Minimizing the Energy Consumption and Exploiting the NLT by E2HCA Model in WSN

Siddamsetty Saritha * and Edara Srinivasa Reddy

Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, AP, India
Email: sarithanune@gmail.com (S.S.); esreddy67@gmail.com (E.S.R.)
*Corresponding author

***Abstract*—In recent years, Wireless Sensor Network (WSN) has arisen as a practical option for many sectors needing smart technology. Despite its impressive credentials, Wireless Sensor Networks' excessive need for power remains a significant limitation. There is a pressing need to create a trustworthy Wireless Sensor Network with efficient energy and network lifetime due to the proliferation of tiny sensors with incomplete resources. One of the actual ways to deal with these matters is to divide the nodes into clusters. So, it is crucial to use existing energy efficiently to prevent energy waste. In this study, we suggest an Energy-Efficient Hybrid Clustering Algorithm to reduce the high-energy ingesting and increase the network lifetime of Wireless Sensor Networks. The position and speed of a gas molecule, among other Kinetic Gas Molecule Optimization particle characteristics, are initially determined by calculating their kinetic energy. Hybridization of Kinetic Gas Molecule Optimization arises from Kinetic Gas Molecule Optimization's faster convergence in space. Pelican Optimization Algorithm is used to fix the problems with Kinetic Gas Molecule Optimization by changing its inertia weight. Throughput the number of live/dead nodes are all validated against other prominent meta-heuristic methodologies in a MATLAB environment simulation of the proposed Kinetic Gas Molecule Optimization—Pelican Optimization Algorithm.**

***Keywords*—Wireless Sensor Network (WSN), kinetic gas molecules optimization, pelican optimization algorithm, network lifetime, energy-efficient hybrid clustering algorithm**

I. Introduction

Based on the net life cycle of Wireless Sensor Networks (WSNs), the excellence of network architecture is reliably quantifiable. This network cycle is typically affected by the period at which a sizeable fraction of the network's sensor nodes dies due to the problem of energy exhaustion [1, 2]. Because of its centrality in maintaining network viability and reducing communication interference [3]. Clustering mechanisms, which provide inspiration for the characteristics of hierarchical topology control algorithms, are deemed important and often employed in this setting. Topology control algorithms rely heavily on the nodes chosen to lead clusters [4, 5].

Moreover, the Cluster Head (CH) issue is found to be NP-complete. The energy, the sensor node's communication unit uses, is more than that of its detecting and processing units [5, 6]. As a result, sensor node clustering is a useful strategy to enhance the WSN's energy efficiency. During the clustering technique, the outlying nodes merge with a nearby Cluster Head (CH) to form a new cluster. The development of clusters like these helps a network use less power overall [7]. Clustering in WSN not only allows for the aggregation of data, scalability, and bandwidth conservation, but it also increases the Localization Technique (LT) of the system by reducing the energy required for the connection between sensor nodes [8].

The clustering procedure divides the sensor nodes into smaller groups, or clusters, each of which is headed by a CH. The Cluster Head (CH) combines data from all of the cluster's sensor nodes and sends it to the Base Station (BS) [9, 10]. The acquired data from the CH is sent to the BS either immediately or via intermediary CHs and/or sensor nodes; this is known as multi-hop communication. The BS sends data to the Cloud, where it can be processed and viewed in greater detail. It is well known that the CH selection is an NP-hard issue [11]. Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Simulated Annealing (SA) are just a few of the searching-based heuristics that have been created since [12, 13].

Both exploration (diversification) and exploitation (intensification) are crucial to the optimization process in any potentially fruitful subsets of the search space, wherein the global optimum may be located. When it comes to exploitation, however, the algorithm actively seeks to comb the area around each solution it unearths during exploration. That's why, early in the optimization process, it's crucial to focus more on exploration than exploitation; later, when it's time to find the best possible answers, exploitation becomes more critical to enhance the likelihood of finding solutions earlier. The procedure's efficiency benefits greatly from a healthy equilibrium between exploring and exploiting.

In KGMO [15, 16], the gas molecules move throughout the search space in order to locate its minimum temperature. Random vectors are used to set each

Manuscript received June 8, 2023; revised July 19, 2023; accepted September 5, 2023; published February 15, 2024.

particle's initial velocity and position inside their respective domains. Here, the inertia mass (think of as weight) that represents the resistance exerted by a gas molecule to its motion. The KGMO's meeting behavior is strongminded by its inertia weight. The optimization converges more slowly when the inertia weight is large, and it gets stuck in a local rut when it's little. Hence, the inertia weight needs to be chosen to optimize the search time/resources ratio. It is necessary to address this shortcoming by boosting the convergence speed of KGMO. The suggested section provides a quick explanation of how a new tweak to the inertia weight can make this a reality.

Our new ideas and work are detailed here. To start, here is how we frame the issue at hand: In order to progress the overall LT of a network and decrease the amount of energy used on transmissions, researchers have turned to an evolutionary algorithm, which takes into account a wide range of characteristics in the fitness function to perform the necessary clustering of sensor nodes. In this research, we introduce the KGMO-POA based Cluster Head Selection (KGMO-POA-CHS) meta-heuristic combined with the Chaotic based POA method in an effort to improve the transmission energy ingesting in WSN.

To better understand the paper's structure, we will look at: The secondary sources that informed the research are presented in Section II. The proposed approach to increase the network's lifetime is presented together with the system model in Section III. Section IV presents the validation analysis with current models in terms of various metrics, and Section V wraps up the work.

II. Literature Review

Least-Square Policy Iteration is proposed by Obi *et al.* [17] as an effective model-free Reinforcement Learning-based technique for optimization in WSNs. The subsequent protocol architecture is a centralized routing procedure for Lifetime Energy Optimization (LEOP) with a GA and Least-Square Policy Iteration (LSPI) for generation and energy optimization (CRPLEOGALSPI). The CRPLEOGALSPI outperforms a current centralized routing protocol for lifespan optimization using genetic algorithms and q-learning, according to simulation data (CRPLOGARL). For this reason, the CRPLEOGALSPI is insensitive to the learning rate, as it selects a routing path in a given state after taking into account all alternative routing paths. Additionally, whereas the CRPLOGARL determines the best course of action based on the Q-values, the adjusts the Q-values according on the most recent knowledge about the network's subtleties using biased purposes.

Energy-efficient routing protocol proposed by Abdulzahra *et al.* [18] is based on Bacterial Foraging Optimization Routing Protocol (BFORP). BFORP is an effort to learn more about the WSN lifespan investigation challenge. By reusing data that frequently travels from the source node to the sink, it can reduce the routing of superfluous messages that can cause significant energy waste. In the suggested method, the node with the lowest traffic load, the maximum residual energy, and the shortest path to the sink can be selected as the preferred node in the sending routes. The simulation findings show that the projected protocol is effective in minimizing energy employment and cutting down on end-to-end delay when associated to the established routing methods already in use.

With the benefits of using both global variety and an accelerated convergence rate that the Improved Bat optimization algorithm and Enhanced Artificial Bee Colony-based Cluster Routing strategy presented for optimal Cluster Head (CH) selection by Janakiraman [19] stands out. For optimal CH selection, it is advised to strike a balance among the two phases of exploration and exploitation. The study aimed to reduce energy consumption by developing a superior CH selection method with the use of an enhanced Bat Optimization Algorithm (IBOA). To improve packet delivery from CH to sink node, the paper also focuses on the design of a sink node mobility strategy based on an EABC. By taking into account node centrality, node degree, distance between CH and BS, this CH selection and sink node mobility strategies contributes to extending the network lifetime via the fitness function. Using MATLAB 2018, modelling experiments demonstrated that placing the BS at the centre of the network increases the number of functional nodes in the network by 39.21%. (100, 100). By shifting where BSs are placed, we may increase the network's lifetime by 23.84 percent, or the number of rounds. When compared to the investigation's baseline CH schemes, the packets conventional at the BS are found to be improved by an average of 26.32%.

To extend the useful life of WSN-based requests and create reliable clusters, Vellaichamy *et al.* [20] present a bio-inspired routing algorithmic strategy to enhance network lifetime. Information retention can be improved by the establishment of groups, and one effective approach for doing so is clustering. Selecting the most qualified cluster leader is achieved through the use of multi-criteria clustering (CH). Following careful identification of the CH, we combine the optimization techniques to determine the best path for data transmission from the CH to the sink, therefore increasing network stability. Many metrics and packet delivery rates are used to compare the proposed method to others. Saving up to 18.6% on energy costs and extending the life of your network by as much as 6% compared to other routing protocols is possible with ours.

Soft C-means have been presented by Viswanathan *et al.* [21]. The Swarming Concurrent Multi-Objective Metaheuristic Dragonfly Optimization (SCMMDO) Approach was developed. The primary objective of the SCMMDO Method is to locate the optimal cluster head for efficient data broadcast in WSN. SCMMDO Technique was used to achieve clustering and optimization in WSN. For starters, the sensor nodes are spread out at random. The sensor nodes are clustered by the soft C-Means technique using three criteria. The availability of bandwidth, the strength of the received signal, and the amount of residual energy. The multi-objective optimization is then applied to select the cluster

head. The source node uses the chosen cluster head to send the data packet to the target node. Energy use, clustering precision, processing time, throughput, and latency are some of the measures used in simulation. The results show that the SCMMDO Method reduces processing time and energy usage while simultaneously improving clustering precision. The proposed approach has a 96% success rate in its clustering endeavors.

With the goal of decreasing node energy consumption and raising data throughput, Chaurasia *et al.* [22] suggested a Meta-heuristic Optimal Routing approach for WSNs (MOCRAW). By employing the Dragonfly Algorithm (DA) and basing its decisions on Local Search Optimization (LSO) and Global Search Optimization (GSO), MOCRAW is able to solve the issues of loop-free routing and the elimination of isolated nodes or hot spots (GSO). This protocol makes use of the most effective implementations of both the Cluster Head Selection Algorithm (CHSA) and the Routing Search Algorithm (RSA). The Energy Level Matrix is used in CHSA. Extreme Learning Machine (ELM) is affected by the distance from the Cluster Head (CH) to the Base Station (BS), the amount of energy left over after each cluster forms, and the number of clusters that form. Using RSA, the inter-cluster levy distribution establishes the best route from source to destination. MOCRAW's efficiency is measured against other clustering and routing protocols using metrics like latency, packet delivery rate, and average energy utilization. The suggested method outperforms its contemporaries in terms of energy efficiency, according to the simulation results.

GWO (EECHIGWO) approach is provided by Reddy *et al*. [23], who aim to remedy the uneven exploitation and exploration, lack of population diversity, and early of the original GWO algorithm. The primary goal of this study is to augment energy efficiency, average throughput, network stability, and durability by optimizing the selection of cluster heads in WSNs with the EECHIGWO technique. The metrics that are used to identify the cluster's frontrunner are the average intra-cluster distance, the sink distance, the residual energy, the cluster head balancing factor, and the cluster head balancing factor. The proposed EECHIGWO-based clustering protocol has been evaluated on a variety of different parameters, including average throughput, number of rounds of operation, energy ingesting, and number of failed nodes. The results of the simulations verify the effectiveness of minimizing energy consumption, avoiding early convergence, and extending the lifetime of WSN networks. The proposed approach improves network stability by 169.29%, 19.03%, 253.73%, 307.89%, and 333.51%, respectively.

The current schemes in the literature suffer from the following major drawbacks: Due to its significance in maintaining a consistent level of network performance and minimizing energy costs, the cluster head selection problem has been identified as a critical issue. Because the selection of CH frequently results in energy imbalance and reduced network longevity, it is also an issue that requires close attention. The bulk of current methods for electing cluster heads cannot maintain a healthy equilibrium between exploitative and exploratory activities. At this juncture, it is thought that combining a local search with a global search algorithm would help keep the ratio of exploitation to exploration stable.

- Hybridizing existing bio-inspired algorithms to find a prospective solution does not maximise the capability of preserving the trade-off between misuse and exploration.
- Current bio-inspired algorithms do not provide enough energy balance to significantly increase or sustain network lifetime.
- The suggested KGMO-POA cluster head selection method was developed in response to the aforementioned constraints.

Despite improvements in energy-efficient protocols for Wireless Sensor Networks (WSNs), there is still a significant research gap in the optimization of energy use while concurrently exploiting Network Lifetime Time (NLT). Existing methods frequently isolate either energy reduction or NLT extension, failing to provide a comprehensive method that successfully unifies both goals. This gap is intended to be filled by the research reported in this work, which suggests the Energy-Efficient Hybrid Clustering Algorithm (E2HCA) paradigm.

III. Proposed System

A. System Model

WSN's system model includes the following components: energy model, mobility model, Long-Lasting Terminal (LLT) model, and free space model. They are used to describe the energy loss that occurs when sending data between nodes. Suppose there are n nodes in the WSN and only one of them is the sink node or the Base Station (BS). Within the range of the wireless links connecting the nodes, only direct communication is possible. As a result, the network topology is composed of evenly-spaced nodes. Clusters are collections of nodes, and each node has a unique identifier. Almost ideally, the sink node should be located where all of the networked sensor nodes can send and receive data packets to it. CH is used for all data transmissions between cluster nodes and the BS. The total sum of accessible network nodes is denoted by the notation,

$$x = \{L_1, L_2, .., L_a, \dots, L_n\} \quad (1)$$

Here, L_n signifies a total sum of sensor nodes.

B. Energy Model

Each network node starts with an unrenewable amount of energy, denoted by the value G_0. Packets lose power when they travel from a normal node to a CH in accordance with multipath fading and the free space mechanism, which depends on the physical separation of the two nodes. In contrast to the receiver, which simply has the radio circuitry for dissipating the energy, the transmitter also has a power amplifier. Node energy dissipation is denoted by when a node transmits q bytes of data:

$$G_{disi}(L_a) = G_{elec} \times q + G_{amp} \times q \times \|L_a - V_b\|^4; \quad if \|L_a - V_b\|^4 \geq c_0 \quad (2)$$

$$G_{disi}(L_a) = G_{elec} \times q + G_{fs} \times q \times \|L_a - V_b\|^2; \quad if \ \|L_a - V_b\|^2 < c_0 \quad (3)$$

where G_{elec} signifies electronic power which is subject to change due to variables like modulation, dispersion, filtering, amplification, and digital coding.

$$G_{elec} = G_{trans} + G_{agg} \quad (4)$$

where G_{trans} indicates transmitter energy, G_{agg} stipulates energy of data aggregation, Gamp indicates the energy of power amplifier, and $\|L_a - V_b\|$ specifies how far apart the given node and CH are from one another. On the other hand, the receiver's energy loss when receiving *q* bytes of data via CH is shown as,

$$G_{disi}(V_b) = G_{elec} \times q \quad (5)$$

Each node's energy value is recalculated when a data transfer of q bytes has been made.

$$G_{d+1}(L_a) = G_d(L_a) - G_{disi}(L_a) \quad (6)$$

$$G_{d+1}(V_b) = G_d(V_b) - G_{disi}(V_b) \quad (7)$$

Repeating the steps in the previous paragraph until all of the nodes have died is the end goal of this data transmission mechanism. Node death occurs when its energy level drops below zero.

C. Mobility Model

For the purpose of defining and specifying the acceleration, location, and velocity changes of sensor nodes across time, the mobility model is employed. If you want to know how well a network will work, look at the mobility pattern. Let's say *a* and *k* started out at coordinates (u_1, v_1) and (u_2, v_2), correspondingly. The vertices *a* and *k*, on the other hand, move with the same variable velocity, but in opposite directions as indicated by the angles _1 and 2. In terms of the Euclidean distance, the pair of nodes *a* and *k* are separated by,

$$D_{(ak,0)} = \sqrt{|u_1 - u_2|^2 + |v_1 - v_2|^2} \quad (8)$$

Here, *D* denotes the Euclidean distance among the nodes.

D. LLT Model

Due to the ever-changing nature of a network's underlying architecture, it's essential that route reliability be determined on the fly. Let's say a and k are two sensor nodes within range of the transmitter and receiver. Over the course of a route request packet's journey, the LLT is calculated at each intermediate hop. However, each node is responsible for determining how long the connection between the current hop and the prior hop will last. Take (M_a, N_a) to be the coordinates of node *a*; (M_k, N_k) to be the coordinates of node *k*. Both node *a* and node *k* have a mobility speed, denoted by the symbols S_a and S_k. Nevertheless, _*a* and _*k* is provided to indicate the distance travelled by sensor node *a* and node *k*, respectively. In order to determine the LLT, we use the formula:

$$LLT = \frac{-(\omega\lambda + \sigma\rho) + \sqrt{(\omega^2 + \sigma^2)\tau^2 - (\omega\rho - \lambda\sigma)^2}}{(\omega^2 + \sigma^2)} \quad (9)$$

where, $\omega = S_a cos\theta_a - S_k cos\theta_k$, $\lambda = M_a - M_k$, $\sigma = S_a sin\theta_a - S_k sin\theta_k$ and $\rho = N_a - N_k$.

E. Proposed Model Using KGMO-POA

Normalization and other forms of data preprocessing are taken into account. After being normalised, the data is given to KGMO, which then makes an estimate of the users' starting velocities and Kinetic Energy (KE). Users' beginning velocities are used to generate a random particle, whose fitness function is then used to evaluate the method in question. Following the determination of the fitness function, the best fitness is chosen to alter the KE and velocity, producing a fresh random particle. This procedure is carried out repeatedly until the highest possible fitness is attained. For this purpose, the KGMO algorithm is used. After the optimal fitness has been determined, the resulting information is sent to the network's data collection unit. Throughput, BER, and latency measures are used to evaluate network performance in relation to user location and mobility management.

This paper presents the POA method for adjusting the inertia weight in an effort to address the fast convergence problems of KGMO. First, the equatorial rationale for KGMO is presented as follows.

1) KGMO—The projected algorithm

The gas are the agents in the proposed KGMO algorithm, and kinetic is the performance metric. The molecules of the gas diffuse across the ampule until they all converge in the region with the lowest energy. To attract one another, gas molecules use very weak electrical intermolecular Van Der Waal forces. It is the presence of both positive and negative charges within the molecules that gives rise to the electrical pressure. Each gas molecule (agent) in the KGMO is described by four parameters: location, kinetic energy, speed, and mass. Each gas molecule's velocity and location are established by its kinetic energy. In order to find the lowest possible temperature, the gas molecules in the algorithm go throughout the whole search space.

In the following, we will think about a system with N agents. The ith agent's character is specified by

$$X_i = \left(X_i^1, .., X_i^d, ... X_i^n\right), \ for \ (i = 1,2, ..., N) \quad (10)$$

where X_i^d Represents the site of the ith agent in the d[th] dimension.

The velocity of the i[th] agent is obtainable by

$$V_i = (v_i^1, ... v_i^d, ... v_i^n), for \ (i = 1,2, ..., N) \quad (11)$$

where v_i^d signifies the velocity of the ith agent in the d[th] dimension.

The Boltzmann distribution governs the motion of the gas molecules in the cylinder, thus their speed is proportional to the exponential of their kinetic energy. What is meant by this kinetic energy is then

$$k_i^d(t) = \frac{3}{2} N b T_i^d(t), K_i = \left(k_i^1, .. k_i^d, ..., k_i^n\right), for \ (i = 1,2.., N) \quad (12)$$

Iteratively updating the molecule's velocity by

$$v_i^d(t+1) = T_i^d(t) w v_i^d(t) + C_1 rand_i(t)(gbest^d - X_i^t(t) + C_2 rand_i(t)(pbest_i^d(t) - X_i^d(t))) \quad (13)$$

where T_i^d decreases exponentially with time for the approaching molecules, as determined by

$$T_i^d(t) = 0.95 \times T_i^d(t-1) \quad (14)$$

In this case, we may write $pbest_i=(pbest_i^1, pbest_i^2, ..., pbest_i^n)$ and gbest = ($gbest^1$, $gbest^2$, ..., $gbest^n$) is the best prior position for all of the gas molecules, and $pbest_i^n$ is the best previous position for the i^{th} gas molecule. Each particle's initial velocity and location are both determined by random vectors inside their respective intervals. The velocity range of the gas molecules is defined here as $[-v_{min}; v_{max}]$. If $|v_i|$ is greater than $|v_{max}|$, then $|v_i| = v_{max}$, where w is the gas molecule's inertia weight. In addition, the search method benefits from the randomness provided, exist.

As only one gas is anticipated to be present in the container at any one time, the mass m of each gas molecule is chosen at random within some range and remains fixed during the algorithm's execution. Variable gases are modelled by the random integer in alternating iterations of the procedure. The location of the molecule is derived from the physics equations of motion.

$$X_{t+1}^i = \frac{1}{2} a_i^d(t+1)t^2 + v_i^d(t+1)t + X_i^d(t) \quad (15)$$

where a_i^d signifies the acceleration of the i^{th} agent in the d^{th} dimension.

From the acceleration equation, we find

$$a_i^d = \frac{(dv_i^d)}{dt} \quad (16)$$

On the additional hand, from Eq. (15) of the gas particle laws, we have

$$dk_d^i = \frac{1}{2} m (dv_i^d)^2 \Rightarrow dv_i^d = \sqrt{\frac{2(dk_i^d)}{m}} \quad (17)$$

Consequently, from Eqs. (16) and (17), the acceleration is clear as

$$a_d^i = \frac{\sqrt{\frac{2(dk_i^d)}{m}}}{dt} \quad (18)$$

In the time intermission Dt, Eq. (18) can be re-written as

$$a_d^i = \frac{\sqrt{\frac{2(\Delta k_i^d)}{m}}}{\Delta t} \quad (19)$$

Thus, in a unit time interval, the hastening would be

$$a_d^i = \sqrt{\frac{2(dk_i^d)}{m}} \quad (20)$$

Then, from Eqs. (15)–(20), the position of the molecule is intended by

$$X_{t+1}^i = \frac{1}{2} a_i^d(t+1)\Delta t^2 + v_i^d(t+1)\Delta t + X_i^d(t) \Longrightarrow$$

$$X_{t+1}^i = \frac{1}{2}\sqrt{\frac{2(\Delta k_i^d)}{m}}(t+1)\Delta t^2 + v_i^d(t+1)\Delta t + X_i^d(t) \quad (21)$$

Last but not least, because the molecule mass (m) is different each time the algorithm is run yet the same for all the molecules in terms of performance, the location is updated for the unit time interval by.

$$X_{t+1}^i = \sqrt{\frac{2(\Delta k_i^d)}{m}(t+1)} + v_i^d(t+1) + X_i^d(t) \quad (22)$$

The smallest fitness function is found by using

$$pbest_i = f(X_i), if\ f(X_i) < f(pbest_i)$$
$$gbest = f(X_i), if\ f(X_i) < f(gbest) \quad (23)$$

In order to obtain the input for the Eq. (14) for inertia weight modification, this research work uses the best fitness function of POA algorithm, where the mathematical equations of POA is explained as follows:

2) Pelican Optimization Algorithm (POA)

In 2022, Dehghani and Trojovsk create an innovative algorithm based on principles found in nature [24]. The primary interest of POA is in the interactions and strategies of pelicans during hunts. Big, bulky birds called pelicans have exceptionally long bills. The pelican's large neck pouch aids in capturing and swallowing its meal. Pelicans flock together frequently. The populace consists of various sized pelicans. Using Eq. (24), we generate a random number to be used as the first seed for the population:

$$x_{i,j} = l_j + rand.(u_j - l_j), \quad i = 1,2,\dots,N; j = 1,2,\dots,m \quad (24)$$

When *N* and *m* stand for the number of population and problem variables, rand is a random number in [0,1], and l_j and u_j stand for the lower and upper limits of problem variables, $x_{i,j}$ represents the value of the j^{th} variable by the i^{th} solution.

In Pelican Optimization Algorithm (POA), we use equation to generate the population matrix that will be used to recognize each associate of the Eq. (25).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \quad (25)$$

Using the following formula, we can calculate the objective Eq. (26).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \quad (26)$$

F is the vector of the function for the ith possible solution. The hunting method for pelicans consists of two stages: scouting and ambush. In the former, the hunter ventures towards the quarry, while in the latter, the hunted soars above the water's surface on the wings of the

predator. First, the pelican gets close to its target after discovering its location. The exploratory capability of POA is enhanced by the random generation of the prey's location. Eq. (27) is a mathematical representation of the initial stage:

$$x^{P_1}{}_{i,j} = \begin{cases} x_{i,j} + rand.\left(p_j - I.x_{i,j}\right), F_p < F_i; \\ x_{i,j} + rand.\left(x_{i,j} - p_j\right), \quad else \end{cases} \quad (27)$$

where I is a random sum and F_p is the value of the objective function for the prey, and $x^{P1}{}_{i,j}$ is the ith pelican's new status in the j^{th} dimension after the first phase. In POA, a pelican's new position is considered valid if and only if function. This is known as efficient updating, because it prevents the algorithm from venturing into suboptimal zones. Mathematically,

$$X_i = \begin{cases} x^{P_1}{}_i, \ F^{P_1}{}_i < F_i; \\ X_i, \quad else, \end{cases} \quad (28)$$

where $x^{P1}{}_i$ is the updated $F^{P1}{}_i$ is the function value for the ith pelican from the first stage. In the second stage, the pelican brings the fish to the surface of the water, where it expands its wings to trap the fish in its neck bag. This allows the pelicans to successfully catch more fish. When the algorithm improves its solutions in the hunting zone, this stage increases POA's potential for exploitation. See below for a logical breakdown of the hunting procedure:

$$x^{P_2}{}_{i,j} = x_{i,j} + R \times \left(1 - \frac{t}{T}\right) \times (2.rand - 1) \times x_{i,j} \quad (29)$$

where $x^{P2}{}_{i,j}$ is the updated status after the second phase, R = 0.2 is a constant, R × (1 – t/T) is the neighbourhood radius of $x_{i,j}$, and t and T denote an iteration counter and a maximum number of iterations. At this point, effective updating new pelican position is formulated by Eq. (30):

$$X_i = \begin{cases} x^{P_2}{}_i, F^{P_2}{}_i < F_i; \\ X_i, \quad else, \end{cases} \quad (30)$$

where $x^{P2}{}_i$ denotes the i-th pelican's new status and $F^{P2}{}_i$ is the *i*-th pelican's objective-function value. Once the full execution is finished, the processes based on Eqs. (27)–(30) are repeated until all members of the population have been updated.

3) Chaotic maps

Non-linear complicated problems with unpredictable outcomes are the focus of chaos theory, a branch of mathematics. Effectively, FA (Fire On the other hand), recent research suggests that chaos has a high amount of mixing capability, thus swapping out a few parameters with a chaotic map can produce solutions that are more adaptable and one-of-a-kind [25]. Major applications of chaos theory can be found in many fields, including biology, engineering, physics, economics, and philosophy. There are a few main characteristics shared by all chaotic maps, and these are: sensitivity to beginning conditions, randomness, and ergodicity. The optimization challenges can be solved using a variety of chaotic maps. Contrarily, this investigation makes use of not one but six distinct chaotic maps. Shown below are the mathematical definitions of these chaotic maps [26, 27]:

TABLE I. CHAOTIC MAPS

Map	Function
Chebyshev-map	$X_{n+1} = cos\left(ncos^{-1}(X_n)\right)$
Circle-Map	$y_{n+1} = y_n + b - \left(a/2\pi\right)\sin(2\pi y) mod(1)$ $a = 0.5, b = 0.2$ *and* $y_n \in (0,1)$
Gauss/Mouse-map	$X_{n+1} = \begin{cases} 0 & X_n = 0 \\ \frac{1}{X_n mod(1)} & otherwise \end{cases}$ $\frac{1}{X_n mod(1)} = \frac{1}{X_n} - \left[\frac{1}{X_n}\right]$
Iterative-Map	$x_{n+1} = sin\left(\frac{a\pi}{x_n}\right)$ Where $\alpha \in (0,1)$
Logistic-map	$x_{n+1} = ax_n(1 - x_n)$ where $a = 4, n$ represents the iteration number and $xxnn$ denotes the nth chaotic number.
Piecewise-map	$x_{n+1} = \begin{cases} \frac{x_n}{P} & 0 \le x_n < P \\ \frac{x_n - P}{0.5 - P} & P \le x_n < \frac{1}{2} \\ \frac{1 - P - x_n}{0.5 - P} & \frac{1}{2} \le x_n < 1 - P \\ \frac{1 - x_n}{P} & 1 - P \le x_n < 1 \end{cases}$ where $P \in [0, 0.5]$ is a control parameter and $x_n \in [0,1]$

We now discuss the suggested Chaotic Pelican Optimization Algorithm (CPOA). Without a doubt, POA offers a close to optimum answer to a given optimization challenge. The convergence rate of POA, however, may be improved by using chaotic maps to get better outcomes with greater efficiency. In this case, POA has been analyzed using six distinct chaotic maps. Applying chaotic maps to POA has had a significant impact on convergence time. This is some pseudocode for (CPOA):

Algorithm 1. The pseudo-code for the CPOA procedure.

Input.
Calculate the POA population − size (N) and the number − of − iterations (NoIT).
Initialize the position of pelicans with chaotic vector (comprises six distinct chaotic maps)and find out the objective − function.
For t = 1: NoIT
Update Chaotic vector for different chaotic maps
Randomly design prey position.
For I = 1: N
For j = 1: m
Compute new − status of the jth − dimension.
End.
Update the ith population member.
For j = 1: m.
Evaluate new − status of the jth − dimension.
End.
Adjust ith population − member.
End.
Update best − candidate − solution.
End.
Return best − candidate solution.

IV. Results and Discussion

Fig. 1 represents the proposed network model. A user's x and y coordinates can be used to pinpoint their exact location on the map. Each network has its own set of CH, all of which are linked to the central hub. Table II shows that on average there are 10 users in each of the clusters. These users are represented by the little circles that are connected to the cluster head. It demonstrates the network requirements for modelling the suggested method. An N, a CH, a BW, and an SNR in the range of –10 to 10 dB stand for the number of users, cluster head, bandwidth, and signal-to-noise ratio, respectively.

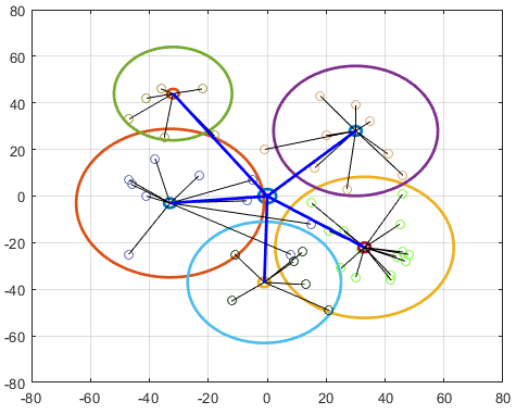

Fig. 1. Proposed network implementation model.

TABLE II. Requirement of Network

Sum of users (N)	300–600
Sum of CH (NC)	30–60
Coverage Area	50 m × 50 m
Bandwidth (BW)	20MHz
Cost Purpose	Least Distance
Optimization	KGMO
Inertia weight	CPOA
SNR Variety	–10 to 10 dB

In this section, the validation analysis of projected model with existing techniques such as standard KGMO and Bacterial Foraging Optimization (BFO) are described in terms of various analysis. The system model is tested with small number of nodes (300) and large number of nodes (600), where Figs. 2–6 presents the validation investigation of projected model for 300 nodes.

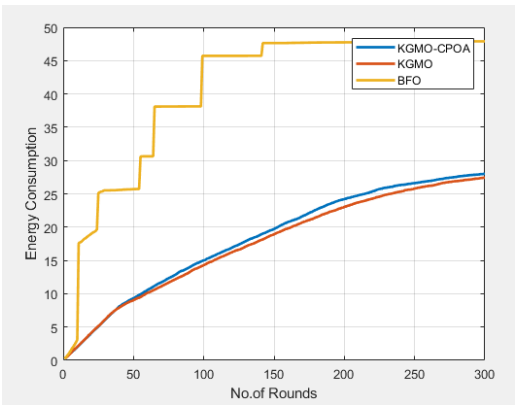

Fig. 2. Analysis of projected model in terms of energy consumption.

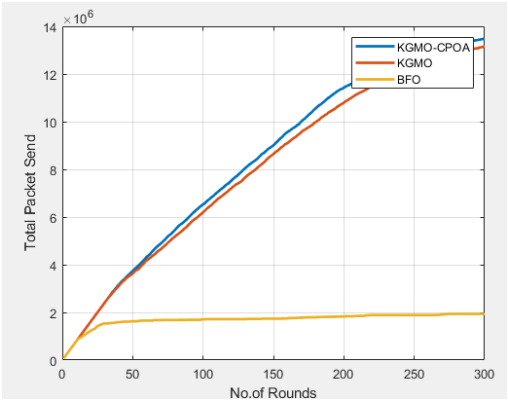

Fig. 3. Analysis based on total packet sent.

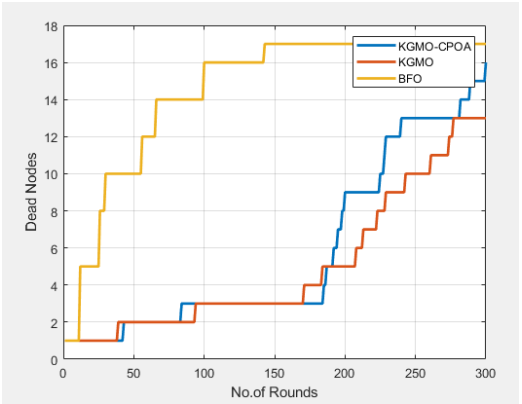

Fig. 4. Performance of proposed model for dead nodes.

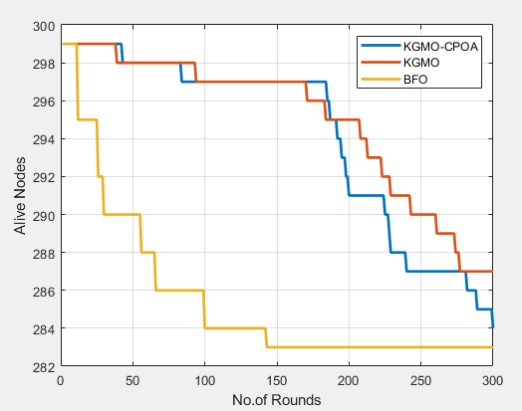

Fig. 5. Comparative analysis of proposed model for alive nodes.

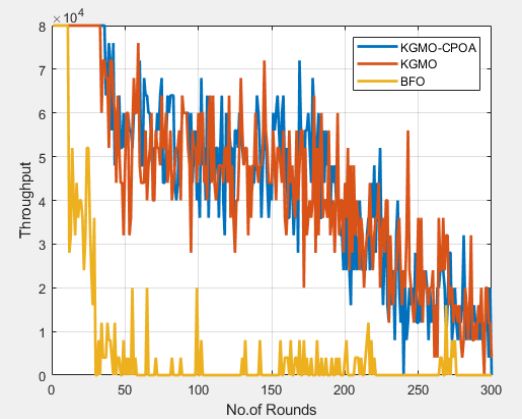

Fig. 6. Analysis of CPOA-KGMO in terms of Throughput.

Since an ineffective cluster head selection procedure drastically decreases the sum of live nodes and the lifetime of the network while simultaneously increasing the number of dead nodes and energy consumption, these factors have been examined in a computer simulation. The first section of the study compared the suggested scheme's dominance across a range of rounds (0–300) based on the proportion of alive nodes to total nodes. Since the suggested technique quickly converges to an estimated optimal solution, the network's living nodes are robustly protected from death. Furthermore, the suggested method safeguards the network's longevity by avoiding the election of ineffective sensor nodes as cluster leaders. As compared to the reference systems, the proposed method is able to keep 15% to 19% of its nodes alive. In order to achieve load balance in the network and ensure that the sensor nodes with the lowest amount of energy do not expire prematurely, it is necessary to pick sensor nodes with energy potential to play the role of cluster head. When compared to popular reference implementations like KGMO and BFO, the suggested strategy significantly reduces the occurrence of newly-emerging dead nodes to between 18% and 23%. Since the suggested method uses the CPOA algorithm to maximise exploration and exploitation without deviating from the optimal path, it is expected to significantly increase throughput on average. Figs. 7–11 compare and contrast the proposed model with already-existing methodologies for various analyses employing 0–600 nodes.

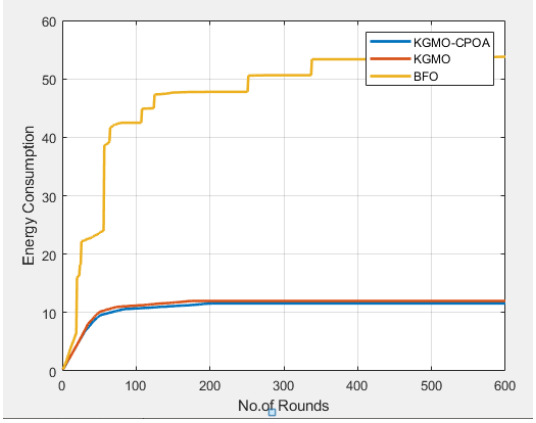

Fig. 7. Analysis of proposed model for energy consumption.

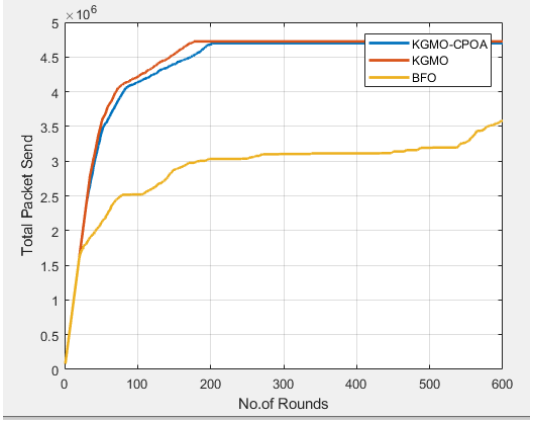

Fig. 8. Comparative analysis in terms of total packet sent.

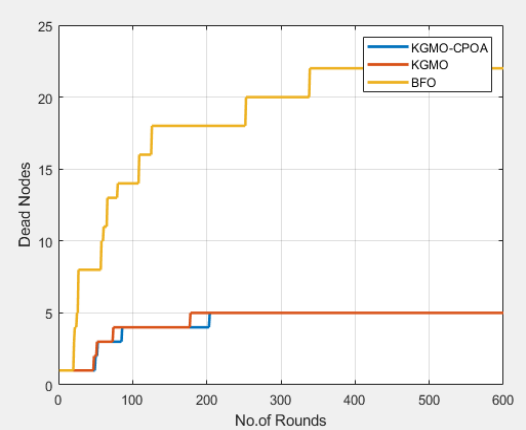

Fig. 9. Analysis of CPOA-KGMO in terms of dead nodes.

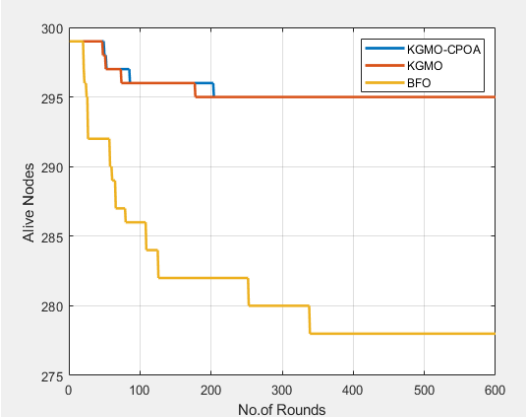

Fig. 10. Validation analysis of proposed model for alive nodes.

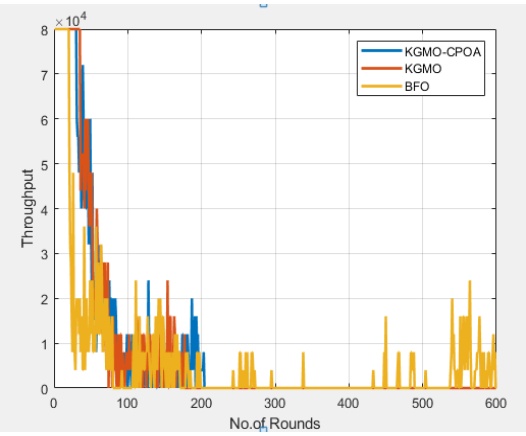

Fig. 11. Throughput comparison for proposed model.

Since the suggested method utilizes the CPOA parameter for adaptive exploitation, which solves the traditional issue of the KGMO algorithm by forcing it to converge, it significantly extends the lifetime of the underlying network. The suggested method facilitates a 6% and 8% improvement in network lifetime, respectively, compared to the reference methods. Since the amount of packet drop is avoided due to significant cluster head selection process, longer average throughput and network lifetime under a variety of sensor node counts. The suggested scheme's temporal complexity is compared to that of the benchmarked systems, both in the best and worst possible scenarios. The suggested model

outperformed KGMO and BFO schemes by 44.76 percentage points and 22.86 percentage points, respectively, in the best-case scenario. The suggested model outperformed the state-of-the-art methods by 55.86% and 51.32%; in the worst-case scenario computation, respectively, the performance of the network is significantly improved. Based on current network conditions, E2HCA uses adaptive techniques to dynamically modify clustering settings. This flexibility makes sure that the model can react to changes in node energy levels and communication patterns in a way that maximizes energy efficiency and minimizes NLT. It helps it be able to produce better results than other existing techniques.

V. Conclusion

When developing clustering algorithms for massive WSNs, minimizing transmission energy usage is a top priority. Current population-based meta-heuristics are intricate, requiring fine-tuning of a variety of parameters in order to maximize energy efficiency. The residual energy of the nodes, the various distance parameters, and the workload on the Cluster Heads (CHs) are not taken into account by state-of-the-art clustering techniques, which limits the network's Capability. The authors of this study propose a better-quality version of the KGMO algorithm to address the energy efficiency concerns associated with CH selection. To address the quick convergence rate of the KGMO algorithm, CPOA is used to adjust the inertia weight of the algorithm. In this manuscript, we develop an original chaotic POA to address the issue of KGMO. Six distinct chaotic maps have been created to enhance POA's functionality. In addition, the six chaotic variants' performance was compared to that of the original POA. Throughput, end-to-end delay, energy usage, etc. are all measured in a series of MATLAB simulations that confirm KGMO-CPOA in comparison to other heuristic algorithms like KGMO and BFO. The outcomes show that the suggested KGMO-CPOA outdid the alternatives in terms of latency, energy consumption, and network longevity. Further study should involve evaluating the suggested model the suggested method safeguards the network's longevity by avoiding the election of ineffective sensor nodes as cluster leaders. As compared to the reference systems, the proposed method is able to keep 15% to 19% of its nodes alive. In order to achieve load balance in the network and ensure that the sensor nodes with the lowest amount of energy do not expire prematurely and in a real-time setting with a larger number of nodes to ensure optimal performance.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

S.S. has written the paper; S.S. and E.S. have conducted the research and analyzed the data; all authors have approved the final version.

References

[1] N.T. Tam, T. H. Hung, and H. T. T. Binh, "A decomposition-based multi-objective optimization approach for balancing the energy consumption of wireless sensor networks," *Applied Soft Computing*, vol. 107, 107365, 2021.

[2] M. Abbas and N. Otayf, "A novel methodology for optimum energy consumption in wireless sensor networks," *Frontiers in Engineering and Built Environment*, vol. 1, no. 1, pp. 25–31, 2021.

[3] R. Vatambeti, S. Sanshi, and D. P. Krishna, "An efficient clustering approach for optimized path selection and route maintenance in mobile ad hoc networks," *J. Ambient Intell Human Comput*., vol. 14, pp. 305–319, 2023.

[4] H. Jafari, M. Nazari, and S. Shamshirband, "Optimization of energy consumption in wireless sensor networks using density-based clustering algorithm," *International Journal of Computers and Applications*, vo. 43, no. 1, pp.1–10, 2021.

[5] S. Sadhana, E. Sivaraman, and D. Daniel, "Enhanced energy-efficient routing for wireless sensor network using extended power-efficient gathering in sensor information systems (E-PEGASIS) protocol," *Smart Systems: Innovations in Computing*, pp. 159–171, 2021.

[6] S. Manikandan and M. Chinnadurai, "Effective energy adaptive and consumption in wireless sensor network using distributed source coding and sampling techniques," *Wireless Personal Communications*, vol. 118, no. 2, pp. 1393–1404, 2021.

[7] R. Kocherla and R. Vatambeti, "An efficient routing strategy for energy management in wireless sensor network using hybrid routing protocols," *Wireless Pers Commun*., vol. 124, pp. 49–73, 2022.

[8] C. Jothikumar, K. Ramana, V. D. Chakravarthy, S. Singh, and I. H. Ra, "An efficient routing approach to maximize the lifetime of IoT-based wireless sensor networks in 5G and beyond," *Mobile Information Systems*, p. 11, 2021.

[9] C. C. P. K. Pareek, V. H. C. D. Albuquerque, A. Khanna, and D. Gupta, "Deep learning technique-based intrusion detection in cyber-security networks," in *Proc. 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, 2022, pp. 1–7.

[10] C. J. Kumar, V. D. Chakravarthy, K. Ramana, P. K. R. Maddikunta, Q. Xin, and G. S. Narayana, "OTP-ER: An ordered transmission paradigm for effective routing in IoT based wireless sensor networks," *Optical and Quantum Electronics*, vol. 54, no. 7, 2022.

[11] C. E. Galarza, J. M. Palma, C. F. Morais, J. Utria, L. P. Carvalho, Bustos, and R. C. Oliveira, "A novel theoretical probabilistic model for opportunistic routing with applications in energy consumption for WSNs," *Sensors*, vol. 21, no. 23, 8058, 2021.

[12] R. Kocherla, M. C. Sekhar, and R. Vatambeti, "Enhancing the energy efficiency for prolonging the network life time in multi-conditional multi-sensor based wireless sensor network," *Journal of Control and Decision*, pp. 72–81, 2022.

[13] H. R. Farahzadi, M. Langarizadeh, M. Mirhosseini, and S. A. Fatemi Aghda, "An improved cluster formation process in wireless sensor network to decrease energy consumption," *Wireless Networks*, vol. 27, pp. 1077–1087, 2021.

[14] E. G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, vol. 74, 2009.

[15] S. J. Hopwood and J. Jeans, *An Introduction to the Kinetic Theory of Gases*, Cambridge University Press, 2009, p. 18.

[16] L. B. Loeb, "The kinetic theory of gases," *Courier Corporation*, pp. 494–495, 2004.

[17] E. Obi, Z. Mammeri, and O. E. Ochia, "A centralized routing for lifetime and energy optimization in WSNs using genetic algorithm and least-square policy iteration," *Computers*, vol. 12, no. 2, 22, 2023.

[18] A. A. Abdulzahra, B. A. Q. Khudor, and S. Alshawi, "Energy-efficient routing protocol in wireless sensor networks based on bacterial foraging optimization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 2, pp. 911–920, 2023.

[19] S. Janakiraman, "Improved bat optimization algorithm and enhanced artificial bee colony-based cluster routing scheme for extending network lifetime in wireless sensor networks," *International Journal of Communication Systems*, vol. 36, no. 5, p. e5428, 2023.

[20] J. Vellaichamy, S. Basheer, P. S. M. Bai, M. Khan, S. K. Mathivanan, P. Jayagopal, and J. C. Babu, "Wireless sensor

networks based on multi-criteria clustering and optimal bio-Inspired algorithm for energy-efficient routing," *Applied Sciences*, vol. 13, no. 5, 2801, 2023.
[21] D. Viswanathan, S. R. Kumari, and P. Navaneetham, "Soft C-means multi objective metaheuristic dragonfly optimization for cluster head selection in WSN," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, pp. 88–95, 2023.
[22] S. Chaurasia, K. Kumar, and N. Kumar, "MOCRAW: A meta-heuristic optimized cluster head selection-based routing algorithm for WSNs," *Ad Hoc Networks*, 103079, 2023.
[23] M. R. Reddy, M. L. R. Chandra, P. Venkatramana, and R. Dilli, "Energy-efficient cluster head selection in wireless sensor networks using an improved grey wolf optimization algorithm," *Computers*, vol. 12, no. 2, 35, 2023.
[24] P. Trojovský and M. Dehghani, "Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications," *Sensors*, vol. 22, no. 3, 855, 2022.
[25] A. H. Gandomi and X. S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, 2014.
[26] S. Arora, M. Sharma, and P. Anand, "A novel chaotic interior search algorithm for global optimization and feature selection," *Applied Artificial Intelligence*, vol. 34, no. 4, pp. 292–328.
[27] I. Fister, M. Perc, S. M. Kamal, and I. Fister, "A review of chaos-based firefly algorithms: Perspectives and research challenges," *Applied Mathematics and Computation*, vol. 252, pp. 155–165, 2015.