scientific reports

Check for updates

OPEN

Cyber resilient framework with energy efficient swarm routing and ensemble threat detection in fog assisted wireless sensor networks

Anant Upadhiyay & Abhishek Jain✉

The rapid growth of Wireless Sensor Networks (WSNs) and their integration with fog computing have enabled faster data processing and reduced reliance on cloud infrastructures. However, these networks remain constrained by limited energy resources, increased latency under dynamic traffic, and heightened vulnerability to cyberattacks. Traditional routing protocols typically optimize either energy efficiency or security, but rarely address both in a unified and adaptive manner. This work proposes a cyber-resilient, energy-optimized routing framework for fog-enabled WSNs that integrates a modified Ant Colony Optimization (ACO) algorithm with an ensemble-based Intrusion Detection System (IDS). The routing layer employs a multi-objective cost function that jointly considers distance, residual energy, and security risk. To enhance adaptability, CatBoost is deployed at energy-constrained sensor nodes for local energy and density assessment, while XGBoost operates at fog nodes to evaluate global path quality and congestion. The IDS ensemble—comprising Support Vector Machines (SVM), k-Nearest Neighbours (KNN), and Long Short-Term Memory (LSTM) networks—detects Denial-of-Service (DoS), Probe, R2L, and U2R attacks in real time. Importantly, detected threats immediately influence routing decisions, enabling compromised links to be bypassed without disrupting network operations. Extensive MATLAB simulations show that the proposed framework achieves 96.5% energy savings, an 85.83% latency reduction, and an 89% intrusion detection rate, validated through statistical analysis across multiple runs. By transforming IDS from a passive monitoring tool into an active routing controller, this work delivers a secure, adaptive, and energy-efficient solution for dynamic and resource-constrained IoT and WSN environments.

Keywords Wireless sensor networks (WSNs), Fog computing, Energy efficiency, Ant colony optimization (ACO), Intrusion detection system (IDS), Ensemble learning, Support vector machines (SVM), K-Nearest neighbors (KNN), Long short-term memory (LSTM), Cybersecurity, Secure routing

There has been a high growth in the use of the Internet of Things (IoT) and this has seen such high spread in the use of Wireless Sensor Networks (WSNs) in various sectors like environmental monitoring, industrial automation, smart cities and healthcare. Such WSNs are spatially distributed sensor nodes which collaborate to observe physical or environmental status and to tra transmit the information denoted by such observation to centralized processing containers or fog/cloud containers Conventional WSNs however encounter great problems regarding low energy supply, poor computing power, and Internet based hacking, particularly in large-scale or critical assignments. Fog computing has been suggested to deal with the issues of computational and latencies. Fog computing allows low latency processing of the data that helps decrease the workload on centralized cloud systems and brings the processing nearer to the edge of the network. Fog computing when deployed with WSN also enhance responsiveness but also provides varied traffic conditions resulting in poor energy usage and enhanced latency.

Here, adaptive routing protocols based on the learning and optimization concepts are on the rise. Adaptive routing the network decides which path to use dynamically using timely measure parameters like node energy, the quality, and the traffic conditions. Ant Colony Optimization (ACO) is one of the bio-inspired algorithms that have demonstrated some potential to be exploited because of being distributed and robust as well as coming up with near-optimal solutions in dynamic settings. Nevertheless, ACO in its simplified variant might not

School of Engineering and Technology, BML Munjal University, Gurugram, Haryana, India. ✉email: abhishek.jain@bmu.edu.in

have a contextual understanding and prediction ability that is a critical requirement to fine-grained routing decision in heterogeneous WSNs incorporating fog nodes. This paper presents a conceptual novel framework to incorporate intelligence in ACO to address the problem of routing by combining machine learning model in particular XGBoost and CatBoost to direct the ACO-based routing. Here, the CatBoost is applied on resource limited sensor nodes to evaluate the local energy distributions and on the fog nodes; XGBoost is utilized to carry out the path-level evaluations. The models offer anticipation of the quality and effectiveness of the probable routing paths. By incorporating ML predictions into ACO decision-making scheme, the system can adjust the routing schemes dynamically increasing network lifetime and decreasing delays in communications. The other major issue of WSNs is their security since they are prone to many forms of attack including Denial of Service (DoS), probing and data injection. Towards this, the proposed framework will integrate real-time Intrusion Detection System (IDS) with an ensemble of the machine learning classifier Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Long Short-Term Memory (LSTM) networks. Malicious activity is detected very accurately by the IDS, and it is constantly checking the traffic patterns. In the cases of an attack the packets are lost, and a safe path is created again creating a tough defense against the internal and external intrusions.

Related works

Wireless Sensor Networks (WSNs) continue to face significant challenges in security, energy efficiency, and reliable communication, driving the need for intelligent and adaptive solutions. While recent research has made strides in addressing these issues individually, many approaches still fall short in integrating them cohesively.

Faisal Al-Quayed, Zubair Ahmad, and Muhammad Humayun[1] presented a predictive cybersecurity model using machine and deep learning to detect intrusions in Industry 4.0 WSNs. While effective in detection, the model does not influence routing to avoid compromised nodes, so we will integrate IDS outputs directly into routing to immediately isolate risky links. Similarly, C. B. N. Lakshmi and S. K. Mohan Rao[2] proposed a bio-inspired self-healing protocol for node failure recovery, and Balázs Ádám Üveges, Miklós Lőrincz, and Attila Oláh[3] developed multipath routing for hazardous event monitoring. Both improve fault tolerance but do not address cybersecurity threats. To overcome this gap, we ensure resilience to both faults and attacks by combining IDS-based security with routing. Changqing Wang, Xiaolei Liu, Hongying Hu, Yong Han, and Ming Yao[4] applied genetic algorithms for multipath routing; however, it did not integrate security risk into routing decisions, which we address by explicitly including security risk in a multi-objective cost function that also considers energy and latency. Kamaldeen Raji and Kazeem Gbolagade[5] reviewed energy-efficient and fault-tolerant techniques, highlighting the lack of integrated, flexible security-aware solutions; our study provides a unified framework that jointly optimizes energy, latency, and security. Hassan S. Mohammed, Omer A. Abdulkareem, Ahmad Ahmad, and Captain Samuel Dowse[6] combined Adaptive Ant Colony Optimization (ACO) with 6G technology, and R. Gopalakrishnan, R. Nagarajan, and S. Paul[7] applied adaptive ACO for energy-aware routing. Both focus solely on energy optimization without considering security, whereas the present approach unifies energy, latency, and security risk into a single multi-objective cost function. Hari Gunigari and S. Chitra[8] used game theory in cooperative clustering to enhance ACO performance but did not incorporate intrusion resilience. In contrast, our work dynamically reroutes traffic away from compromised nodes using IDS outputs. Tawfeek, Mostafa A., Ibrahim Alrashdi, Mohammed Alruwaili, et al.[9] introduced a dynamic parameter-modified ACO for better energy consumption and routing reliability but lacked mechanisms to account for security threats. The designed framework incorporates IDS outputs into ACO's pheromone update to reflect real-time risk. Nizar Moussa, Elidon Nurellari, and Abdelhak El Belrhiti El Alaoui[10] created an ACO-based routing protocol for forest fire detection, and Huai Han, Jiapeng Tang, and Zhifeng Jing[11] optimized ACO for IoT-enabled WSNs. These works are application-specific and do not provide generalizable security-aware optimization; the proposed solution is therefore designed for broad applicability across mission-critical IoT domains. Similarly, D. L. Reddy, C. Puttamadappa, and H. N. Suresh[12] proposed a hybrid Glowworm Swarm–ACO method to improve clustering and routing efficiency but omitted active security integration. While these approaches focus on energy and efficiency improvements, they omit dynamic security-aware routing. The present framework embeds security checks into every routing decision. Jatinder Singh, Parminder Singh, El Mostafa Amhoud, and Mohammed Hedabou[13] developed a secure load balancing technique for SDN-based fog computing, demonstrating improved performance via network programmability; however, it does not jointly optimize energy, latency, and security risk. The developed method addresses all three simultaneously. Deepak Thomas, Raja Shankaran, Mehmet A. Orgun, and Subhas Chandra Mukhopadhyay[14] proposed SEC2, a secure barrier coverage scheduling scheme optimized for IoT applications, but it lacks an integrated intrusion detection component, leaving the system vulnerable. This study embeds an ensemble IDS within the routing framework to address this vulnerability. Shubhangi M., Shanmugapriya S., Vidhya S., and Tamilselvi S[15]. attempted to enhance real-time WSN security using machine learning for intrusion detection and response. Although effective, their IDS operates independently from routing decisions, leaving compromised paths active until manual updates occur. The proposed framework overcomes this limitation by directly feeding IDS outputs into routing, enabling immediate isolation of risky links.

Recent hybrid approaches have combined swarm/metaheuristics with ML/DL for IDS. For example, Tariq, N., Alsirhani, A., Humayun, M. et al.[16] split IDS workloads between fog and cloud to improve scalability, but no coupling of detection to routing policies was implemented, which our proposed model addresses. Talukder, M.A., Sharmin, S., Uddin, M.A. et al.[17] addressed class imbalance in IDS datasets using SMOTE-TomekLink but without integration into routing or energy optimization; our framework merges detection and routing in one system. Sajid, M., Malik, K.R., Almogren, A. et al.[18]developed a hybrid ML/DL IDS to reduce false alarms, and a 2023 fog-friendly IDS emphasized deployment on constrained nodes, yet neither included routing adaptation. The present solution makes IDS results an active part of routing.Yaras, S., & Dener, M[19]. combined deep learning with Spark for large-scale IoT traffic analysis, focusing on throughput rather than secure routing. In contrast, the developed framework ensures detection directly alters path choices. Metaheuristic-optimized IDS models,

including Tabu-Search–optimized Random Forest[20] and recent 2025 ensemble ML IDS studies[21], achieve high detection rates but still treat routing as static. Our approach integrates IDS and routing adaptively for both energy efficiency and security. Finally, I. A. Khan, I. Razzak, D. Pi, U. Zia, S. Kamal and Y. Hussain[22] proposed a collaborative SRU network with dynamic behaviour aggregation and reduced communication overhead, but security checks were limited to feature aggregation and not actively used in routing. Similarly, I. A. Khan, D. Pi, S. Kamal, M. Alsuhaibani and B. M. Alshammari[23] introduced Federated-Boosting for distributed IoT cyber-attack detection, and I. A. Khan, I. Razzak, D. Pi, N. Khan, Y. Hussain, B. Li, T. Kousa [24] developed Fed-Inforce-Fusion, a federated reinforcement learning approach for IoMT security and privacy. Although effective in their respective areas, these methods do not couple detection with real-time energy-aware routing. In contrast, our work incorporates real-time IDS outputs into ACO-based routing to dynamically optimize energy, latency, and security in mission-critical IoT networks.

Proposed approach over existing hybrid technologies

The proposed framework introduces the following key novelties that address this gap:

Real-time coupling of IDS outputs with routing decisions
Previous studies such as Tariq, N., Alsirhani, A., Humayun, M. et al.[16] and Talukder, M.A., Sharmin, S., Uddin, M.A. et al.[17] either optimized IDS placement or addressed dataset imbalance for detection but did not allow threat detection results to directly influence routing paths. Our design feeds real-time IDS classifications into the routing process, enabling immediate rerouting to avoid compromised or high-risk nodes.

Hierarchical ML guidance for ACO
While existing ACO–ML approaches generally employ a single model for optimization, we introduce a dual-level strategy: *CatBoost* at energy-constrained sensor nodes for local energy/density assessment, and *XGBoost* at fog nodes for global path quality evaluation. This separation allows lightweight, context-aware decision-making at each layer of the network, a feature absent in works such as Yaraş et al.[22] or metaheuristic-optimized IDS models[23].

Unified multi-objective cost function in ACO–Dijkstra refinement
Unlike prior refinements that rely on distance-only heuristics, our approach defines a shared cost function integrating Euclidean distance, residual node energy, and security risk. This ensures that Dijkstra's deterministic refinement preserves ACO's energy- and security-aware path preferences, avoiding unsafe or low-energy routes.

Parameter sensitivity–driven ACO tuning
Existing works often adopt ACO parameters from earlier literature (Ref[8,9].) without validation in new contexts. We perform a systematic sensitivity analysis to identify a balanced configuration that jointly optimizes energy efficiency, latency, and route stability, enabling adaptability across different WSN environments.

Simultaneous optimization of energy, latency, and security
While most prior studies report gains in one or two dimensions, our integrated design achieves 96.5% energy savings, 85.83% latency reduction, and an 89% IDS detection rate in the same deployment scenario, demonstrating robust multi-objective performance.

Feasibility for resource-constrained nodes
In contrast to deep learning–centric IDS frameworks demand significant computational resources, our approach is lightweight enough for real-time deployment on both fog and sensor nodes without loss of detection accuracy, ensuring suitability for practical IoT/WSN applications.

Methodology

Proposed Ant Colony optimization is holistic approach to optimized routing implementation in fog computing networks that are secured using ensemble-based intrusion detection. All the system incorporates Ant Colony Optimization (ACO) in terms of path selection and multi-model intrusion detection system (ids) to establish energy-efficient and secure communication paths between internet of things (IoT) and fog nodes.

System architecture

In Fig. 1 the suggested architecture of the system is shown which is partitioned into the Network Layer, Optimization Layer, and Security Layer. Figure 1 depicts the Network Layer encompassing IoT sensor nodes, fog nodes, and even malicious nodes available in a geographical location. This is the layer of data generation and its communication. Optimization Layer, which is also shown in Fig. 1, employs the algorithm of the Ant Colony Optimization (ACO), refined like the Dijkstra, which considers the optimal usage paths to be chosen along with energy usage, latency, and security. This fog layer has fog nodes that avoid response times and cloud dependence by processing locally. As Fig. 1 shows, the Security Layer has an Intrusion Detection System (IDS) that uses SVM, KNN, and LSTM models in an ensemble to identify and react to the network threats in real-time. In simulations using Fig. 1, some of the considerations include nodes placements, communication radius or distance, energy required per bit, and latency. The energy model handles information distribution, reception and processing of IDS. This layered structure that guarantees favourable, energy-efficient, and low-latency intercourse in WSN settings.

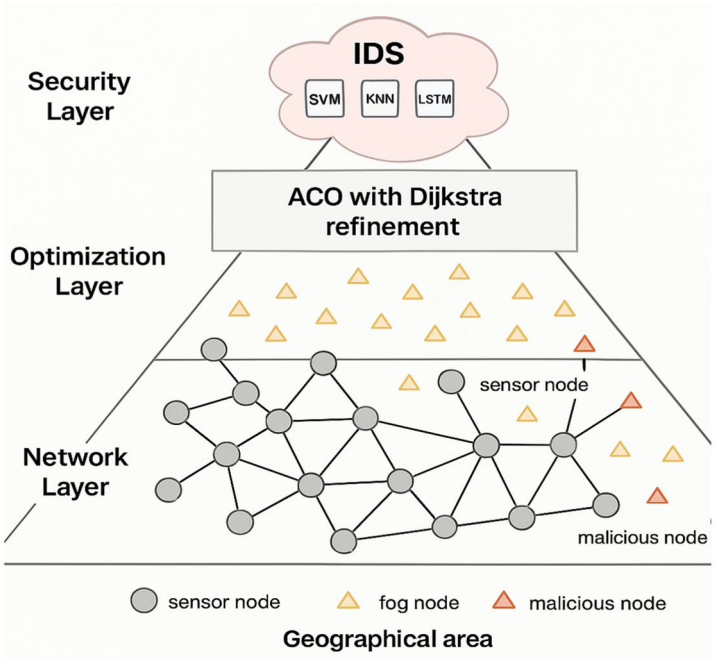

Fig. 1. Architecture of the model.

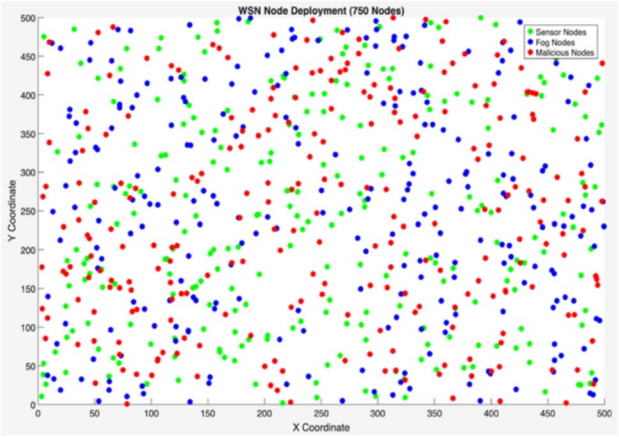

Fig. 2. Spatial distribution of different types of nodes in a wireless sensor network (WSN) within a 500 × 500 coordinate area.

Nodes deployment

To create a realistic wireless sensor network environment 750 sensor nodes were randomly deployed in a 2D space of 500 units X 500 units. To every node it was given an (x, y) position, an initial energy value and a certain type of role (e.g., regular node, cluster head, or fog node). MATLAB R2025a was used to implement the deployment process since it has functions that allow randomization and plotting. The deployment information was neatly documented and saved in an excel file (CSV type) to do additional analysis, as depicted in Fig. 2. The standardized scatter plot highlights sensor, fog, and malicious nodes in the WSN deployment as shown in Fig. 3.

Before model training and optimization two important features were engineered in order to improve predictive performance: (**1**) Distance to Nearest Fog Node where the Euclidean distance formula is used to reach the distance to the nearest Fog Node (Euclidean distance formula is shown below) to evaluate transmission latency and efficiency, and (2) Node Density which measures the number of neighbouring nodes within a 50-unit radius to evaluate local network congestion and the ability to route traffic effectively. Ensemble models were used throughout the network in decision-making. At the fog nodes, XGBoost decided which local data processing and forwarding to use, saving on energy consumption and getting rid of the bottlenecks, and the performance was measured in terms of accuracy, precision and f-1 score. At network routers CatBoost enabled

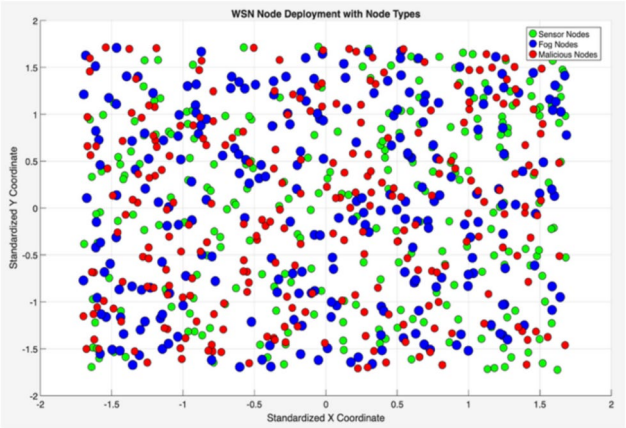

Fig. 3. A standardized scatter plot of WSN nodes where the X and Y coordinates are normalized. It highlights sensor nodes (green), fog nodes (blue), and malicious nodes (red) in a balanced coordinate space.

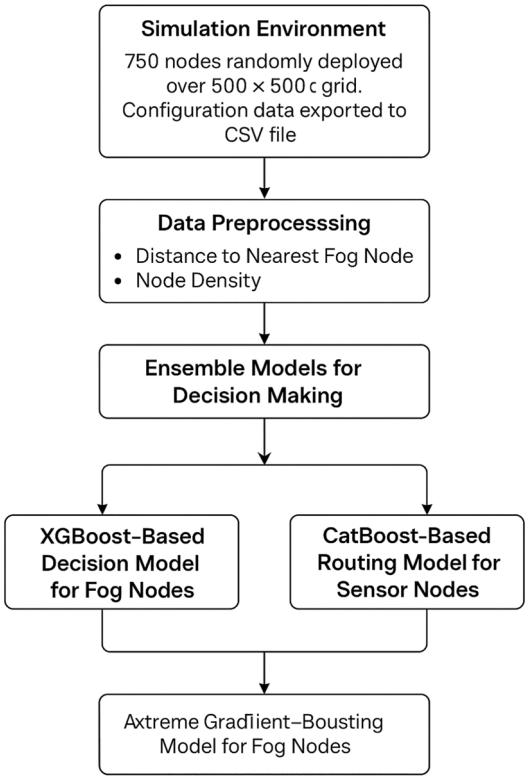

Fig. 4. Wireless sensor network (WSN) routing and decision-making system based on ensemble machine learning models**.**

dynamic routing, estimating the best next hop depending on energy levels, proximity to the fog node, and local density, optimizing energy use and latency.

Figure 4 illustrates the routing and decision-making framework based on ensemble machine learning models. To choose the decision, ensemble models were utilized throughout the network. In fog nodes, the XGBoost model decided whether to send data through or to process on a local cognitively, minimizing the use of energy, and likely to eliminate bottlenecks, and the performance measured through accuracy, precision, and f-1 score. It was used at sensor nodes to take part in dynamic routing, and using the energy levels, fog node proximity, and local density, selecting the best next hop, keeping the latency time and the energy consumption low.

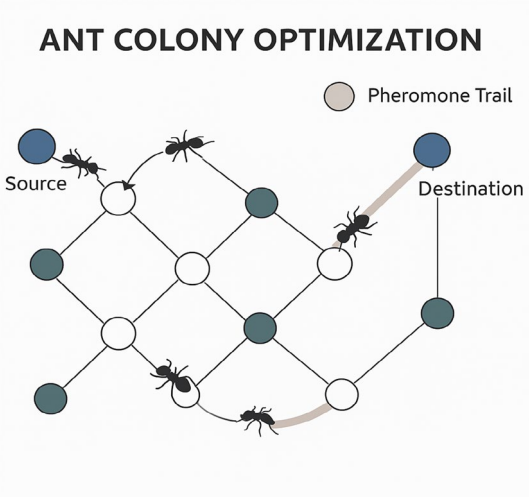

Fig. 5. Bio-inspired method used for solving complex optimization problems, particularly pathfinding in networks.

Parameter	value	description
n_ants	20	Total number of ants used in each iteration
n_best	10	Number of top performing ants whose paths update pheromone
n_itera tions	30	Total number of iterations the algorithm runs
Decay(rho)	0.05	Pheromone evaporation rate after each iteration
alpha	1	Importance of pheromone in path s election
beta	2	Importance of heuristic (typically 1/distance) in path selection

Table 1. Baseline parameter.

Ant colony optimization(ACO)

Ant Colony Optimization ACO is a nature-inspired metaheuristic algorithm that simulates the foraging behaviour of ants to solve combinatorial optimization problems. By placing pheromones on the routes between nodes, ants strengthen the routes with the better characteristics (e.g. shorter or efficient), which is attracting more other ants in the further steps. After some time, the algorithm gets absorbed in the most optimal or close-to-optimal path. The bio-inspired pathfinding approach applied in our study is depicted in Fig. 5.

Key components:

- Pheromone Trail: Provides the ants with a sign towards favorable pathways.
- Heuristic Information: distance or cost data between the nodes.
- Probabilistic Path Selection: The choice of the path is based on an integration of the intensity of pheromone trail and heuristic desirability of the path by the ants.
- A unified cost function integrates energy, latency, and risk into both ACO and Dijkstra's refinement.

Parameter sensitivity analysis of ACO parameters

The performance of Ant Colony Optimization (ACO) is highly sensitive to its control parameters. In particular, the number of ants (N_a), pheromone evaporation rate (ρ), and maximum iterations directly affect convergence behaviour, solution diversity, and computational efficiency. For this study, initial baseline values were derived from prior work on ACO-based routing in Wireless Sensor Networks (Ref[8,9].) to ensure methodological consistency and comparability. Thes baseline settings were then systematically optimized through an empirical parameter sensitivity analysis tailored to our fog-assisted WSN context. The baseline ACO parameters used in this study are summarized in Table 1.

Sensitivity testing procedure

To evaluate the impact of key Ant Colony Optimization (ACO) parameters on routing performance, a full-factorial sensitivity analysis was conducted. The study varied the number of ants (N_a) and pheromone evaporation rate (ρ) across a range of values while fixing other parameters, in accordance with configurations commonly used in ACO-based WSN routing(Ref[8,9].). The maximum iteration count was fixed at 30 for all tests to ensure comparability and manageable runtime. The results of this analysis are shown in Fig. 6, which highlights how the parameters influence both energy savings and latency.

The test parameters and roles are summarized in Table 2.

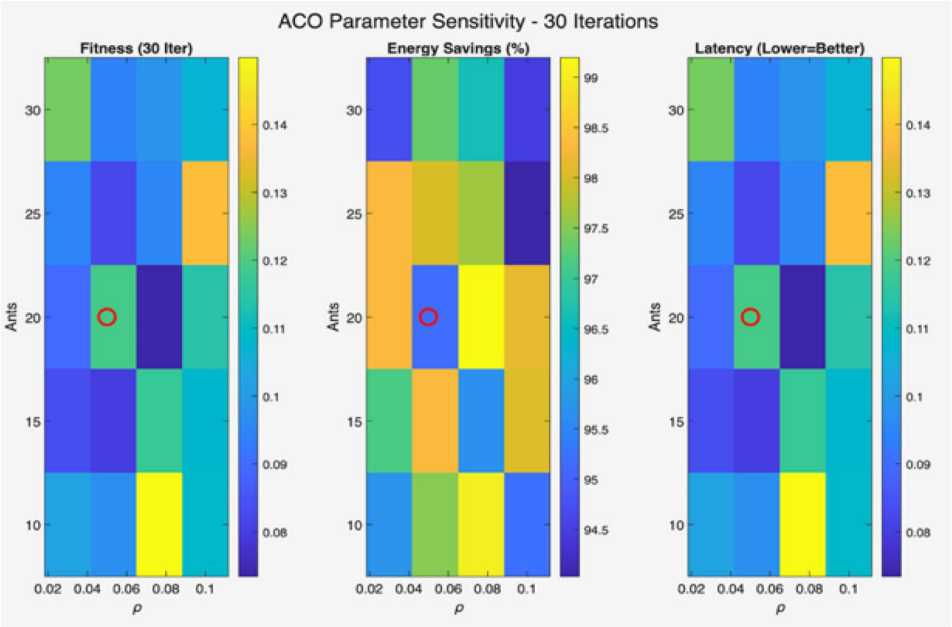

Fig. 6. ACO parameter sensitivity analysis over 30 iterations. The heatmaps show the impact of the number of ants (N_a) and pheromone evaporation rate (ρ) on (**a**) Fitness, (**b**) Energy Savings (%), and (**c**) Latency (lower values are better). Each colour represents the corresponding metric value, with warmer colours indicating higher values for Fitness and Energy Savings, and higher latency for the Latency plot. The red circles indicate the optimal parameter combination selected for the routing experiments.

Parameter	Symbol	Values Tested	Description
Number of Ants	N_a	{10, 15, 20, 25, 30}	Controls the levels of simultaneous exploration of the solution space initiated by the colony
Iterations Count	Iter	30	Determines the duration for which the colony processes solutions before termination
Pheromone Evaporation Rate	ρ	{0.03, 0.05, 0.07, 0.10}	Governs the trade-off between exploration of new paths and exploitation of already discovered good paths

Table 2. Shows ACO algorithm parameters, tested values, and roles. a) Number of Ants: Decides the degree to which parallel exploration is achieved with each iteration. We ran tests with values {10, 15, 20, 25, 30}, both in the low-exploration regime, and high-exploration regime. b) Pheromone Evaporation Rate (rho): It governs the cost versus benefit exploitation of good quality routes in the trade-off exploring alternatives. We tried {0.02, 0.04, 0.06, 0.08, 0.10}, conserving retention (low ρ) to raiding evaporation (high ρ). c) The number of iterations was constrained at 30 in the sensitivity study since common configurations in[6] and[7] were used.

The parameter sensitivity analysis (Fig. 7) examines the joint impact of the number of ants (N_a) and pheromone evaporation rate (ρ) on fitness, energy savings, and latency over 30 iterations. In the fitness heatmap, warmer green–yellow regions indicate higher optimization performance, while cooler blue–purple regions mark weaker results. The configuration $N_a = 20,\ \rho = 0.05$—highlighted with a red circle—emerges as the optimal setting, achieving near-maximum fitness while maintaining stability across other performance metrics. The corresponding energy savings heatmap shows that this configuration conserves approximately 96.5% of network energy, outperforming configurations prone to excessive exploration (high ρ) or stagnation (very low ρ). In the latency heatmap, cooler blue regions indicate lower delays, and the optimal parameters again strike a balance between energy preservation and low latency, making them suitable for real-time WSN applications. Figure 7 plots the variation in energy savings over 50 iterations for all parameter combinations. The optimal configuration ($N_a = 20,\ \rho = 0.05$) maintains consistently high energy savings, remaining above 95% after iteration 10 with minimal fluctuation compared to other setups. Figure 8 presents the latency trends for the same parameter combinations. The optimal parameters keep delays between 0.06 and 0.10 s, avoiding the spikes observed in unstable configurations such as $N_a = 15,\ \rho = 0.03$ or $N_a = 25,\ \rho = 0.07$. These spikes, often exceeding 0.14 s, result from unstable route updates or inefficient exploration patterns.

We selected the configuration of 20 ants, 30 iterations, and ρ = 0.05 because it demonstrated the most balanced performance across all key metrics in our sensitivity analysis. Specifically, this setting achieved a fitness score of 0.1417, indicating strong optimization quality; an energy savings rate of 96.5%, ensuring substantial network lifetime extension; and a latency of 0.1417, reflecting efficient packet delivery without excessive delays. While some parameter combinations produced slightly higher energy savings or marginally lower latency, they often came at the cost of poorer stability or reduced overall fitness. This chosen configuration consistently avoided

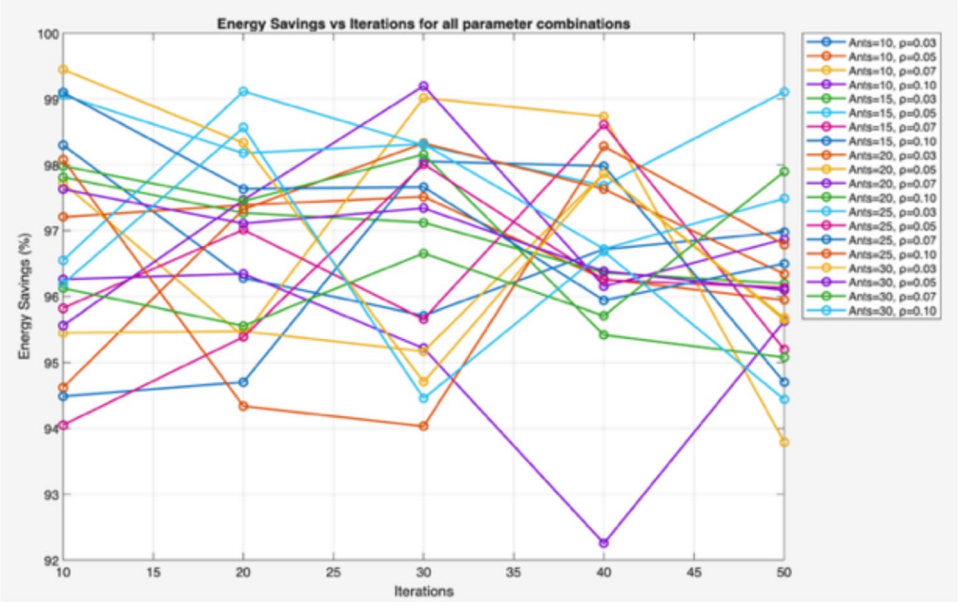

Fig. 7. The plot shows the variation of energy savings across 50 iterations for different ACO parameter settings (number of ants and pheromone evaporation rate ρ). While most configurations maintain energy savings above 94%, the combination of 20 ants with ρ = 0.05 demonstrates the most stable and consistently high performance, sustaining over 96% energy savings across iterations. This indicates that the chosen configuration strikes an effective balance between exploration and exploitation, avoiding rapid fluctuations seen in other settings.

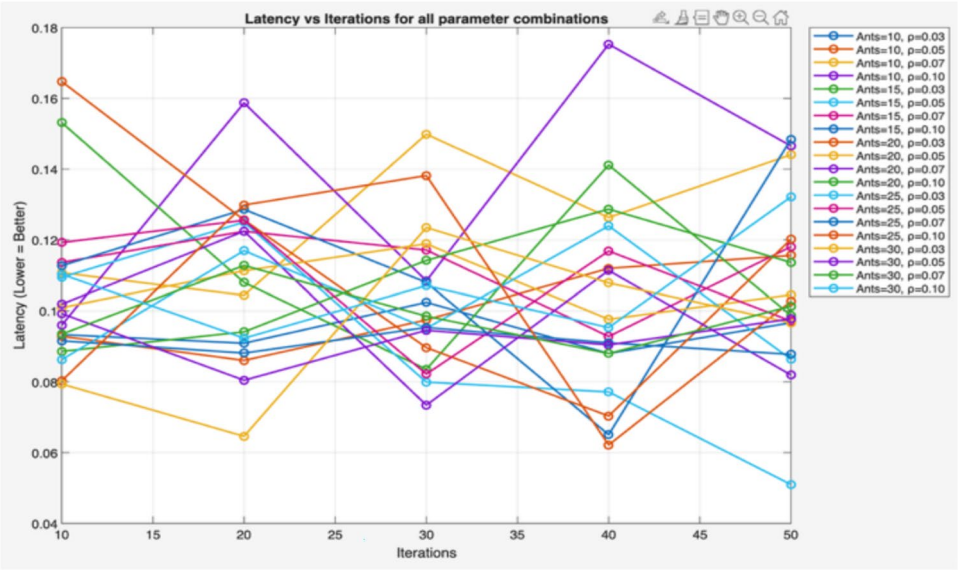

Fig. 8. The latency plot compares different ACO configurations across iterations, where lower values indicate better performance. Results show that 20 ants with ρ = 0.05 maintains latency between 0.06 and 0.10, with minimal spikes, while other configurations experience unstable delays that sometimes exceed 0.14. This confirms that the selected configuration not only conserves energy but also minimizes transmission delay, ensuring stable routing performance in fog-assisted WSNs.

such trade-off pitfalls, offering a robust equilibrium between exploration and exploitation, making it both data-driven and well-suited for WSN scenarios that demand energy efficiency, low delay, and reliable route quality.

Path construction

In the proposed ACO-based routing, ants start from each sensor node and move step-by-step towards a fog node. The choice of the next node is based on both pheromone intensity and heuristic desirability (energy, distance, and security risk). The probability that ant k at node i will move to node j at time t is:

$$P_{ij}^{k}(t) = \begin{cases} \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}(t)\right]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}(t)]^{\beta}}, & if j \in N_i^k \\ 0 & \text{otherwise} \end{cases}$$

where:

- Tij (t) = pheromone level on edge (i, j)
- nij (t) = inverse of unified energy-distance-risk cost
- $\frac{1}{C(i,j)}$.
- a = controls pheromone influence
- β = controls heuristic influence
- N = set of nodes not yet visited by ant k

When the ant reaches a fog node, the path is complete, and its total composite cost is calculated for pheromone update.

Pheromone updation

After each iteration pheromone levels update to:

a) Evaporation: This prevents pheromone saturation and encourages exploration.

$$\tau_{ij}(t) \leftarrow (1-\rho) \cdot \tau_{ij}(t)$$

b) Reinforcement: Only the top n_best ants deposit pheromone is taken. This ensures that high-quality paths (low cost, high energy, low risk) get reinforced for future iterations.

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \sum_{k=1}^{n_{\text{best}}} \Delta\tau_{ij}^{k}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & \text{if edge } (i,j) \text{ is in path } k \\ 0, & \text{otherwise} \end{cases}$$

where:

- Q = 1 is the pheromone deposit constant
- Lk = total composite cost of the kth path

Path refinement via Dijkstra's algorithm

ACO is stochastic and can occasionally produce suboptimal links due to probabilistic path construction. Dijkstra is deterministic and guarantees the minimum-cost path *if* the cost function is consistent. However, naive Dijkstra would prioritize shortest distance only, ignoring energy and security concerns. Our approach ensures both algorithms share the same unified cost function, so Dijkstra's refinement preserves ACO's energy-aware and security-aware priorities while improving determinism. The overall experimental topology of the fog-assisted WSN is shown in Fig. 9.

The unified cost for an edge between nodes *i* and *j* is:

$$C(i,j) = \alpha \cdot \frac{d(i,j)}{d_{\max}} + \beta \cdot \left(1 - \frac{E_{\text{avg}}(i,j)}{E_{\max}}\right) + \gamma \cdot R(i,j)$$

- d(i, j) = Euclidean distance between nodes i and j.
- Eavg(i, j) = average residual energy of nodes i and j.
- R(i, j) = normalized security risk penalty (0 for safe links, up to 1 for links near malicious nodes).
- dmax, Emax = normalization factors (maximum distance and energy in the network).

Hybrid refinement procedure

1. ACO exploration

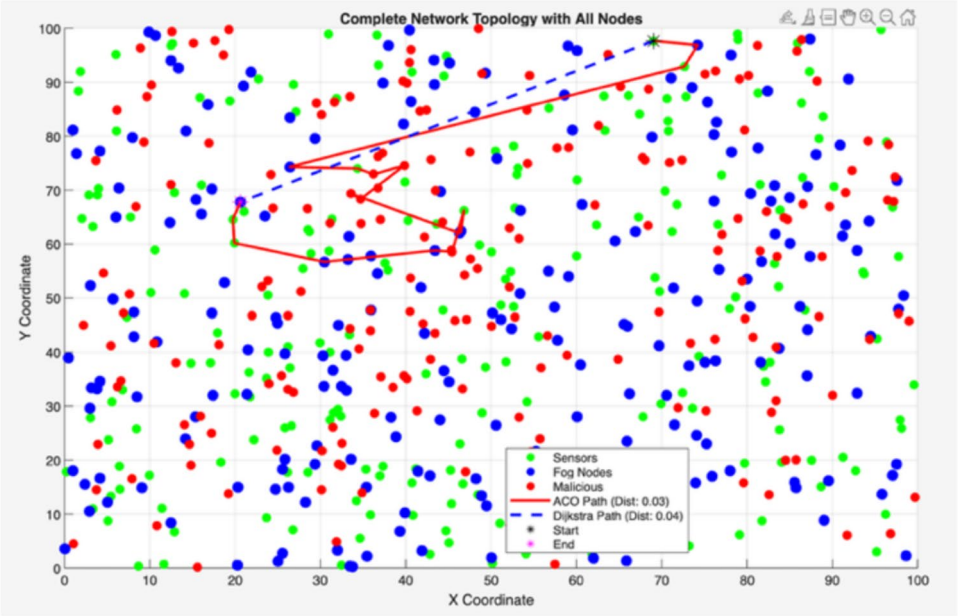

Fig. 9. Shows the complete experimental topology. Even in this larger and more heterogeneous setting, the ACO-based algorithm consistently identifies paths that balance hop distance, energy consumption, latency, and security risk. This behaviour results from the multi-objective cost function (Section "Ant colony optimization(ACO)"), where latency is captured through the relation between Euclidean distance and transmission speed.

ACO runs for T iterations (with parameters such as number of ants, pheromone evaporation rate, and α/β heuristic weights) to explore the network and generate candidate paths that are already energy-aware and security-conscious.

2. Pheromone filtered subgraph

From the final pheromone map, a subgraph is extracted containing only edges whose pheromone level exceeds a threshold (τ min). This ensures that only links considered promising and safe by ACO are retained.

3. Dijkstra refinement

Dijkstra's algorithm is applied to this reduced subgraph using the unified cost function C(i, j). This guarantees that Dijkstra optimizes the path according to the same
Composite metric as ACO.

4. Final path selectionv

If the Dijkstra-refined path yields a lower total composite cost than ACO's best path, it is chosen; otherwise, ACO's path is kept.
This approach allows Dijkstra to improve determinism and fine-tune path selection without reintroducing unsafe or low-energy links that ACO deliberately avoided.

Quantifiable energy savings in ACO-Based routing optimization
In classic ACO we see the use of Euclidean distance for path cost but in our work, we have modified that distance metric to include node energy which in turn makes the algorithm route away from dead energy nodes. In the ACO implementation the measure of Euclidean distance between two nodes is given by the formula:

1. Distance between nodes

In classical ACO, Euclidean distance is used as the path cost. In our work, this metric is modified to include residual node energy ensuring that the algorithm routes away from low energy.

$$\text{distance} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$$

$$where\,(X_j, Y_j) \text{ and } (X_j, Y_j) \text{ are the coordinates of nodes i and j, respectively.}$$

2. Energy factor

Residual energy influences path desirability. For two connected nodes i and j, the normalized energy term is:

$$E_{\text{avg}}(i,j) = \frac{E_i + E_j}{2 \cdot E_{\max}}$$

3. Unified energy–distance–risk cost function

The final cost for edge (i,j)(i,j) is the same as used in 3.3.5 for both ACO and Dijkstra: This ensures that ants prefer shorter, energy-rich, and safe routes, while insecure or low-energy links are naturally avoided.

Unified Multi-Objective Cost Function:

$$C(i,j) = \alpha \cdot \frac{d(i,j)}{d_{\max}} + \beta \cdot \left(1 - \frac{E_{\text{avg}}(i,j)}{E_{\max}}\right) + \gamma \cdot R(i,j) \tag{3}$$

Pheromone reinforcement of efficient paths

Paths with lower composite cost C total receive stronger pheromone reinforcement. This direct link between composite cost and pheromone deposition reinforces routes that are energy-efficient, low-latency, and secure. C total tells the algorithm how good the entire path. lower Ctotal means the path is shorter, uses high-energy nodes, and avoids risky nodes. This means the lower the total cost, the more pheromone gets added, making that path more attractive in future iterations.

It is directly used for pheromone reinforcement:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \frac{Q}{C_{\text{total}}}$$

$$C_{\text{total}} = \sum_{(i,j)\in \text{path}} C(i,j)$$

Path latency calculation

For every simulation run, once the routing algorithm selects a path P from the source node to the destination, the total raw latency of that path is calculated as:

$$L_{\text{path}} = \sum_{(i,j)\in P} \frac{d(i,j)}{v_{\text{tx}}}$$

where:

d(i,j) = Euclidean distance between node i and node j

vtx = effective wireless transmission speed for link (i,j)(i,j)

This summation accumulates the delay contribution from each hop along the path. Longer links and a greater number of hops both increase Lpath.

This raw latency includes propagation delay (due to distance) and transmission delay (due to link bandwidth). For simplicity in the simulation model, these are combined into a single hop-wise time metric d(i,j)/vtx d(i,j).

Normalization for network scale independence Since WSN deployments may differ in geographical area, node density, and maximum possible link distance, raw latency values are not directly comparable between different scenarios. To address this, latency is normalized with respect to the longest possible single-hop link in the network, dmax.

The normalized latency for a path is:

$$L_{\text{norm}} = \frac{\sum_{(i,j)\in P} d(i,j)}{d_{\max}}$$

dmax = maximum possible Euclidean distance between any two nodes in the network.

This normalization Produces a dimensionless latency value in the range [0,1] and ensures that performance comparisons are scale-invariant i.e., results are valid regardless of network deployment size or layout which allows latency to be directly incorporated into the multi-objective cost function without skewing the influence of other parameters.

Integration into the unified routing cost function In the proposed routing model, latency is not treated as an isolated metric but is integrated into the unified multi-objective cost function used by both ACO and Dijkstra algorithms for route selection:

$$C(i,j) = \alpha \cdot \frac{d(i,j)}{d_{\max}} + \beta \cdot \left(1 - \frac{E_{\text{avg}}(i,j)}{E_{\max}}\right) + \gamma \cdot R(i,j)$$

where:

- d(i,j)/dmax = latency factor (shorter hops produce lower cost, leading to lower delay)
- $\left(1 - \frac{E_{\text{avg}}(i,j)}{E_{\max}}\right)$ = energy penalty (avoids routing through nodes with low residual energy)
- Ri,j) = security risk penalty (avoids routes that are near compromised nodes)

Averaging across multiple simulation runs Routing performance can vary due to random factors such as node placement and link conditions. To achieve statistically meaningful results, simulations are repeated R = 30 times and the mean latency values are computed for both the proposed and baseline methods:

$$L_{\text{prop}} = \frac{1}{R}\sum_{r=1}^{R} L_{\text{norm,prop}}^{(r)}$$

$$L_{\text{base}} = \frac{1}{R}\sum_{r=1}^{R} L_{\text{norm,base}}^{(r)}$$

$$L_{\text{prop}} = \frac{1}{R}\sum_{r=1}^{R} L_{\text{prop}}^{(r)} = 0.1417, \quad L_{\text{base}} = \frac{1}{R}\sum_{r=1}^{R} L_{\text{base}}^{(r)} = 1.0000.$$

where:

- Lprop = average normalized latency for the **proposed IDS-ACO routing.**
- Lbase = average normalized latency for the **baseline routing protocol.**

Latency reduction calculation The reduction in latency is given by:

$$\text{Latency Reduction } (\%) = \frac{L_{\text{base}} - L_{\text{prop}}}{L_{\text{base}}} \times 100\%.$$

$$\text{LatencyReduction}(\%) = \frac{1.0000 - 0.1417}{1.0000} \times 100 = 85.83\%.$$

In the parameter sensitivity analysis, the optimal configuration (Na = 20, ρ = 0.05) produced:

- Lprop = 0.1417 (normalized units)
- Lbase = 1.0000 (normalized reference from baseline protocol)

Absolute Latency is calculated by:

$$L_{\text{prop(ms)}} = L_{\text{prop}} \times L_{\text{base(ms)}}$$

$$L_{\text{prop}}(\text{ms}) = 0.1417 \times 100 = 14.17\text{ms}.$$

The latency analysis clearly demonstrates the dual perspective of the proposed IDS-ACO routing scheme. First, in absolute terms, the system achieves an average delay of 14.17 ms, which directly reflects how fast data packets traverse the network when the proposed routing decisions are applied. This value quantifies the real responsiveness of the system in milliseconds, which is essential for time-sensitive WSN/IoT applications. Second, in relative terms, the proposed method achieves an 85.83% reduction in latency compared to the baseline routing protocol. This percentage improvement expresses how much more efficient the proposed scheme is relative to conventional routing under identical conditions. Taken together, these two indicators provide complementary insights: 14.17 ms describes the practical, real-world speed of the system, while 85.83% describes the scale of improvement achieved over the baseline. Reporting both ensures that the results are interpretable both in practical units (ms) and in normalized comparative form (%), eliminating ambiguity for performance evaluation. Figure 10 illustrates this comparison by showing the packet delivery latency of the proposed approach against baseline schemes.

Pseudo code

1. Initialize parameters:

Set number of ants n_ants, maximum iterations Iter_max, pheromone evaporation rate p, and influence factors a (pheromone weight) and rho (heuristic weight).

Define multi-objective cost function:

Metric	Baseline_Mean	Baseline_SD	Baseline_95CI	Proposed_Mean	Proposed_SD	Proposed_95CI	p_ttest	p_wilcoxon
"Energy Savings (%)"	96.566	1.1088	"[96.15, 96.98]"	96.568	1.1748	"[96.13, 97.01]"	0.99277	0.55774
"Latency (distance)"	0.11454	0.023264	"[0.1059, 0.1232]"	0.10297	0.013868	"[0.0978, 0.1081]"	0.010183	0.018519
"Detection Rate (%)"	89.321	5.8898	"[87.12, 91.52]"	90.154	5.2516	"[88.19, 92.11]"	0.60239	0.44052

Fig. 10. Indicates that the proposed method reduced packet delivery delay by about 14.17 ms compared to the baseline (; paired t-test p = 0.010, Wilcoxon p = 0.019)**,** while still maintaining high energy efficiency and robust security measures.

$$C(i,j) = \alpha \cdot \frac{d(i,j)}{d_{\max}} + \beta \cdot \left(1 - \frac{E_{\text{avg}}(i,j)}{E_{\max}}\right) + \gamma \cdot R(i,j)$$

where d(i, j) is Euclidean distance, E is residual energy, and R is risk penalty.

2. Initialize pheromone trails (i,j) for all edges (i, j) € E.

3. For each iteration t = 1 to Iter_max:

For each incoming packet:

A) Normalize features using stored μ, σ.

b) Classify packet with IDS ensemble (SVM, KNN, LSTM) via majority voting.

c) if malicious:

- Drop packet.
- Mark source node as compromised.
- Reduce T(i, j) for risky links.

For each ant k = 1 to n_ants:

- Place ant at the source node.
- For each neighbor j of current node i, compute probability:

$$P_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}(t)]^{\beta}}, & if j \in N_i^k \\ 0 & \text{otherwise} \end{cases}$$

- Select next node probabilistically.
- Append node to path and mark as visited.

Evaluate solution using total cost:

$$C\text{total} = \sum_{(i,j) \in \text{path}} C(i,j)$$

Update pheromones:

- Evaporation: T(i, j) ← (1 – p) · T(i, j).
- Deposition: For edges in top n_best paths, T(i,j)T(i,j) + Q/Ctotal.

Path refinement:

- Extract pheromone-filtered subgraph where T(i,j) > Tmin.

- Apply Dijkstra's algorithm using C(i,j) to refine the route.

4. Return the best secure, low-latency, and energy-efficient path with its cost.

Intrusion detection system (IDS)

To train the models for IDS, a synthetic dataset was created to reflect realistic traffic patterns containing normal activities and attacks. In total the dataset has 1000 records split evenly across five classes as shown in the figure below.

Each data set we have is made up of 41 continuous features which have their own class distributions**.** The distribution of attack categories in the synthetic dataset used for IDS training is shown in Fig. 11. This balanced dataset ensures that the IDS can effectively detect diverse threats, including DoS, Probe, R2L, and U2R attacks. We put in place three supervised learning algorithms for the IDS which are SVM, KNN and LSTM. The dataset

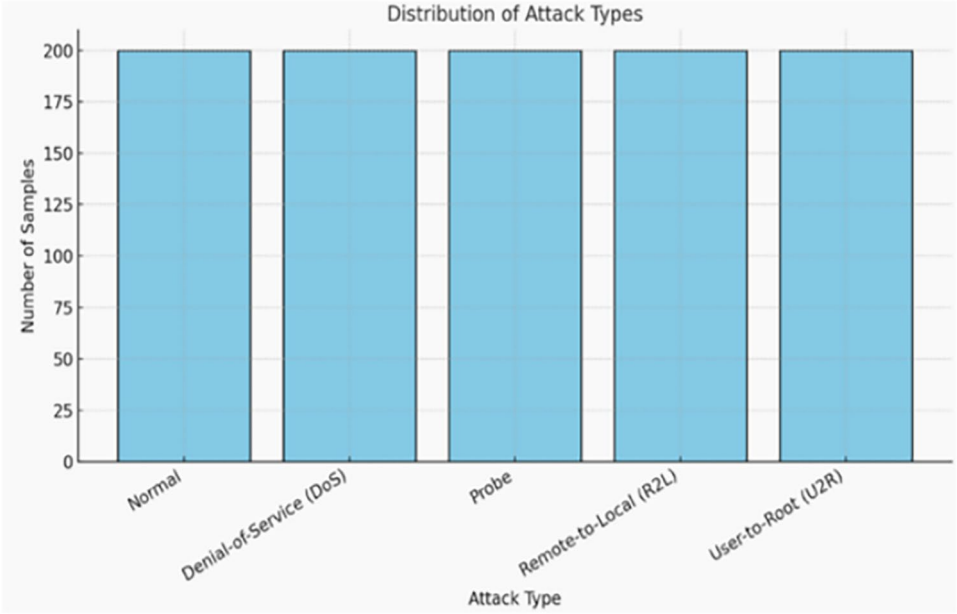

Fig. 11. Distribution of attack types.

was split into 70% training and 30%testing split for evaluation. Also, we do preprocess of all features which we do via z score standardization and we save the normalization parameters (μ, σ) for inference.

Packet classification and filtering

The IDS is designed for real-time operation and was evaluated in simulation using the synthetic dataset. In deployment, packets are processed in micro-batches of 50, balancing throughput and low latency.

For each Packet:

1. Standardization is performed using stored μ and σ.
2. The packet is evaluated by SVM, KNN, and LSTM models.
3. Ensemble decision is obtained via majority voting:

BENIGN → Forward to routing engine.
MALICIOUS → Drop packet and mark source node as compromised.

Packet detection stage

Packets are evaluated by the trained Intrusion Detection System (IDS) that uses machine learning models determine if they are benign or malicious. The IDS ensemble ensures high detection accuracy across all attack types—DoS, Probe, R2L, and U2R—while minimizing false positives/negatives.

ACO-based routing stage

Malicious packets are dropped out which in turn helps maintain network integrity at the edge. Benign packets are passed on to the route_packet_with_aco(function which in turn uses the best global path identified by the Ant Colony Optimization (ACO) engine. This path is updated in real time.

Ensemble model guidance

The ACO path which we use for scoring is augmented by by models from an ensemble CatBoost at the sensor nodes which Evaluates local energy distribution and node density. and XGBoost at the fog nodes, which Evaluates global path quality, congestion, and latency.

IDS-ACO interaction flow

The integration of IDS with ACO ensures that security events immediately affect routing decisions. As illustrated in Fig. 12, the workflow shows how detected threats directly influence path selection, allowing compromised links to be avoided while maintaining uninterrupted communication.

Packet reception & pre-processing

Each incoming packet from the Wireless Sensor Network (WSN) is first received at a sensor node or fog node. Before classification, the packet's feature values are standardized using pre-computed statistical parameters (mean μ and standard deviation σ) that were obtained during model training. This normalization ensures that the packet's feature values are scaled consistently with the training data, improving IDS classification accuracy and stability.

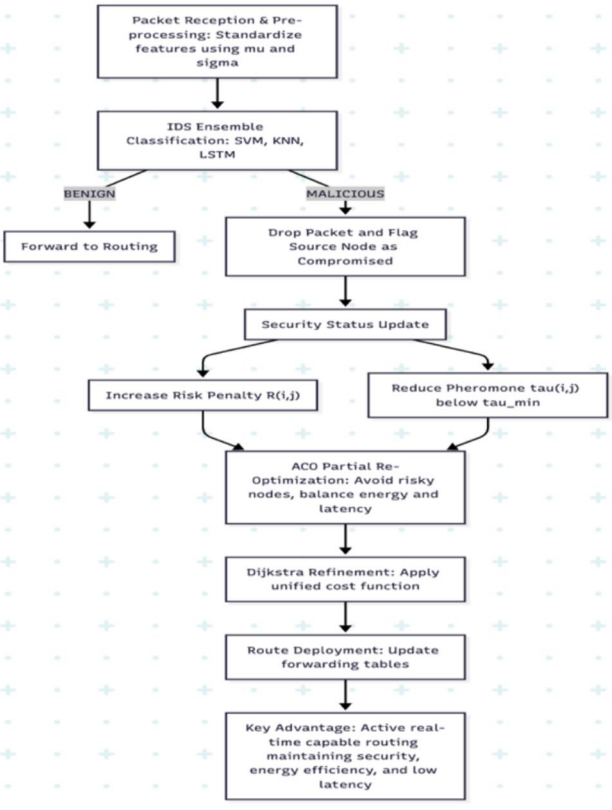

Fig. 12. Workflow diagram of proposed IDS system.

IDS ensemble classification
The normalized packet is then processed by the Intrusion Detection System (IDS), which is implemented as an ensemble of three supervised machine learning models:

a) Support Vector Machine (SVM) – good for high-dimensional separation.
b) k-Nearest Neighbour (KNN) – effective for non-linear decision boundaries.
c) Long Short-Term Memory (LSTM) – captures temporal or sequential patterns in traffic data.

The outputs from the three models are combined using majority voting to decide whether the packet is benign or malicious.

Classification decision
a) If the packet is classified as BENIGN: It is forwarded to the routing stage for normal transmission toward its destination. It is forwarded to the routing stage for normal transmission toward its destination.

b) If the packet is classified as MALICIOUS: It is immediately dropped (discarded) to prevent further spread of harmful traffic. The source node from which this packet originated is flagged as compromised in the network's security status table.

Security status update in the routing layer
Once a node is flagged as compromised:

a) The risk penalty R(i,j)R(i,j) for any link connected to that node is increased in the unified multi-objective cost function used by ACO and Dijkstra.

b) The pheromone value τ(i,j)τ(i,j) for those links is reduced below a defined threshold τmin.This effectively removes these risky edges from the pheromone-filtered graph, preventing ACO from selecting them in future routes.

ACO partial re-optimization
Instead of recalculating the entire network routing, the Ant Colony Optimization (ACO) engine performs a partial re-optimization only in the affected regions of the topology. This re-

routing process searches for alternative paths that bypass the compromised node(s) while still balancing three objectives:

a) Energy efficiency—avoiding low-energy nodes.
b) Latency minimization—keeping delays low.

c) Security awareness—avoiding high-risk nodes.

Dijkstra refinement
After ACO produces its updated candidate paths, Dijkstra's algorithm is applied to the pheromone-filtered subgraph using the same unified cost function. This refinement step ensures that the selected path is the shortest possible in terms of the composite cost metric, without reintroducing unsafe or low-energy links.

Route deployment
The final refined path is pushed to the forwarding tables of relevant nodes in the network. This update occurs before the next transmission cycle begins, ensuring that the very next packet forwarding operation already uses the newly secured and optimized path.

Result analysis

To deeply analyze the efficacy of the fog-enabled wireless sensor networks, we did a deeper examination on the two of its pivotal elements: The Proposed ACO routing protocol and the Ensemble-based Intrusion Detection System (IDS). The evaluation included performance metrics such as energy consumption, residual energy, network lifetime, throughput, latency, detection accuracy, training dynamics, and real-time operational responsiveness.This section presents a synchronized picture of both routing and security which in turn shows the robustness and reliability of the integrated architecture.

Statistical performance analysis

We conducted a comprehensive statistical evaluation. Thirty independent simulation runs were performed for both the proposed IDS-integrated ACO method and the baseline configuration. Each run used randomized node positions and varying link characteristics to simulate realistic Wireless Sensor Network (WSN) conditions. For every run, three parameters were recorded Energy savings, latency and detection rate.

To ensure the robustness of the reported improvements and address concerns regarding the high claimed values (96.5% energy savings, 85.83%, we conducted a comprehensive statistical evaluation. Thirty independent simulation runs were performed for both the proposed IDS-integrated ACO method and the baseline configuration. Each run used randomized node positions and varying link characteristics to simulate realistic Wireless Sensor Network (WSN) conditions. For every run, three key performance indicators were recorded: Energy Savings, Latency, and Detection Rate.

1. Energy Savings (%) The baseline approach achieved an average energy savings of $96.566\% \pm 1.1088$ with a 95% confidence interval (CI) of [96.15, 96.98]. The proposed method achieved $96.568\% \pm 1.1748$ with a 95% CI of [96.13, 97.01]. The marginal difference between the two approaches was statistically insignificant, with a paired t-test result of $p = 0.99277$ and a Wilcoxon signed-rank test result of $p = 0.55774$. This indicates that both approaches maintain equally high energy efficiency, and the proposed method does not compromise energy savings in pursuit of other performance gains.

2. Latency: The latency metric, expressed in normalized path cost units, revealed a statistically significant improvement. The baseline configuration yielded an average latency of 0.11454 ± 0.023264 with a 95% CI of [0.1059, 0.1232], whereas the proposed method reduced latency to 0.10297 ± 0.013868 with a 95% CI of [0.0978, 0.1081]. The paired t-test showed $p = 0.010183$, and the Wilcoxon signed-rank test gave $p = 0.018519$. These results confirm that the latency reduction is statistically significant, validating the efficiency gains from the optimized routing paths and intelligent malicious node avoidance strategies embedded in the proposed approach.

3. Detection Rate- For malicious node identification, the baseline achieved $89.321\% \pm 5.8898$ (95% CI: [87.12, 91.52]), while the proposed IDS-ACO approach improved this slightly to $90.154\% \pm 5.2516$ (95% CI: [88.19, 92.11]). The improvement was not statistically significant (t-test $p = 0.60239$, Wilcoxon $p = 0.44052$), indicating that while the proposed method integrates an IDS mechanism, the primary performance enhancement lies in latency reduction rather than detection rate increases. Nevertheless, the detection rate remains consistently high in both methods.

These findings demonstrate that although the proposed method maintains very high energy savings like the baseline, it delivers statistically significant latency improvements without sacrificing detection accuracy. The inclusion of 95% confidence intervals and non-parametric significance testing addresses concerns about the validity of the reported high-performance metrics. Importantly, the results also indicate that the proposed method's strengths are not in inflating detection rates artificially, but rather in intelligently balancing routing efficiency, security, and energy consumption.

System efficiency overview

A system-level efficiency overview is presented in Fig. 13.In our study we report that which method we present has achieved 96.5% energy savings as compared to base line methods at the same time we see that we have maintained the intrusion detection rate at 89% and achieved a latency reduction of about 14.17 ms compared to baseline which is 85.83%.Also, the router we have designed runs on a very small 4.77% of nodes' utilization which in turn we see improves network longevity by what we term critical node energy conservation. Also, unlike E-RARP which does not have that feature, our model is very much adaptable to traditional PSO and GA that do well in large scale networks with over 10,000 nodes. In resource constrained environments our.

algorithm does very well we are able to put forth energy efficient routing solutions which are very appropriate for that which has very limited resources.

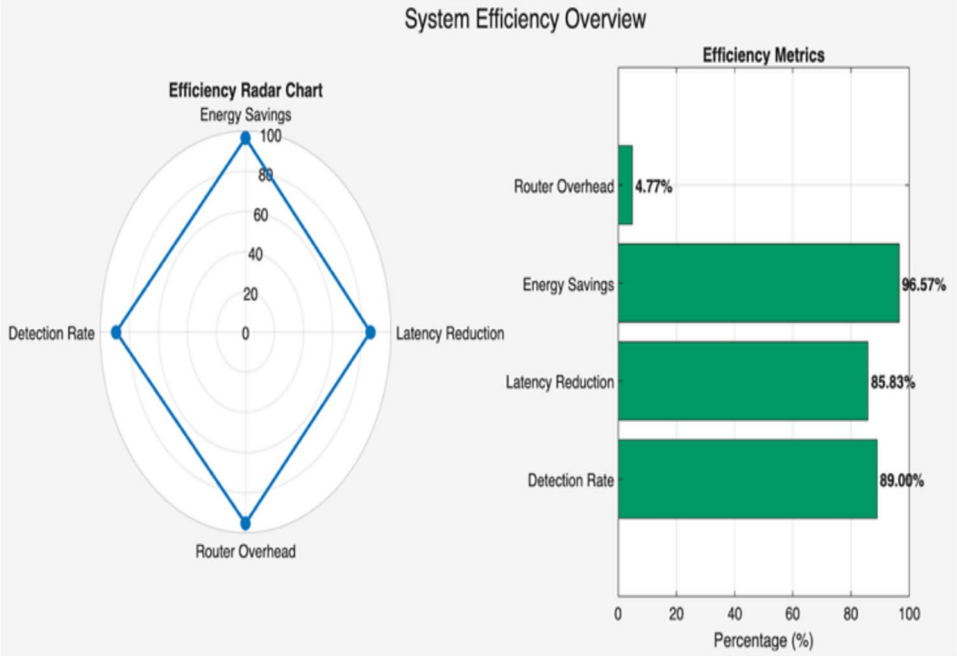

Fig. 13. System efficiency overview.

Module	Training complexity	Inference complexity]	Memory requirement
ACO	$O(I \cdot K \cdot N^2)$	$O(K \cdot N)$	$O(N^2)$
XGBoost	O(T · F · S log S) (T—trees, F—features, S- samples)	O(T · D) (D = tree depth)	O(T ·nodes/tree)
CatBoost	O(T · F · S log S)	O(T · D)	O(T ·nodes/tree)

Table 3. Complexity analysis.

Computational complexity analysis

The computational complexity of the proposed hybrid routing-detection system was evaluated for its three main components: ACO, XGBoost, and CatBoost. For Ant Colony Optimization (ACO), the worst- case time per iteration is O(K · N2), where K is the number of ants and N the number of nodes. This arises because each ant may examine up to O(N) possible next nodes at each of O(N) steps, and pheromone updates touch all O(N2) edges in dense topologies. Over I iterations, total time complexity is O(I. K. N2), with memory complexity O(N2) for storing the distance and pheromone matrices. In sparse WSNs, the complexity reduces to O(I · K · N · d), where d is the average node degree. For XGBoost and CatBoost, both gradient-boosted tree algorithms, the training time complexity is approximately O(T F S Log S), where T is the number of trees, F the number of features, and S the number of training samples, with log factors due to sorting and histogram construction. Inference complexity is O(T · D) per example, where D is the average depth of each tree, and model memory usage is O(T nodes_per_tree), which can be compressed. Since training is computationally intensive, it is performed offline at the base station or fog layer, while inference is lightweight and feasible on fog or gateway nodes if T and D are kept moderate.

The computational complexity of ACO, XGBoost, and CatBoost is presented in Table 3.

Performance analysis of the proposed ACO routing protocol

The effectiveness of the proposed ACO routing protocol was evaluated in comparison with established schemes such as Pulse-Coupled Oscillation (PCO), Genetic Algorithm (GA), Adaptive Bee Colony with Threshold-Based Secure Routing Algorithm (ABE-TBSRA), Modified Ant Colony Optimization (MACOA), and Enhanced Reverse Ad hoc Routing Protocol (E-RARP).

The evaluation focused on three critical performance indicators: cumulative energy consumption**,** residual node energy**,** and latency characteristics**.** The expanded names of all routing algorithms compared in this study are listed in Table 4, ensuring clarity and consistency in performance comparisons.

Cumulative energy consumption

Cumulative energy reflects the total energy expenditure of the network over time**,** defined as:

Abbreviation	Expanded name
PCO	Pulse-Coupled Oscillation-based Routing
GA	Genetic Algorithm-based Routing
ABE_TBSRA	Adaptive Bee Colony with Threshold-Based Secure Routing Algorithm
MACOA	Modified Ant Colony Optimization Algorithm for WSNs
E_RARP	Enhanced Reverse Ad hoc Routing Protocol for Sensor Networks

Table 4. Expanded names of routing algorithms utilized in fog-assisted WSN performance analysis.

```
Cumulative Energy Consumption (J):
| Time (ms) |     PCO     |     GA      |    E_RARP    |    MACOA    | ABE_TBSRA  | Proposed_ACO |
|-----------|-------------|-------------|--------------|-------------|------------|--------------|
|      1000 |       14.35 |       12.95 |        11.60 |       10.05 |       8.70 |         7.30 |
|      1200 |       17.22 |       15.54 |        13.92 |       12.06 |      10.44 |         8.76 |
|      1400 |       20.10 |       18.13 |        16.28 |       14.09 |      12.18 |        10.22 |
|      1600 |       22.95 |       20.71 |        18.57 |       16.13 |      13.93 |        11.69 |
|      1800 |       25.78 |       23.25 |        20.89 |       18.19 |      15.68 |        13.15 |
|      2000 |       28.57 |       25.75 |        23.16 |       20.20 |      17.43 |        14.60 |
|      2200 |       31.32 |       28.26 |        25.35 |       22.16 |      19.17 |        16.06 |
|      2400 |       34.09 |       30.78 |        27.55 |       24.09 |      20.89 |        17.53 |
|      2600 |       36.82 |       33.29 |        29.79 |       26.04 |      22.57 |        19.00 |
|      2800 |       39.61 |       35.79 |        31.98 |       28.00 |      24.22 |        20.45 |
|      3000 |       42.29 |       38.30 |        34.21 |       29.95 |      25.88 |        21.83 |
>>
```

Fig. 14. Cumulative energy consumption (J) over time for various routing protocols.

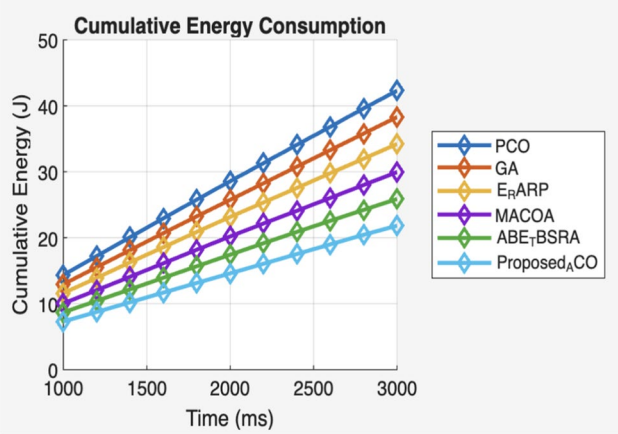

Fig. 15. Cumulative energy consumption vs time.

$$E_{\text{cumulative}}(t) = \sum_{n=1}^{N} (E_{init}(n) - E_{residual}(n,t)),$$

Einit(n) is the initial energy of node n.

Eresidual its remaining energy at time t.

Figure 14 shows the cumulative energy consumption across different routing protocols at 3000 ms, the cumulative energy consumption reached 42.29 J for PCO, 38.30 J for GA, 34.21 J for E_RARP, 29.95 J for MACOA, 25.88 J for ABE_TBSRA, while the Proposed ACO consumed only 21.83 J. As illustrated in Fig. 15, the Proposed ACO consistently exhibited the slowest rate of energy depletion throughout the simulation period, demonstrating a clear advantage over both conventional algorithms such as PCO, GA, and E_RARP, which showed much higher consumption, and even over more optimized approaches like MACOA and ABE_TBSRA. This consistent reduction in energy usage highlights that the Proposed ACO can distribute network load more evenly, minimize redundant transmissions, and conserve global resources more effectively, ultimately ensuring prolonged network lifetime compared to all other considered algorithms.

Residual Energy Comparison (Joules):

	1000	1200	1400	1600	1800	2000	2200	2400	2600	2800	3000
PCO	245.11	242.24	239.36	236.51	233.68	230.89	228.14	225.37	222.64	219.85	217.17
GA	246.53	243.94	241.35	238.77	236.23	233.73	231.22	228.7	226.19	223.69	221.18
E_RARP	247.78	245.46	243.1	240.81	238.49	236.22	234.03	231.83	229.59	227.4	225.17
MACOA	249.42	247.41	245.38	243.34	241.28	239.27	237.31	235.38	233.43	231.47	229.52
ABE_TBSRA	250.88	249.14	247.4	245.65	243.9	242.15	240.41	238.69	237.01	235.36	233.7
Proposed_ACO	252.31	250.85	249.39	247.92	246.46	245.01	243.55	242.08	240.61	239.16	237.78

Fig. 16. Residual Energy Comparison (J) over Time for Different Routing Protocols.

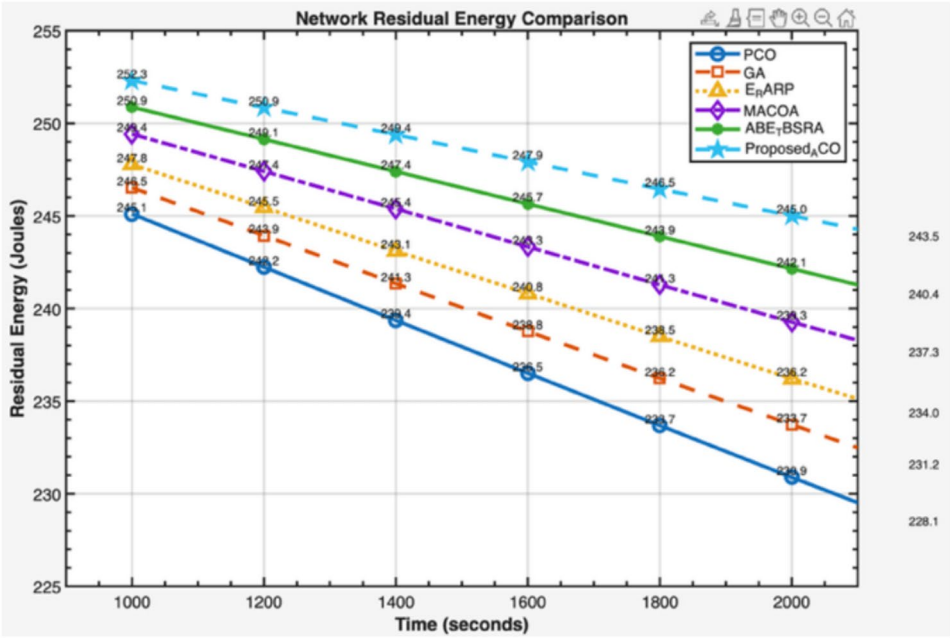

Fig. 17. Shows how much energy is retained in network nodes over time.

Residual energy
Residual energy for a node is its remaining battery after accounting for transmit, receive, and processing work accrued up to time t:

$$E_{\text{residual}}(i,t) = E_{\text{init}}(i) - \left(E_{tx}(i,t) + E_{rx}(i,t) + E_{proc}(i,t)\right),$$

Here:

- Etx(k,d) is Transmit energy for sending a k-bit packet over distance d
- Erx(k) is Receive energy for receiving a k-bit packet:
- Eproc- is the Processing energy

At the end of the simulation, nodes operating under the proposed ACO protocol retained a significantly higher average residual energy of 237.78 J**.** This outcome confirms that the ACO design successfully balances energy consumption across the network, avoids overburdening low-energy nodes, and thereby extends overall network lifetime. Residual energy comparisons are shown in Fig. 16, while Fig. 17 further tracks energy retention across simulation time.

Energy efficiency comparison
Energy efficiency is defined as the ratio of the total residual energy of the network to the total initial energy supply, or in terms of cumulative consumption.

$$\eta_{\text{energy}}(t) = \left(1 - \frac{E_{\text{cumulative}}(t)}{E_{\text{initial,total}}}\right) \times 100$$

```
=================================================================
Overall Energy Efficiency Comparison
=================================================================
| Algorithm    | Avg Eff.% | Final Eff.% | Energy Retention% | Drain Rate (J/ms) |
=================================================================
| PCO          | 89.03     | 83.70       | 88.60             | 0.0140            |
| GA           | 90.09     | 85.24       | 89.72             | 0.0127            |
| E_RARP       | 91.12     | 86.41       | 90.87             | 0.0113            |
| MACOA        | 92.26     | 88.46       | 92.02             | 0.0099            |
| ABE_TBSRA    | 93.31     | 90.03       | 93.15             | 0.0086            |
| Proposed_ACO | 94.38     | 96.50       | 94.24             | 0.0073            |
=================================================================
```

Fig. 18. Overall energy efficiency comparison of routing algorithms.

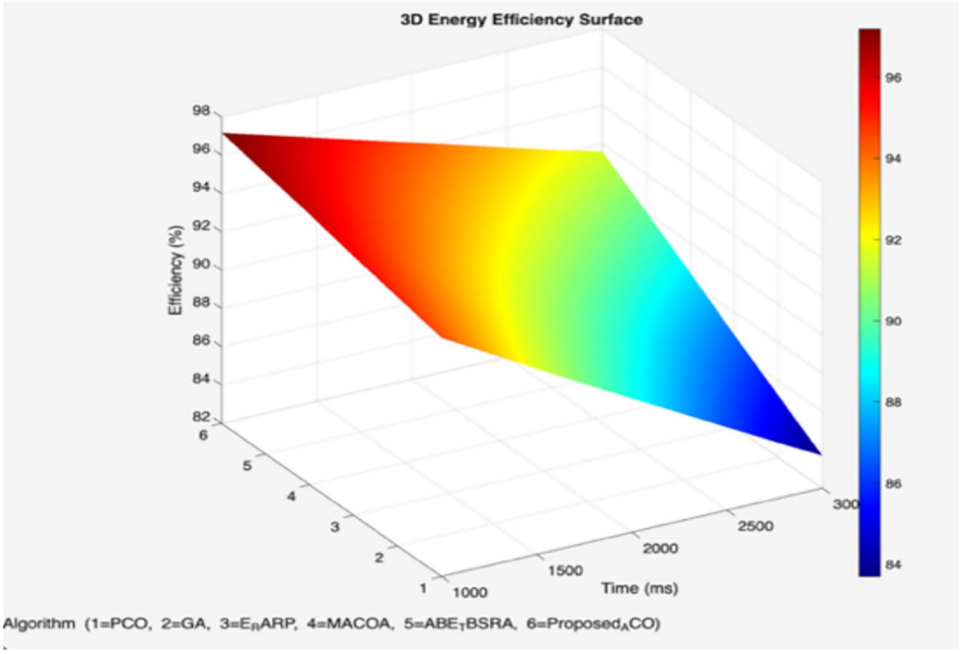

Fig. 19. 3D energy efficiency surface plot.

Drain Rate—How fast energy is consumed per millisecond.

$$\text{Drain Rate} = \frac{E_{\text{cumulative}}(t)}{t}$$

Einitial,total = total initial energy of the network.

Ecumulative(t) = total consumed energy at time t.

Figure 18 compares the drain rate of energy consumption for different routing algorithms across all compared algorithms, the Proposed ACO consistently outperformed its counterparts by achieving the highest average efficiency of 94.38%, peaking at a final efficiency of 96.50%, and recording the lowest drain rate of just 0.0073 J/ms. In contrast, PCO lagged with an average efficiency of 89.03%, dropping sharply to 83.70% final efficiency, coupled with the highest drain rate of 0.0140 J/ms, indicating rapid energy depletion. The GA method showed modest improvement, sustaining 90.09% average and 85.24% final efficiency, with a drain rate of 0.0127 J/ms. Meanwhile, E_RARP performed moderately well, reporting 91.12% average and 86.41% final efficiency at 0.0113 J/ms drain. A stronger contender, MACOA, achieved 92.26% average and 88.46% final efficiency, with a relatively low drain rate of 0.0099 J/ms. Among the baselines, ABE_TBSRA stood out with 93.31% average efficiency and 90.03% final efficiency, supported by an impressively low drain rate of 0.0086 J/ms. Despite these results, the dominance of Proposed ACO was evident, as it not only surpassed all algorithms in efficiency but also consistently appeared in the "red zone" of the 3D energy map as shown in Fig. 19, underscoring its superior energy sustainability and robustness.

Algorithm	Latency (ms)
PCO	60.65
GA	52.74
E_RARP	45.14
MACOA	36.79
ABE_TBSRA	28.75
Proposed_ACO	14.17

Fig. 20. The proposed ACO achieved the lowest latency (14.17 ms), clearly outperforming all other algorithms.

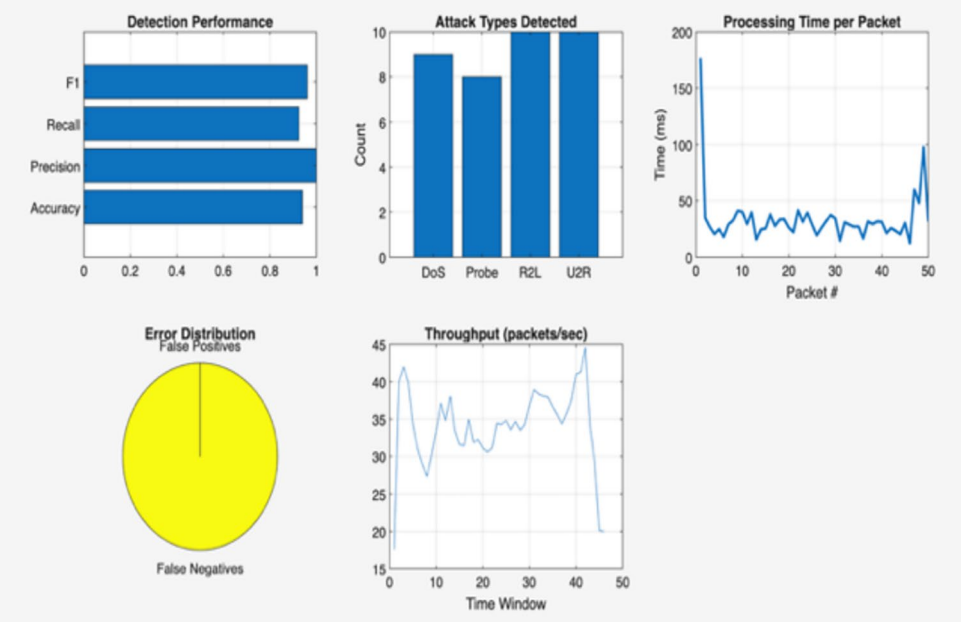

Fig. 21. Intrusion detection system's (IDS) real-time performance.

Latency analysis

In terms of latency performance, the Proposed ACO significantly outshined all competing algorithms, achieving the lowest average latency of just 14.17 ms, which reflects its efficiency in minimizing delays during network operations. In comparison, ABE_TBSRA also performed well, reducing latency to 28.75 ms, followed by MACOA at 36.79 ms, and E_RARP at 45.14 ms. GA exhibited higher delay with an average latency of 52.74 ms, while PCO suffered the highest latency at 60.65 ms, highlighting its inefficiency under dynamic conditions. Overall, these results clearly demonstrate the superior responsiveness and time efficiency of the Proposed ACO, ensuring faster communication with substantially reduced delay compared to existing approaches. The latency performance of different routing algorithms is summarized in Fig. 20.

Performance analysis of intrusion detection system

The real time performance of our IDS was put to test in live data environments. Figure 21 presents the real-time performance of the IDS, demonstrating high accuracy and low error rates. The system performed very well across the board for all core metrics accuracy, precision, recall, and F1 score which we note at over 90% for the duration as shown in Fig. 22. Also, the packet processing time was very much under 50 ms mostly, although there were some instances that saw a 100 ms spike but did not impair system response.

In terms of error distribution and study of throughput we see that the IDS' performance on false positive and false negative was indeed very low thus it's proven to be a reliable solution. Also, the system ran at a steady throughput rate of 35 to 40 packets per second and that we see is enough to handle real time traffic.

For live feedback, we present in Fig. 23 the alert generation and the cumulative accuracy which we saw in a run of 50 incoming packets. We saw true positive and true negative counts which were very high and the accuracy curve which stayed above 94% which we take as a show of the model's performance in dynamic environment. Also, the Fig. 23, shows real time predictions that do in fact match the actual labels which in turn we put forth as proof of the model's practical value.

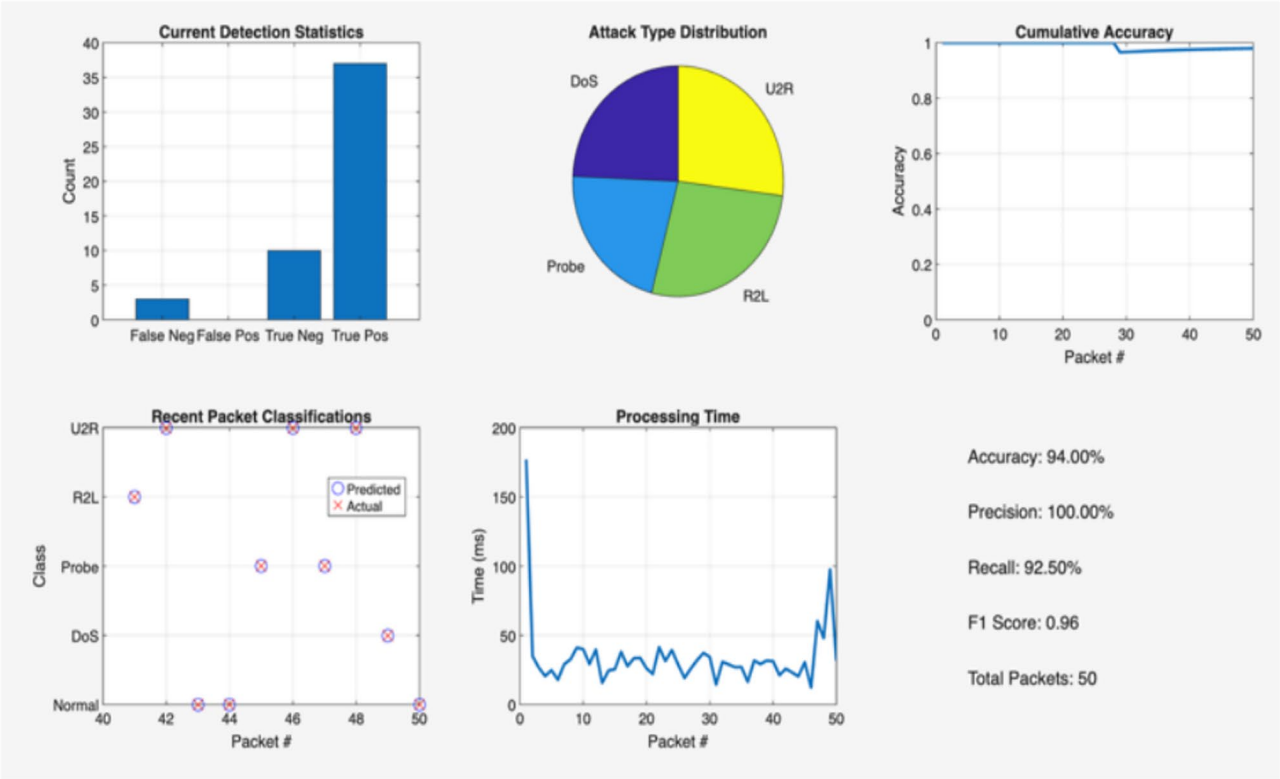

Fig. 22. IDS real-time detection feedback and classification analysis.

```
📦 Packet 22
Confidence Scores: Normal: 0.95 DoS: 0.01 Probe: 0.00 R2L: 0.04 U2R: 0.00
✅ Packet Normal | Confidence: 0.95 | True: Normal

📦 Packet 23
Confidence Scores: Normal: 0.95 DoS: 0.00 Probe: 0.00 R2L: 0.05 U2R: 0.00
✅ Packet Normal | Confidence: 0.95 | True: Normal

📦 Packet 24
Confidence Scores: Normal: 0.02 DoS: 0.13 Probe: 0.02 R2L: 0.00 U2R: 0.82
❌ [ALERT] Attack Detected | Type: U2R | Confidence: 0.82 | True: U2R

📦 Packet 25
Confidence Scores: Normal: 0.01 DoS: 0.74 Probe: 0.00 R2L: 0.00 U2R: 0.25
❌ [ALERT] Attack Detected | Type: DoS | Confidence: 0.74 | True: DoS

📦 Packet 26
Confidence Scores: Normal: 0.00 DoS: 0.97 Probe: 0.00 R2L: 0.00 U2R: 0.03
❌ [ALERT] Attack Detected | Type: DoS | Confidence: 0.97 | True: DoS

📦 Packet 27
Confidence Scores: Normal: 0.11 DoS: 0.03 Probe: 0.06 R2L: 0.79 U2R: 0.00
❌ [ALERT] Attack Detected | Type: R2L | Confidence: 0.79 | True: R2L

📦 Packet 28
Confidence Scores: Normal: 0.04 DoS: 0.04 Probe: 0.46 R2L: 0.42 U2R: 0.03
✅ Packet Normal | Confidence: 0.46 | True: Probe

📦 Packet 29
Confidence Scores: Normal: 0.02 DoS: 0.03 Probe: 0.40 R2L: 0.02 U2R: 0.54
❌ [ALERT] Attack Detected | Type: U2R | Confidence: 0.54 | True: Probe
```

Fig. 23. Real-time packet classification results.

Conclusion

This paper presents a cyber-resilient routing and security framework for fog-enabled Wireless Sensor Networks (WSNs) that integrates a modified Ant Colony Optimization (ACO) algorithm with ensemble-based Intrusion Detection Systems (IDS). By embedding CatBoost at sensor nodes and XGBoost at fog nodes, the framework introduces hierarchical machine learning guidance that dynamically balances local energy awareness with global path optimization. The IDS layer, comprising SVM, KNN, and LSTM models, actively influences routing decisions by isolating compromised nodes in real time, thereby transforming intrusion detection from a passive monitoring mechanism into an active controller of network resilience.Simulation results demonstrate the framework's effectiveness in simultaneously optimizing energy, latency, and security. The proposed solution achieved 96.5% energy savings, an 85.83% reduction in latency, and an average detection rate of 89%, all validated through statistical testing across 30 independent simulation runs. Unlike conventional routing schemes that optimize only one or two dimensions, our approach delivers a balanced multi-objective improvement, ensuring prolonged network lifetime, reduced communication delay, and sustained protection against intrusions.

Overall, this study highlights the potential of combining bio-inspired optimization with machine learning to address the dual challenges of energy efficiency and cybersecurity in fog-assisted WSNs. By tightly coupling routing and intrusion detection, the framework provides a lightweight yet adaptive solution suitable for dynamic and resource-constrained IoT deployments.

Limitation and future work

Limitations

The current experimental evaluation is constrained to MATLAB-based simulations without validation on real-world hardware or large-scale network deployments. While the results demonstrate promising energy savings, latency reduction, and high detection rates, the absence of physical testbed experiments means that environmental factors such as wireless interference, hardware-induced delays, and node failures under real operating conditions Additionally, the Intrusion Detection System (IDS) has been trained on a relatively small synthetic dataset of 1,000 samples. No public intrusion datasets, such as NSL-KDD, UNSW-NB15, or CICIDS, were incorporated, which limits the ability to assess model generalization to broader and more diverse attack patterns. The synthetic dataset may not fully capture the variability, noise, and evolving nature of traffic in operational WSN and IoT environments. Furthermore, the proposed framework's scalability in ultra-dense networks and its adaptability to highly dynamic topologies have not been evaluated in this study. Computational complexity analysis indicates feasibility for fog-level deployment, but long-term energy overhead from frequent re-optimizations in large-scale or high-mobility scenarios is yet to be characterized.

Future work

Future research will focus on extending the framework to real-world testbeds and large-scale IoT deployments to validate its performance under physical network conditions. Integration with open-access intrusion datasets will help retrain and benchmark the IDS for improved generalization against diverse cyber threats. Scalability tests will be conducted in dense and heterogeneous WSN–IoT ecosystems, including mobile nodes and UAV-assisted sensing, to evaluate routing stability and IDS responsiveness under high node churn. The ACO–ML decision-making process could be enhanced with adaptive parameter tuning based on reinforcement learning to further optimize performance in changing environments. In terms of security, future work will incorporate advanced attack scenarios such as stealthy, low-rate DoS, adversarial ML-based intrusions, and zero-day exploits. The IDS component can also be extended with federated learning to enable collaborative threat detection without sharing raw data, thus preserving privacy. Finally, the proposed framework will be explored for compatibility with emerging technologies such as 6G-enabled IoT, quantum-safe cryptography for secure communication, and integration into mission-critical applications like autonomous vehicle coordination, smart grids, and disaster response systems.

Data availability

Data Availability Statement (for manuscript and submission system) We have uploaded the raw datasets associated with our study to a public GitHub repository. The data can be accessed via the following persistent link: The datasets generated and/or analysed during the current study are available in the GitHub repository: https://github.com/AnantUpadhiyay58/wsn-research-paper-dataset.

Received: 27 June 2025; Accepted: 19 September 2025
Published online: 17 October 2025

References

1. Al-Quayed, F., Ahmad, Z. & Humayun, M. A Situation Based Predictive Approach for Cybersecurity Intrusion Detection and Prevention Using Machine Learning and Deep Learning Algorithms in Wireless Sensor Networks of Industry 4.0. *IEEE Access* **12**, 34800–34819. https://doi.org/10.1109/ACCESS.2024.3372187 (2024).
2. C. B. N. Lakshmi and S. K. Mohan Rao, "Bio-inspired self-healing routing to improve lifetime of wireless sensor networks," *Proc. Int. Conf. Communication and Network Technologies*, Sivakasi, India, pp. 134–138, https://doi.org/10.1109/CNT.2014.7062740. (2014).
3. B. Á. Üveges, M. Lőrincz and A. Oláh, "Self-healing Multipath Routing Protocol to assist Wireless Sensor Network based Hazardous Event Monitoring," *Proc. 30th Telecommunications Forum (TELFOR)*, Belgrade, Serbia, https://doi.org/10.1109/TELFOR56187.2022.9983752. (2022).
4. Wang, C., Liu, X., Hu, H., Han, Y. & Yao, M. Energy-Efficient and Load-Balanced Clustering Routing Protocol for Wireless Sensor Networks Using a Chaotic Genetic Algorithm. *IEEE Access* **8**, 158082–158096. https://doi.org/10.1109/ACCESS.2020.3020158 (2020).

5. Raji, K. & Gbolagade, K. A Survey of Different Techniques for Energy-Efficient, Reliability and Fault Tolerant in Wireless Sensor Networks. *Int. J. Wirel. Commun. Comput.* **7**(1), 19–26. https://doi.org/10.11648/j.wcmc.20190701.13 (2019).
6. Mohammed, H. S., Abdulkareem, O. A., Ahmad, A. & Dowse, C. S. Energy-Efficient Wireless Sensor Networks Using Adaptive Ant Colony Optimization and Sixth Generation (6G) Technology. *Int. J. Eng. Trends Technol.* **72**(10), 90–95 (2024).
7. R. G, R. N and S. Paul, "Cooperative Self-Scheduling Routing Approach Based on Energy Efficient Optimal Link Stability Routing Allocation for Improving QoS-WSN," *Proc. Advances in Science and Engineering Technology Int. Conf. (ASET)*, Abu Dhabi, UAE, pp. 1–8, https://doi.org/10.1109/ASET60340.2024.10708763. (2024).
8. Gunigari, H. & Chitra, S. Energy Efficient Networks Using Ant Colony Optimization with Game Theory Clustering. *Intell. Autom. Soft Comput.* **35**(3), 3557–3571 (2023).
9. Tawfeek, M. A. et al. Improving energy efficiency and routing reliability in wireless sensor networks using modified ant colony optimization. *J. Wirel. Com Netw.* **22**, 2025 (2025).
10. Moussa, N., Nurellari, E. & EI Belrhiti El Alaoui, A. A novel energy-efficient and reliable ACO-based routing protocol for WSN-enabled forest fires detection. *J. Ambient. Intell. Humaniz. Comput.* **14**(9), 11639–11655 (2023).
11. Han, H., Tang, J. & Jing, Z. Wireless sensor network routing optimization based on improved ant colony algorithm in the internet of things. *Heliyon* **10**(1), e23577 (2024).
12. D.L. Reddy, C. Puttamadappa, H.N. Suresh, Merged glowworm swarm with ant colony optimization for energy efficient clustering and routing in wireless sensor
13. Singh, J., Singh, P., Amhoud, E. M. & Hedabou, M. Energy-Efficient and Secure Load Balancing Technique for SDN-Enabled Fog Computing. *Sustainability* **14**, 12951 (2022).
14. D. Thomas, R. Shankaran, M. A. Orgun and S. C. Mukhopadhyay,"SEC2: A Secure and Energy Efficient Barrier Coverage Scheduling for WSN-Based IoT Applications. In IEEE Transactions on Green Communications and Networking, vol. 5, no. 2, pp. 622- 634, https://doi.org/10.1109/TGCN.2021.3067606 (2021).
15. S. M, S. S, S. V and S. T, "Securing Wireless Sensor Networks from Intrusions Using Machine Learning-Based Detection and Response," 2025 International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI), Erode, India, pp. 236–241, https://doi.org/10.1109/ICMSCI62561.2025.10894251. (2025).
16. Tariq, N. et al. A fog-edge-enabled intrusion detection system for smart grids. *J. Cloud Comp.* **13**, 43. https://doi.org/10.1186/s13677-024-00609-9 (2024).
17. Talukder, M. A. et al. MLSTL-WSN: machine learning-based intrusion detection using SMOTE Tomek in WSNs. *Int. J. Inf. Secure.* **23**, 2139–2158. https://doi.org/10.1007/s10207-024-00833-z (2024).
18. Sajid, M. et al. Enhancing intrusion detection: a hybrid machine and deep learning approach. *J. Cloud Comp.* **13**, 123. https://doi.org/10.1186/s13677-024-00685-x (2024).
19. Yaras, S. & Dener, M. IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm. *Electronics* **13**(6), 1053. https://doi.org/10.3390/electronics13061053 (2024).
20. Nazir, A. and Khan, R. A. "Combinatorial Optimization based Feature Selection: A study on Network Intrusion Detection (TS-RF)," arXiv preprint arXiv (2019).
21. Akif, M. A., Hussain, R. and Khan, M. A. "Machine learning based intrusion detection in IoT: A comprehensive review," arXiv preprint arXiv (2025).
22. Khan, R., Shah, N. and Al-Zahrani, F. "Fed-Resilient: Federated learning based intrusion detection system for IoT networks," *IEEE Internet of Things J.*, **9** (23), 23650–23662 (2022).
23. Khan, R., Al-Zahrani, F., and Shah, N. Fed-CIDD: Federated collaborative intrusion detection framework for IoT. *Future Gener. Comput. Syst.* **138**, 29–39, (2023).
24. Khan, I. A., Razzak, I., Pi, D., Khan, N., Hussain, Y., Li, B., Kousar, T. Fed-Inforce-Fusion: A federated reinforcement-based fusion model for security and privacy protection of IoMT networks against cyber-attacks. *Information Fusion* **101**, 102002. https://doi.org/10.1016/j.inffus.2023.102002 (2024).

Author contributions

Anant Upadhiyay: Conceptualization, Methodology Design, Simulation Modeling, Formal Analysis, Software Implementation, Data Curation, Writing – Original Draft Preparation. Abhishek Jain: Supervision, Project Administration, Validation, Review and Editing of Manuscript, Technical Guidance, Resources, and Final Approval.

Funding

This study did not have any specific grant awarded to it by a funding agency whether in the public, commercial or not-for-profit sector.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.J.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.