.

Table 3 compares the present study's RBNG model with published studies using R^2 and RMSE as evaluation metrics. (Zhu and Zhu, 2024) achieved the highest R^2 (0.982) and an RMSE of 0.078 with the RFGH model, showcasing strong performance. (Gounari and Kanzilieris, 2024) reported slightly lower accuracy ($R^2 = 0.975$, RMSE = 0.083) using the DTRT model, while (Ileri, 2024) demonstrated significantly lower performance ($R^2 = 0.9043$, RMSE = 0.349) with the LGBM-PSO model. The RBNG model in the present study achieved an R^2 of 0.978 and the lowest RMSE of 0.069, indicating excellent error minimization and competitive predictive accuracy. These results highlight the RBNG model's ability to provide a balanced trade-off between explanatory power and prediction error, making it a reliable and efficient approach for similar applications.

5. Conclusion

This study addresses the challenge of accurately predicting Allocative Localization Error (ALE) in Wireless Sensor Networks (WSNs), which arises due to the spatial distribution of nodes, resource limitations, and dynamic environmental conditions. High ALE negatively impacts network performance, resulting in inefficient routing, inaccurate data accumulation, and reduced dependability in applications like environmental monitoring and target acquisition. Reducing ALE is, therefore, essential to improving the accuracy and reliability of WSNs, particularly through the deployment of advanced machine learning (ML) models and optimization techniques. The proposed approach employed a Radial Basis Function (RBF) Neural Network Regression model enhanced by three optimization algorithms—Coot Optimization Algorithm (COA), Goshawk Optimization Algorithm (GOA), and Smell Agent Optimization (SAO)—to form hybrid models named RFCO, RFNG, and RBSO, respectively. An ensemble framework, RSNC, combining these optimizers, was also introduced. Experimental results demonstrated that the hybrid models significantly outperformed the stand-alone RBF model. Among the hybrid approaches, the RFNG model achieved the lowest Mean Squared Error (MSE) of 0.005, followed by RFCO with an MSE of 0.009, compared to the RBF model's MSE of 0.013. These results highlight the effectiveness of hybrid models in improving ALE prediction by leveraging the strengths of nature-inspired optimization techniques. The findings underscore the superiority of the RFNG model, which exhibited the best balance between prediction accuracy and computational efficiency. The improved accuracy is attributed to the ability of GOA to balance exploration and exploitation effectively, allowing it to optimize key RBF parameters, such as the spread and centers, under diverse WSN scenarios. Similarly, the RFCO and RBSO models demonstrated consistent improvements, showcasing the adaptability of hybrid ML models in addressing ALE challenges. This study also highlights the inherent complexities of ALE prediction in WSNs. Factors such as the dynamic topology of WSNs, limited computational resources, and real-time environmental changes pose significant challenges for ML models. Moreover, the need for extensive and diverse training data to ensure generalization across varied network scenarios remains a limitation. Despite these challenges, the proposed hybrid models demonstrate the potential to enhance ALE prediction accuracy and reliability, enabling better node placement, reduced energy consumption, and extended network lifespan. Future work should explore

further optimization techniques, the integration of online learning mechanisms to address network dynamics, and strategies to reduce the computational overhead of ML models for real-time applications. The outcomes of this study establish a solid foundation for advancing ML-based ALE prediction and resource management in WSNs, contributing to more efficient and reliable network operations.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Availability of data and materials

Data can be shared upon request.

CRediT authorship contribution statement

Guo Li: Writing – original draft. **Hongyu Sheng:** Writing – original draft.

Competing interests

The author declare no competing interests.

Acknowledgements

We would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

References

- Adday, GH, Subramaniam, SK, Zukarnain, ZA, & Samian, N. (2022). Fault Tolerance Structures in Wireless Sensor Networks (WSNs): Survey, Classification, and Future Directions. *Sensors*, 22, 6041.
- Ahmadi, H, & Bouallegue, R. (2017). Exploiting machine learning strategies and RSSI for localization in wireless sensor networks: A survey. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)* (pp. 1150–1154). IEEE.
- Alrajeh, NA, Marwat, U, Shams, B, & Shah, SSH. (2015). Error correcting codes in wireless sensor networks: an energy perspective. *Applied Mathematics & Information Sciences*, 9, 809.
- Balasubramanian, DL, & Govindasamy, V. (2023). Design of improved deer hunting optimization enabled multihop routing protocol for wireless sensor networks. *International Journal of Cognitive Computing in Engineering*, 4, 363–372.
- Bhatti, MA, Riaz, R, Rizvi, SS, Shokat, S, Riaz, F, & Kwon, SJ. (2020). Outlier detection in indoor localization and Internet of Things (IoT) using machine learning. *Journal of Communications and Networks*, 22, 236–243.
- Broomhead, DS, & Lowe, D. (1988). Radial basis functions, multi-variable functional interpolation and adaptive networks. In *Royal Signals and Radar Establishment Malvern (United Kingdom)*.
- Cheng, J, & Xia, L. (2016). An effective cuckoo search algorithm for node localization in wireless sensor network. *Sensors*, 16, 1390.
- Dardari, D, Conti, A, Buratti, C, & Verdone, R. (2007). Mathematical evaluation of environmental monitoring estimation error through energy-efficient wireless sensor networks. *IEEE Trans Mob Comput*, 6, 790–802.
- Friery, AC, Ramos, H, Alencar-Neto, J, & Nakamura, E (2008). Error estimation in wireless sensor networks. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 1923–1928).
- Gebremariam, GG, Panda, J, & Indu, S. (2023). Secure localization techniques in wireless sensor networks against routing attacks based on hybrid machine learning models. *Alexandria Engineering Journal*, 82, 82–100.
- Gharghan, SK, Nordin, R, & Ismail, M. (2016). A wireless sensor network with soft computing localization techniques for track cycling applications. *Sensors*, 16, 1043.
- Gopakumar, A, & Jacob, L. (2008). Localization in wireless sensor networks using particle swarm optimization. In *2008 IET international conference on wireless, mobile and multimedia networks* (pp. 227–230). IET.
- Gounari, I, & Kanzilieris, M. (2024). Wireless Sensor Networks Focusing on Predicting Average Localization Error Through Machine Learning Applications. *Journal of Artificial Intelligence and System Modelling*, 1, 104–118.
- Goyal, S, & Patterer, MS. (2014). Wireless sensor network localization based on cuckoo search algorithm. *Wirel Pers Commun*, 79, 223–234.
- Guo, S, Zhang, H, Zhong, Z, Chen, J, Cao, Q, & He, T. (2014). Detecting faulty nodes with data errors for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 10, 1–27.
- He, X, Zhou, X, Anwar, K, & Matsumoto, T. (2013). Estimation of observation error probability in wireless sensor networks. *IEEE Communications Letters*, 17, 1073–1076.
- Hu, G, Du, B, Li, H, & Wang, X. (2022). Quadratic interpolation boosted black widow spider-inspired optimization algorithm with wavelet mutation. *Math Comput Simul*, 200, 428–467.
- Ileri, K. (2024). LightGBM: Predicting Average Localization Error Through Particle Swarm Optimization in Wireless Sensor Networks. In *2024 International Conference on Wireless Communications Signal Processing and Networking (WISPNET)* (pp. 1–4). IEEE.
- Imran, M, Said, AM, & Hasbullah, H. (2010). A survey of simulators, emulators and testbeds for wireless sensor networks. In *2. 2010 International Symposium on Information Technology* (pp. 897–902). IEEE.
- Kadel, R, Paudel, K, Guruge, DB, & Halder, SJ. (2020). Opportunities and challenges for error control schemes for wireless sensor networks: A review. *Electronics (Basel)*, 9, 504.
- Kim, T, Vecchietti, LF, Choi, K, Lee, S, & Har, D (2020). Machine learning for advanced wireless sensor networks: A review. *IEEE Sens J*, 21, 12379–12397. <https://doi.org/10.1109/JSEN.2020.3035846>
- Kumar, A, & Nayyar, A. (2014). Energy efficient routing protocols for wireless sensor networks (WSNs) based on clustering. *Int J Sci Eng Res*, 5, 440–448.
- Li, B, Li, Z, Yang, P, Xu, J, & Wang, H. (2022). Modeling and optimization of the thermal-hydraulic performance of direct contact heat exchanger using quasi-opposite Jaya algorithm. *International Journal of Thermal Sciences*, 173, Article 107421.
- Luan, F, Zhang, XY, Zhang, HX, Zhang, RS, Liu, MC, Hu, ZD, et al. (2006). QSPR study of permeability coefficients through low-density polyethylene based on radial basis function neural networks and the heuristic method. *Comput Mater Sci*, 37, 454–461. <https://doi.org/10.1016/j.commatsci.2005.11.003>
- Matin, MA, & Islam, MM. (2012). Overview of wireless sensor network. *Wireless Sensor Networks-Technology and Protocols*, 1.
- Minakov, I, Passerone, R, Rizzardi, A, & Sicari, S. (2016). A comparative study of recent wireless sensor network simulators. *ACM Transactions on Sensor Networks (TOSN)*, 12, 1–39.
- Morelande, MR, Moran, B, & Brazil, M. (2008). Bayesian node localisation in wireless sensor networks. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 2545–2548). IEEE.
- Motwakel, A, Hashim, AHA, Alamro, H, Alqahtani, H, Alotaibi, FA, & Sayed, A. (2023). Chaotic Mapping Lion Optimization Algorithm-Based Node Localization Approach for Wireless Sensor Networks. *Sensors*, 23, 8699.
- Mukherjee, A, Barma, PS, Dutta, J, Panigrahi, G, Kar, S, & Maiti, M. (2021). A multi-objective antlion optimizer for the ring tree problem with secondary sub-depots. *Operational Research*, 1–39.
- Muralidharan, K, Ramesh, A, Rithvik, G, Prem, S, Reghunaath, AA, & Gopinath, MP (2021). 1D Convolution approach to human activity recognition using sensor data and comparison with machine learning algorithms. *International Journal of Cognitive Computing in Engineering*, 2, 130–143.
- Muruganantham, P, & Balakrishnan, SM. (2021). A survey on deep learning models for wireless capsule endoscopy image analysis. *International Journal of Cognitive Computing in Engineering*, 2, 83–92.
- Naruei, I, & Keynia, F. (2021). A new optimization method based on COOT bird natural life model. *Expert Syst Appl*, 183, Article 115352.
- Nayak, P, Swetha, GK, Gupta, S, & Madhavi, K. (2021). Routing in wireless sensor networks using machine learning techniques: Challenges and opportunities. *Measurement*, 178, Article 108974. <https://doi.org/10.1016/j.measurement.2021.108974>
- Nayyar, A, & Sharma, S. (2014). A survey on coverage and connectivity issues surrounding wireless sensor network. *IJRCCCT*, 3, 111–118.
- Nayyar, A, & Singh, R. (2014). A comprehensive review of ant colony optimization (ACO) based energy-efficient routing protocols for wireless sensor networks. *International Journal of Wireless Networks and Broadband Technologies (IJWNBT)*, 3, 33–55.
- Nayyar, A, & Singh, R. (2015). A comprehensive review of simulation tools for wireless sensor networks (WSNs). *Journal of Wireless Networking and Communications*, 5, 19–47.
- Nidhya, R, Pavithra, D, Vinothini, C, & Maragatham, T. (2023). Hybrid Tasmanian Devil and Improved Simulated Annealing-Based Clustering Algorithm for Improving Network Longevity in Wireless Sensor Networks (WSNs). *Wirel Pers Commun*, 132, 1553–1576.
- Pandey, D, & Kushwaha, V. (2019). Experimental tools and techniques for wireless sensor networks. *Int J Recent Technol Eng(IJRTE)*, 8, 1674–1684.
- Salawudeen, AT, Mu'azu, MB, Sha'aban, YA, & Adedokun, EA (2018). On the development of a novel smell agent optimization (SAO) for optimization problems. In *2nd International Conference on Information and Communication Technology and its Applications (ICTA 2018)*.
- Salawudeen, AT, Mu'azu, MB, Yusuf, A, & Adedokun, AE (2021). A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application. *Knowl Based Syst*, 232, Article 107486.
- Salawudeen, AT, Mu'azu, MB, Yusuf, A, & Adedokun, EA (2018). From smell phenomenon to smell agent optimization (SAO): a feasibility study. *Proceedings of ICGET*.
- Sankar, S, Somula, R, Parvathala, B, Kolli, S, & Pulipati, S. (2022). SOA-EACR: Seagull optimization algorithm based energy aware cluster routing protocol for wireless sensor networks in the livestock industry. *Sustainable Computing: Informatics and Systems*, 33, Article 100645.

- Sedaghat, B, Javadzade Khiavi, A, Naeim, B, Khajavi, E, & Taghavi Khanhah, AR (2023). Evaluation of Object-Based and Pixel-Based Technique for Extracting Snow Cover Surface Using Landsat 8 Satellite Images (Case Study Damavand Mountain Range). *Advances in Engineering and Intelligence Systems*, 2.
- Sharma, H, Haque, A, & Blaabjerg, F. (2021). Machine learning in wireless sensor networks for smart cities: a survey. *Electronics (Basel)*, 10, 1012.
- Silmi S, Doukha Z, Kemcha R, Moussaoui S. Wireless sensor networks simulators and testbeds. ArXiv Preprint ArXiv:200903640 2020.
- Singh, A, Kotiyal, V, Sharma, S, Nagar, J, & Lee, C-C. (2020). A machine learning approach to predict the average localization error with applications to wireless sensor networks. *IEEE Access*, 8, 208253–208263.
- Singh, S, Anand, V, & Bera, PK. (2023). A Delay-Tolerant low-duty cycle scheme in wireless sensor networks for IoT applications. *International Journal of Cognitive Computing in Engineering*, 4, 194–204.
- Sobehi, A, Hou, JC, Kung, L-C, Li, N, Zhang, H, Chen, W-P, et al. (2006). J-Sim: a simulation and emulation environment for wireless sensor networks. *IEEE Wirel Commun*, 13, 104–119.
- Sundani, H, Li, H, Devabhaktuni, V, Alam, M, & Bhattacharya, P. (2011). Wireless sensor network simulators a survey and comparisons. *International Journal of Computer Networks*, 2, 249–265.
- Vuran, MC, & Akyildiz, IF. (2009). Error control in wireless sensor networks: a cross layer analysis. *IEEE/ACM Transactions on Networking*, 17, 1186–1199.
- Wang, L, Er, MJ, & Zhang, S. (2020). A kernel extreme learning machines algorithm for node localization in wireless sensor networks. *IEEE Communications Letters*, 24, 1433–1436.
- Wang, L, Er, MJ, & Zhang, S. (2020). A kernel extreme learning machines algorithm for node localization in wireless sensor networks. *IEEE Communications Letters*, 24, 1433–1436.
- Zhu, Z, & Zhu, M. (2024). Utilizing Machine Learning Approach to Forecast Average Location Determination Errors in Wireless Sensor Networks. *Journal of Artificial Intelligence and System Modelling*, 1, 83–98.