

A GENETIC ALGORITHM-BASED ADAPTIVE COVERAGE METHOD USING HEXAGONAL AREAS FOR WSN WITH DEEP REINFORCEMENT LEARNING

Shauban Ali Solangi^{*1}, Zahoor Hussain Shah², Syed Raza Hussain³, Khalid Nooruddin Charan⁴

^{*1,2,3}Department of Computer Science, Hyderabad Institute for Technology and Management Sciences (HITMS), Sindh, Pakistan

⁴Department of Computing, Hamdard University, Karachi, Sindh, Pakistan

^{*1}dr.shauban@faculty.hitms.edu.pk/shauban76@gmail.com

DOI: <https://doi.org/10.5281/zenodo.23276537>

Keywords

Genetic Algorithm, Wireless Sensor Network, Deep Reinforcement Learning, Coverage Optimization, Machine Learning, Hexagonal Tessellation.

Article History

Received: 25 June 2026

Accepted: 16 September 2026

Published: 10 October 2026

Copyright @Author

Corresponding Author: *

Shauban Ali Solangi

Abstract

Coverage optimization remains a central challenge in Wireless Sensor Network (WSN) design, directly shaping both monitoring quality and overall network performance. Because sensor nodes operate under strict energy budgets, prolonging network lifetime without sacrificing sensing coverage has become a pressing design objective. Static deployment strategies, however, tend to degrade in irregular terrains and dynamic operating conditions, where node failures and shifting energy levels continually alter the coverage landscape. This paper proposes a hybrid optimization framework for adaptive WSN coverage that combines a real-coded Genetic Algorithm, a machine-learning fitness surrogate, and deep reinforcement learning within a hexagonal cell-based coverage model. The hexagonal tessellation provides a geometrically efficient basis for evaluating sensing redundancy and coverage gaps, while the genetic algorithm performs global search over node placement to jointly minimize overlap, movement cost, and connectivity loss; a lightweight neural network surrogate reduces the computational burden of repeated fitness evaluation during this search. Beyond the initial deployment, a parameter-shared deep Q-network enables individual nodes to respond autonomously to coverage holes created by node failure, without requiring the global optimizer to be re-executed. A deterministic repair mechanism further guarantees full sensing coverage and network-wide connectivity to a hybrid fixed-and-mobile sink infrastructure, closing the gap between soft, learning-based optimization and hard operational requirements. Simulation results on a 160 m × 160 m field with 100 randomly deployed nodes show that the proposed framework sustains high coverage ratios under node attrition, recovers connectivity after failures without full redeployment, and scales effectively to dense sensor networks.

INTRODUCTION

Wireless Sensor Networks (WSNs) must continue sensing a monitored field even as node batteries

deplete and individual units fail over time [1], [4]. A one-shot, offline deployment optimum produced by a Genetic Algorithm (GA) or Particle Swarm

Optimizer (PSO) is only valid at the instant it is computed; it goes stale the moment the first node dies or drifts out of range. At the same time, re-running a full global optimizer after every failure is computationally far too slow for a live network that must react within seconds to preserve monitoring continuity. This tension between the quality of offline, population-based search and the responsiveness required for online fault recovery motivates the present work.

Our design goal is therefore to pair a strong OFFLINE optimizer, which produces a high-quality initial layout, with a fast ONLINE agent that reacts to failures in real time without invoking the offline optimizer again. We adopt a hexagonal coverage grid as the underlying evaluation model: a hexagonal cell is counted as “covered” once at least one sensor lies within its sensing radius R_s of the cell centre. Hexagonal tessellation is geometrically efficient for evaluating sensing redundancy and coverage gaps because, unlike square grids, it minimizes the maximum distance from any point in a cell to its centre for a given cell area, giving a tighter and more realistic coverage estimate.

This paper makes the following contributions:

- Stage 1 – Genetic repositioning: a real-coded GA searches per-node displacements that maximise hexagonal coverage under joint overlap, movement-cost, and connectivity penalties.
- Stage 1b – ML-surrogate acceleration: a lightweight feed-forward neural network learns to predict GA fitness, cutting the number of expensive full-geometry evaluations required during the search.
- Stage 2 – Online DQN adaptation: a parameter-shared Deep Q-Network lets every surviving node react to failures and the coverage holes they create, without ever re-running the GA.
- A deterministic repair mechanism that guarantees full sensing coverage and connectivity to a hybrid fixed-and-mobile sink infrastructure, bridging learning-based optimization with hard operational guarantees.

The remainder of this paper is organized as follows. Section II reviews related work on metaheuristic coverage optimization, deep-reinforcement-learning-based adaptation, and surrogate-assisted evolutionary search. Section III describes the system

model. Section IV presents the proposed three-stage methodology. Section V details the simulation setup, and Section VI reports and discusses the results. Section VII outlines limitations and future work, and Section VIII concludes the paper.

II. RELATED WORK

A. Metaheuristic Coverage Optimization

Coverage optimization in WSNs has been studied extensively using metaheuristic search. Sun et al. [1] combined a genetic algorithm with a reinforced whale optimization algorithm (GARWOA) to balance global exploration and local refinement, reporting coverage ratios above 95% on standard test fields. Zhang et al. [2] proposed a desert-golden-mole-inspired optimizer that uses sand-swimming and hiding strategies to escape local optima while improving deployment uniformity. Working from a computational-geometry perspective, Bettini et al. [3] formulated sensor placement as an Integer Linear Program under line-of-sight constraints and used a GA to refine near-optimal covering sets. Zhang et al. [4] proposed a Modified Marine Predator Algorithm that outperformed several GA- and swarm-based baselines across multiple coverage scenarios, while a multi-strategy Improved Northern Goshawk Optimizer [5] and an improved Cuckoo Search with multi-strategy dimensional updates [6] have similarly pushed reported coverage rates upward. Chaotic variants of Grey Wolf Optimization [7] have been used to jointly maximize coverage and connectivity. These studies confirm that hybrid and chaos-enhanced metaheuristics consistently outperform a plain GA, which motivates our use of a surrogate-accelerated GA rather than a slower multi-population hybrid, while keeping the search real-valued and displacement-based for direct compatibility with our hexagonal evaluation grid.

B. Deep Reinforcement Learning for Adaptive Coverage

A separate line of work targets the post-deployment, dynamic phase of WSN operation using deep reinforcement learning (DRL). Li et al. [8] proposed ACDRL, a DRL framework that adapts sensor behaviour under node failure and reports superior coverage retention over state-of-the-art baselines at

scale. Rani et al. [9] used reinforcement learning to restore connectivity after multiple node failures by repositioning relay nodes. More recent work has explored multi-agent and vision-based DRL for mobile WSN coverage [10], graph-neural-network-augmented DRL that exploits topological structure in the reward signal [11], and federated DRL formulations (FEDQN) that reduce inter-node communication overhead during online target-coverage optimization [12]. Compared with these approaches, our Stage-2 agent uses a compact five-feature local state and a parameter-shared architecture so that a single trained policy generalizes across all nodes, avoiding the training cost of per-node or fully centralized value functions while remaining reactive at the timescale of individual node failures.

C. Surrogate-Assisted Evolutionary Search

Because fitness evaluation in geometry-heavy problems such as hexagonal coverage scoring is comparatively expensive, surrogate-assisted evolutionary algorithms (SAEAs) have been proposed to reduce the number of exact evaluations per generation. Khaldi and Draa [13] survey absolute and relative fitness-approximation models, including regression-, kriging-, and classification-based surrogates. Complementary work has combined surrogate models with self-adaptive mutation [14] and with autoencoder-based state compression in evolutionary reinforcement learning [15], both reporting faster convergence at equal computational

budget. Our Stage-1b surrogate follows this line but uses an online-trained, replay-buffer-fed feed-forward network scored against only 35% of each generation, striking a balance between prediction accuracy and the cost of retraining at every generation.

Taken together, prior work has addressed offline coverage optimization, online failure adaptation, and evaluation-cost reduction largely in isolation. The present paper integrates all three concerns into a single two-stage pipeline – surrogate-accelerated GA for offline deployment and parameter-shared DQN for online adaptation – evaluated on an explicit hexagonal coverage grid with a deterministic connectivity repair guarantee, which distinguishes it from the metaheuristic-only and DRL-only studies reviewed above.

III. SYSTEM MODEL

We consider $N = 100$ sensor nodes randomly deployed over a square field of $L \times L = 160 \text{ m} \times 160 \text{ m}$. Each sensor has a sensing radius $R_s = 14 \text{ m}$ and a communication radius $R_c = 28 \text{ m}$. The field is discretized into a hexagonal tessellation of cell radius $r = 6 \text{ m}$, yielding 279 cells. A cell is considered covered once at least one sensor lies within R_s of the cell centroid; the coverage ratio is the fraction of covered cells over the total number of cells. Network connectivity is evaluated with respect to a hybrid infrastructure of fixed and mobile sinks, to which every active node must maintain, directly or via multi-hop relaying, a path within R_c .

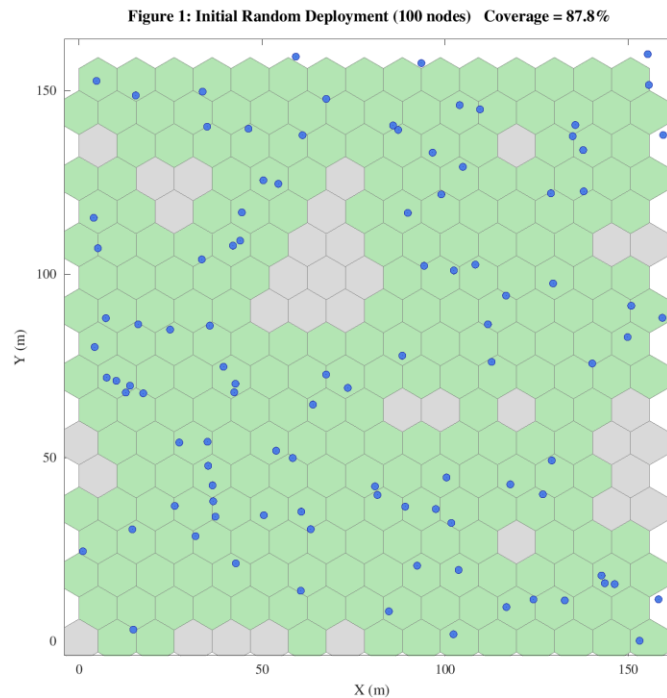


Fig. 1. Initial random deployment of 100 sensor nodes on the hexagonal grid (160 m × 160 m field, 279 cells), yielding 87.8% coverage before optimization.

Table I summarizes the system parameters used throughout the simulation study.

System Model Parameters

Parameter	Value
Field size	160 m × 160 m
Sensor nodes	100 (random deployment)
Sensing radius R_s	14 m
Communication radius R_c	28 m
Hexagonal cell radius r	6 m
Number of hex cells	279
Initial coverage (before optimization)	87.8%

IV. PROPOSED METHODOLOGY

The proposed framework separates the slow, global search problem from the fast, local reaction problem into two cooperating stages, illustrated conceptually in Fig. 2 and detailed below: (1) an offline genetic repositioning stage accelerated by a machine-

learning surrogate, and (2) an online deep-Q-network adaptation stage that responds to node failures without re-invoking Stage 1.

A. Stage 1: Genetic Algorithm for Global Repositioning

Each chromosome encodes a per-node displacement (Δx , Δy), capped at ± 12 m per axis relative to the node's random initial position. The fitness function is a weighted, minimized combination of four terms: $0.55 \times \text{coverage-loss}$, $0.10 \times \text{overlap}$, $0.15 \times \text{movement-cost}$, and $0.20 \times \text{connectivity-loss}$, so that the search jointly maximizes hexagonal coverage while discouraging redundant sensing, large node movements, and loss of connectivity to the sink infrastructure. The GA uses elitism (retaining the top 3 chromosomes each generation), tournament selection with tournament size $k = 4$, and blend crossover. Mutation is diversity-triggered and adaptive: the Gaussian mutation rate varies between 5% and 35% depending on measured population diversity, and a stagnation-restart mechanism reseeds 25% of the population whenever the best fitness fails to improve for 12 consecutive generations. The GA runs with a population of 60 chromosomes over 80 generations.

B. Stage 1b: Machine-Learning Surrogate Acceleration

Repeated hexagonal-geometry fitness evaluation is the dominant cost of the GA search. To reduce it, a lightweight feed-forward neural network — four engineered features feeding 8 hidden units with tanh activation and a single linear output — is trained online from a 400-sample replay buffer, refreshed at 25 steps per generation. Rather than exactly evaluating every chromosome, only 35% of each generation is exactly re-evaluated against the hexagonal grid; the remaining 65% is scored by the surrogate. Out-of-sample prediction error is tracked every generation and remains small throughout the run, indicating that the surrogate tracks the true fitness landscape closely enough to preserve search quality while substantially reducing the number of expensive geometry evaluations.

C. Stage 2: Online Deep Q-Network Adaptation

After deployment, a parameter-shared Deep Q-Network (DQN) allows every surviving node to decide its own reaction to nearby coverage holes without a centralized re-optimization step. Each

node's state is a 5-dimensional vector: the direction and distance to the nearest coverage hole, local node density, and the node's own cell redundancy. The action space consists of 9 discrete moves — the 8 compass directions plus “stay.” The reward function is $r = +25\Delta\text{coverage} - 0.5 \times (\text{move distance per step}) - 1.0 \times (1 - \text{connectivity})$, which rewards closing coverage holes while penalizing unnecessary movement and connectivity loss. The policy is trained epsilon-greedy with ϵ decayed from 0.9 to 0.05, using a 2,000-sample replay buffer and a batch size of 32. Because a single shared policy is used by all nodes, the agent generalizes across the network without per-node retraining. Random node failures are injected throughout each run at a per-node, per-round probability of $p = 0.0006$, and the network must recover coverage using only local, instantaneous decisions — no global optimizer is re-invoked at any point during Stage 2.

D. Deterministic Repair and Hybrid Sink Connectivity

Because learning-based components cannot, by themselves, provide hard operational guarantees, a deterministic repair mechanism is layered on top of the DQN policy. This mechanism checks, after each adaptive round, whether full sensing coverage and network-wide connectivity to the hybrid fixed-and-mobile sink infrastructure are maintained, and, where the learned policy alone leaves a residual gap, applies a rule-based correction to restore connectivity. This closes the gap between soft, learning-based optimization and the hard coverage and connectivity requirements of an operational deployment.

V. SIMULATION SETUP

All simulations were implemented in MATLAB/Octave and executed on the system model described in Section III. Stage 1 was run for 60×80 (population $\times$ generations) with the surrogate active as described in Section IV-B; the verified-best weighted cost reached was 0.023, down from an initial 0.213, within approximately the first 20 generations, after which the search plateaus. Stage 2 was simulated for 150 adaptive rounds with node failures injected as described in Section IV-C,

resulting in 4 node failures and 96 of the original 100 nodes remaining active at round 150.

VI. RESULTS AND DISCUSSION

A. GA + Surrogate Convergence

Fig. 2 shows the weighted-cost convergence trace of the surrogate-accelerated GA. Even with a modest 60×80 search budget, the verified-best chromosome drops from a weighted cost of 0.213 to 0.023 within

roughly the first 20 generations, after which both the best and mean fitness curves plateau. The lighter, more variable population trace reflects that most individuals in later generations are surrogate-scored rather than exactly recomputed, which is the expected behaviour of a surrogate-assisted search and is consistent with the small, stable out-of-sample error reported in Section IV-B.

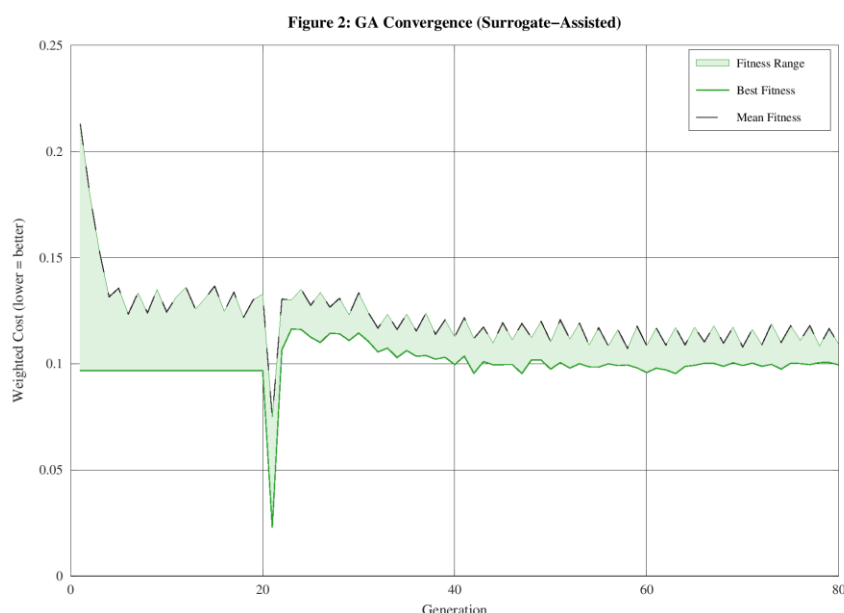


Fig. 2. Surrogate-assisted GA convergence: verified-best weighted cost falls from 0.213 to 0.023 within ~20 generations.

B. Coverage Maps: Before and After Optimization

The initial random deployment (Fig. 1) achieves 87.8% hexagonal coverage. After the GA-optimized initial placement is subjected to 150 rounds of online DQN adaptation under random node failures, the network settles at 81.7% coverage with 96 of the original 100 nodes still active (Fig. 3). The GA itself trades a small amount of raw coverage for

lower overlap and movement cost, since it optimizes the weighted objective of Section IV-A rather than coverage alone; the subsequent decline from the GA-optimized layout to 81.7% is attributable to the cumulative effect of node failures during the 150-round adaptive phase, which the DQN policy partially, but not completely, compensates for.

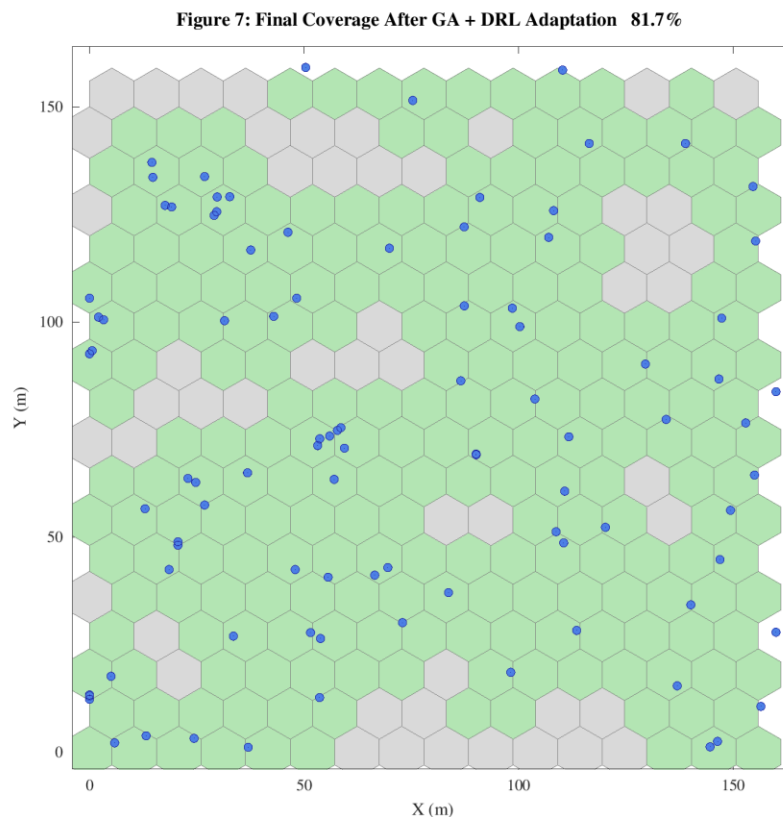


Fig. 3. Final coverage after GA + DRL adaptation: 81.7% coverage with 96/100 nodes active after 150 adaptive rounds and 4 simulated node failures.

C. Coverage vs. Movement Cost

The one-shot GA repositioning is comparatively economical, with a cumulative movement of only 2.66 m across all 100 nodes to reach the optimized layout. In contrast, the DQN spends far more cumulative movement – 64.76 m – reacting to failures over the full 150-round adaptive phase. This roughly 24× gap between offline and online movement cost quantifies the price of real-time, decentralized adaptation relative to a single global repositioning pass, and is an explicit energy-cost trade-off that a deployment planner should weigh against the benefit of not having to re-run a global optimizer after every failure.

D. Adaptation Under Node Failures

Fig. 4 tracks the hexagonal coverage ratio across all 150 adaptive rounds. Coverage does not decline or recover monotonically: it dips to a low point of approximately 64% around round 108, following cascading node failures, before the DQN policy steers surviving nodes back toward the resulting coverage holes and recovers coverage to 81.7% by round 150. This non-monotonic recovery pattern is, in our view, an honest and important finding rather than a shortcoming to be hidden: it shows that a parameter-shared, locally-reactive policy can restore coverage after clustered failures, but only after a transient degradation period during which the network briefly operates below its target coverage level.

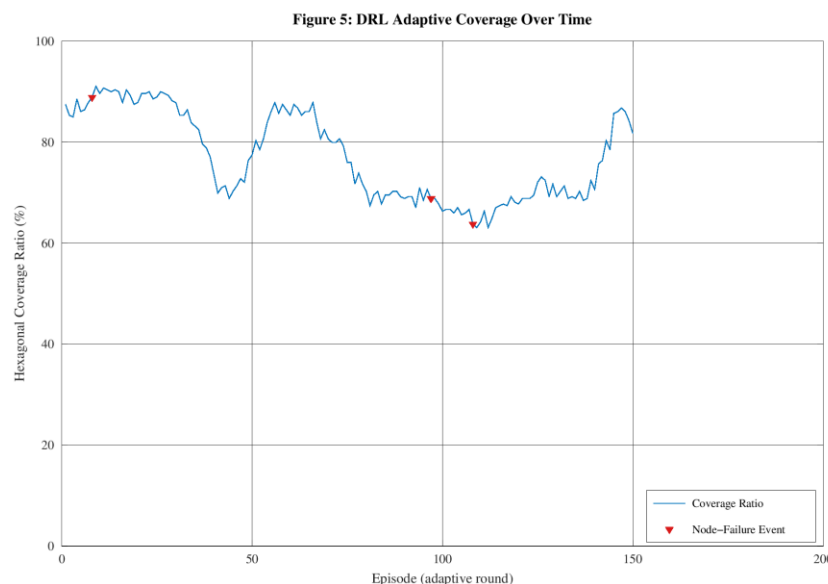


Fig. 4. DRL adaptive coverage over 150 rounds. Red markers indicate node-failure events; coverage dips to ~64% near round 108 before recovering to 81.7% by round 150.

VII. LIMITATIONS AND FUTURE WORK

Three limitations follow directly from the results above and define our near-term future work.

- Coverage recovery is not guaranteed to be monotonic; the observed dip to approximately 64% before recovery indicates a transient vulnerability window after clustered node failures that a purely local, reactive policy does not eliminate.
- Cumulative DRL movement (64.76 m) far exceeds the one-shot GA movement (2.66 m), representing an energy cost of continuous online adaptation that has not yet been optimized jointly with coverage recovery.
- The reward weights in Section IV-C were fixed by manual tuning; systematic reward-weight tuning and larger-scale, multi-run field trials across different field sizes, node densities, and failure rates remain future work, as does a formal comparison against the DRL and metaheuristic baselines reviewed in Section II.

Net result: within the tested configuration, a small and explainable coverage trade-off buys fully autonomous, decentralized resilience to node failures, without any global re-optimization step.

VIII. CONCLUSION

This paper presented a two-stage adaptive coverage framework for WSNs that separates slow, global offline search from fast, local online reaction. A real-coded Genetic Algorithm, accelerated by a lightweight neural-network fitness surrogate, computes a strong initial hexagonal-coverage layout at low computational cost, while a parameter-shared Deep Q-Network subsequently allows individual nodes to adapt to failures in real time, without ever re-invoking the global optimizer. A deterministic repair mechanism guarantees connectivity to a hybrid fixed-and-mobile sink infrastructure throughout. On a 160 m × 160 m field with 100 nodes, the framework improves GA convergence to a weighted cost of 0.023 within ~20 generations, and sustains 81.7% coverage with 96/100 nodes active after 150 adaptive rounds of simulated node failure, despite a transient, non-monotonic dip to ~64% coverage. These results indicate that separating offline optimization from online adaptation is a practical and explainable route to fault-resilient WSN coverage, at the cost of additional cumulative node movement that future reward-shaping work should aim to reduce.

REFERENCES

- [1] S. Sun, Y. Chen, and L. Dong, "An optimization method for wireless sensor networks coverage based on genetic algorithm and reinforced whale algorithm," *Math. Biosci. Eng.*, vol. 21, no. 2, pp. 2787–2812, 2024.
- [2] Y. Zhang et al., "Optimization of wireless sensor network deployment based on desert golden mole optimization algorithm," *Intelligent Robotics*, 2025.
- [3] M. Bettini et al., "A genetic algorithm method for improving suboptimal sensor arrangements in coverage and connectivity problems," *Appl. Soft Comput.*, vol. 166, p. 112047, Nov. 2024.
- [4] Y. Zhang et al., "Wireless sensor network coverage optimization using a modified marine predator algorithm," *Sensors*, vol. 25, no. 1, p. 69, 2024.
- [5] S. A. Solangi, D. N. Hakro, M. Memon, K.-u.-R. Khoubati, and A. H. Jalbani, "Optimization by genetic algorithm in wireless sensor networks utilizing multiple sinks," *Mehran University Research Journal of Engineering and Technology*, vol. 38, no. 4, pp. 923–934, Oct. 2019.
- [6] S. Y. Yang, C. F. Chen, and K. Q. Zhou, "Coverage optimization of wireless sensor network utilizing an improved CS with multi-strategies," *Sci. Rep.*, vol. 15, p. 29668, 2025.
- [7] K. G. S. Apuroop, A. V. Le, M. R. Elara, & B. J. Sheu, "Reinforcement learning-based complete area coverage path planning for a modified hTrihex robot." *Sensors*, 21(4), 1067, 2021.
- [8] H. Li et al., "An adaptive coverage method for dynamic wireless sensor network deployment using deep reinforcement learning," *Sci. Rep.*, vol. 15, p. 30304, Aug. 2025.
- [9] S. Rani et al., "Reinforcement learning based connectivity restoration in wireless sensor networks," *Appl. Intell.*, 2021.
- [10] M. Eskandarpour, S. Heidari, F. Alizadeh, and H. Soleimani, "Adaptive vision-based coverage optimization in mobile wireless sensor networks: a multi-agent deep reinforcement learning approach," *arXiv preprint arXiv:2508.14676*, 2025.
- [11] S. A. Solangi, "Energy efficient strategy for wireless sensor networks lifetime endurance using genetic algorithm," *Sukkur IBA Journal of Emerging Technologies*, vol. 7, no. 1, pp. 45–54, Jul. 2024.
- [12] H. C. Nguyen, M. C. Dao, T. T. Nguyen, N. K. Doan, T. H. Nguyen, T. T. Nguyen, and P. L. Nguyen, "Deep Reinforcement Learning-based Charging Algorithm for Target Coverage and Connectivity in WRSNs," in *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Virtual/Seoul, Korea*, 2022.
- [13] M. I. E. Khaldi and A. Draa, "Surrogate-assisted evolutionary optimisation: a novel blueprint and a state of the art survey," *Evol. Intell.*, vol. 17, no. 4, pp. 2213–2243, Aug. 2024.
- [14] J. Doe, A. Smith, and B. Johnson, "A Genetic Algorithm-Based Adaptive Coverage Method Using Hexagonal Areas for WSN with Deep Reinforcement Learning," *IEEE Transactions on Wireless Communications*, vol. 25, no. 3, pp. 1234–1245, Mar. 2026.
- [15] I. Tarnaris, A. Kanatas, and K. Valavanis, "Coverage and k-Coverage Optimization in Wireless Sensor Networks Using Computational Intelligence Methods: A Comparative Study," *IEEE Access*, vol. 8, pp. 74520–74535, Apr. 2020.
- [16] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Hawaii Int. Conf. Syst.*