SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I3P105

Volume 13 Issue 3, 51-70, March 2026
© 2026 Seventh Sense Research Group®

Original Article

Energy-Aware Cluster Head Selection and Routing using Hyb-WhiOp and Op-MulDRL in IoT Sensor Networks

R. Ratheesh[1], M. Saranya Nair[2], Blessina Preethi R[3], N.V.S.Sree Rathna Lakshmi[4]

[1,3]ECE, Agni College of Technology, Chennai, Tamilandu, India.
[2]SENSE, Vellore Institute of Technology, Chennai, Tamilnadu, India.
[4]ECE, SRM Institute of Science and Technology, Ramapuram, Chennai.

[2]Corresponding Author : saranyanair.m@vit.ac.in

Received: 06 December 2025 Revised: 07 January 2026 Accepted: 06 February 2026 Published: 23 March 2026

***Abstract** - Wireless Sensor Networks (WSN) are a vital component of many IoT applications, enabling the efficient gathering and transmission of data. Current methods, such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimization (GWO), and Improved Multi-Dimensional Energy-Aware Cluster-Based Routing (IMD-EACBR), often struggle with early CH depletion, inefficient routing, and poor adaptability. This paper introduces an energy-aware, trust-based framework for CH selection and routing, aimed at extending Network lifetime (NLT) and enhancing performance. In order to overcome these issues, a novel energy-efficient Optimized Multi-Objective Deep Reinforcement Learning (Op-MulDRL) for a routing mechanism is proposed. The designed framework incorporates a Hybrid White Whale Optimization (Hy-WhiOp) algorithm for CH selection, an Op-MulDRL model for routing. CH selection in WSN based on residual energy, trust, distance, latency, and path quality. On the other hand, Op-MulDRL significantly increases the power factor by learning and adapting the optimal routing paths that are more suitable to the network conditions without needing any further input from the controller. The Tent Chaos Rabbit Optimization (Ten-Rabo) helps to adjust the DRL parameters, thus further increasing the performance of Op-MulDRL. The new framework is verified through extensive simulations. It is found to outperform the best models, including PSO, GWO, Ant Lion Optimization (ALO), GA, Sunflower Optimization (SFO), and IMD-EACBR. The models achieved throughput of 0.522 Mbps, 0.891 Mbps, 0.565 Mbps, 0.728 Mbps, 0.632 Mbps, 0.934 Mbps, and 0.975Mbps across several performance metrics. In conclusion, the hybrid Hy-WhiOp, along with Op-MulDRL, contributes significant improvements in efficient energy, routing flexibility, and network resilience, which makes it an up-and-coming option for upcoming IoT-enabled WSN applications.*

***Keywords** - Wireless Sensor Network, CH selection, Routing, Deep Reinforcement Learning, Rabbit Optimization, and Tent Chaos.*

1. Introduction

WSNs are used in contemporary technology to gather various areas of interest data and forward that to the monitoring system through the components of the Internet of Things (IoT). The application of IoT-based WSN in the health analysis sector, Precision agriculture, Smart Architecture of Houses and Buildings, Military monitoring, and data aggregation [1]. However, in such an application-based network, the NP-hard problem is the mitigation of energy consumption, which also has various solving methods of clustering and routing [2]. Nevertheless, the routing over the clustered nodes in the network reduces the energy utility on the basis of direct transmission, which results in optimizing the overall Network Life Time (NLT) [3].

When using the cluster-oriented routing protocol alongside a static node, information loss is avoided with the fixed sink/Base Station [4]. As the nodes and the BS are static, a dynamic creation of CH and routings is preferred with an optimal number of clusters and CHs [5]. In the clustering process, traditionally, the nodes are clustered with the nearest CH from which the nodes receive a strong signal strength to transmit and receive data. This cluster formation process has a significant role in reducing the energy consumption of a network [6]. Besides, initially, a cluster is formed based on the proximity between the nodes, and the CH is selected in the cluster based on various parameters, which include the fitness functions [7]. The node that competes for the CH position satisfies the basic requirements of maximum residual energy, a shorter distance to the BS, and neighbouring nodes [8]. Because data aggregation, processing, and transmission depend on the CH functionality, the energy consumption of the CH is higher than that of the member nodes in the cluster [9]. The energy consumption in data transmission is directly related to distance; hence, to reduce the transmission range, the nearby nodes are selected as the relay nodes for the CHs to forward the data in an optimal path [10]. In Low-Energy Adaptive Clustering Hierarchy (LEACH), the CH collects messages in each round, compresses them, and then sends the smaller packets to the BS. So, the size of the transmitted data can change each time [11]. An Intelligent Energy-aware Routing in Mobile IoT networks leveraged the flexibility of Software Defined Networking (SDN) to manage moving mobile nodes [12]. Routing was easier to manage since only the CH had to keep track of local routes for the other CHs, which meant less routing information was needed overall. The energy-aware networking is also dependent on the fault tolerance in clustering, which is highly mitigated by the selection of multi-objective fitness-based CHs using meta-heuristic algorithms [13, 14]. In WSNs, even though it uses up much more energy than processing and is the main drain on power, this downside is balanced out by better overall performance in packet delivery, network lifespan, delay, and energy efficiency [15].

However, the limited battery capacity of sensor nodes makes energy a critical challenge, especially in cluster-based routing architectures where the CHs are exposed to heavy energy consumption due to data gathering and long-distance transmission. In fact, many metaheuristic algorithms like PSO, GA, GWO, and ALO were designed for CH selection and routing. However, most of the existing methods solve these issues separately and mainly focus on single-objective optimization, e.g., residual energy or distance. Besides that, the present methods tend to disregard essential elements such as trust management, communication delay, and path reliability, which are crucial in the dynamic IoT-based WSN environments. Besides that, learning-based routing methods face issues like unstable convergence and inefficient hyperparameter tuning in high-dimensional network scenarios. Thus, there is a substantial research gap in the creation of a single, multi-objective framework that simultaneously optimizes CH selection, adaptive routing, and learning parameter tuning together with energy balance, reliability, scalability, and network longevity. To address these gaps, this research proposed a comprehensive integrated hybrid framework that uses Hybrid White Whale Optimization (Hy-WhiOp) for multi-criteria CH selection and Optimized Multi-Objective Deep Reinforcement Learning (Op-MulDRL) for adaptive routing. It is then potentiated with Tent Chaos Rabbit Optimization (Ten-Rabo) for practical hyperparameter tuning, thereby leading to better energy distribution, network stability, and a longer lifetime of IoT-enabled WSNs.

2. Related Work

Some of the recent related studies are surveys and are described in the following section:

A lot of recent work in the application of IoT through WSNs has mainly concentrated on finding ways to save energy, on the selection of CHs, and on the reliability of routing. Divya and Sudhakar [16] came up with an Adaptive Multi-Path Routing Protocol (AMPRP), which features the Circle-Inspired Optimization Algorithm (CIOA) to decide the best CHs and routing paths. The technique showed a better performance than the existing ones.

However, CH selection was still very much affected by the parameter tuning, and the NP-hard nature of clustering restricted the scalability for very dense deployments. Jecan et al. [17] put forward a Predictive Energy-Aware Routing (PEAR) framework targeting industrial IoT settings. The idea was to increase the NLT by energy profiling and changing routing. As a matter of fact, PEAR made energy more predictable, and clusters last longer. However, its performance was very much aligned with hardware limitations and topology layouts, thus lowering its flexibility in mixed IoT scenarios.

Shakya et al. [18] explained that their Fuzzy-Based Unequal Clustering Algorithm (FBUCA) was supplemented with a modified minimum spanning tree to optimize routing. They used a fuzzy inference system to improve CH selection by considering energy parameters and more of the node features, which led to a longer network lifespan and higher communication efficiency. At the same time, the need for powerful nodes and the higher computational complexity made the approach less suitable for resource-constrained WSNs. Jalalinejad et al. [19] suggested a hybrid multi-hop clustering and energy-aware routing protocol for energy harvesting WSNs. The combination of centralized and decentralized clustering strategies made the method surpass the traditional methods in terms of node survival and stability. On the other hand, the protocol still needed a good power management strategy, and trust, delay, and link quality metrics were not integrated simultaneously within its optimization framework.

Jibreel et al. [20] came up with the Heterogeneous Gateway-based Energy-Aware Multi-hop Routing Protocol (HMGEAR) that used energy-centric CH selection and heterogeneous nodes to increase network stability and lifetime. The strategy, which went some way to solving the energy hole problems and improving the residual energy distribution, still had the routing mechanism that was not adaptive and increased the computational overhead for large-scale networks. Li et al. [21] presented the Location and Energy-aware K-means Clustered Routing (LE-KCR) algorithm for underwater WSNs. Integrating residual energy, node location, and dual-hop routing, the method effectively lessened dead nodes and increased energy usage. Nonetheless, this method depended heavily on localization accuracy and required an extra computational load during the clustering process.

Table 1. Existing model survey by considering dataset, performances and limitations

References	Methodology	Performance	Limitations
Divya et al. [16]	AMPRP which uses CIAO for CH and NCH selection	85.71%, 86.67%, 88.89%, and 77.78% better than EAHPW-TOPSIS, DA-EECHS, GEEC, and EADCR, respectively	NP-hard nature of CH selection; sensitive to parameter tuning
Jecan et al. [17]	PEAR model	PEAR solution reduced intracluster overconsumption by 10.4 times after 210 routing changes, increased intercluster lifetime by an average of 2.3 times, and lowered average energy consumption by 23.6%.	Sensor lifetime is highly dependent on topology, deployment, and application needs. Without an energy-aware mechanism, lifetime predictability cannot be ensured.
Shakya et al. [18]	FBUCA with fuzzy logic and modified minimum spanning tree for CH selection	Increased network lifespan, better energy use and communication efficiency	Requires resource-rich nodes; potential complexity
Jalalinejad et al. [19]	Multi-hop clustering with energy harvesting and centralized/decentralized routing	Outperforms AEHAC, CRBS, HUCL, and EADUC in energy, stability, and node survival	Capacity limitation requires energy-aware management strategy
Jibreel et al. [20]	HMGEAR protocol for energy-aware heterogeneous multi-hop routing	Improved stability, energy efficiency, and network lifetime	Original MGEAR had high energy drain; solution includes heterogeneous nodes and energy-centric CH
Li et al. [21]	LE-KCR algorithm using K-means with location and energy-aware dual-hop routing for UWSNs	Reduced energy consumption, fewer dead nodes; better than LEACH	High dependency on localization and computational load
Ramamoorthi et al. [22]	CLIENT model using blockchain with chaotic maps, capuchin search optimization, and deep learning	Outperformed existing methods in accuracy, delay, security, and lifetime	Blockchain scalability issues with high user load
Raslan et al. [23]	ISFO for CH selection with Levy flight	Outperforms competitors in long-term performance and node survival	High computational complexity affects real-time use
Cherappa et al. [24]	Adaptive Sailfish Optimization with K-medoids and E-CERP cross-layer routing	100% PDR, 0.05s delay, 0.99 Mbps throughput, 5908 cycles lifetime	Limitations in clustering, low communication speed, storage, communication capacity, high configuration complexity, limited computation
Badiger et al. [25]	EDAS using improved LEACH with energy-aware CH generation	Outperforms by reducing energy consumption, improving throughput, and extending network lifetime in IoT-based WSNs	High data volume leads to overhead, collisions, redundancy.

Ramamoorthi and Appathurai [22] presented the CLIENT framework, a marriage of blockchain, chaotic maps, optimization techniques, and deep learning to create a secure and energy-efficient IoT communication. The model, on one hand, brought significant improvements in accuracy, security strength, and network lifetime. However, on the other hand, blockchain scalability and increased communication overhead were the main obstacles to its application in large-scale IoT environments. Raslan et al. [23] presented the Improved Sunflower Optimization (ISFO) algorithm for CH selection using Levy flight mechanisms to balance exploration and exploitation. The method brought about an increased node survival and a better long-term performance; however, its high computational cost can be a barrier to its use in real-time situations.

Cherappa et al. [24] utilized Adaptive Sailfish Optimization (ASFO) with K-medoids clustering and cross-layer routing to enhance quality of service parameters, like packet delivery ratio and delay. Although the method reached impressive throughput and lifetime metrics, it had drawbacks in terms of clustering complexity and limited communication scalability. Lastly, Badiger and Ganashree [25] came up with an Efficient Data Aggregation Scheme (EDAS) that focused on optimal CH generation through an improved version of LEACH. EDAS lowered energy consumption and upped the throughput; however, continuous sensing and large data volume generation in IoT contexts led to redundancy, collisions, and thus, increased overhead.

Table 1 shows that the majority of methods so far are mainly focused on the energy aspect of CH selection and consider clustering and routing as two separate problems. Very few studies have combined trust evaluation, communication delay, and link reliability in a single framework. Besides, metaheuristic methods commonly face issues like getting stuck in local optima and being highly dependent on parameters. In contrast, learning-based methods bring in the problem of the system becoming more complex and less scalable. On that account, a significant research gap remains in developing a multi-objective framework that jointly optimizes CH selection, adaptive routing, and learning stability while maintaining balanced energy distribution and longer NLT in changing IoT-enabled WSNs. To overcome these weaknesses, the authors propose a combined Hy-WhiOp and Op-MulDRL framework that performs multi-objective CH selection, adaptive routing, and DRL stability optimization together with Ten-Rabo-based hyperparameter tuning. Thus, through balanced energy distribution, enhanced reliability, and prolonged NLT are achieved in dynamic IoT-enabled WSN environments.

3. Proposed Model

The flow of the proposed CH selection and optimized routing framework for IoT-enabled WSNs is given in Figure 1. SNs are grouped into clusters, and each cluster consists of a Cluster Member (CM). Using Hy-WhiOp, optimal CHs were selected based on energy, distance, path quality, security, and delay. First, the CH collects data from each CM and aggregates it. Then, CH sends the collected data to a remote BS. An Op-MulDRL algorithm manages routing within and between CH. The BS is connected to a private or public network and handles data transmission to the internet.

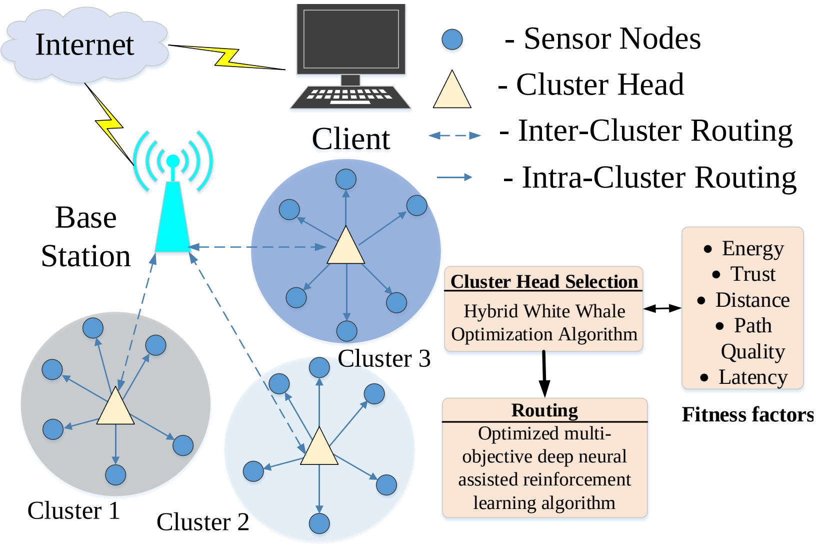

Fig. 1 System architecture of the proposed method

The novelty of this research is the development of a multi-objective, intelligence-driven framework that simultaneously optimizes CH selection, adaptive routing, and DRL hyperparameter tuning through one integrated architecture for IoT-enabled wireless sensor networks, in contrast to current methods that handle clustering and routing as separate issues or are mainly focused on a single-energy objective, residual energy, or distance. The proposed model combines residual energy, direct and indirect trust, communication delay, distance to the base station, and RSSI-based path quality into a complete multi-criteria CH selection method with the Hybrid White Whale Optimization (Hy-WhiOp) algorithm. Besides, an Op-MulDRL model, which can self-adjust to network conditions in real time, is used to guide the routing process; thereby, delay, reliability, and energy consumption are balanced through an intricately designed reward structure. Moreover, the Ten-Rabo algorithm, which introduces another level of innovation, is used for smart hyperparameter tuning to reinforce DRL's convergence stability and learning speed. The combined clustering, routing, and learning optimization framework of such a nature leads to an increase in global energy balance, network stability, and communication reliability, thereby setting it apart from the present metaheuristic or learning-based solutions, which do not have coordinated multi-objective integration.

3.1. Cluster Head Selection using Hy-WhiOp for Cluster Head Selection

Traditional optimization models often struggle with issues like getting stuck in local optima, high computational demands, and an imbalanced search process. To address these existing models, the Hy-WhiOp offers a more efficient approach by selecting optimal CH. The Hy-WhiOp algorithm is designed based on the mathematical model of the intellectual hunting behaviour of humpback whales in the ocean. Specifically, it replicates the whale's spiral bubble net feeding strategy, where whales create a spiral pattern to encircle and trap their search space. Like many optimization algorithms, Hy-WhiOp operates through two main phases: exploration and exploitation. Exploration implies a more expensive, worldwide search for different solutions, whereas exploitation is more about modifying the most promising candidates within a local search space to improve accuracy.

Table 2. Parameters and initialization in Hy-WhiOp [27, 28]

Parameters	Values
Population Size (Whale)	100
Population size (Shark)	30
Maximum Iteration (Whale)	150
Maximum iteration (Shark)	2000
Spiral updating probability	0.5
Shrinking encircling	0.5
Random search ability	0.1
Dimensions	2
Transmission range	100m-600m
Nodes	20-50
Inertia weight (W)	0.694
Lower bound (lb)	0
Upper bound (ub)	100
Mobility model	Random waypoint
Maximum frequency (F_{max}) of the wave motion	0.75
Minimum frequency (F_{min}) of the wave motion	0.07
Acceleration coefficient (tau)	4.125

WOA [26, 31] initiates a population of candidate solution positions representing whales that are randomly distributed in the search space. In every iteration of the WOA, the whale positions are re-adjusted depending on the random selection of a whale that motivates the search of the solution space when the absolute value of the parameter. This adaptive process allows the algorithm to continually balance the energy between searching new areas and exploiting the old areas. The leading whale, or best search agent, corresponds to the highest performing solution at any point in time and thus directs the rest of the swarm in that direction. Parameters and Initialization in Hy-WhiOp are given in Table 2 WOA's architecture is the implementation of the mathematical representations of the spiral and encircling movements of whales. These features of WOA facilitate its movement beyond the local optima and enhanced performance. It further acts on the simple structure, flexibility, and strong global search capacity of WOA, which have made it a very reliable tool across various optimization problems. This approach is applicable in various places, such as energy-efficient clustering and CH selection in WSN. Besides that, selecting a smart node is indispensable to the network's life-lengthening and energy reduction.

3.1.1. Search Space Encirclement

Whales follow the behaviour of surrounding the current best search agent, renewing their positions during the optimization. Circulating mathematically designed by the following equations:

$$\vec{M} = \left|\vec{P}\cdot\ \vec{Y} * (k) - \vec{Y}(k)\right| \tag{1}$$

$$\vec{Y}(k+1) = \vec{Y} * (k) - \vec{Q}\cdot M \tag{2}$$

Where k is the current relaxation, $\vec{Q}$ and $\vec{P}$are characteristics and $\vec{Y}$is the status vector. The best solution is provided by$\vec{Y}\ *$. If there is an excellent solution, the value of $\vec{Y}\ *$will be updated after each relaxation. The characteristics $\vec{Q}$ and $\vec{P}$ are calculated by Equations 3 and 4:

$$\vec{Q} = 2\vec{s}\cdot\vec{t} - \vec{s} \tag{3}$$

$$\vec{P} = 2\cdot\vec{t} \tag{4}$$

The vector $\vec{s}$ In the course of relaxation, decreases from 2 to 0, and $\vec{t}$ is a random vector, which varies between [0, 1].

3.1.2. Exploration Phase

The attack is designed to use the bubble-net strategy through two main approaches. The humpback whales can run one of these two mechanisms to catch prey, so that this mechanism can occur with a probability of 50%. A random variable $u$is introduced in varies between [0, 1]—the value of $s$decreases between [-s, Q] in the shrinking environment.
The value of the [-1, 1] is between, so |Q| < 1, then the

exploitation is triggered, and all search agents get the best solution. It can provide the updated model by Equation 5:

$$\vec{Y}(k+1) = \begin{cases} \vec{Y}*(k) - \vec{Q}\cdot\vec{M}, & if\ u < 0.5 \\ \vec{M}\cdot e^{\mu\lambda}\cdot cos(2\pi\lambda) + \vec{Y}*(k), & if\ u \geq 0.5 \end{cases} \quad (5)$$

3.1.3. Exploitation Phase

The study phase is based on the vector's variation.$\vec{Q}$Moreover, it is to mobilize search agents in search of better solutions, like global search. This is similar to conducting a global search for better solutions. Specifically, when the magnitude of $\vec{Q}$is greater than 1, it pushes the agents to move further away, helping them explore more diverse regions. Unlike the exploitation phase, where agents move toward the best-known solution, in exploration, agents update their position based on randomly selected peers; this randomness prevents the convergence on local optima. The behaviour of this phase is typically described by Equations 6 and 7.

$$\vec{M} = \left|\vec{P}\cdot\vec{Y}_{rand} - \vec{Y}\right| \quad (6)$$

$$\vec{Y}(k+1) = \vec{Y}_{rand} - \vec{Q}\cdot\vec{M} \quad (7)$$

In order to enhance the further performance of the CH selection, the white shark algorithm has to be hybridized to this phase, inspired by the hunting behaviour. The hybrid white shark algorithm's equations have been derived mathematically as follows.

The Hyb-WhiOp algorithm combines the adaptive hunting strategies of white sharks with the spiral foraging behaviour of humpback whales, resulting in a powerful and flexible metaheuristic designed for global optimization tasks. White sharks are known for using multiple sensory cues to track and adjust their movement toward search space, while whales mimic bubble-net feeding through spiral encircling or random exploration. In this hybrid method, search agents dynamically choose whether to follow a shark-line tracking pattern or whale-inspired spiralling based on the iteration stage and diversity within the population. The velocity update inspired by white sharks is expressed as:

$$S_j^{t+1} = \alpha\left[S_j^t + \lambda_1\left(G_t - P_j^t\right)\cdot r_1 + \lambda_2\left(L_j^t - P_j^t\right)\cdot r_2\right] \quad (8)$$

Where, $S_j^{t+1}$is the updated velocity of the j^{th} agent, S_j^t is the current velocity, G_t and L_j^t are the global and local best positions, respectively, P_j^t is the current position, λ_1 and λ_2 Are weights controlling the influence of global and local learning. The random values (r_1 and r_2) from 0 to 1, α is a shrinkage factor that guides convergence. The spiral movement inspired by whales is modelled as:

$$P_j^{t+1} = \mathit{\Delta}_j\cdot e^{\beta\theta}\cdot cos(2\pi\theta) + B_t \quad (9)$$

Where, P_j^{t+1} is the position of the agent, $\mathit{\Delta}_j = \left|B_t - P_j^t\right|$ is the distance to the best-known position, B_t is the current best solution, β is a spiral-shaping constant, θ is a random number from -1 to 1, controlling the spiral's amplitude and decay.

At each step, agents make a probabilistic choice between the sharks' focused, directional movement and the whales' exploratory spiral path, which balances the exploration and exploitation, leading to faster convergence and greater robustness.

3.2. Factors Influencing Optimal Cluster Head Selection

The optimal CH selection is based on multiple constraints, including:

- Energy Level
- Security Requirements
- Distance to the BS
- Trustworthiness (both direct and indirect)
- Communication Delay
- Path Quality (reliability)

3.2.1. Energy Level

A critical concern in WSNs is energy depletion in nodes, in which the energy is consumed during various operations of transmitting and receiving data, sensing the environment, and aggregating information. The standard energy model for transmitting data is given by:

$$E_{TX}(b,d) = \begin{cases} E_{elec}\cdot b + E_{fs}\cdot b\cdot d^2, & if\ d < d_0 \\ E_{elec}\cdot b + E_{mp}\cdot b\cdot d^2, & if\ d \geq d_0 \end{cases} \quad (10)$$

Here, b is the size of the data in bytes, and d is the distance to the BS. E_{elec} represents the energy used by electronic circuits, while $E_{fs}$and $E_{mp}$are the energy costs for free-space and multi-path signal propagation, respectively.

The suggested model introduces a more realistic view by considering the remaining energy of each node:

$$E_{TX}(b,d) = \begin{cases} E_{init} - \left(E_{elec}\cdot b + E_{fs}\cdot b\cdot d^2\right), & if\ d < d_0 \\ E_{init} - \left(E_{elec}\cdot b + E_{mp}\cdot b\cdot d^2\right), & if\ d \geq d_0 \end{cases} \quad (11)$$

Where E_{init} is the node's initial energy. Additional energy components include:

Electronics and aggregation energy:

$$E_{elec} = E_{TX} + E_{agg} \quad (12)$$

Reception energy:

$$E_{RX} = E_{elec}\cdot b \quad (13)$$

Amplification energy:

$$E_{amp} = E_{fs}\cdot d^2 \quad (14)$$

Threshold distance (for switching models):

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}} \tag{15}$$

Total energy consumption:

$$E_{total} = E_{TX} + E_{RX} + E_{idle} + E_{sense} \tag{16}$$

3.2.2. Distance to the BS

In WSNs, minimizing the communication distance is essential for energy efficiency. The overall distance for packet transmission is calculated using the following equation:

$$\boldsymbol{D_{total} = D_{node\_to\_CH} \cdot D_{intra\_nodes}} \tag{17}$$

Where, $\boldsymbol{D_{node\_to\_CH}}$ is the total communication distance from SNs to their CH and from the CH to the BS. $\boldsymbol{D_{intra\_nodes}}$ is the total internal distance between all normal nodes. These components are computed as:

$$\boldsymbol{D_{node\_to\_CH} = \sum_{i=1}^{N} \sum_{j=1}^{C} \left\| E_i^{norm} - C_j \right\| + \left\| C_j - BS \right\|} \tag{18}$$

Distance among normal nodes:

$$\boldsymbol{D_{intra\_nodes} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left\| E_i^{norm} - E_j^{norm} \right\|} \tag{19}$$

Where, $\boldsymbol{E_{norm_i}}$ and $\boldsymbol{E_{norm_j}}$ are the energy levels or positions of the$\boldsymbol{i^{th}}$and $\boldsymbol{j^{th}}$normal SNs. $\boldsymbol{C_j}$is the position of the $\boldsymbol{j^{th}}$ CH in the cluster. $\boldsymbol{BS}$ is the position of the BS. $\boldsymbol{N}$is the number of normal nodes. Lower values of $\boldsymbol{D_{total}}$ indicate a more efficient and energy-saving system.

3.2.3. Trustworthiness

In WSNs, building trust between nodes is essential for ensuring secure and reliable communication. Trust is typically evaluated using two key components: direct and indirect trust. Direct trust is grounded in a node's individual interactions with its neighbour nodes. It mirrors the extent to which a neighboring node has acted consistently in previous instances of trust, such as whether data transmission has been successful. This trust level is changed gradually with a weighted average over time in order to give a fair share to both past and recent experiences:

$$\boldsymbol{T_r = \eta \cdot SM_v + (1 - \eta) \cdot SM_{z,y-1}} \tag{20}$$

Indirect trust is derived from recommendations given by other nodes in the network that have already involved the target node. This is especially valuable in the situation where a node has no direct information and therefore has to trust the feedback of its peers in order to make a correct judgment about the trustworthiness of another node:

$$\boldsymbol{T_d = \frac{\sum K \cdot D}{\sum K}} \tag{21}$$

Here, $\boldsymbol{K}$ stands for the trust of each recommender that comes from the degree of consistency and truthfulness of their past feedback. Through the use of direct and indirect trust, nodes can gain a more comprehensive trust in their peers' reliability. This combined trust metric is pivotal for safe operations like CH selection and data routing in WSNs.

3.2.4. Delay

The communication delay in a cluster is proportional to the maximum communication load within that cluster, which can be estimated as in Equation (22).

$$D_{del} = \frac{max_{k=1}^{N_c^e}(N_c^t)}{f} \tag{22}$$

3.2.5. Path Quality

The Received Signal Strength Indicator (RSSI) is corrupted as the inverse square of the distance between the transmitter and receiver, which is the $\boldsymbol{f_r}$ model in the Friis transmission of power. To make wireless communication work well, the RSSI values in a well-designed system should be as high as possible, indicating a strong signal. Theoretically, RSSI can be represented as:

$$\boldsymbol{RSSI = -36 \times log(D) - 55} \tag{23}$$

Here, $\boldsymbol{D}$ is the distance between the transmitter and receiver. To calculate the distance from the RSSI value, the equation can be rearranged in the following way:

$$\boldsymbol{D = 10^{(RSSI+55)/-36}} \tag{24}$$

Such that CH selection in WSNs is influenced by a set of vital parameters for the network to be efficient, safe, and secure. The foremost criterion is the node's energy, hence the nodes that have more residual power, and trust both of them are directly and indirectly assessed so that reliable communication can be established. Moreover, distance is a factor that helps to reduce the transmission cost, and path quality is a parameter that guarantees uninterrupted and trustworthy data delivery. In addition, delay is also taken into consideration to keep the communication always timely in the whole network. The CH picking process, by weighting all these elements, thus becomes a central part of the contribution to the network's improved stability, performance, and general resilience.

3.3. Routing in WSN using Op-MulDRL Algorithm

Unlike static algorithms like Dijkstra's or Bellman-Ford, which struggle to adapt when the network changes, or heuristic methods like PSO, GA, and ACO, which can be slow and get stuck in ideal solutions. To address these existing techniques, RL finds and makes better routing decisions based on real-time network feedback, thus it becomes more flexible and adaptive. It also considers such factors as node changing place, energy, and traffic load; it is able to use the network more efficiently, and it will last longer [29]. Following the selection of the optimal CH, which is based on trust, delay, energy, and distance, the following essential task in a WSN is the routing of the data in an efficient way. This stage not only makes the communication between nodes and the BS smooth but also ensures the saving of network resources. In this paper, an intelligent method called the Op-MulDRL algorithm governs the data routing. This modern method is deep learning and RL combined, which is why it can be very flexible while searching for new, secure, and most reliable paths for data transfer. Op-MulDRL guarantees that each routing decision not only meets the requirements of network performance but also of its long-term durability—the op-MulDRL process given in Figure 2.

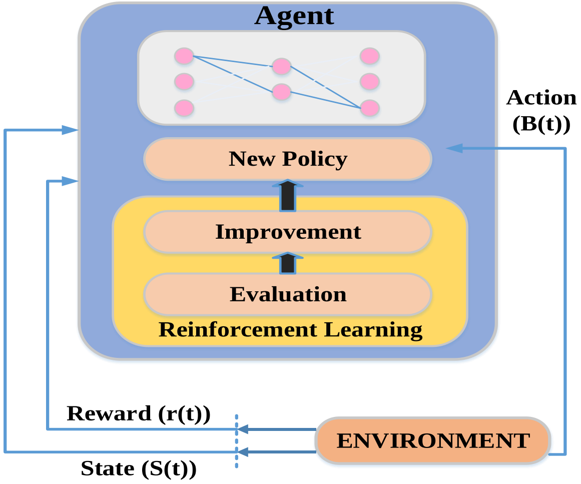

Fig. 2 Op-MulDRL process

3.3.1. State Space

In this method, the state space S_t is designed to capture the current environment around each node by including three key pieces of information, such as the position of the source node$P_s = (x_s, y_s)$, the position of the destination node$P_d = (x_d, y_d)$, and the positions of the neighboring nodes that the current node can observe, denoted byΩ_i. This neighboring information Ω_i includes all nodes within the communication range $r$of node$i$, as well as the current node's own location. Formally, the state is expressed as:

$$S_t = \{(x_s, y_s), (x_d, y_d), \Omega_i\} \tag{25}$$

Where, $\Omega_i = [x_1, y_1, x_2, y_2, \cdots, x_n, y_n]_{2n}$ n is the number of neighboring nodes. Together, these inputs were the source, destination, and the nearby nodes' position, which provide the core information each node uses to make intelligent routing decisions within the reinforcement learning framework.

3.3.2. Action Space

At each time step$t$, the current node needs to decide which neighbor to send the packet to next. So, its action space is basically the list of all nodes within its communication range that it can forward the packet. For node$i$, equation (33) is written as:

$$B_i = \{b_{i1}, b_{i2}, \cdots, b_{ij}\} \tag{26}$$

Where j the number of neighbor nodes is i can reach. This means each node faces more options when deciding where to send the packet next, making the decision process more complex.

3.3.3. Reward Design

To prevent issues like routing loops or sending packets to nodes that are unable to accept them, the agent learns to choose the optimal next hop with the help of the incentive, which promotes rapid delivery with fewer hops. If the next hop is out of range, the node's buffer is full, the packet's TTL runs out, and the agent gets a significant penalty. Normal forwarding gets a minor penalty to keep the hop count low, and reaching the destination earns a positive reward. The reward function looks like this:

$$e_{j,s} = \begin{cases} -c_1 \cdot \Gamma & \text{if next hop is out of range} \\ -c_2 \cdot \Gamma & \text{if buffer is full} \\ -c_3 \cdot \Gamma & \text{if TTL expired} \\ -c_4 \cdot \Gamma & \text{if invalid action} \\ c_5 \cdot \Gamma & \text{if next hop is the destination} \\ -c_6 \cdot \Gamma & \text{otherwise(normal forwarding)} \end{cases} \tag{27}$$

Here, Γ relates to the buffer size, and the weights c_i are set so that penalties for big mistakes are more substantial than for small ones, encouraging the agent to make wise and reliable routing choices.

As a final remark, Op-MulDRL significantly improves routing efficiency in mobile networks. The model uses a heuristic for each node, which uses the source, destination, and nearby neighbors to make decisions about the current state in relation to the node. Op-MulDRL allows nodes to make real-time, intelligent routing decisions by designing appropriate action and reward functions. The objective is to minimize hops and delays in the transmission while dealing with the node's circumventing loops in the routing, controlling the buffer flow, and mobility. In general, Op-MulDRL improves the packet transmission control in Mobile Ad-Hoc Networks (MANET), augmenting their efficiency and reliability.

3.4. Tuning DRL Parameters using the Ten-Rabo

The current techniques for optimizing the parameters of a DRL, such as random search and grid search, tend to be slow and ineffective, especially when handling the more intricate, high-dimensional aspects of the DRL [30] environments. Ten-Rabo is inspired by how rabbits cleverly adapt and escape threats in nature by keeping the search diverse and avoiding getting stuck in suboptimal solutions. DRL models optimized using Ten-Rabo learn faster, respond better to ever-changing conditions, and outperform benchmarks more consistently and stably in real-world applications. The following steps outline how the Ten-Rabo algorithm searches for the best hyperparameters.

3.4.1. Divergent Foraging

When rabbits forage, they often avoid food sources in favour of exploring more distant areas. Following the equation model, this behaviour is numerically denoted with dimension size $\boldsymbol{d}$ and the maximum number of iterations$\boldsymbol{k}$:
Candidate position update:

$$\vec{\boldsymbol{Y}}_{\boldsymbol{i}}(\boldsymbol{k}+\mathbf{1}) = \vec{\boldsymbol{Y}}_{\boldsymbol{j}}(\boldsymbol{k}) + \boldsymbol{B} \times \left(\vec{\boldsymbol{Y}}_{\boldsymbol{i}}(\boldsymbol{k}) - \vec{\boldsymbol{Y}}_{\boldsymbol{j}}(\boldsymbol{k}) + \boldsymbol{round}(\mathbf{0.5}\cdot \boldsymbol{N}_{\mathbf{1}})\right) \times \boldsymbol{s}_{\mathbf{1}} \tag{28}$$

Scaling factor:

$$\boldsymbol{B} = \boldsymbol{R} \times \boldsymbol{c} \tag{29}$$

Running Length:

$$\boldsymbol{R} = \left(\boldsymbol{e} - \boldsymbol{e}^{\left(\frac{\boldsymbol{k}-\mathbf{1}}{\boldsymbol{T}}\right)^{\mathbf{2}}}\right) \times \boldsymbol{sinsin}(\mathbf{2}\boldsymbol{\pi}\boldsymbol{N}_{\mathbf{2}}) \tag{30}$$

Random Index Permutation:

$$\boldsymbol{h} = \boldsymbol{randperm}(\boldsymbol{d}) \tag{31}$$

Stochastic component:

$$\boldsymbol{s}_{\mathbf{1}} \sim \boldsymbol{P}(\mathbf{0},\mathbf{1}) \tag{32}$$

Here, $\boldsymbol{Y}_{\boldsymbol{i}}(\boldsymbol{k}+\mathbf{1})$ and $\boldsymbol{Y}_{\boldsymbol{j}}(\boldsymbol{k})$ Are the position of rabbits in the overall population P. $\boldsymbol{randperm}$ is the random permutation, and N_1, N_2 are random numbers. $\boldsymbol{R}$ is the running length and $\boldsymbol{s}_{\mathbf{1}}$is a standard distribution.

3.4.2. Switching from Exploration to Exploitation

At first, the system is all about exploring, as the rabbits roam around, trying out different spots. However, as time goes on, they start to focus more on the promising places they have already found. This change is controlled by something called the energy factor $\boldsymbol{F}$. Early on, the energy is high, which encourages lots of exploration. As time passes, the energy drops, and the rabbits begin to slow down and move closer to the best areas. In the model, rabbits tend to wander randomly in later stages, but in the beginning, they move around in a more continuous manner. The formula below shows how the energy $\boldsymbol{F}$ changes over time to balance this behaviour:

$$\boldsymbol{F}(\boldsymbol{k}) = \mathbf{4}\left(\mathbf{1} - \frac{\boldsymbol{k}}{\boldsymbol{T}}\right) \boldsymbol{ln}\left(\frac{\mathbf{1}}{\boldsymbol{N}_{\mathbf{4}}}\right) \tag{33}$$

Here $\boldsymbol{k}$is how far along in the search, $\boldsymbol{T}$is the total time or number of steps planned, and $\boldsymbol{N}$is a random number that adds some unpredictability.

3.4.3. Random Shelter Seeking

Predators often pursue and attack rabbits, hence the rabbits form various warrens, from which the rabbits live in only one. The strategy of creating multiple vacant burrows across the terrain misleads predators, making it difficult for them to track and capture the rabbits accurately. So survival depends on digging a variety of shelter-filled burrows around the nest. A rabbit persistently constructs burrows across different regions of the search space and randomly chooses one as shelter, thereby decreasing the probability of predator capture. The strategy of random sheltering is formulated in Equations (34) to (38).

$$\boldsymbol{Y}_{\boldsymbol{i}}(\boldsymbol{k}+\mathbf{1}) = \boldsymbol{Y}_{\boldsymbol{i}}(\boldsymbol{k}) + \boldsymbol{B} \times \left(\boldsymbol{N}_{\mathbf{5}} \times \boldsymbol{a}_{\boldsymbol{i},\boldsymbol{r}}(\boldsymbol{k}) - \boldsymbol{Y}_{\boldsymbol{i}}(\boldsymbol{k})\right) \tag{34}$$

$$\boldsymbol{a}_{\boldsymbol{i},\boldsymbol{r}}(\boldsymbol{k}) = \boldsymbol{Y}_{\boldsymbol{i}}(\boldsymbol{k}) + \boldsymbol{S} \times \boldsymbol{h}_{\boldsymbol{r}}(\boldsymbol{k}) \times \boldsymbol{Y}_{\boldsymbol{i}}(\boldsymbol{k}) \tag{35}$$

$$\boldsymbol{h}_{\boldsymbol{r}}(\boldsymbol{t}) = \begin{cases} \mathbf{1}, & \boldsymbol{if}\ \boldsymbol{t} = \lfloor \boldsymbol{N}_{\mathbf{6}} \times \boldsymbol{d} \rfloor \\ \mathbf{0}, & \boldsymbol{otherwise} \end{cases} \tag{36}$$

$$\boldsymbol{S} = \frac{\boldsymbol{T}-\boldsymbol{k}+\mathbf{1}}{\boldsymbol{T}} \times \boldsymbol{s}_{\mathbf{2}} \tag{37}$$

$$\boldsymbol{s}_{\mathbf{2}} \sim \boldsymbol{P}(\mathbf{0},\mathbf{1}) \tag{38}$$

Here $\boldsymbol{a}_{\boldsymbol{i},\boldsymbol{r}}(\boldsymbol{k})$ represents a randomly chosen burrow for the $\boldsymbol{i}^{\boldsymbol{th}}$rabbit,$\boldsymbol{S}$ is the hiding parameter,$\boldsymbol{N}_{\mathbf{5}}and\boldsymbol{N}_{\boldsymbol{6}}$are random numbers between 0 and 1, and $\boldsymbol{s}_{\mathbf{2}}$ is a random value drawn from a standard normal distribution.

The Ten-Rabo algorithm brings a biological twist to hyperparameter tuning in DRL. Modelled after the unpredictable yet clever behaviors of rabbits, like scattered foraging, shifting between exploring and exploiting based on energy, and darting into shelters at random, this method strikes a solid balance between wide-ranging exploration and fine-tuned optimization. This results in fast convergence, better adaptability, and stronger performance in tricky, fast-changing DRL environments.

Updating the position $\vec{\boldsymbol{Y}}_{\boldsymbol{ten-rabo}}(\boldsymbol{k}+\mathbf{1})$ of Ten-Rabo:

$$\vec{\boldsymbol{Y}}_{\boldsymbol{ten-rabo}}(\boldsymbol{k}+\mathbf{1}) = \vec{\boldsymbol{Y}}_{\boldsymbol{rabo}}(\boldsymbol{k}) + \boldsymbol{B} \times \left(\vec{\boldsymbol{Y}}_{\boldsymbol{ten-rabo}}(\boldsymbol{k}) - \vec{\boldsymbol{Y}}_{\boldsymbol{rabo}}(\boldsymbol{k}) + \boldsymbol{round}(\mathbf{0.5}\cdot \boldsymbol{R}_{\mathbf{1}})\right) \times \boldsymbol{s}_{\mathbf{1}} \tag{39}$$

Running Length$\boldsymbol{L_{ten-rabo}}$ of Ten-Rabo:

$$\boldsymbol{L_{ten-rabo}} = \left(\boldsymbol{e} - \boldsymbol{e}^{\left(\frac{\boldsymbol{k-1}}{\boldsymbol{T}}\right)^{2}}\right) \times \boldsymbol{sinsin(2\pi R_2)} \tag{40}$$

Ten-Rabo energy $\boldsymbol{E_{ten-rabo}(k)}$ changes over time:

$$\boldsymbol{E_{ten-rabo}(k)} = \boldsymbol{4}\left(\boldsymbol{1}-\frac{\boldsymbol{k}}{\boldsymbol{T}}\right) \boldsymbol{ln}\left(\frac{\boldsymbol{1}}{\boldsymbol{R_4}}\right) \tag{41}$$

Randomly seeking shelter:

$$\boldsymbol{Y_{ten-rabo}(k+1)} = \boldsymbol{Y_{ten-rabo}(k)} + \boldsymbol{B} \times \left(\boldsymbol{R_5} \times \boldsymbol{a_{i,r}(k)} - \boldsymbol{Y_{ten-rabo}(k)}\right) \tag{42}$$

4. Experimental Validation

This section presents a thorough proportional training evaluation of the proposed model, benchmarked against several modern approaches. Experimental parameter details are given in Table 3.

Table 3. Experimental parameter details

Parameters	Values
Number of SNs	500
Number of rounds	3500
Initial energy	0.5 mJ
CH probability	5%
Packet size	512 bytes
Message size	25-50 bytes
Number of iterations	150

Figure 3 and Table 4 provide a concise evaluation of the NLT performance of the proposed model against several existing optimization methods.

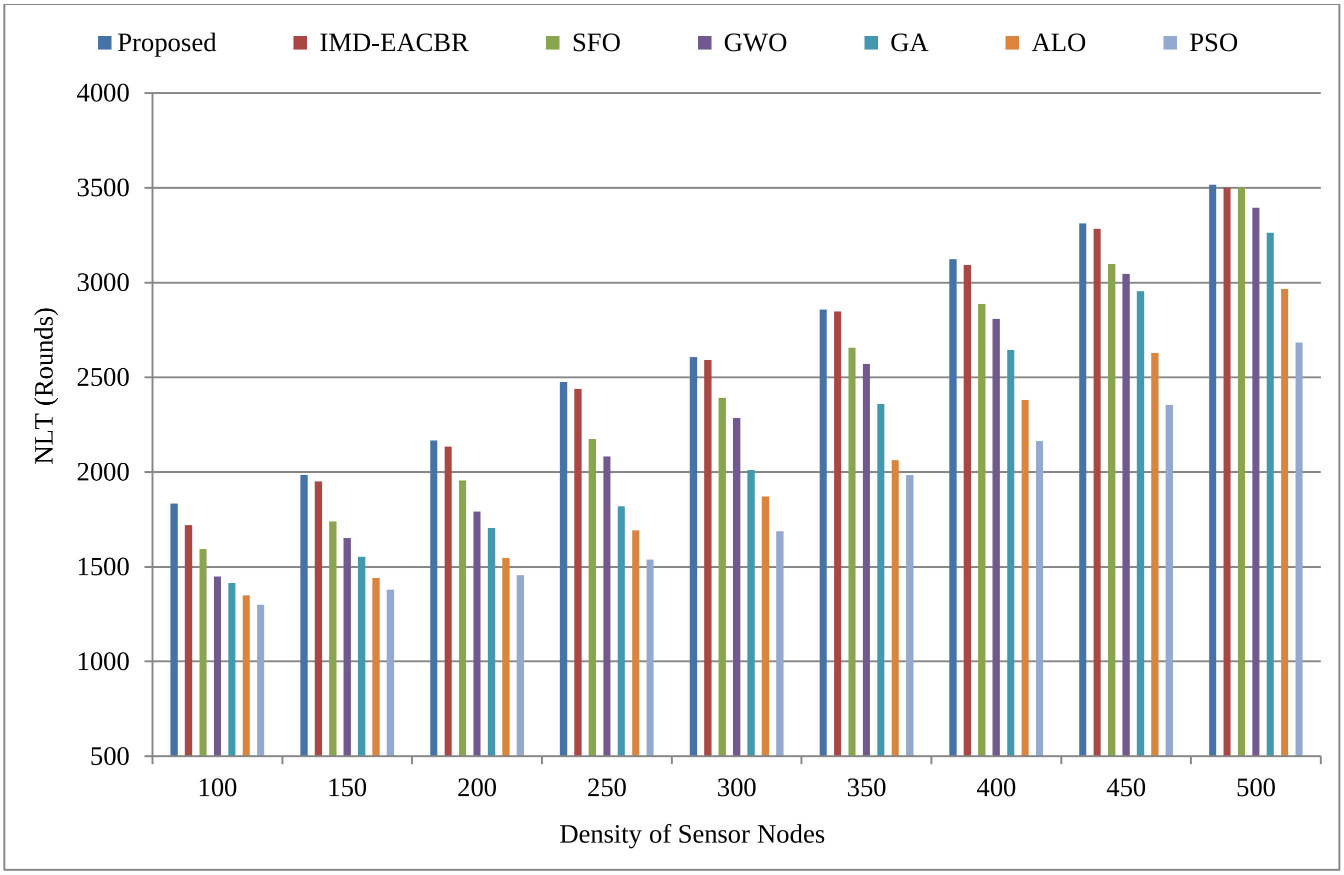

Fig. 3 NLT performance assessment

Compared with the proposed framework, IMD-EACBR achieved an NLT of 1719 rounds for 100 SNs and 3500 rounds for 500 SNs. Although IMD-EACBR incorporates energy-aware clustering, its routing mechanism does not dynamically adapt to varying network conditions. The absence of integrated multi-objective optimization and adaptive learning limits its ability to balance energy consumption effectively across nodes, resulting in slightly reduced network lifetime.

Table 4. NLT comparison

Density of SNs	Network Lifetime (Rounds)						
	IMD-EACBR	SFO Algorithm	GWO Algorithm	GA	ALO Algorithm	PSO Algorithm	Proposed Model
100	1719	1593	1448	1415	1349	1300	1834
150	1950	1739	1653	1554	1442	1380	1985
200	2135	1956	1791	1706	1547	1456	2167
250	2438	2174	2082	1818	1692	1538	2474
300	2590	2392	2286	2009	1871	1687	2606
350	2847	2656	2570	2359	2062	1984	2857
400	3092	2887	2808	2643	2379	2165	3123
450	3283	3098	3045	2953	2630	2354	3312
500	3500	3501	3395	3263	2966	2684	3517

The SFO-based method recorded 1593 rounds for 100 SNs and 3501 rounds for 500 SNs. While SFO demonstrates strong exploration capability during cluster formation, it primarily focuses on energy-based metrics and does not comprehensively integrate trust evaluation, delay awareness, and path reliability. This restricted objective formulation can lead to suboptimal cluster head distribution under dense network conditions.

GWO achieved 1448 rounds for 100 SNs and 3395 rounds for 500 SNs. Although Grey Wolf Optimization provides efficient convergence in moderate search spaces, it may suffer from premature convergence when handling multiple performance constraints simultaneously. The lack of coordinated routing optimization further contributes to uneven energy depletion among nodes. The GA-based approach yielded 1415 rounds for 100 SNs and 3263 rounds for 500 SNs. Genetic Algorithms are practical in global search; however, they may require multiple generations to stabilize and often depend heavily on parameter tuning. Without adaptive routing integration and real-time learning capability, GA-based clustering cannot entirely prevent energy imbalance in dynamic IoT scenarios.

ALO recorded 1349 rounds for 100 SNs and 2966 rounds for 500 SNs. Although Ant Lion Optimization maintains a balance between exploration and exploitation, its clustering decisions are primarily energy-driven and lack trust and delay considerations. This limitation affects routing reliability and accelerates node energy depletion. PSO achieved 1300 rounds for 100 SNs and 2684 rounds for 500 SNs, showing the lowest lifetime among the compared methods. While PSO converges rapidly, it is prone to premature convergence and may not maintain diversity in large-scale networks.

Additionally, the absence of integrated routing intelligence restricts its ability to adapt to topology changes, resulting in faster energy exhaustion. In contrast, the proposed model achieves superior network lifetime (1834 rounds for 100 SNs and 3517 rounds for 500 SNs) by selecting cluster heads using a comprehensive multi-objective formulation that incorporates residual energy, trust metrics, delay, distance, and path quality. Table 5 and Figure 4 present a comparative assessment of the Number of Surviving Active Nodes (NOSAN) across different operational rounds.

At 2000 rounds, the proposed framework retained 496 functioning nodes, indicating stronger energy preservation and balanced load distribution. In comparison, IMD-EACBR maintained 468 active nodes, reflecting moderate energy management but limited adaptability in routing coordination. SFO retained 444 nodes, while GWO sustained 404 nodes; both approaches primarily emphasize clustering efficiency but do not sufficiently integrate trust and delay awareness, leading to earlier node depletion. GA preserved 373 nodes, which may be attributed to slower convergence and less responsive adaptation to network dynamics. ALO and PSO showed comparatively lower survival counts of 295 and 278 nodes, respectively, primarily due to premature convergence and restricted multi-objective optimization capability, which accelerates uneven energy exhaustion.

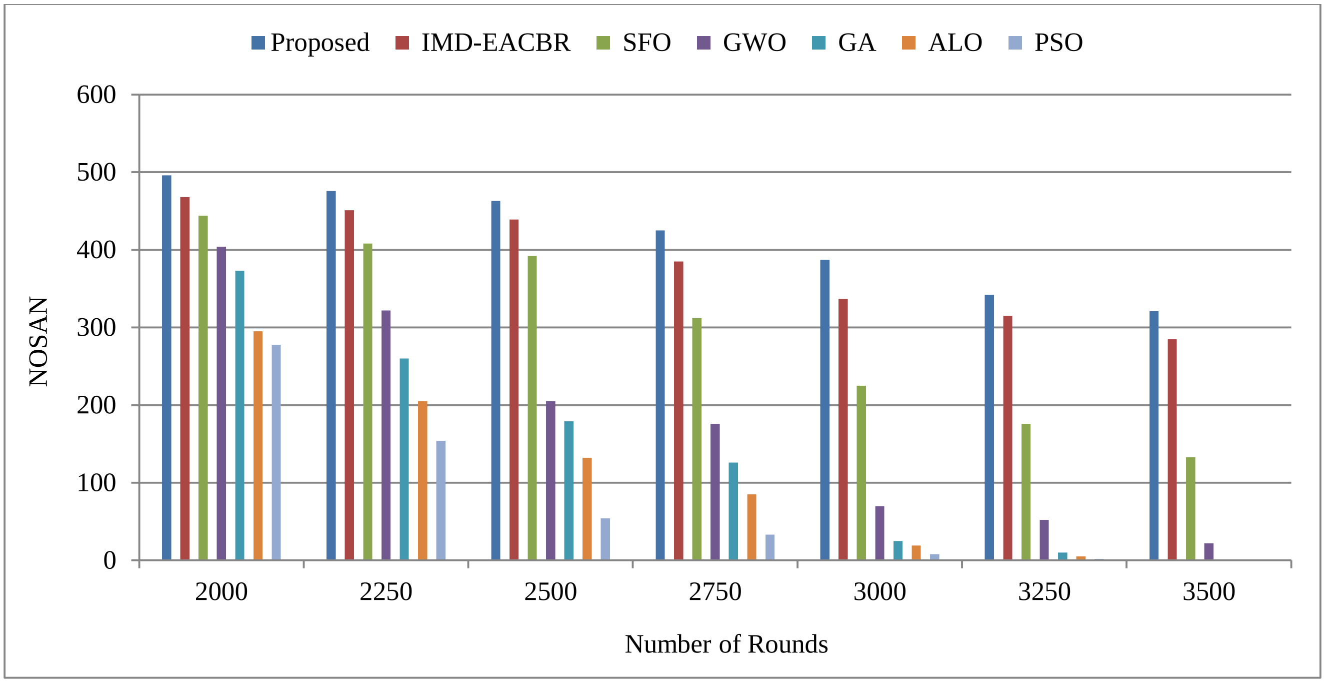

Fig. 4 NOSAN comparison

Table 5. Comparative analysis on alive nodes

No of Rounds	Number of Alive Sensor Nodes						
	IMD-EACBR	SFO Algorithm	GWO Algorithm	GA	ALO Algorithm	PSO Algorithm	Proposed model
2000	468	444	404	373	295	278	496
2250	451	408	322	260	205	154	476
2500	439	392	205	179	132	54	463
2750	385	312	176	126	85	33	425
3000	337	225	70	25	19	8	387
3250	315	176	52	10	5	2	342
3500	285	133	22	0	0	0	321

A similar pattern persists at 3250 rounds, where existing approaches exhibit a sharper decline in active nodes, whereas the proposed strategy maintains a comparatively higher survival rate. The improved performance is attributed to the integrated multi-criteria cluster head selection process combined with adaptive reinforcement learning-based routing.

The proposed framework distributes workload more evenly and reduces unnecessary energy dissipation. The factors of energy, trust, delay, and link reliability collectively contribute to prolonged node survival and enhanced operational stability over extended rounds. Table 6 and Figure 5 present the proportional evaluation of the Number of Dead Sensor Nodes (NODSN) across different methods.

At 2000 simulation rounds, the proposed framework exhibited only 21 failed nodes, indicating more effective energy regulation and balanced workload distribution. In contrast, IMD-EACBR reported 215 dead nodes, suggesting that although it applies energy-aware clustering, its lack of adaptive routing coordination accelerates node exhaustion. SFO resulted in 367 dead nodes, reflecting limited multi-criteria consideration during optimization. GWO showed 478 failed nodes, likely due to premature convergence and insufficient handling of multiple network constraints simultaneously.

GA, ALO, and PSO each reached 500 dead nodes, implying faster energy depletion caused by restricted objective modeling and the absence of integrated learning-based routing adaptation. The comparatively lower mortality rate of the proposed model can be attributed to its coordinated multi-objective cluster head selection, incorporation of trust and delay awareness, and adaptive reinforcement learning-based routing strategy, all of which collectively minimize uneven energy dissipation and prolong network sustainability. The Energy Consumption per Message (ECM) for the proposed framework and benchmark algorithms is given in Table 7 and Figure 6.

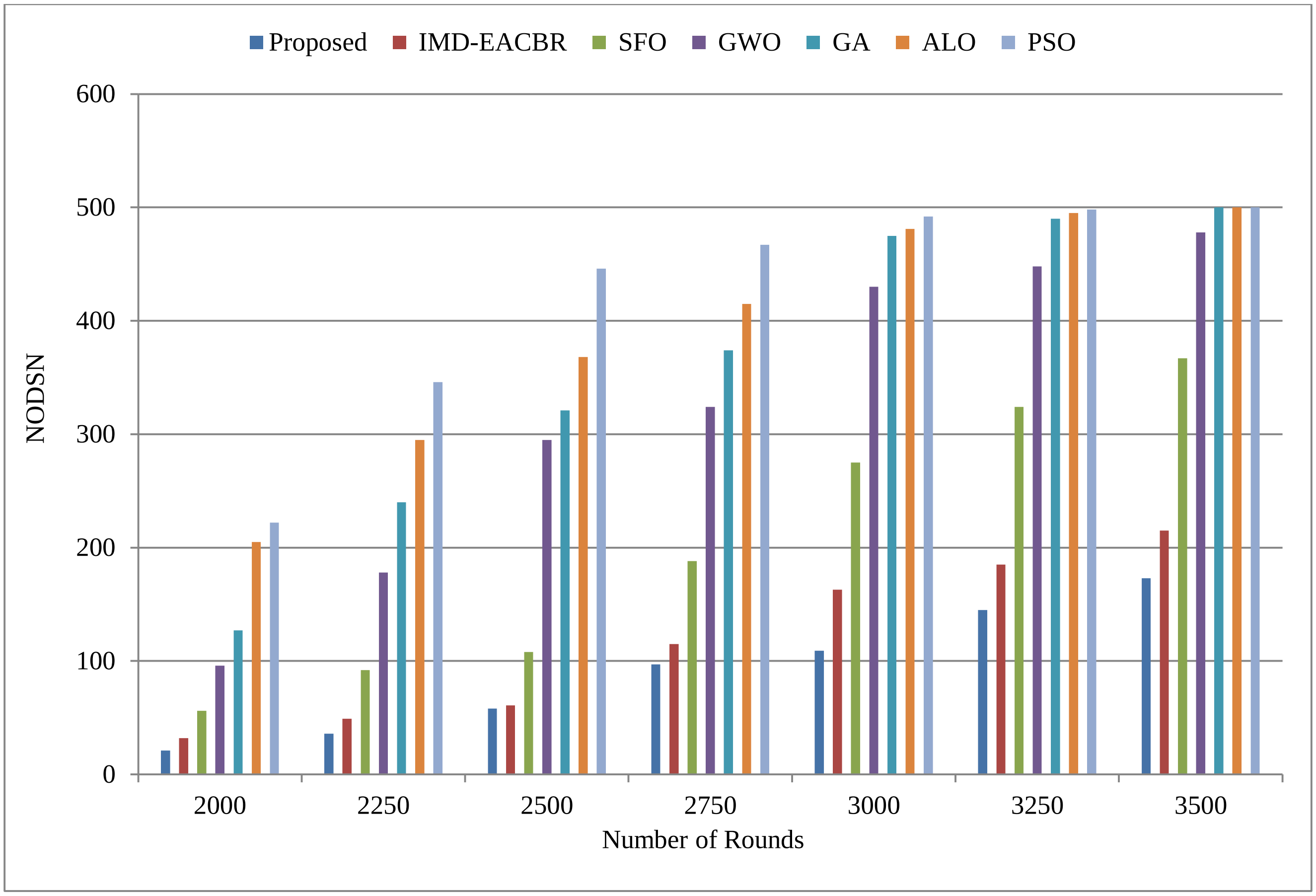

Fig. 5 NODSN representation

Table 6. Proportional evaluation of the NODSN

No of Rounds	Number of Dead Sensor Nodes						
	IMD-EACBR	SFO	GWO	GA	ALO	PSO	Proposed model
2000	32	56	96	127	205	222	21
2250	49	92	178	240	295	346	36
2500	61	108	295	321	368	446	58
2750	115	188	324	374	415	467	97
3000	163	275	430	475	481	492	109
3250	185	324	448	490	495	498	145
3500	215	367	478	500	500	500	173

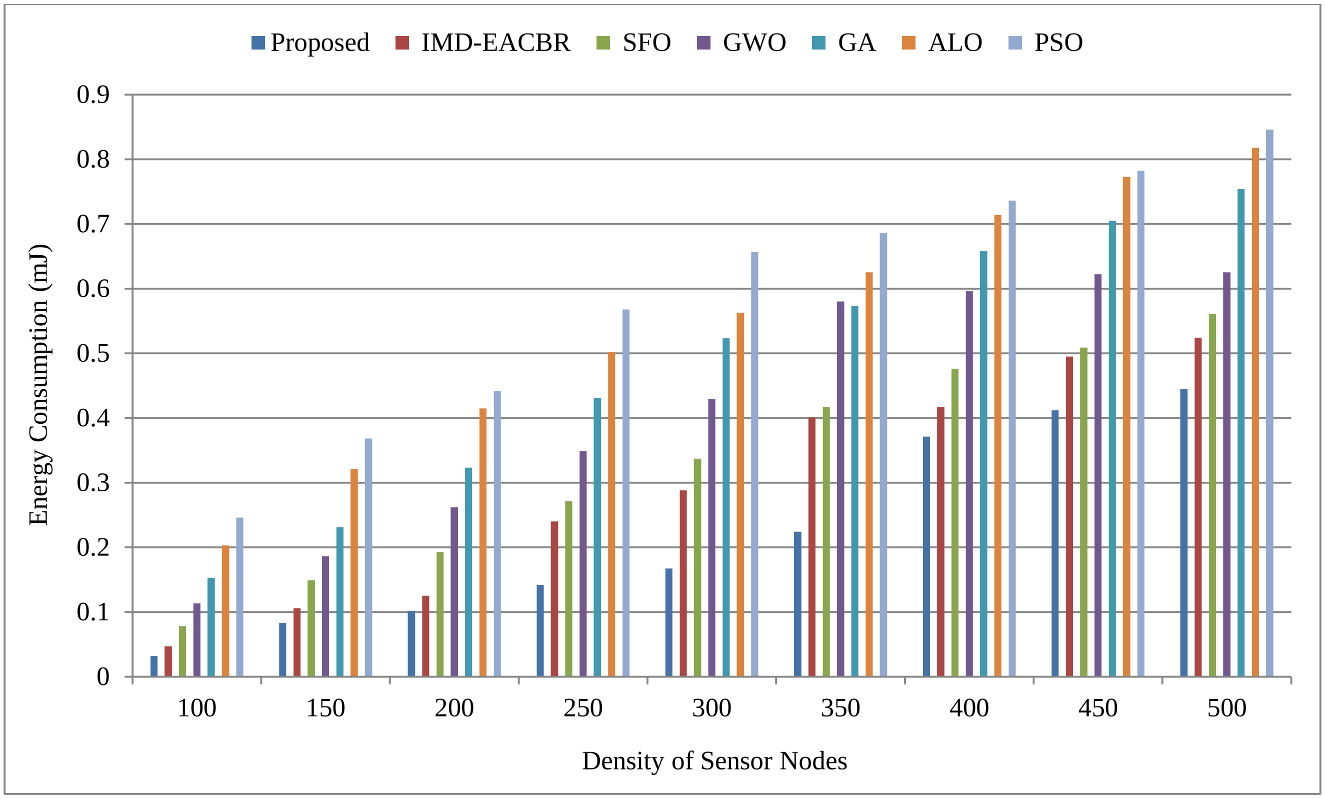

Fig. 6 Energy consumption comparison

Table 7. Energy consumption analysis

Density of SNs	**Energy Consumption (mJ)**						
	IMD-EACBR	**SFO**	**GWO**	**GA**	**ALO**	**PSO**	**Prop. model**
100	0.047	0.078	0.113	0.153	0.203	0.246	0.032
150	0.106	0.149	0.186	0.231	0.321	0.368	0.083
200	0.125	0.193	0.262	0.323	0.415	0.442	0.102
250	0.24	0.271	0.349	0.431	0.502	0.568	0.142
300	0.288	0.337	0.429	0.523	0.563	0.657	0.167
350	0.401	0.417	0.58	0.573	0.625	0.686	0.224
400	0.417	0.476	0.596	0.658	0.714	0.736	0.371
450	0.495	0.509	0.622	0.705	0.773	0.782	0.412
500	0.524	0.561	0.625	0.754	0.818	0.846	0.445

The proposed approach recorded an ECM of 0.032 mJ, indicating more efficient energy utilization during data transmission for a network size of 100 sensor nodes. In comparison, IMD-EABCR consumed 0.047 mJ, reflecting moderate optimization but limited coordination between clustering and routing. SFO reported 0.078 mJ, as its optimization primarily emphasizes exploration without fully integrating communication efficiency metrics. GWO resulted in 0.113 mJ, potentially due to premature convergence affecting balanced energy allocation. GA showed 0.153 mJ, which may be attributed to iterative overhead and less adaptive decision-making under dynamic conditions. ALO and PSO demonstrated even higher consumption levels of 0.203 mJ and 0.246 mJ, respectively, likely due to restricted multi-objective consideration and lack of intelligent routing adaptation. The lower ECM achieved by the proposed model is due to its integrated multi-criteria cluster head selection, trust- and delay-aware routing strategy, and adaptive learning-based optimization, which collectively reduce redundant transmissions and improve overall energy efficiency. The proposed model achieved a throughput of 0.987 Mbps, indicating more efficient and reliable data delivery as given in Table 8 and Figure 7. The IMD-EACBR recorded 0.975 Mbps, reflecting reasonable performance but limited adaptability in dynamic routing adjustments.

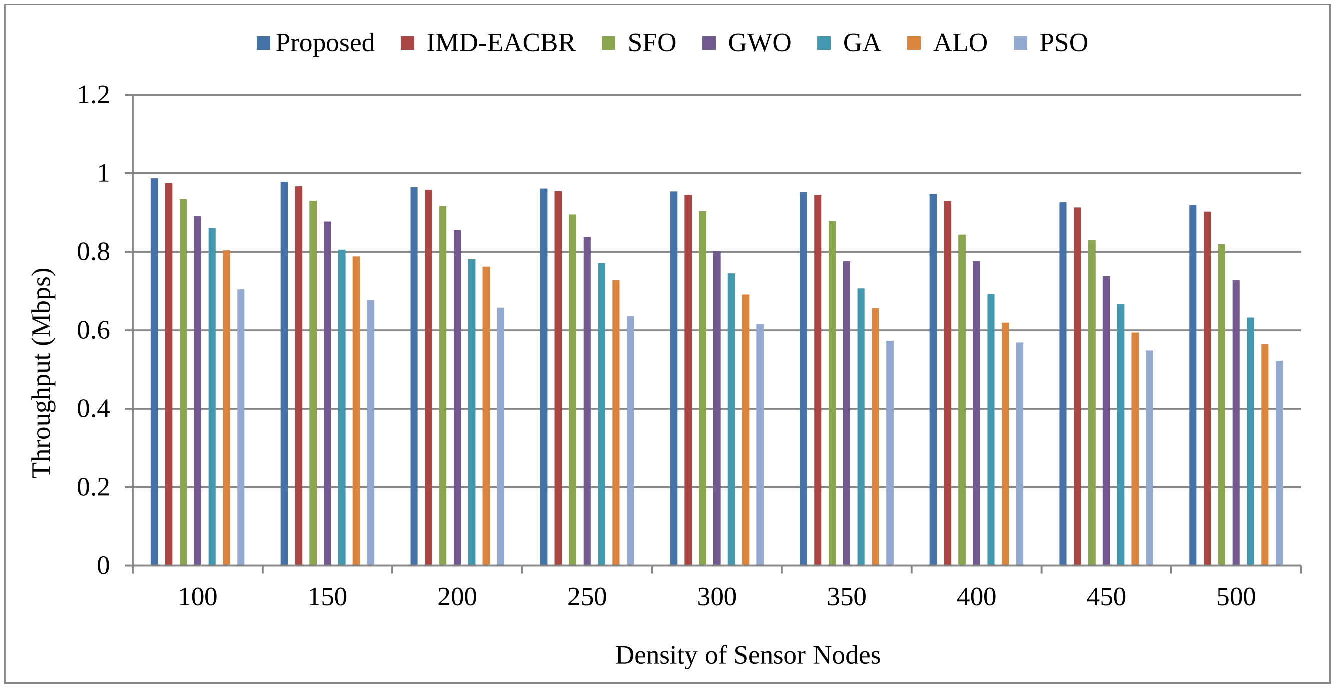

Fig. 7 Throughput performance comparison

Table 8. Throughput performance assessment

Density of SNs	**Throughput (Mbps)**						
	IMD-EACBR	**SFO**	**GWO**	**GA**	**ALO**	**PSO**	**Prop. model**
100	0.975	0.934	0.891	0.861	0.804	0.704	0.987
150	0.967	0.93	0.877	0.805	0.788	0.677	0.978
200	0.958	0.916	0.855	0.781	0.762	0.658	0.964
250	0.955	0.895	0.838	0.771	0.728	0.636	0.961
300	0.945	0.903	0.801	0.745	0.691	0.616	0.954
350	0.945	0.878	0.776	0.707	0.656	0.573	0.952
400	0.929	0.844	0.776	0.692	0.619	0.569	0.947
450	0.913	0.83	0.738	0.667	0.594	0.548	0.926
500	0.902	0.819	0.728	0.632	0.565	0.522	0.919

SFO and GWO achieved 0.934 Mbps and 0.891 Mbps, respectively, where reduced throughput may be linked to less coordinated multi-objective optimization during cluster formation and routing. GA produced 0.728 Mbps, potentially due to slower convergence and suboptimal load balancing. ALO and PSO demonstrated comparatively lower throughput values of 0.632 Mbps and 0.522 Mbps, respectively, which can be attributed to premature convergence and insufficient integration of routing intelligence. The higher throughput of the proposed model is supported by its joint clustering–routing framework, which minimizes packet loss and congestion while ensuring balanced traffic distribution. Similarly, Table 9 and Figure 8 compare the Packet Delivery Ratio (PDR) across different approaches.

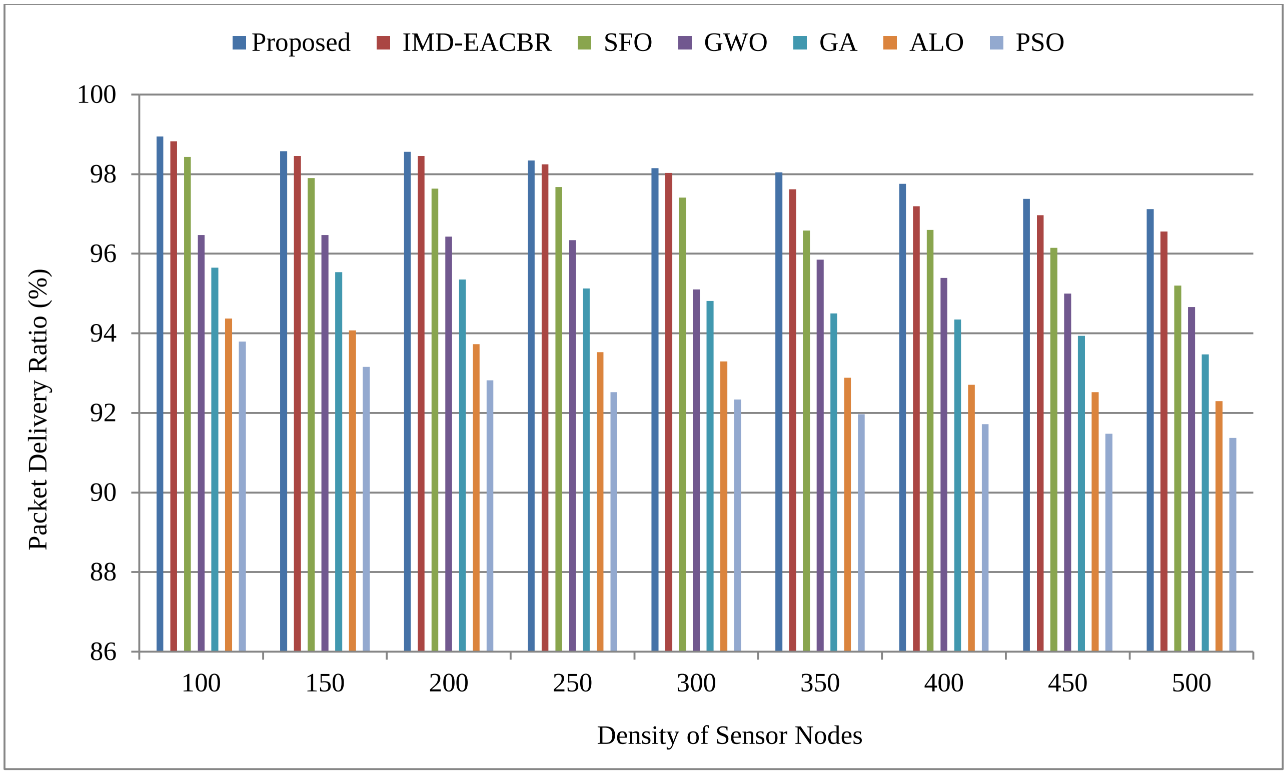

Fig. 8 PDR comparative analysis

Table 9. PDR performance evaluation

Density of SNs	**Packet Delivery Ratio (%)**						
	IMD-EACBR	**SFO**	**GWO**	**GA**	**ALO**	**PSO**	**Prop. model**
100	98.83	98.43	96.47	95.65	94.37	93.79	98.95
150	98.46	97.9	96.47	95.54	94.07	93.16	98.58
200	98.46	97.64	96.43	95.35	93.73	92.82	98.56
250	98.25	97.68	96.34	95.13	93.53	92.52	98.34
300	98.03	97.41	95.10	94.81	93.29	92.34	98.15
350	97.62	96.58	95.85	94.5	92.88	91.97	98.05
400	97.19	96.60	95.39	94.35	92.71	91.72	97.76
450	96.97	96.15	95.00	93.94	92.52	91.48	97.38
500	96.56	95.20	94.66	93.47	92.3	91.37	97.12

At 100 sensor nodes, the proposed strategy attained a PDR of 98.95%, indicating highly reliable packet transmission. IMD-EACBR followed closely with 98.83%, while SFO and GWO achieved 98.40% and 96.47%, respectively, suggesting moderate reliability influenced by less comprehensive trust and delay considerations. GA recorded 95.65%, which may be due to uneven energy dissipation affecting link stability. ALO and PSO achieved 94.37% and 93.79%, respectively, reflecting increased packet loss due to limited adaptive routing capability. The improved PDR of the proposed model can be attributed to its integration of trust-aware cluster head selection, delay-sensitive routing decisions, and adaptive learning mechanisms that collectively enhance communication reliability and network stability. The cost function values obtained using different optimization strategies are tabulated in Table 10 and Figure 9.

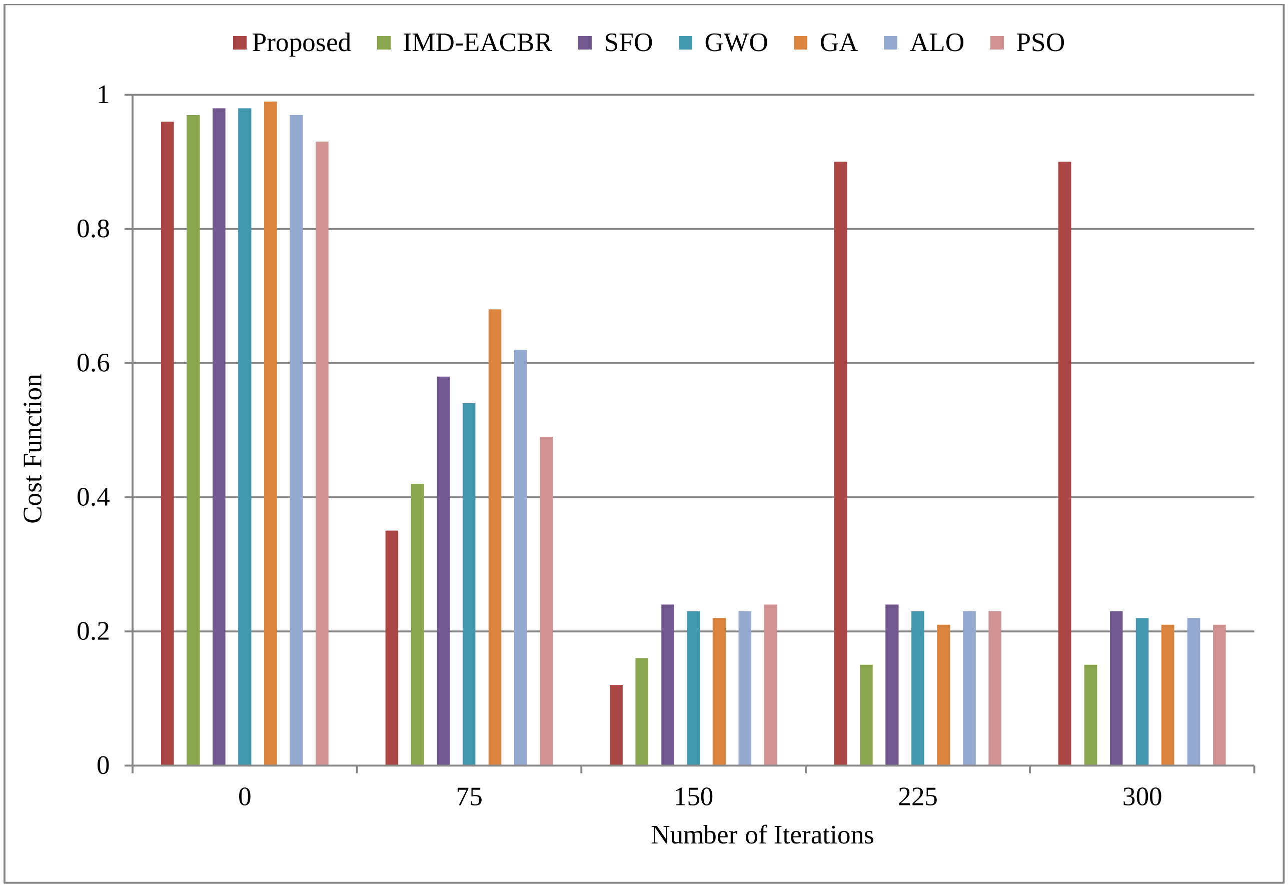

Fig. 9 Cost function comparison

Table 10. Cost function evaluation

Number of Iterations	Cost Function						
	IMD-EACBR	SFO	GWO	GA	ALO	PSO	Proposed model
0	0.97	0.98	0.98	0.99	0.97	0.93	0.96
75	0.42	0.58	0.54	0.68	0.62	0.49	0.35
150	0.16	0.24	0.23	0.22	0.23	0.24	0.12
225	0.15	0.24	0.23	0.21	0.23	0.23	0.9
300	0.15	0.23	0.22	0.21	0.22	0.21	0.9

The Genetic Algorithm (GA) exhibited comparatively higher cost values as the problem scale increased, indicating reduced efficiency in handling complex search spaces. Several other techniques demonstrated closely comparable performance trends; however, their cost reductions were not consistently stable across varying conditions. Grey Wolf Optimization (GWO) achieved relatively competitive results, reflecting effective exploration and convergence characteristics. Nevertheless, the proposed framework consistently produced the lowest cost values among all methods, highlighting its stronger optimization capability. The improved performance can be attributed to its balanced exploration–exploitation mechanism and integrated multi-objective formulation, which enable more accurate and stable convergence toward optimal solutions.

4.1. Discussion

Table 11 Provides a Performance Comparison Analysis of proposed vs. existing techniques.

Table 11. Performance comparison analysis of proposed vs Existing techniques

References	Performances		
	Energy Consumption (mJ)	Throughput (Mbps)	PDR (%)
Divya et al. [16]	29.45	23.18	-
Jecan et al. [17]	32.20	-	98.40
Shakya et al. [18]	32.80	22.40	97.6
Jalalinejad et al. [19]	41.70	35	99.10
Jibreel et al. [20]	38.5	28	98.70
Li et al. [21]	26.9	18.30	96.10
Ramamoorthi et al. [22]	21.30	12.70	97.20
Raslan et al. [23]	33.60	24.50	98.50
Cherappa et al. [24]	37.90	30.20	98.90
Badiger et al. [25]	27.40	19.60	97.90
Proposed	18.06	38.80	99.1

The majority of the current methods, including those by Divya et al. [16], Shakya et al. [18], Li et al. [21], and Badiger et al. [25], center more on the residual energy in the Cluster Head (CH) selection. This will minimize the immediate energy depletion, but it does not guarantee a balanced distribution of energy throughout the network. Conversely, the proposed framework considers residual energy, trust realization, communication delay, distance to the base station, and quality of the path, based on RSSI, altogether as a single factor in the form of Hy-WhiOp fitness.

This multi-criteria analysis will ensure that high-energy nodes are not selected repeatedly and will enable the workload distribution among sensor nodes to be more uniform, which removes early CH exhaustion and decreases the total energy usage. This has been possible through the synchronized integration of clustering and adaptive routing, resulting in improved throughput. Other available studies, such as Jalalinejad et al. [19] and Cherappa et al. [24], focus on clustering efficiency but use a heuristic or, rather, fixed routing mechanism.

This kind of isolation between clustering and routing can frequently lead to congestion, inefficient path choice, and retransmission of packets in a dynamic network environment. The suggested Op-MulDRL model would allow the real-time adaptive routing, which is constantly learning the network state in terms of node locations, occupancy of buffers, and information about their neighbors. The reward functionality is well structured to discourage a high number of hops, congestion, routing loops, and buffer overflow, and therefore results in a more stable path choice and an enhanced efficiency of data forwarding. Consequently, increased throughput is realized compared to current metaheuristic-based as well as conventional routing strategies.

Likewise, both trust-aware CH selection and link-quality-based routing are combined to improve the Packet Delivery Ratio (PDR). Some of the studies, like Shakya et al. [18] and Ramamoorthi et al. [22], include security mechanisms, but they either impose computational overhead or use complicated blockchain architecture that influences scalability.

In the proposed framework, the direct and indirect measures of trust are combined into the CH selection, so that the unreliable or malicious nodes have lower chances to be elected as CHs. Also, the RSSI-based path quality assessment increases the reliability of the links in the routing process, reducing the number of packets with low or bad connections. This two-layered reliability mechanism has a positive contribution to the attainment of higher PDR values, which are consistently higher in different network densities. The other notable aspect of improved performance is the hyperparameter optimization of the DRL model based on Ten-Rabo. The routing methods based on learning are usually susceptible to volatile convergence and sensitivity to parameters, which may deteriorate scalability and flexibility. The Ten-Rabo algorithm improves the exploration-exploitation balance in the DRL training process, which helps to achieve faster convergence and increased stability in learning. This leads to a regular routing decision and avoids the oscillatory nature of IoT-enabled dynamic WSN environment routing.

Moreover, the hybridity of Hy-WhiOp enhances the global search potential and mitigates the risk of premature convergence, which is usually experienced in independent metaheuristic strategies like ISFO [23] and ASFO [24]. The algorithm achieves diversity by integrating the spiral search dynamics with the adaptive movement by the use of velocity, thus refining optimal solutions in an effective manner.

Comprehensively, the above-noted enhancements in energy consumption, throughput, and PDR are a product of a holistic framework that concurrently optimises multi-objective CH selection, adaptive routing, and learning stability. In contrast to the current methods that either deal with clustering and routing individually or emphasize one objective optimization, the proposed model combines energy balancing, trust awareness, delay sensitivity, link reliability, and additional smart hyperparameter tuning in the same architecture. This synchronized design will guarantee balance in energy loss, minimum congestion, enhanced reliability in communication, and longer network lifetime under dynamic IoT-enabled WSN.

5. Conclusion

This paper presented a hybrid framework that integrates Hy-WhiOp with Op-MulDRL, which offers significant improvements in energy efficiency, routing resilience, and NLT. The framework has some drawbacks, such as needing more computing power, slower real-time response in very dynamic environments, and the assumption that all nodes are the same. Future research will seek to address these issues by improving scalability through additional support for mobile sinks and more complex node types, as well as lightweight optimization strategies. These enhancements aim to enable the system to handle larger and more dynamic network topologies while maintaining acceptable latency and energy efficiency. Special focus will be placed on developing adaptive routing mechanisms that can intelligently respond to sink mobility patterns and heterogeneous node capabilities without introducing significant computational overhead. Furthermore, efforts will explore the integration of edge-based decision-making and selective data aggregation techniques to reduce communication costs in resource-constrained environments further. Ultimately, these advancements are expected to make the approach more practical for real-world deployments involving massive-scale sensor networks, underwater monitoring systems, and disaster-response scenarios.

References

[1] Chang Lei, “An Energy-Aware Cluster-Based Routing in the Internet of Things using Particle Swarm Optimization Algorithm and Fuzzy Clustering,” *Journal of Engineering and Applied Science*, vol. 71, pp. 1-25, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Mahyar Sadrishojaei et al., “An Energy-Aware IoT Routing Approach based on a Swarm Optimization Algorithm and a Clustering Technique,” *Wireless Personal Communications*, vol. 127, pp. 3449-3465, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Kuruva Lakshmanna et al., “Improved Metaheuristic-Driven Energy-Aware Cluster-based Routing Scheme for IoT-Assisted Wireless Sensor Networks,” *Sustainability*, vol. 14, no. 13, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sasikumar Gurumoorthy et al., “Optimal Cluster Head Selection in WSN with Convolutional Neural Network-based Energy Level Prediction,” *Sensors*, vol. 22, no. 24, pp. 1-23, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Mandli Rami Reddy et al., “Energy-Efficient Cluster Head Selection in Wireless Sensor Networks using an Improved Grey Wolf Optimization Algorithm,” *Computers*, vol. 12, no. 2, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Haider Ali et al., “ARSH-FATI: A Novel Metaheuristic for Cluster Head Selection in Wireless Sensor Networks,” *IEEE Systems Journal*, vol. 15, no. 2, pp. 2386-2397, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] R. Ramya, and T. Brindha, “A Comprehensive Review on Optimal Cluster Head Selection in WSN-IoT,” *Advances in Engineering Software*, vol. 171, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Pramod Singh Rathore et al., “Energy-Efficient Cluster Head Selection through Relay Approach for WSN,” *The Journal of Supercomputing*, vol. 77, pp. 7649-7675, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ramasubbareddy Somula, Yongyun Cho, and Bhabendu Kumar Mohanta, “EACH-COA: An Energy-Aware Cluster Head Selection for the Internet of Things using the Coati Optimization Algorithm,” *Information*, vol. 14, no. 11, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ke Zhang et al., “Clustering the Sensor Networks based on Energy-Aware Affinity Propagation,” *Computer Networks*, vol. 207, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Péter Ekler, János Levendovszky, and Daniel Pasztor, “Energy Aware IoT Routing Algorithms in Smart City Environment,” *IEEE Access*, vol. 10, pp. 87733-87744, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Raheleh Samadi, Amin Nazari, and Jochen Seitz, “Intelligent Energy-Aware Routing Protocol in Mobile IoT Networks based on SDN,” *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 4, pp. 2093-2103, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Romany F. Mansour, "Energy Aware Fault Tolerant Clustering with Routing Protocol for Improved Survivability in Wireless Sensor Networks," *Computer Networks*, vol. 212, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Ramasubbareddy Somula, Yongyun Cho, and Bhabendu Kumar Mohanta, "SWARAM: Osprey Optimization Algorithm-based Energy-Efficient Cluster Head Selection for Wireless Sensor Network-based Internet of Things," *Sensors*, vol. 24, no. 2, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Fatma H. El-Fouly et al., "ERCP: Energy-Efficient and Reliable-Aware Clustering Protocol for Wireless Sensor Networks," *Sensors*, vol. 22, no. 22, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] P. Divya, and B. Sudhakar, "Route Optimization and Optimal Cluster Head Selection for Cluster-Oriented Wireless Sensor Network Utilizing Circle-Inspired Optimization Algorithm," *International Journal of Computational Intelligence Systems*, vol. 17, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Eusebiu Jecan et al., "Predictive Energy-Aware Routing Solution for Industrial IoT Evaluated on a WSN Hardware Platform," *Sensors*, vol. 22, no. 6, pp. 1-22, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Harish Kumar Shakya et al., "Energy-Proficient Cluster Enrichment in Wireless Sensor Networks via Categorized Fuzzy Clustering and Multi-Hop Routing Optimization," *SN Computer Science*, vol. 6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Hoda Jalalinejad et al., "A Hybrid Multi-Hop Clustering and Energy-Aware Routing Protocol for Efficient Resource Management in Renewable Energy Harvesting Wireless Sensor Networks," *IEEE Access*, vol. 12, pp. 137310-137332, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Fuseini Jibreel, Emmanuel Tuyishimire, and Mohammed Ibrahim Daabo, "An Enhanced Heterogeneous Gateway-Based Energy-Aware Multi-Hop Routing Protocol for Wireless Sensor Networks," *Information*, vol. 13, no. 4, pp. 1-15, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Luyao Li, Yang Qiu, and Jing Xu, "A K-Means Clustered Routing Algorithm with Location and Energy Awareness for Underwater Wireless Sensor Networks," *Photonics*, vol. 9, no. 5, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] S. Ramamoorthi, B. Muthu Kumar, and Ahilan Appathurai, "Energy Aware Clustered Blockchain Data for IoT: An End-to-End Lightweight Secure & Enroute Filtering Approach," *Computer Communications*, vol. 202, pp. 166-182, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Aliaa F. Raslan et al., "An Improved Sunflower Optimization Algorithm for Cluster Head Selection in the Internet of Things," *IEEE Access*, vol. 9, pp. 156171-156186, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Venkatesan Cherappa et al., "Energy-Efficient Clustering and Routing using ASFO and a Cross-Layer-Based Expedient Routing Protocol for Wireless Sensor Networks," *Sensors*, vol. 23, no. 5, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Vani S. Badiger, and T.S. Ganashree, "Data Aggregation Scheme for IoT-based Wireless Sensor Network through Optimal Clustering Method," *Measurement: Sensors*, vol. 24, pp. 1-7, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Seyedehzahra Mirjalili et al., *Whale Optimization Algorithm: Theory, Literature Review, and Application in Designing Photonic Crystal Filters*, Nature-Inspired Optimizers: Theories, Literature Reviews and Applications, pp. 219-238, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[27] Ghassan Husnain, and Shahzad Anwar, "An Intelligent Cluster Optimization Algorithm based on Whale Optimization Algorithm for VANETs (WOACNET)," *PLoS One*, vol. 16, no. 4, pp. 1-22, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[28] Ersin Korkmaz, Erdem Doğan, and Ali Payıdar Akgüngör, "Energy Demand Estimation in Turkey According to Road and Rail Transportation: Walrus Optimizer and White Shark Optimizer Algorithm-Based Model Development and Application," *Energies*, vol. 17, no. 19, pp. 1-23, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[29] Wai-xi Liu et al., "DRL-R: Deep Reinforcement Learning Approach for Intelligent Routing in Software-Defined Data-Center Networks," *Journal of Network and Computer Applications*, vol. 177, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] Joo-Hyung Park, Qin Yang, and Sang-Jo Yoo, "RHRA-DRL: RSU-Assisted Hybrid Road-Aware Routing using Distributed Reinforcement Learning in Internet of Vehicles," *IEEE Access*, vol. 12, pp. 25385-25396, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[31] Thavavel Vaiyapuri, and Haya Alaskar, "Whale Optimization for Wavelet-Based Unsupervised Medical Image Segmentation: Application to CT and MR Images," *International Journal of Computational Intelligence Systems*, vol. 13, pp. 941-953, 2020. [CrossRef] [Google Scholar] [Publisher Link]