ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# Deep Reinforcement Learning for Mitigating Selective Forwarding Attacks in ACO

Ibrahim Samson Salifu[1]; Gaddafi Abdul-Salaam[2]

[2]Supervised

[1](PG6250724), [2]Professor

Kwame Nkrumah University of Science and Technology Kumasi, Ghana

A Thesis Submitted to the Department of Computer Science, College of Science, in Partial Fulfilment of the Requirements for the Degree of Master of Philosophy in Cyber-Security and Digital Forensics.

Publication Date: 2026/11/16

**How to Cite:** Ibrahim Samson Salifu; Gaddafi Abdul-Salaam (2026) Deep Reinforcement Learning for Mitigating Selective Forwarding Attacks in ACO. *International Journal of Innovative Science and Research Technology*, 11(10), 3311-3365. https://doi.org/10.38124/ijisrt/26oct1627

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# DECLARATION

I declare that this thesis is my own work. To the best of my knowledge, it contains no material previously published or written by another person, and no material that has been accepted for another degree or diploma, except where the source has been acknowledged.

Student: Ibrahim Samson Salifu

Signature: _____ Date: _____

Supervisor: Prof. Gaddafi Abdul-Salaam

Signature: _____ Date: _____

Head of Department: Prof. Michael Asante

Department of Computer Science, KNUST

Signature: _____ Date: _____

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# ABSTRACT

**Wireless Sensor Networks (WSNs) and Internet of Things (IoT) networks rely on resource-constrained devices and unstable wireless links, making them vulnerable to selective forwarding attacks. Conventional routing methods often prioritise delay, energy and link quality, so a route may remain attractive even when a compromised node drops some packets while forwarding others. This study designed and evaluated a DQN-supported, evidence-gated security mechanism for mitigating selective forwarding attacks in Ant Colony Optimisation (ACO) routing, implemented as the Adaptive Bioinspired Routing Framework (ABRF). The DQN was trained offline on synthetic node-state transitions and, during simulation, used live routing observations to propose one of five graded actions. The evidence gate validated each proposed action using forwarding and route-advertisement evidence before the action modified pheromone or route preference. Evaluation was conducted in a Python discrete-event simulator using smart-home, smart-traffic and wearable-IoT scenarios. The main experiment included 2,250 attack runs and 450 clean-control runs and compared ABRF with simplified RPL-like, AODV-like, ACO-only and traditional-IDS baselines. ABRF achieved a mean attack-step detection rate of 0.847, a hard false-action rate of 0.033 and a packet delivery ratio of 0.665, compared with 0.681, 0.126 and 0.579 for the traditional IDS. Its soft false-action rate was 0.312, indicating that cautious pheromone reductions occurred more often than hard isolation. Dynamic INT8 quantisation reduced the model file size by about 60 percent, although inference was tested on a Colab CPU rather than physical IoT hardware.The findings show that validated DRL security actions can mitigate selective forwarding by directly influencing ACO pheromone decisions while supporting secure route selection under the tested simulation conditions. The conclusions remain limited by the purpose-built simulator, simplified baselines and offline-trained policy.**

***Keywords:*** *Ant Colony Optimisation, Deep Q-Network, Deep Reinforcement Learning, Evidence-Gated Routing, Selective Forwarding Attack, Internet of Things, Wireless Sensor Networks.*

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# ACKNOWLEDGEMENTS

I am sincerely grateful to my supervisor, Prof. Gaddafi Abdul-Salaam, for the scholarly guidance, constructive criticism, and consistent encouragement that shaped this research from its earliest conception to its final form. I thank the Department of Computer Science at Kwame Nkrumah University of Science and Technology for providing the academic environment in which this research was conducted, and the faculty and staff whose teaching and administrative support made this work possible.

I am grateful to the opensource communities behind Python, PyTorch, NumPy, and SciPy, without whose work the experimental component of this thesis would not have been possible.

Finally, I thank my family especially my uncle (Mr Francis B. Danyi) for their patience, sacrifice, and unwavering support throughout the duration of this programme. This work is dedicated to them.

Ibrahim Samson Salifu

Department of Computer Science

Kwame Nkrumah University of Science and Technology

Kumasi, Ghana

June 2026.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# DEDICATION

This thesis is dedicated to my family especially my uncle (Mr Francis B. Danyi), teachers and colleagues whose support made this work possible.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# LIST OF ABBREVIATIONS

| Abbreviation | Meaning |
|---|---|
| ABRF | Adaptive Bioinspired Routing Framework |
| ACO | Ant Colony Optimisation |
| AODV | Ad hoc On-Demand Distance Vector |
| DQN | Deep Q-Network |
| DRL | Deep Reinforcement Learning |
| FPR | False Positive Rate / false-action rate, as defined in this study |
| IDS | Intrusion Detection System |
| IoT | Internet of Things |
| MDP | Markov Decision Process |
| PDR | Packet Delivery Ratio |
| RPL | Routing Protocol for Low-Power and Lossy Networks |

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# TABLE OF CONTENTS

**DECLARATION** ..... **3312**
**ABSTRACT** ..... **3313**
**ACKNOWLEDGEMENTS** ..... **3314**
**DEDICATION** ..... **3315**
**LIST OF ABBREVIATIONS** ..... **3316**
**TABLE OF CONTENTS** ..... **3317**
**LIST OF FIGURES** ..... **3320**
**LIST OF TABLES** ..... **3321**
**CHAPTER ONE INTRODUCTION** ..... **3322**
Background of the Study ..... 3322
Statement of the Problem ..... 3323
General Objective ..... 3323
Objectives of the Study ..... 3323
Research Questions ..... 3323
Significance of the Study ..... 3323
Scope and Delimitations of the Study ..... 3324
Justification of the Study ..... 3325
Organisation of the Thesis ..... 3325
**CHAPTER TWO LITERATURE REVIEW** ..... **3326**
Introduction ..... 3326
How the Review Was Done ..... 3326
Conceptual and Theoretical Review ..... 3326
IoT Routing and Low-Power and Lossy Networks ..... 3326
RPL and AODV Routing ..... 3326
Routing-Layer Attacks ..... 3327
What Security-Enabled Routing Means ..... 3327
Ant Colony Optimisation Routing ..... 3328
Trust and Reputation in Secure Routing ..... 3328
Machine Learning and Deep Reinforcement Learning ..... 3328
Deep Q-Network and Evidence-Gated Security Actions ..... 3328
Evaluation of Secure Routing ..... 3329
Relationship Between Routing Metrics and Security Risk ..... 3329
Detection, Response and Route Recovery ..... 3329
Need for Graded Security Actions ..... 3329
Limits of Simulation-Based Secure Routing Evidence ..... 3330
Review of Existing Systems and Studies ..... 3330
Synthesis of Protocol and Detection Studies ..... 3333
Synthesis of Trust-Based Routing Studies ..... 3333
Synthesis of ACO Routing Studies ..... 3333
Synthesis of DRL Secure-Routing Studies ..... 3333
Synthesis of ACO-Learning Hybrid Studies ..... 3333
Quick-Reference Summary ..... 3334
Critical Review of Literature ..... 3335
Research Gap ..... 3335
Theoretical Framework ..... 3336
Ant Colony Optimisation Theory ..... 3336
Markov Decision Process and Deep Q-Network ..... 3336
Joint Security and Routing Decision ..... 3336

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

Conceptual Framework ..... 3336
Chapter Summary ..... 3337
**CHAPTER THREE METHODOLOGY ..... 3338**
Introduction ..... 3338
Research Design ..... 3338
Experimental Variables and Controls ..... 3338
System Architecture ..... 3338
Decision Cycle of the Framework ..... 3338
ACO Routing Module ..... 3339
Route Heuristic and Weight Change ..... 3339
DQN Security Agent ..... 3341
State Representation ..... 3341
Action Space and Reward Logic ..... 3341
Offline Training Procedure ..... 3342
Evidence Gate ..... 3343
Purpose and Decision Rules ..... 3343
Simulation Platform and Network Scenarios ..... 3344
Baseline Routing Methods ..... 3345
Attack Models ..... 3345
Selective Forwarding ..... 3345
Sinkhole ..... 3345
Wormhole ..... 3346
Coordinated and Adaptive Attacks ..... 3346
Experiment Design ..... 3346
Evaluation Measures ..... 3347
Data Analysis ..... 3348
Descriptive and Inferential Analysis ..... 3348
Ethical and Reproducibility Considerations ..... 3348
Chapter Summary ..... 3348
**CHAPTER FOUR RESULTS AND DISCUSSION ..... 3349**
Introduction ..... 3349
DQN Training and Pre-Deployment Validation ..... 3349
Interpretation of the Training Curve ..... 3349
Overall Security Performance ..... 3350
Detection Performance ..... 3350
Hard and Soft False Actions ..... 3350
Inferential Statistical Results ..... 3351
Overall Routing Performance ..... 3351
Packet Delivery ..... 3351
Delay and Energy ..... 3352
Performance by Attack Type ..... 3353
Clean-Control Results ..... 3353
Pheromone Behaviour ..... 3354
Ablation Analysis ..... 3355
Scalability ..... 3355
Sensitivity Analysis ..... 3356
Model Quantisation ..... 3357
Transfer Projection ..... 3358
Answers to the Research Questions ..... 3358
Answer to Research Question One ..... 3358
Answer to Research Question Two ..... 3358

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

Answer to Research Question Three ..... 3358
Discussion in Relation to Existing Studies ..... 3358
Chapter Summary ..... 3359
**CHAPTER FIVE SUMMARY, CONCLUSION AND RECOMMENDATIONS ..... 3360**
Introduction ..... 3360
Summary of the Study ..... 3360
Main Findings ..... 3360
Security Finding ..... 3360
Routing Finding ..... 3360
Deployment Finding ..... 3360
Contribution of the Study ..... 3361
Conclusion ..... 3361
Limitations ..... 3361
Simulator and Baseline Limitations ..... 3361
Learning Limitations ..... 3361
Measurement Limitations ..... 3361
Generalisation Limitations ..... 3362
Recommendations ..... 3362
Recommendations for Model Development ..... 3362
Recommendations for Network Evaluation ..... 3362
Recommendations for Deployment ..... 3362
Final Statement ..... 3363
**REFERENCES ..... 3364**

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# LIST OF FIGURES

Figure 1. Conceptual framework for DQN-supported, evidence-gated security in ACO routing 3337
Figure 2. Main architecture of the ABRF 3339
Figure 3. Security-enabled pheromone integration 3340
Figure 4. ACO route construction process 3341
Figure 5. Eleven-dimensional node-state vector 3342
Figure 6. DQN training and use in the simulation 3343
Figure 7. Evidence-gated security decision process 3344
Figure 8. Example network layouts for the three deployment scenarios 3345
Figure 9. Experiment design matrix 3347
Figure 10. DQN training reward 3349
Figure 11. Detection rate by attack type 3350
Figure 12. Detection and false-action trade-off 3351
Figure 13. Packet delivery across deployment scenarios 3352
Figure 14. Energy use and end-to-end delay 3353
Figure 15. Mean pheromone behaviour under selected conditions 3354
Figure 16. Ablation results 3355
Figure 17. Scalability results 3356
Figure 18. Sensitivity of selected security settings 3357

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# LIST OF TABLES


| | | |
|---|---|---|
| **Table 1** | Main routing-layer attacks considered in this study | 3327 |
| **Table 2** | Summary of key studies on secure and intelligent IoT routing | 3334 |
| **Table 3** | Route-heuristic weights applied under normal and threat conditions | 3340 |
| **Table 4** | DQN action space and the routing effect of each action | 3343 |
| **Table 5** | Deployment-scenario parameters for the three simulated networks | 3344 |
| **Table 6** | Baseline routing methods compared in this study | 3345 |
| **Table 7** | Attack models and their simulated behaviour | 3346 |
| **Table 8** | Experiment design and number of runs per experiment | 3346 |
| **Table 9** | Evaluation measures and their preferred direction | 3348 |
| **Table 10** | DQN validation results on held-out synthetic states | 3349 |
| **Table 11** | Security-performance comparison between the traditional IDS and the full ACO-DQN framework | 3350 |
| **Table 12** | Summary of paired ACO-DQN and traditional-IDS comparisons | 3351 |
| **Table 13** | Routing-performance comparison across the compared protocols | 3352 |
| **Table 14** | Quantisation results for model size and CPU inference latency | 3357 |

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# CHAPTER ONE
# INTRODUCTION

➢ *Background of the Study*

The Internet of Things (IoT) connects sensors, actuators, gateways and other small devices so that they can collect and exchange data. These devices support applications in homes, health systems, transport, industry and public infrastructure. Their usefulness depends on reliable communication. Data must reach the intended destination within a reasonable time, with acceptable delay and without exhausting the limited energy available to the devices. Routing is therefore a central function in an IoT network. Routing in IoT environments is difficult because many devices have limited battery power, memory and processing capacity. Wireless links may also change because of interference, movement, sleeping nodes or device failure. The Routing Protocol for Low-Power and Lossy Networks (RPL) was developed for networks with these constraints, while the Ad hoc On-Demand Distance Vector protocol (AODV) discovers routes when they are required (Winter et al., 2012; Perkins, Belding-Royer and Das, 2003). Both approaches support communication in changing networks, but neither removes the security risks created by compromised routing nodes. A route is normally selected from information supplied by nodes and links. This information may include hop count, delay, link quality, residual energy or a protocol-specific rank. The routing method assumes that the information is sufficiently reliable for path selection. When a compromised node advertises false information or behaves differently after joining a route, the path that appears efficient may become unsafe. Routing-layer attacks exploit this dependence on route information. A sinkhole node advertises an attractive route so that nearby nodes send traffic through it. A selective-forwarding node receives packets but drops some of them. A wormhole uses a hidden link between malicious nodes to make a false path appear unusually short or reliable. Coordinated and adaptive attackers can combine these behaviours or change their intensity over time. Such attacks can reduce packet delivery, increase delay, waste energy and disturb the network structure (Alfriehat et al., 2024). They are difficult to identify because packet loss and route changes may also occur naturally in a weak wireless network. Several methods have been proposed to improve routing security. Protocol-based methods add fixed checks to a routing protocol. Trust-based methods observe node behaviour and reduce the use of nodes with poor trust values. Machine learning and deep reinforcement learning methods use several network features to detect or respond to suspicious behaviour. These approaches can improve security, but many operate as detection systems beside the routing process. A warning may be generated without immediately changing the route value that made the suspicious path attractive (Cevallos Moreno et al., 2023; Jamshidi et al., 2025). Ant Colony Optimisation (ACO) offers a practical way to connect security evidence with route selection. ACO uses artificial ants to explore possible paths. Better paths receive more pheromone and are more likely to be selected later. Pheromone evaporation allows the network to move away from old or weak routes. ACO is suitable for IoT routing because it is distributed and can combine several routing factors, including energy, delay and link quality (Han, Tang and Jing, 2024; Tawfeek et al., 2025). However, the same mechanism can reinforce an unsafe path when performance information appears attractive but the forwarding behaviour is malicious. Recent studies have combined ACO with trust, reinforcement learning or other security mechanisms. Sharmin et al. (2023) incorporated trust into ACO-based IoT routing. Dubey et al. (2023) combined reinforcement learning and ACO for path selection in IoT-based wireless sensor networks. Kandhoul and Dhurandher (2022) used deep Q-learning for secure routing in opportunistic IoT networks. These studies confirm that learning, security and routing can be combined, but they do not examine the same arrangement used in the present study. The present work focuses on a security-specific Deep Q-Network (DQN) that proposes graded actions from live routing observations. An evidence gate checks whether the proposed action is supported by current forwarding and route-advertisement behaviour. The validated action then changes ACO pheromone or route availability before later paths are selected. This arrangement treats routing and security as connected stages of one decision process. The study therefore developed an Adaptive Bioinspired Routing Framework (ABRF) for security-enabled ACO routing in IoT networks. ACO performs route selection, the trained DQN proposes a security response, and the evidence gate validates the

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

response before it affects the route. The framework was evaluated under selected routing-layer attacks and clean-control conditions in a purpose-built simulator.

➢ *Statement of the Problem*

Conventional ACO-based IoT routing mainly rewards routes that appear efficient in terms of delay, link quality and residual energy. However, a compromised forwarding node can exploit this mechanism by remaining attractive while quietly dropping, delaying or redirecting some packets. As a result, unsafe nodes can continue to receive pheromone reinforcement, causing packet delivery to drop even though the route still appears useful. This can increase retransmissions, waste limited device energy and weaken route recovery. Existing trust-aware ACO, reinforcement-learning and deep-reinforcement-learning approaches have shown that security and learning can improve route selection, but important limitations remain. Trust-based approaches can depend on fixed rules, while learning-based security methods may operate as separate detection modules without directly controlling the pheromone values that determine later ACO route choices (Sharmin et al., 2023; Dubey et al., 2023; Kandhoul and Dhurandher, 2022). A strong automated response that is not checked against current evidence may also isolate a normal node, whereas a weak response may leave a malicious route attractive. Therefore, the network requires security evidence to directly influence route preference. The unresolved problem is how to use a DQN-supported, evidence-gated controller so that live security evidence can validate graded actions and directly modulate ACO pheromone reinforcement and evaporation before subsequent routes are selected. This study addresses that problem through the Adaptive Bioinspired Routing Framework (ABRF).

➢ *General Objective*

The aim of the study is to design a DQN-supported, evidence-gated security mechanism for mitigating selective forwarding attacks in Ant Colony Optimisation routing.

➢ *Objectives of the Study*

- To design a DQN evidence-gated controller for detecting selective forwarding attacks in ACO based routing through quantitative experimentation.
- To integrate security evidence directly into Ant Colony Optimisation (ACO) routing by modulating pheromone reinforcement and evaporation mechanisms to isolate unsafe nodes and enforce secure route selection.
- To evaluate the integrated framework against RPL-like, AODV-like, ACO-only and IDS-augmented routing using security, packet delivery, delay, energy and scalability measures.

➢ *Research Questions*

- How can a DQN-based evidence-gated controller be designed to improve the detection of selective forwarding in ACO-based IoT routing?
- How can security evidence be directly integrated into ACO routing by modulating pheromone reinforcement and evaporation to isolate unsafe nodes and enforce secure route selection?
- How does the integrated framework compare with RPL-like, AODV-like, ACO-only and IDS-augmented routing in terms of security, packet delivery, delay, energy use and scalability?

➢ *Significance of the Study*

The study is significant because it connects attack response directly with route selection. In many secure-routing designs, a detector reports suspicious behaviour after a path has already been selected or reinforced. In ABRF, the validated response changes pheromone, restricts a node or quarantines it before later routing decisions are made. The study therefore demonstrates a practical mechanism through which security evidence can affect the next route choice rather than remaining a separate alert. The work also provides a balanced

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

evaluation of security and network performance. A method that detects attacks but causes excessive delay, packet loss or energy consumption may not be useful in a resource-constrained IoT network. The evaluation therefore considers attack-step detection, hard and soft false actions, packet delivery, end-to-end delay, energy use and scalability. Looking at all of these measures at once helps show the balance between protection and communication quality. The paper also breaks the method into clear parts, instead of treating it as one block. It separates the DQN, the evidence gate, and the ACO module. First, the DQN suggests an action using the state of the current node. Next, the evidence gate reviews that action with the latest proof gathered from forwarding and route signals. After that, the ACO module takes the approved choice and carries it out. It uses pheromone updates and route control to do the final decision step. This layout makes the method easier to understand. Each piece has a specific job. That means later testing can check each part on its own. The work also offers a simulation setup that others can repeat. It runs the proposed framework against four baselines. The comparison uses the same network setups, the same attack types, and the same random seed plan. In addition, the design can be used in future studies that use full protocol implementations. It can also fit online reinforcement learning. The same approach can be moved to real IoT test devices. It can even support stronger ablation tests that focus on single modules. The paper adds to secure routing research as well. It draws a line between mild route penalties and strict isolation. It reports soft false-action rates and hard false-action rates. The results suggest that the security cost changes with how strong the response is. This also matters because a low hard isolation error rate can still hide many penalty events on nodes that are not actually malicious.

➢ *Scope and Delimitations of the Study*

This work mainly looked at selective forwarding attacks that happen at the routing level in wireless sensor and IoT networks. It also checked what happens when some packets get dropped and how that changes which route gets picked and whether packets arrive. It did not investigate application security, cloud security, user authentication, device firmware, physical tampering or data-privacy policy.

The framework was evaluated in a Python discrete-event simulator. The simulated environments represented smart-home, smart-traffic and wearable-IoT settings. Selective forwarding was the primary attack scenario. Sinkhole, wormhole, coordinated and adaptive attacks were retained as secondary robustness tests, while clean-network controls were included to examine benign behaviour. These scenarios created varied test conditions, but they do not represent every IoT device, topology, protocol or attack strategy.

The routing comparisons used simplified RPL-like and AODV-like baselines, ACO-only routing and an IDS-augmented routing method. The simplified protocol baselines represented their main route-selection ideas within one common simulator. They were not complete standards-compliant implementations of all messages, timers, objective functions or recovery procedures in RPL and AODV. The findings must therefore be interpreted within the implemented baseline definitions. The DQN was trained offline using synthetically generated node-state transitions and a predefined reward function. During the main simulation runs, the trained model received live routing observations and selected an initial security action, but its weights were not updated online. An evidence gate then validated the proposed action before it influenced routing. The study therefore evaluates a fixed DQN policy operating within a live routing simulation, not end-to-end online reinforcement learning. Performance was assessed using attack-response measures, hard and soft false-action measures, packet delivery ratio, end-to-end delay, energy use and scalability. Attack-step detection recorded whether at least one malicious node was identified during an attack control step. It was not a node-level recall measure. Detection latency was expressed in simulation steps because the simulator did not assign a universal real-time duration to every step. The study did not deploy the framework on physical IoT hardware. Model-size and CPUinference measurements provide preliminary computational evidence only. They do not establish production readiness, total gateway memory use, radio overhead or energy consumption on an actual device. Dynamic INT8 quantisation was assessed for stored model size and inference on a Colab CPU, not for execution on specialised embedded hardware. The ablation analysis compared the complete framework with reduced baselines but did not fully isolate the independent contribution of every component. The scalability experiment increased node count while the physical area and communication range remained fixed, so network

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

density also changed. These delimitations are stated to prevent the simulation results from being interpreted as universal field performance.

➢ *Justification of the Study*

The study is justified by the need for a routing mechanism that can respond to selective forwarding while route selection is still active. In IoT networks, malicious packet dropping can resemble ordinary wireless loss.

A compromised node may also remain attractive when routing decisions rely mainly on delay, energy and link quality. A security mechanism should therefore use current behaviour and ensure that validated evidence changes future route preference.

Integrating a DQN, an evidence gate and ACO provides a practical way to address this need. The DQN gets feedback from real observations. The evidence gate blocks choices that are not well supported. Then the ACO uses the checked output to guide pheromones and the route. We compare this full setup with common baselines to see if we can raise security levels without hurting packet delivery, delay, or energy use. The simulation also provides a controlled and reproducible basis for later physical-IoT evaluation.

➢ *Organisation of the Thesis*

This thesis is organised into five chapters. Chapter One introduces the study by presenting the background, problem, aim, objectives, research questions, significance, scope and delimitations, and justification. Chapter Two reviews literature on IoT routing, routing-layer attacks, Ant Colony Optimisation, trust, Deep Q-Networks and evidence-based action validation, and identifies the research gap addressed by ABRF. Chapter Three describes the research design, simulation platform, network scenarios, attack models, routing baselines, ACO module, DQN security agent, evidence gate, experiment design, evaluation measures and data-analysis procedures. Chapter Four presents and discusses the results, including DQN training, security and routing performance, clean-control behaviour, ablation, scalability, sensitivity, quantisation and answers to the research questions. Chapter Five summarises the study, presents the main findings and contribution, concludes the thesis, states the limitations and gives recommendations. The reference list presents the sources cited throughout the thesis.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# CHAPTER TWO
# LITERATURE REVIEW

➢ *Introduction*

This chapter reviews studies and concepts on deep reinforcement learning for mitigating selective forwarding attacks in wireless sensor and IoT networks. The review focuses on a simple question: how can an IoT routing method choose efficient paths while avoiding nodes whose behaviour may be harmful? The chapter explains the concepts behind the study, discusses the most relevant solutions, compares what they achieved, and shows what remains unresolved.

The discussion is arranged so that the research gap develops from the evidence. Section 2.2 explains IoT routing, routing attacks, trust, Ant Colony Optimisation and deep reinforcement learning. Section 2.3 reviews key studies by stating what each study wanted to achieve, what it did, what it found and what it did not address. Section 2.4 brings the findings together. Sections 2.5 to 2.7 present the research gap, theoretical framework and conceptual framework.

- *How the Review Was Done*

The literature search used IEEE Xplore, ScienceDirect, SpringerLink, PeerJ, MDPI, Google Scholar and official Internet Engineering Task Force documents. Search terms included IoT secure routing, RPL attacks, selective forwarding, sinkhole, wormhole, trust-based routing, Ant Colony Optimisation routing, reinforcement learning routing, deep reinforcement learning security, DQN intrusion detection and ACO-DRL routing.

Priority was given to peer-reviewed work published from 2020 to July 2026. Older sources were retained only when they defined a standard or theory that remains central to the study, such as RPL, ACO and DQN. Studies were selected when their methods and findings had direct value to the security agent, pheromone control, routing process or evaluation used in this thesis.

➢ *Conceptual and Theoretical Review*

- *IoT Routing and Low-Power and Lossy Networks*

Many IoT networks are built from small devices that have limited battery power, memory and processing capacity. Their wireless links may also be weak or unstable. The Internet Engineering Task Force describes these environments as Low-Power and Lossy Networks. A routing method for such a network must therefore use little control traffic, adapt when links change, and avoid draining a small group of nodes too quickly (Winter et al., 2012).

Routing becomes a security issue when a node can influence the path chosen by other nodes. A route may appear short, fast or energy-efficient but still pass through a compromised device. The main problem is therefore not only finding a path. It is finding a path that remains useful when some of the information used for route selection may be false or when a forwarding node may drop packets.

- *RPL and AODV Routing*

RPL is the standard IPv6 routing protocol for low-power and lossy networks. It organises nodes in a Destination-Oriented Directed Acyclic Graph that leads towards a root or gateway. Each node selects a preferred parent according to an objective function, which may consider rank, link quality, energy or another routing measure. This design reduces routing state and control traffic, but it also means that false rank or route information can affect many later decisions (Winter et al., 2012).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

AODV follows a different approach. It discovers a route only when a node needs one, using route-request and route-reply messages. This can suit mobile or changing networks because the protocol does not maintain every route all the time. However, a malicious node can still give a false reply, attract traffic or drop packets after joining the route. RPL and AODV are therefore useful baselines, but neither automatically provides adaptive protection against internal routing attacks (Perkins, Belding-Royer and Das, 2003; Alfriehat et al., 2024).

- *Routing-Layer Attacks*

Routing-layer attacks misuse the process used to create or maintain paths. They are difficult to handle because normal link loss, low energy and mobility can produce behaviour that looks similar to an attack. Recent reviews also show that many proposed defences focus on one attack or one fixed topology, which limits their use when network conditions or attacker behaviour change (Alfriehat et al., 2024).

Table 1 Main Routing-Layer Attacks Considered in this Study

| Attack | What the attacker does | Main effect on routing | Why detection is difficult |
|---|---|---|---|
| Selective forwarding | Drops some packets while forwarding others. | Reduces packet delivery and may waste energy through retransmission. | Its behaviour may look like ordinary wireless loss. |
| Sinkhole | Advertises an attractive route so that nearby nodes send traffic through it. | Concentrates traffic on a malicious node and supports later packet dropping or alteration. | The false route may use values that look valid to the protocol. |
| Wormhole | Creates a hidden tunnel between two distant malicious nodes. | Makes a harmful path appear unusually short or reliable. | Each malicious node may look normal when viewed alone. |
| Coordinated attack | Several malicious nodes support one another or combine attack methods. | Damages more routes and weakens local monitoring. | Evidence is spread across several nodes and events. |
| Adaptive attack | Changes the level or timing of harmful behaviour to avoid fixed rules. | Reduces the value of static thresholds and fixed trust rules. | The attack deliberately stays close to normal behaviour. |

Selective forwarding, sinkhole and wormhole attacks affect routing in different ways, but they share one feature. The path can still look attractive to the routing protocol after it has become unsafe. A security-enabled routing method must therefore use current behaviour, not only normal route-quality measures, when it updates path preference.

- *What Security-Enabled Routing Means*

Security-enabled routing means that security information changes the route-selection process itself. This is different from placing an intrusion detector beside the routing protocol. A separate detector may raise an alert, but the routing algorithm may continue to reward or select the same path until another mechanism removes it. In security-enabled routing, evidence about suspicious behaviour directly reduces the preference given to the affected node or link.

The security response should also be proportional to the evidence. A weak signal may justify closer monitoring or a small route penalty. Strong evidence may justify isolation or quarantine. This graded response is useful in IoT networks because immediate removal of every uncertain node can create false alarms, break useful routes and increase delay.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

- *Ant Colony Optimisation Routing*

Ant Colony Optimisation takes its idea from ants that leave pheromone and later use those traces to form good paths. In a network, software ants try many possible ways to move. When an ant must pick the next hop, it looks at two things. One is the pheromone level on that link. The other is a heuristic score that hints at how good the current route is. Links that lead to good results get stronger pheromone. At the same time, pheromone fades away over time, so old traces do not stay too strong (Dorigo and Stützle, 2004).

ACO can fit IoT routing well. It is spread out across the system. It can also react when links change, instead of relying on one fixed global route map. The heuristic can blend multiple inputs. For example, it can weigh residual energy, delay, link quality, and distance. This same flexibility can also hurt security. If pheromone is updated only from raw performance, a bad node could look attractive. It might seem fast or near, and the traffic may keep going its way. To reduce that risk, security needs to be placed in the heuristic, the pheromone update, or in both parts.

- *Trust and Reputation in Secure Routing*

Trust-based routing assigns a score to a node from its past behaviour. The score may use packet forwarding, route messages, recommendations from neighbours and remaining energy. Routes through low-trust nodes receive less preference. This is useful against internal attackers because a node may hold valid credentials but still behave badly.

Trust is not automatically adaptive. Many schemes use fixed formulas, fixed thresholds or slowly changing averages. A node may behave well long enough to build a strong score and then begin an attack. A new network also has little history from which to calculate trust. Trust can improve route safety, but a fixed trust rule may not respond quickly when behaviour changes.

- *Machine Learning and Deep Reinforcement Learning*

Machine learning can use several network features at the same time to identify patterns that simple thresholds miss. Most intrusion-detection models produce a label such as normal or malicious. The label is useful, but it does not by itself state how strongly routing should react. The detector and the routing process may still operate as separate parts of the system.

Deep reinforcement learning learns a policy for choosing actions. The agent observes a state, selects an action and receives a reward or penalty. Reviews of DRL-based IoT intrusion detection show that the method can support adaptive decisions, but they also identify weak reproducibility, unrealistic environments and unclear links between detection and operational response as continuing problems (Cevallos Moreno et al., 2023; Jamshidi et al., 2025).

- *Deep Q-Network and Evidence-Gated Security Actions*

A Deep Q-Network uses a neural network to estimate the value of each available action in the current state. DQN made it possible to use Q-learning with large state spaces by combining value learning with deep neural networks (Mnih et al., 2015). In this study, the model was trained offline on synthetically generated node states. During evaluation, it received live forwarding, packet-drop, delay, energy, link and route-advertisement observations from the simulator.

The DQN first proposes one of five graded actions: monitor, watch, reduce pheromone, isolate or quarantine. Before the action affects routing, an evidence gate checks live forwarding and route-advertisement evidence. It can reduce a severe action when evidence is weak or strengthen the response when clear evidence is present. The validated action then changes the ACO routing response. This design separates weak suspicion from strong evidence and reduces unsupported isolation.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

- *Evaluation of Secure Routing*

Security is not the only thing to look at in routing. You also need to check the network side. Detection rate tells you how many attacks are caught. False positives tell you how often safe traffic gets flagged by mistake. Packet delivery ratio matters because it shows if messages still reach the destination. Delay matters too, since extra waiting can hurt use. Energy per delivered packet tells you what the extra checks cost. Network lifetime helps show if the method drains nodes too fast. Scalability shows if it still works when the network gets bigger. If you only report detection accuracy, you get a partial picture.

- *Relationship Between Routing Metrics and Security Risk*

Energy, delay, reliability, and security rarely line up on one path. Sometimes a fast route goes through a node that looks risky. Other times the safer option adds extra hops and uses more power. Because of that, routing cannot just chase one goal. It has to balance them.

ACO can help with this balance since its path rule can mix more than one value. The hard part is how those values shift when new threat signs show up. If the security weight stays the same, it may not react enough when an attack is under way. At other times it can push too hard when things are normal. For this reason, the method uses separate weights for normal use and for threat periods. It also changes how pheromone is updated, at the node or at the link, based on the evidence gate.

This split is important in practice. A single security weight for the whole network shifts how much security matters across all choices. A local penalty on pheromone instead reduces how appealing one suspicious route looks. Using both tools lets the system react to the broad threat state and also to the actions of a particular node.

- *Detection, Response and Route Recovery*

Detection checks if something odd is happening. Response states what should happen next. Route recovery explains how traffic should go after the change. These pieces work together, but they are not the same thing. Even if detection finds an attack well, routing can still turn out badly. That can happen when the response comes late. It can also happen when the new path is not strong.

Some intrusion detection work lists accuracy, precision, recall, or an F1 score. Yet it does not show how an alert shifts the choice of routes. Other secure routing studies report packet delivery or energy use. Still, they do not show how the security decision was formed. A full framework should link what is seen, what action is taken, how pheromone is updated, and what the later route outcome looks like.

This study uses that idea in a linked chain. Current signals go into the trained DQN. The evidence gate then reviews the suggested action. After that, the verified response updates the pheromone levels. Later results, like delivery, delay, and energy use, indicate how well the route recovered. This setup lets the study judge both security and routing in a clear way.

- *Need for Graded Security Actions*

A binary rule sorts each odd event into just two bins: safe or serious. That split can go wrong in two ways. One issue is that mild attacks might not get enough attention. Another issue is that normal packet loss could trigger an isolation step that was not needed. Graded responses sit between doing nothing and cutting a node off.

Reducing pheromones helps a lot when a middle step is needed. It lowers how much a node is wanted, yet it does not erase the node. Later, if the warning signs get clearer, the system can shift to isolation or quarantine. If the signs fade, the node can stay in the mix. This idea supports careful routing when bad links can seem like harmful forwarding.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

In this work, the soft false-action rate tracks how often the system applies these mild penalties to nodes that are actually fine. The hard false-action rate tracks the times it uses stronger actions that push toward isolation. Both numbers matter. A low hard false-action rate can still hide that the soft penalties happen a lot.

- *Limits of Simulation-Based Secure Routing Evidence*

A simulator lets us compare protocols and attacks using the same setup. It also makes it easier to repeat tests and keep the random inputs the same from run to run. Even with that, a simulator still does not capture everything. Radio behavior is simplified, delays are modeled in a rough way, and device faults are not fully represented. Protocol details are also limited. Because of this, the output only supports conclusions that match the assumptions built into the tool.

A simulator made for a specific goal can help when checking the decision mechanism. Still, it is not the same as a full build that matches the full standards. For example, baselines that follow RPL-like or AODV-like ideas can show the main route concept. They may not include every message flow or every timer behavior. So the thesis needs to spell out this gap. It should not suggest direct field performance.

After this, real testbeds matter. With physical devices, you can measure memory use, energy use, radio interference, and timing that happens on the hardware. The work here is meant to set up the next step in experiments. It is not meant to replace that next step.

➢ *Review of Existing Systems and Studies*

This section discusses the studies that are closest to the present work. Each discussion states the aim, method, main result and remaining limitation. The studies move from rule-based and trust-based security to ACO, DRL and recent hybrid routing methods.

- *Zaminkar and Fotohi (2020): SoS-RPL*

Zaminkar and Fotohi (2020) aimed to protect RPL networks from sinkhole attacks. They developed SoS-RPL, which rated and ranked nodes using distance information and packet-transmission behaviour. The method was tested in NS-3 and improved measures such as detection, packet delivery and packet loss when compared with ordinary RPL. Its contribution was to show that routing information can be checked before a suspicious node is accepted as a preferred route. The remaining limitation is that the defence was designed for one known attack and used fixed rating rules rather than a learning agent that could choose different levels of response.

- *Kandhoul and Dhurandher (2022): DQNSec*

Kandhoul and Dhurandher (2022) aimed to secure opportunistic IoT routing against sinkhole, hello-flood and distributed denial-of-service attacks. They modelled the routing problem as a Markov Decision Process and used an actor-critic deep Q-learning approach to detect and isolate attackers. Simulations using an opportunistic-network environment showed that the method could improve secure forwarding when compared with selected machine-learning routing schemes. The study is important because it placed DRL inside a secure-routing decision. However, it addressed opportunistic store-carryforward routing and did not use ACO pheromone as the route-control mechanism.

- *Alansari et al. (2023): RPLAD3*

Alansari et al. (2023) wanted to detect blackhole, grayhole and selective-forwarding attacks in RPL-based IoT networks. They built a lightweight anomaly-detection method that considered mobility and several routing features. The study showed that internal packet-dropping attacks can be detected without relying on one delivery threshold. Its main value to the present research is the use of several behaviour indicators. However, RPLAD3 remained an intrusion-detection layer. It did not place the detector output inside an ACO pheromone update or study how graded security actions change route selection.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

- *Sharmin et al. (2023): Trust-Aware ACO Routing*

Sharmin et al. (2023) aimed to improve secure IoT routing by combining residual energy and trust in a modified ACO method. Their design used primary and secondary trust measures and changed pheromone management so that routes through trusted, energy-efficient nodes became more attractive. The reported results showed improvements in throughput, delay, energy use and processing time against the selected secure-routing methods. This work is one of the closest studies to the present thesis because security information directly affected pheromone. The remaining limitation is that the trust score came from a fixed evaluation model, not from a DRL agent that could select graded actions from changing network behaviour.

- *Dubey et al. (2023): PPO-ACO*

Dubey et al. (2023) wanted to improve path selection in 5G-enabled IoT wireless sensor networks. They combined Proximal Policy Optimisation with ACO. PPO learned a path-selection policy, while ACO used pheromone to guide the search. The simulations reported better residual energy and a larger number of active nodes than the comparison methods. The study proves that reinforcement learning and ACO can work together in IoT routing. However, its main aim was path and energy optimisation. Security was treated as part of the general trade-off and the study did not use a separate security agent to detect routing attacks and apply graded pheromone penalties.

- *Liu et al. (2023): Deep Learning, Trust and ACO*

Liu et al. (2023) aimed to improve secure routing in satellite networks where ordinary trust models had difficulty identifying unknown attacks. They cleaned trust evidence with Dempster-Shafer theory, used a variational autoencoder to identify abnormal behaviour, and added the resulting security factor to ACO pheromone. The method improved delay, packet loss and throughput in the reported simulations. This is the closest architectural example found in the review because learned security information changed pheromone. Its limits are equally clear: it used a variational autoencoder rather than DRL, focused on satellite networks rather than IoT low-power networks, and did not use graded routing responses.

- *Han et al. (2024): Improved ACO for IoT Routing*

Han et al. (2024) aimed to improve routing in IoT wireless sensor networks. They changed the transfer probability, pheromone update and route objective used by ACO. Their experiments reported lower energy use and delay, fewer dead nodes and higher throughput than the comparison methods. The study shows that ACO can combine several routing measures and adapt to network conditions. However, the optimisation remained performance-based. It did not assess malicious forwarding behaviour and did not include a security value in the route decision.

- *Fu et al. (2024): Energy-Aware Two-Way Trust Routing*

Fu et al. (2024) aimed to protect IoT-fog routing while controlling energy use. They combined direct trust, indirect trust and energy trust, and used a Generative Flow Network to predict available node energy. The route was then selected through an AODV-based multipath process. The results showed better malicious-node detection, energy efficiency and data transmission than the selected trust-based methods. The study gives a strong example of joint security and energy routing. However, security still depended on a designed trust model, and the routing engine was AODV rather than security-enabled ACO.

- *Mahjoub et al. (2024): Adversarial Reinforcement Learning IDS*

Mahjoub et al. (2024) wanted to improve intrusion detection in IoT under changing and unbalanced attack data. They combined supervised learning with an adversarial reinforcement-learning environment and compared the result with other machine-learning models. The proposed method improved the classification of harmful traffic in the reported tests. The study supports the use of reinforcement learning when the threat environment changes. Its limitation for this thesis is that the output was an intrusion class. The work did not show how the action of the security agent should update routing preference or ACO pheromone.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

- *Wang et al. (2024): DRL-Based RPL Optimisation*

Wang et al. (2024) aimed to make RPL routing more reliable and energy-efficient in Industrial IoT. They formulated routing as a partially observable decision problem and proposed a scalable multi-attention actor-double-critic method. The simulations showed improvements in energy use, network lifetime, queue loss and packet delivery. The study confirms that DRL can make routing decisions in a large and changing IoT network. However, its objective was reliability and energy optimisation, not attack detection. It did not connect a security-specific action to ACO pheromone.

- *Omar, Soudan and Altaweel (2024): UOS_IOTSH_2024*

Omar, Soudan and Altaweel (2024) aimed to improve research on sinkhole detection by providing a more varied RPL dataset. They used COOJA to collect 1,771,880 traffic samples from 60 scenarios covering different network sizes, DODAG structures, attacker numbers and attacker positions. The dataset gives researchers a stronger basis for testing sinkhole-detection methods. Its limitation is that it is a dataset rather than a routing solution. It helps a detector learn or evaluate attack patterns, but it does not show how a detected threat should change a route during operation.

- *Tawfeek et al. (2025): Modified ACO for Reliability and Energy*

Tawfeek et al. (2025) aimed to improve energy efficiency and routing reliability in wireless sensor networks. Their Modified Ant Colony Optimisation Algorithm used energy, reliability, bandwidth and path distance in one heuristic and adjusted pheromone decay according to network conditions. The reported results showed better network life and packet delivery than the comparison methods. The study shows how ACO can support several routing objectives in one formula. However, reliability did not mean attack-aware security. The model did not use live behavioural evidence from a learning agent to penalise suspicious nodes.

- *Jamshidi et al. (2025): Systematic Review of DRL-Based IoT IDS*

Jamshidi et al. (2025) reviewed the use of deep reinforcement learning in IoT intrusion detection. The review found that DRL can support adaptive threat detection and can reduce unnecessary alerts when the state, action and reward design are suitable. It also identified continuing problems with datasets, reproducibility and practical integration. This review supports the decision to use graded security actions instead of only a binary label. It also shows that strong detection results do not by themselves prove that a model can improve routing. The link between the security policy and the route-control mechanism still needs direct testing.

- *Bala et al. (2026): Federated Multi-Agent DRL Secure Routing*

Bala et al. (2026) aimed to provide secure and energy-efficient routing in IoT-enabled wireless sensor networks. They combined clustering, cluster-head optimisation and federated multi-agent deep reinforcement Q-learning. Security entered the reward through malicious path choices, abnormal packet drops and unstable links. NS-3 simulations with 100 nodes reported improved delay, throughput, residual energy and network stability. The study is important because it directly uses DRL for secure IoT routing. However, it selects secure paths through a multi-agent routing policy and does not use a security-specific DRL action to control ACO pheromone.

- *Pidy Pidy et al. (2026): CARAC-MDRL*

Pidy Pidy et al. (2026) aimed to improve routing in highly mobile vehicular networks. They added two DRL agents to a cluster-based ant-colony method. One agent controlled cluster stability, while the other adjusted ACO parameters, and federated learning supported cooperation. NS-3 simulations reported better delay, throughput and packet delivery than the selected baselines. The work shows that several DRL agents can tune an ant-based routing system. However, it studied vehicular performance rather than IoT routing-layer security, and its agents did not produce graded security actions for malicious nodes.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

✓ *Synthesis of Protocol and Detection Studies*

SoS-RPL, RPLAD3 and related protocol studies show that routing information can be checked for known harmful behaviour. Their main strength is closeness to the routing process. Their common weakness is limited adaptation: the rules are designed for selected attacks and may not provide several levels of response when behaviour changes.

The studies also show that no single feature is sufficient. Packet loss can come from an attack, congestion or a weak link. Route advertisements can be false, but a sudden change may also come from mobility. This supports the use of a state vector that combines forwarding, route, energy and link information rather than relying on one threshold.

✓ *Synthesis of Trust-Based Routing Studies*

Sharmin et al. (2023) and Fu et al. (2024) move security closer to route selection by using trust. They show that the route engine can consider behaviour as well as energy or path quality. This is an important foundation for the present work.

The main difference is how the security value is produced. A fixed trust equation gives a score according to designed rules. A DQN gives an action according to learned state-action values, while an evidence gate checks the action against live observations. The present study therefore builds on trust-aware routing but uses a different decision process.

✓ *Synthesis of ACO Routing Studies*

Han et al. (2024) and Tawfeek et al. (2025) show that ACO can improve energy use, delay, reliability and network lifetime. Their work confirms that pheromone can represent several route qualities and can respond to changing network conditions.

The limitation is not poor routing performance. It is the absence of a tested mechanism for turning live malicious behaviour into a graded pheromone response. Performance-based ACO may reward a route that looks efficient even when the route includes a compromised node. This is the point at which the present work adds security control.

✓ *Synthesis of DRL Secure-Routing Studies*

Kandhoul and Dhurandher (2022), Wang et al. (2024) and Bala et al. (2026) show that DRL can support routing and security decisions. Their studies are important because they move beyond a separate classifier and allow learning to affect forwarding or route choice.

These studies also prevent an exaggerated novelty claim. DRL-based secure routing already exists. The present contribution must therefore be stated more narrowly: the study evaluates a DQN-supported, evidence-gated action that directly changes ACO pheromone in the tested IoT scenarios.

✓ *Synthesis of ACO-Learning Hybrid Studies*

Dubey et al. (2023), Liu et al. (2023) and Pidy Pidy et al. (2026) confirm that learning and ant-based routing can work together. Dubey et al. use reinforcement learning for path optimisation, Liu et al. place learned security information in pheromone, and Pidy Pidy et al. use DRL to adjust ant-routing parameters.

The reviewed hybrids use different networks, learning methods and objectives. None of them evaluates the same combination of an offline-trained DQN security agent, live evidence validation, five graded actions and direct pheromone penalties across the security and routing measures used in this thesis. This is the exact boundary of the research gap.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

✓ *Quick-Reference Summary*

Table 2 places the reviewed studies side by side. It shows the movement from fixed rules and trust scores to learning-based routing. It also shows why the present gap is narrower than simply saying that ACO and DRL have never been combined.

Table 2 Summary of Key Studies on Secure and Intelligent IoT Routing

| Study | What the work aimed to do | What was done | Main finding | What remained unresolved |
|---|---|---|---|---|
| Zaminkar & Fotohi (2020) | Detect sinkhole attacks in RPL | Node rating and packet-transmission rules | Improved routing and attack measures in NS-3 | Single attack; fixed rules; no learning-based pheromone control |
| Kandhoul & Dhurandher (2022) | Secure OppIoT routing | Actor-critic deep Q-learning | Detected and isolated three attack types | OppIoT setting; no ACO pheromone mechanism |
| Alansari et al. (2023) | Detect packet-dropping attacks in RPL | Lightweight anomaly detection with mobility features | Detected blackhole, grayhole and selective forwarding | Detection remained separate from route optimisation |
| Sharmin et al. (2023) | Add trust to secure IoT ACO routing | Trust and residual energy in pheromone management | Reported better throughput, delay and energy use | Trust model was fixed rather than DRL-based |
| Dubey et al. (2023) | Improve IoT path and energy selection | PPO combined with ACO | Improved residual energy and active-node life | Security was not a live attack-response signal |
| Liu et al. (2023) | Improve secure satellite routing | VAE anomaly detection plus trust and ACO pheromone | Improved delay, packet loss and throughput | Satellite setting; VAE not DRL; no graded actions |
| Han et al. (2024) | Improve IoT WSN routing | Improved ACO probability, objective and pheromone update | Reduced energy and delay; improved throughput | No security criterion |
| Fu et al. (2024) | Combine secure and energy-aware IoT-fog routing | Two-way trust, energy prediction and AODV multipath | Improved malicious-node detection and transmission | Fixed trust model; not ACO |
| Mahjoub et al. (2024) | Improve IoT intrusion detection | Adversarial reinforcement learning IDS | Improved harmful-traffic classification | Produced a class, not a routing action |
| Wang et al. (2024) | Improve IIoT RPL routing | Scalable actor-double-critic DRL | Improved reliability and energy measures | No routing-layer attack response |
| Omar et al. (2024) | Support sinkhole detection research | COOJA dataset with 60 RPL scenarios | Produced 1,771,880 varied traffic samples | Dataset only; no live mitigation |

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

| | | | | |
|---|---|---|---|---|
| Tawfeek et al. (2025) | Improve WSN reliability and energy | Multi-objective modified ACO | Improved network life and delivery | Reliability did not include attack behaviour |
| Jamshidi et al. (2025) | Review DRL-based IoT intrusion detection | Systematic review | Identified adaptive value and practical gaps | Did not provide a route-control mechanism |
| Bala et al. (2026) | Secure and optimise IoT routing | Federated multi-agent deep Q-learning | Improved secure routing and energy measures | No security-specific control of ACO pheromone |
| Pidy Pidy et al. (2026) | Improve dynamic vehicular ant routing | Two DRL agents tune clustering and ACO | Improved latency, throughput and PDR | Performance focus; not IoT attack response |

➢ *Critical Review of Literature*

The reviewed studies solve important parts of the problem, but their limits form a clear pattern.

- Protocol and anomaly-detection studies can identify known routing behaviour, but many use fixed rules, one attack or one topology. Their security result is often an alert rather than a direct change to the route-selection formula.
- Trust-based routing brings security closer to routing. Sharmin et al. (2023) and Fu et al. (2024) show that trust and energy can improve route choice. The weakness is that the trust rule is designed in advance and may respond slowly when the attack changes.
- Performance-based ACO studies show that pheromone can combine energy, delay, reliability and distance. Han et al. (2024) and Tawfeek et al. (2025) improve routing performance, but they do not test whether the same pheromone process can avoid nodes that behave maliciously.
- DRL studies show that an agent can detect attacks or make routing decisions. Mahjoub et al. (2024) focus on detection, while Wang et al. (2024), Kandhoul and Dhurandher
- (2022), and Bala et al. (2026) place learning closer to routing. These studies do not use a security-specific DRL action as a direct pheromone control in IoT ACO routing.
- Hybrid studies prove that ACO and learning can be combined. Dubey et al. (2023) use PPO-ACO for path selection, Liu et al. (2023) place a learned security factor in satellite ACO pheromone, and Pidy Pidy et al. (2026) use DRL to tune ant routing. No single study reviewed here combines the exact IoT setting, a DQN-based initial security decision, evidence-gated validation, graded responses and direct pheromone penalties used in the present work.

The literature therefore supports all the building blocks of the proposed system. It also shows that the main contribution cannot be stated as simply combining ACO and DRL. That combination already exists. The contribution must be stated at the level of how security evidence is produced and how it changes pheromone during routing.

➢ *Research Gap*

The reviewed literature shows three connected gaps. First, secure ACO studies normally use fixed trust or reputation values. These values can make routes safer, but they do not provide an adaptive action policy for changing attack behaviour. Second, DRL-based secure-routing studies select routes or isolate attackers, but they do not use a security-specific action to change the pheromone process of ACO. Third, ACO-DRL studies mainly optimise energy, stability or path quality rather than routing-layer attack response.

Liu et al. (2023) is the closest work identified because a learned security factor enters ACO pheromone. However, that study used a variational autoencoder and trust evidence in a satellite network. It did not use a

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

DQN security agent, graded actions or an IoT low-power routing environment. Sharmin et al. (2023) is also close because trust changes pheromone, but its trust model is fixed.

The gap addressed by this study is therefore the limited tested evidence on an IoT routing framework in which a trained DQN observes live routing behaviour, proposes a graded security action, and passes that action through an evidence gate before the validated response changes ACO pheromone. No study identified in this review evaluates this exact mechanism across security, packet delivery, delay, energy, network lifetime and scalability measures. The proposed Adaptive Bioinspired Routing Framework addresses this gap.

➢ *Theoretical Framework*

- *Ant Colony Optimisation Theory*

ACO explains how route preference is built and changed in the proposed system. An ant selects a next hop using pheromone and current heuristic information. A successful path receives a positive update, while evaporation reduces the value of old paths. In this study, the same theory allows security to influence routing. A suspicious node receives a pheromone penalty, so later ants are less likely to use it. The routing system can therefore move away from unsafe paths without waiting for a completely new routing protocol (Dorigo and Stützle, 2004).

- *Markov Decision Process and Deep Q-Network*

The DRL component is guided by the Markov Decision Process. The state represents node and network behaviour, while the action represents the level of security response. In the implementation used by this study, the DQN was trained offline on synthetically generated state transitions. Its reward function encouraged suitable responses to malicious states and penalised missed attacks and unnecessary actions. During the main simulation experiments, the trained model received live network observations and produced an initial action, but its weights were not updated. DQN supports this process by using a neural network to estimate action values for a large state space (Mnih et al., 2015).

- *Joint Security and Routing Decision*

ACO, DQN and the evidence gate play different but connected roles. The DQN proposes the level of security response from observed behaviour. The evidence gate checks whether the live network evidence supports that action. ACO then decides which route should be preferred after the validated response changes pheromone. The combined framework links security and routing in one repeated decision process while reducing unsupported isolation.

➢ *Conceptual Framework*

Figure 1 shows the flow of the proposed framework. The IoT network produces routing observations such as forwarding rate, packet drops, delay, energy, link quality and route-advertisement behaviour. These observations form the state presented to the DRL security agent.

The trained DQN proposes a graded action. Monitoring and watching keep the node under observation, pheromone reduction lowers its routing influence, and isolation or quarantine removes stronger threats from active routing. Before any action changes routing, the evidence gate checks live forwarding and route-advertisement indicators. The validated action is then passed to the ACO module, which updates pheromone and next-hop preference before the next routing decision.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

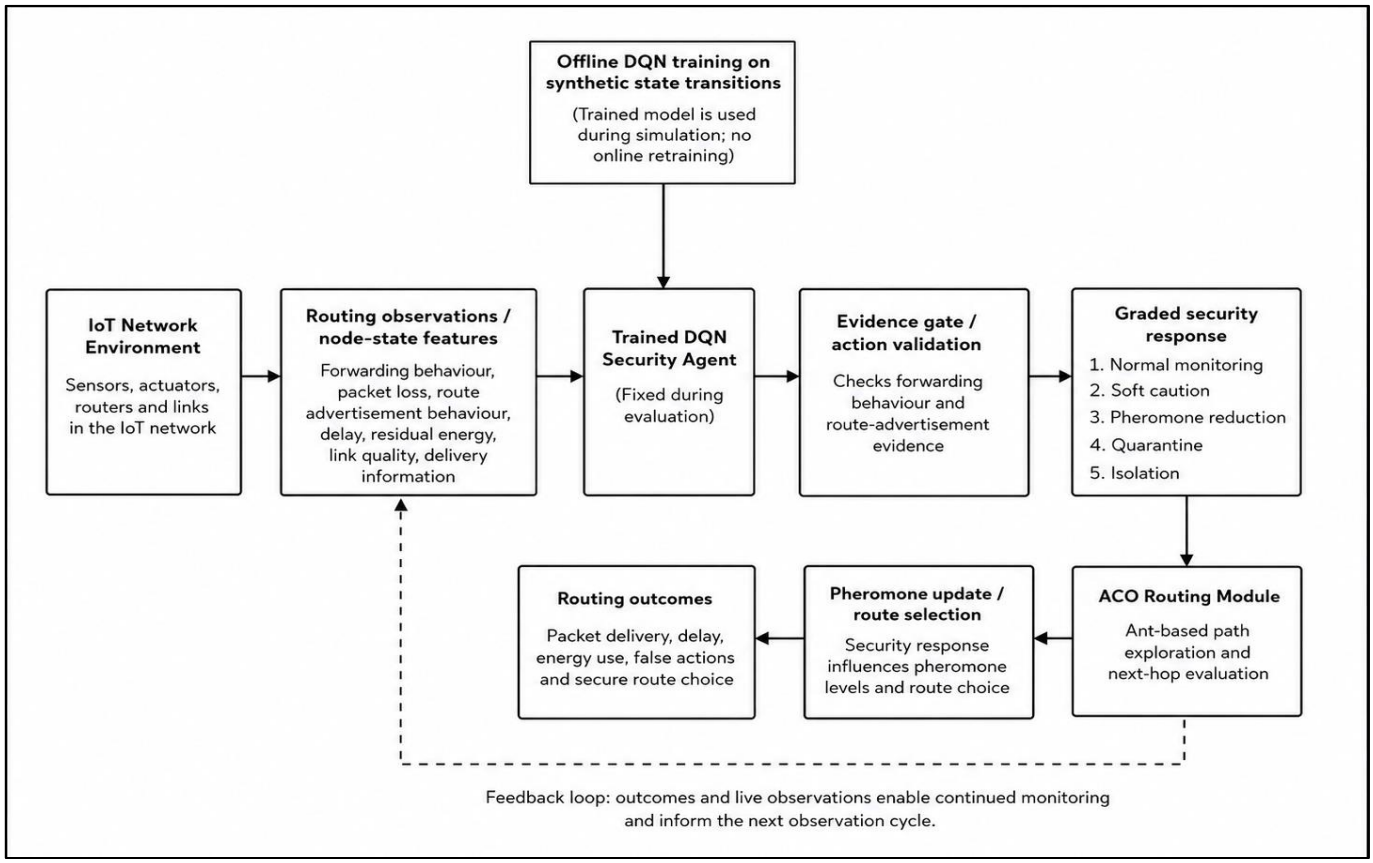


Fig 1 Conceptual Framework for DQN-Supported, Evidence-Gated Security in ACO Routing

After packets are forwarded, the network produces new routing and security outcomes. These outcomes become the next live observations used for another security and routing decision. The feedback changes later actions and routes, but it does not retrain the DQN during the evaluation runs. This is the main difference between the proposed framework and an intrusion detector that only reports an attack after routing has already taken place.

➢ *Chapter Summary*

This chapter reviewed the ideas and studies behind security-enabled ACO routing. RPL and AODV support constrained and changing networks but remain open to internal route manipulation. Rule-based and trust-based methods improve security, but many depend on fixed attack assumptions. ACO provides flexible multi-criteria routing, while DRL provides action-based learning from network state.

The closest studies confirm that trust can change ACO pheromone, reinforcement learning can work with ACO, and DRL can support secure routing. The remaining gap is narrower: limited evidence exists on an IoT system where a trained DQN proposes graded security actions, an evidence gate validates them against live network behaviour, and the validated response directly changes ACO pheromone during routing attacks. The theoretical and conceptual frameworks show how the proposed ABRF connects these parts. Chapter Three presents the implementation and evaluation method.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# CHAPTER THREE
# METHODOLOGY

## ➢ *Introduction*

This chapter explains how the proposed routing framework was designed and tested. It describes the research design, simulator, routing methods, DQN security agent, evidence gate, attack models, experiment plan and evaluation measures. The chapter also states the limits of the method so that the results can be interpreted correctly.

## ➢ *Research Design*

- *Experimental Variables and Controls*

The independent variables were the routing method, attack type, deployment scenario and random seed. The main dependent variables were attack-step detection, hard and soft false-action rates, packet delivery ratio, end-to-end delay, energy per delivered packet and network lifetime. The same scenario settings and seed were used across methods where possible so that the comparison reflected the routing method rather than a different network realisation.

The experiment controlled the number of simulation steps, route-update interval, security-control interval and ant-exploration interval. Each main configuration used 30 independent seeds. Repetition was needed because node position, link quality, traffic and attack effects include random variation.

We kept the attack tests apart from the clean checks. In the attack part, each run tested whether detection and routing worked when things were hostile. In the clean part, we looked at how often security took extra steps when there was no malicious node at all. We did this on purpose. A system can look safer if it applies harsh penalties to many nodes, even when the threat is not there. The clean checks made the true cost of that kind of handling clear.

The study used a quantitative experimental design. A Python discrete-event simulator was developed to provide the same network conditions to five routing methods. Each configuration was repeated with independent random seeds. This made it possible to compare security and routing outcomes under controlled conditions.

The work followed a design-and-evaluation approach. The research artefact was the ABRF. Its main parts were an ACO routing module, a trained DQN security agent and an evidence-gating mechanism. The artefact was evaluated against four baselines and across several attack and deployment conditions.

## ➢ *System Architecture*

- *Decision Cycle of the Framework*

Each decision cycle began with observations from the network. The observations described recent forwarding, packet drops, route advertisements, energy, link quality and delivery conditions. These values were prepared as the state expected by the trained DQN.

The DQN produced an initial action for a node. The evidence gate then compared the action with current forwarding and route-advertisement evidence. The validated result changed the node security status or pheromone value. The ACO module used the updated information during later route construction.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

Packet forwarding produced new observations for the next cycle. This feedback made route decisions responsive to changing network behaviour. It did not update the neural-network weights. The distinction between live decision feedback and online DQN learning was maintained throughout the methodology.

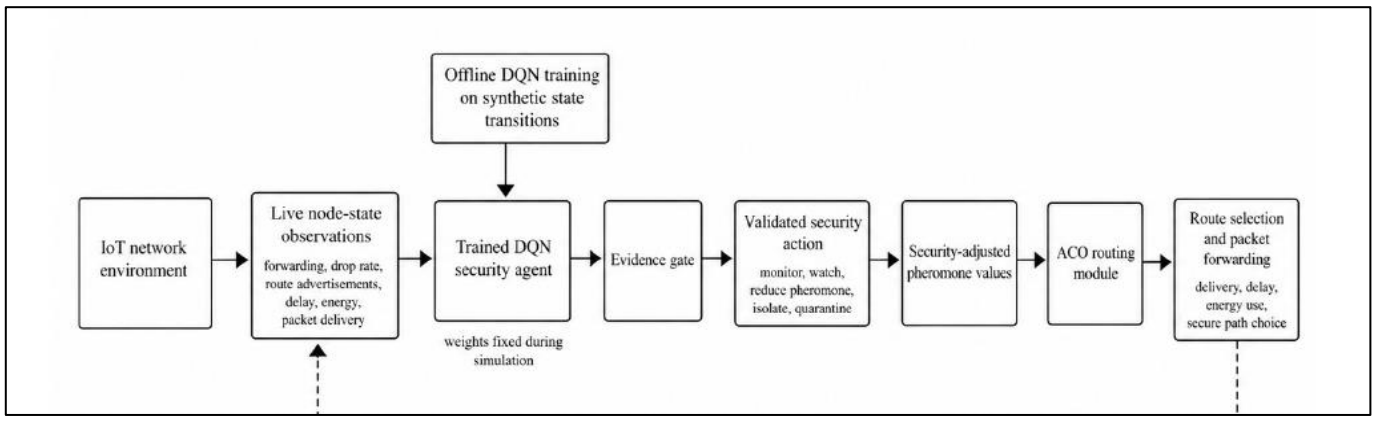


Fig 2 Main Architecture of the ABRF

The IoT network produced live node and route observations. The trained DQN used these observations to propose a security action. The evidence gate then checked whether the current forwarding and route-advertisement evidence supported that action. The validated action changed the security value used by ACO. ACO used the updated pheromone values to select later routes.

The model weights remained fixed during the evaluation runs. The feedback loop supplied new observations for later decisions, but it did not retrain the DQN online.

➢ *ACO Routing Module*

- *Route Heuristic and Weight Change*

The route heuristic combined four normalised measures: residual energy, delay, reliability and security. Residual energy favoured nodes that could continue forwarding. Delay favoured timely paths. Reliability represented recent link and forwarding success. Security represented the effect of the validated security action.

Under normal conditions, energy, delay and reliability received most of the weight, while security remained present at a lower level. Under a threat condition, the security weight increased to 0.45. This did not mean that every low-energy or longer route became preferred. It meant that suspicious behaviour had enough influence to overcome a route that looked attractive only from performance measures. The reliability weight was reduced because recent forwarding reliability could itself be misleading during sinkhole, wormhole or selective-forwarding behaviour. Under threat, the framework therefore gave more weight to validated security evidence while still retaining reliability as part of the route score.

Pheromone evaporation reduced the influence of old routes. A successful forwarding path received reinforcement, while security penalties reduced the attractiveness of suspicious nodes or links. Because evaporation occurred with path updates, its practical effect depended on traffic and update frequency. This implementation detail was kept constant during the main comparisons.

ACO represented the network as a graph. Artificial ants explored possible next hops and preferred links with stronger pheromone and better current routing conditions. Pheromone increased on useful paths and reduced through evaporation. This allowed old or weak paths to lose influence over time.

The route score combined residual energy, delay, reliability and security. Under normal conditions, the weights were 0.35 for energy, 0.30 for delay, 0.25 for reliability and 0.10 for security. Under threat conditions,

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

the security weight increased to 0.45, while the other weights were reduced. This change allowed suspicious behaviour to influence path selection without removing the other routing needs.

The normalised route heuristic for a candidate next hop j was represented as

$$b\,ij \;+\; wR\,Rij \;+\; wS\,Sj\,, \eta ij \;=\; wE\,Ej \;+\; wD\,(1 \;-\; D \qquad (1)$$

B ij is normalised delay, Rij is reliability, Sj is the security where Ej is residual energy, D value, and the weights sum to one. The probability of choosing candidate j from node i was

$$\tau ij\alpha\ \eta ij\beta\ Pij = P\ \alpha\ \beta\ , \qquad (2)$$

$$k\in Ni\ \tau ik\ \eta ik$$

Where τij is pheromone and Ni is the set of available next hops. Pheromone was then updated through evaporation, route reinforcement and any validated security penalty:

$$\tau ij \;\leftarrow\; (1 \;-\; \rho)\tau ij \;+\; \Delta\tau ij \;-\; \lambda Aij\,, \qquad (3)$$

Where ρ is the evaporation rate, Δτij is the route-quality reinforcement, and Aij is the penalty produced by the validated security action.

Table 3 Route-Heuristic Weights Applied Under Normal and Threat Conditions

| Condition | Energy | Delay | Reliability | Security |
|---|---|---|---|---|
| Normal | 0.35 | 0.30 | 0.25 | 0.10 |
| Threat | 0.20 | 0.25 | 0.10 | 0.45 |

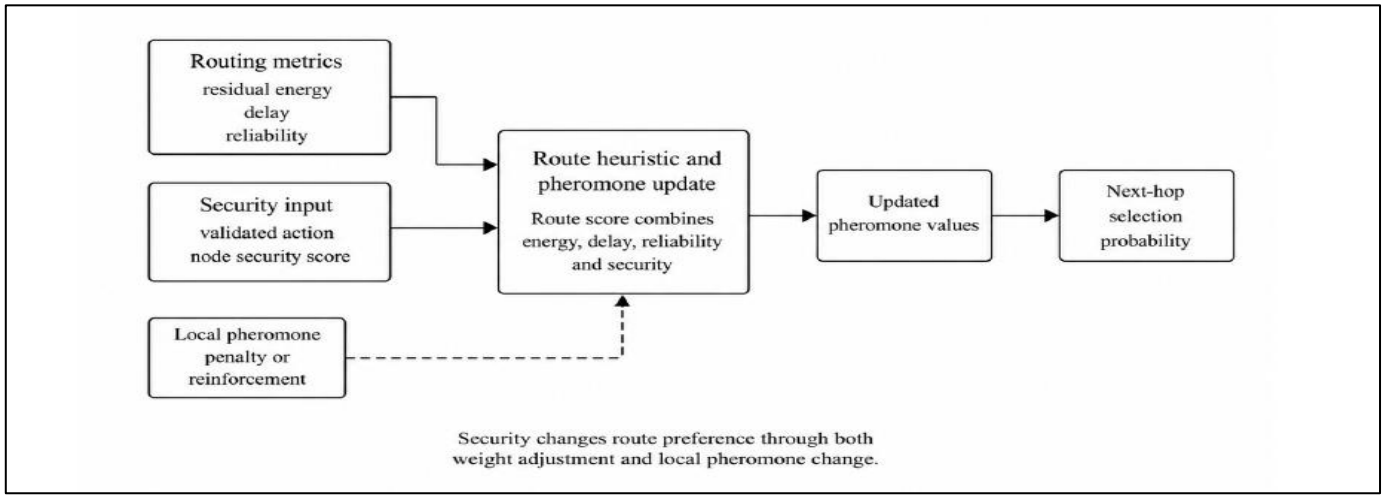



Fig 3 Security-Enabled Pheromone Integration

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

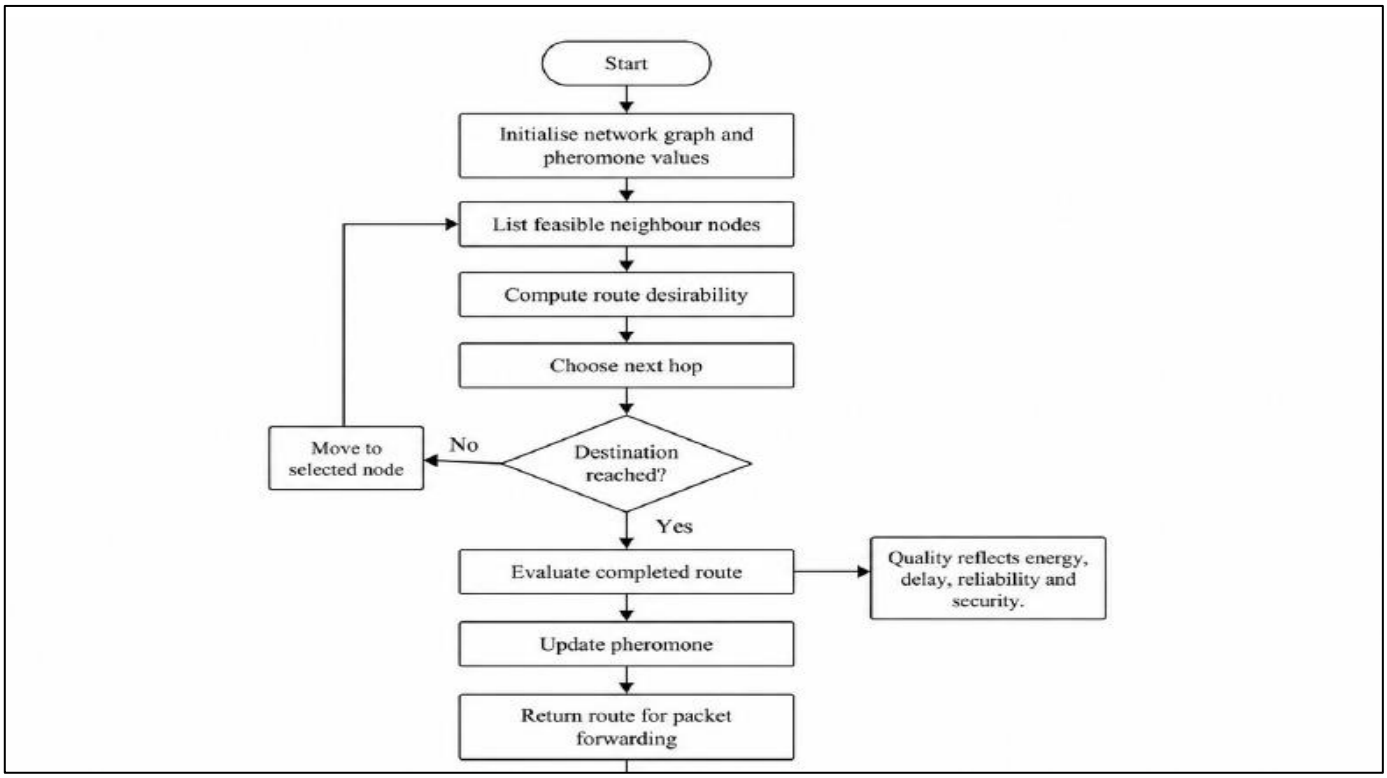


Fig 4 ACO Route Construction Process

➢ *DQN Security Agent*

- *State Representation*

The 11-value state vector was designed to describe both security and routing condition. Forwarding and drop features represented packet-handling behaviour. Route-advertisement features represented claims that could attract traffic. Energy and link features helped the model distinguish malicious behaviour from ordinary weakness. Delivery-related features described the recent effect on network service.

All features were scaled into ranges suitable for the neural network. The same order and meaning were used during training and evaluation. A mismatch between training and live feature order would make the policy unreliable, so the state construction was defined once and reused by the simulator.

- *Action Space and Reward Logic*

The system used five actions in a set order. Action 0 left normal monitoring as it was.

✓ Action 1 raised attention, but route availability stayed the same.
✓ Action 2 lowered pheromone.
✓ Action 3 cut the node out of the active route choices.
✓ Action 4 used the highest quarantine level.

In offline training, the malicious synthetic states got better scores for stronger actions. Benign states did better when the monitoring option was used. If an attack was missed, that led to a penalty. If a strong action was used without need, that also led to a penalty. The scoring scheme was meant to support a stepped policy. Still, the values were picked by the researcher. They were not learned from real network cost. That choice is a key constraint of this training setup.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

- *Offline Training Procedure*

Training used a replay buffer, epsilon-greedy action selection and a target network. The replay buffer stored state, action, reward and next-state samples. Epsilon-greedy selection allowed the model to explore different actions early in training and rely more on learned values later. The target network improved stability by providing a slower-changing estimate for the update target.

For each sampled transition (st , at , rt , st+1 ), the target value was

$$yt = rt + \gamma \, max' \, Qtarget \, (st + 1 \, , a' \, ), \tag{4}$$

And the network minimised the mean squared temporal-difference loss

$$L(\theta) = E \, (yt - Q(st \, , at \, ; \, \theta))2 \, . \tag{5}$$

The discount factor γ controlled the value placed on later rewards.

Synthetic next states were created by adding controlled variation to the current state. This produced many training combinations without running the full network simulator inside the training loop. The advantage was simpler and faster training. The disadvantage was a gap between the synthetic transition process and the true effect of actions on the simulated network.

After training, the model was validated on held-out synthetic states and then used in the live simulator. The model weights were fixed for all main experiments. The evaluation therefore tested policy deployment under changing network observations, not continued reinforcement learning from each simulation outcome.

The security agent used an 11-value state vector. The values described forwarding behaviour, route advertisements, energy, link quality and recent delivery conditions. The model contained two hidden layers with 96 and 64 units and an output layer with five actions.

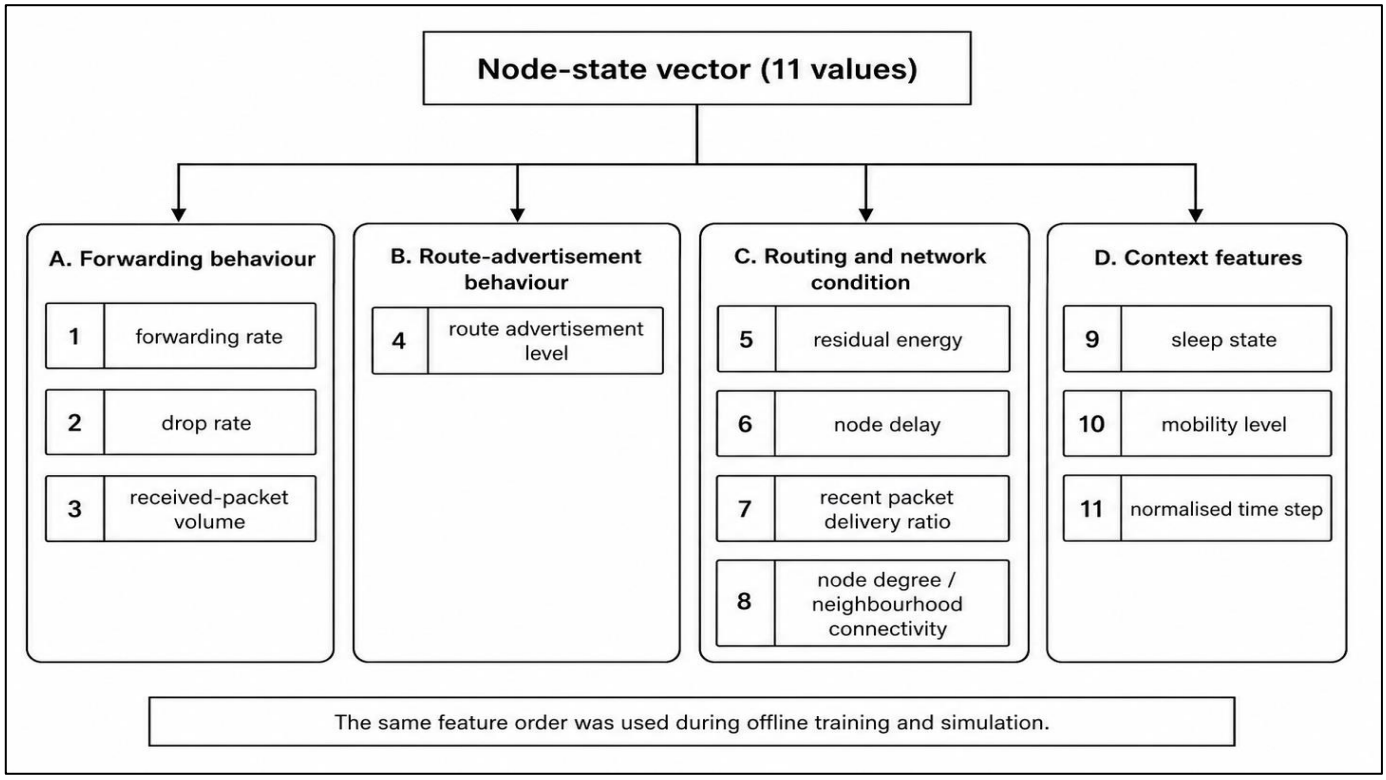



Fig 5 Eleven-Dimensional Node-State Vector

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

Table 4 DQN Action Space and the Routing Effect of Each Action

| Action | Meaning | Routing effect |
|---|---|---|
| 0 | Normal monitoring | No security penalty |
| 1 | Watch / soft caution | Node remains under closer observation |
| 2 | Reduce pheromone | Lowers the node or link routing influence |
| 3 | Isolate | Removes the node from active route choice |
| 4 | Quarantine | Applies the strongest security restriction |

The DQN was trained offline using synthetic node-state transitions. The reward values encouraged stronger actions for malicious states and discouraged unnecessary strong actions for benign states. The next states were generated by adding controlled noise to the current synthetic states. The training process therefore followed a DQN-style update, but it was not end-to-end reinforcement learning from live simulator interactions.

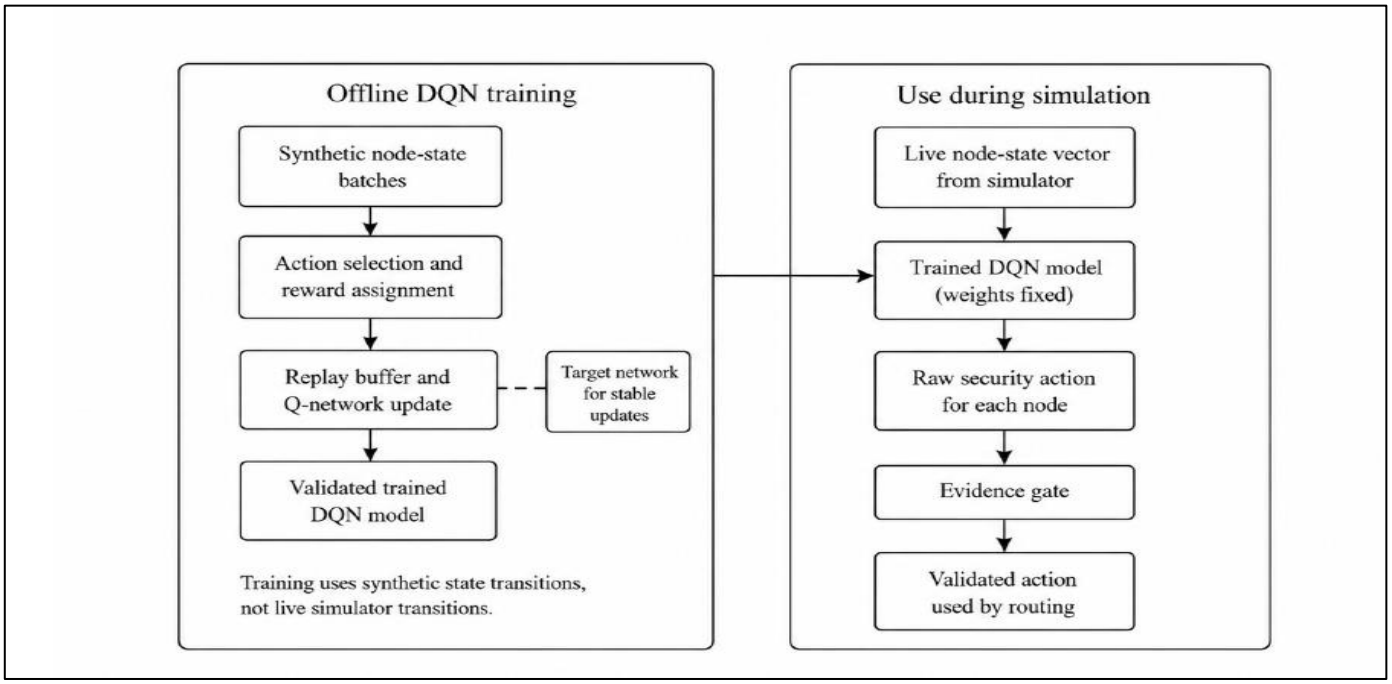



Fig 6 DQN Training and use in the Simulation

➢ *Evidence Gate*

- *Purpose and Decision Rules*

The evidence gate addressed the risk that a neural-network action could be stronger than the live evidence justified. It used explicit checks on forwarding and route-advertisement behaviour. When the DQN proposed isolation but the evidence was weak, the gate could reduce the action. When the evidence clearly showed harmful behaviour, the gate could support a stronger response.

The gate also allowed a cautious pheromone-reduction action when the state was suspicious but not strong enough for isolation. This supported route avoidance without immediately removing the node. The mechanism was especially relevant in lossy IoT networks where ordinary packet loss can resemble selective forwarding.

The final system should therefore be understood as DQN-supported and evidence-gated. Performance cannot be assigned to the DQN alone because the gate changes some actions. This is why the ablation analysis compared reduced variants of the system.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

The evidence gate reduced the risk of applying a strong action based only on the DQN output. It checked live forwarding and route-advertisement indicators before the action changed routing. Weak evidence could reduce an aggressive action, while strong evidence could support isolation. The gate therefore formed part of the final decision mechanism and not only a post-processing report.

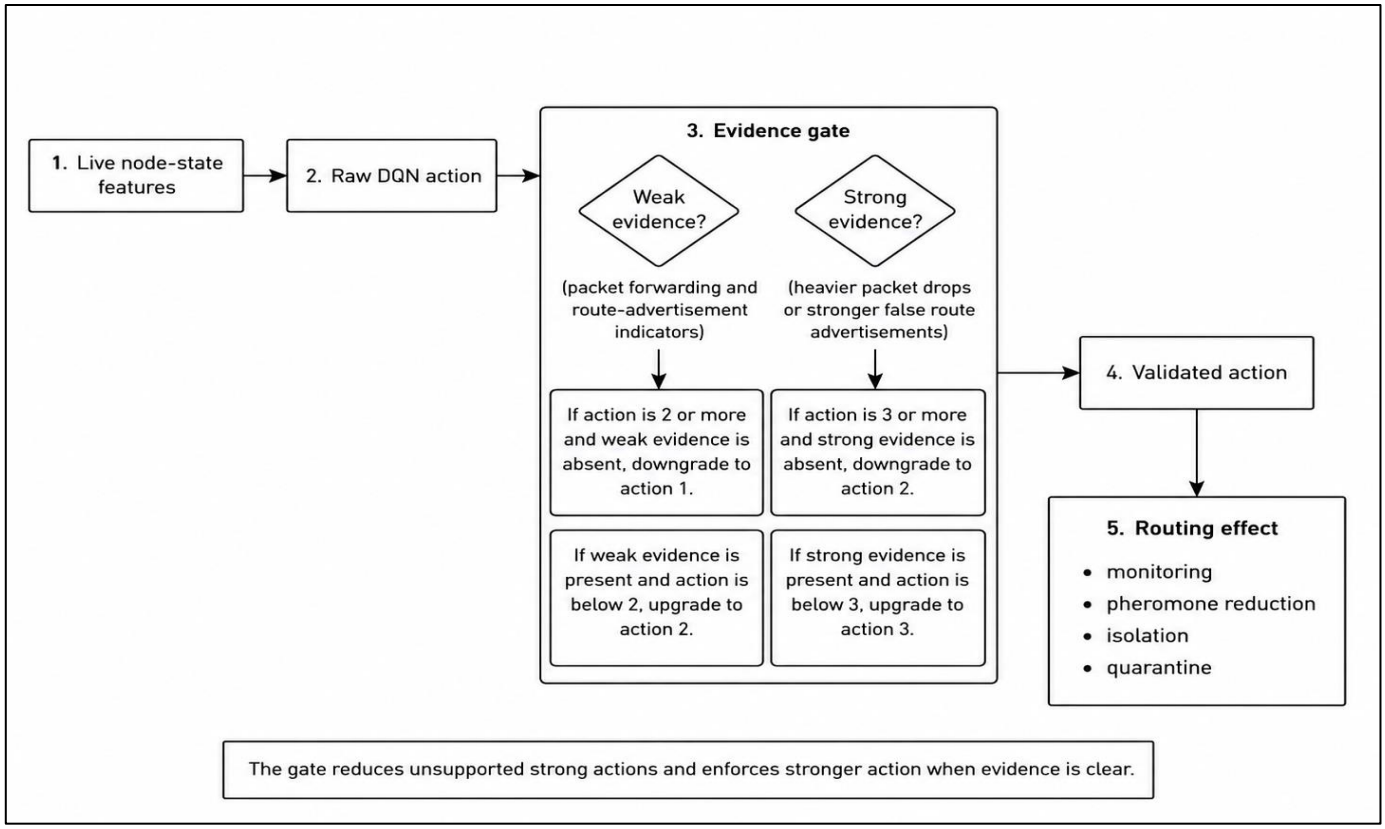



Fig 7 Evidence-Gated Security Decision Process

➢ *Simulation Platform and Network Scenarios*

The simulator was written in Python. It modelled node position, wireless range, link quality, energy use, sleeping nodes, packet generation, packet forwarding and route updates. It was a purpose-built discrete-event environment rather than a full implementation of the RPL and AODV standards.

Table 5 Deployment-Scenario Parameters for the Three Simulated Networks

| Scenario | Nodes | Area | Range | Mobility | Sleep probability | Initial energy | Traffic rate |
|---|---|---|---|---|---|---|---|
| Smart home | 50 | 150 x 150 | 58 | 0.35 | 0.18 | 1.00 | 5 |
| Smart traffic | 150 | 2000 x 2000 | 390 | 7.50 | 0.02 | 3.00 | 12 |
| Wearable IoT | 25 | 20 x 20 | 11.5 | 0.90 | 0.25 | 0.55 | 3 |

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

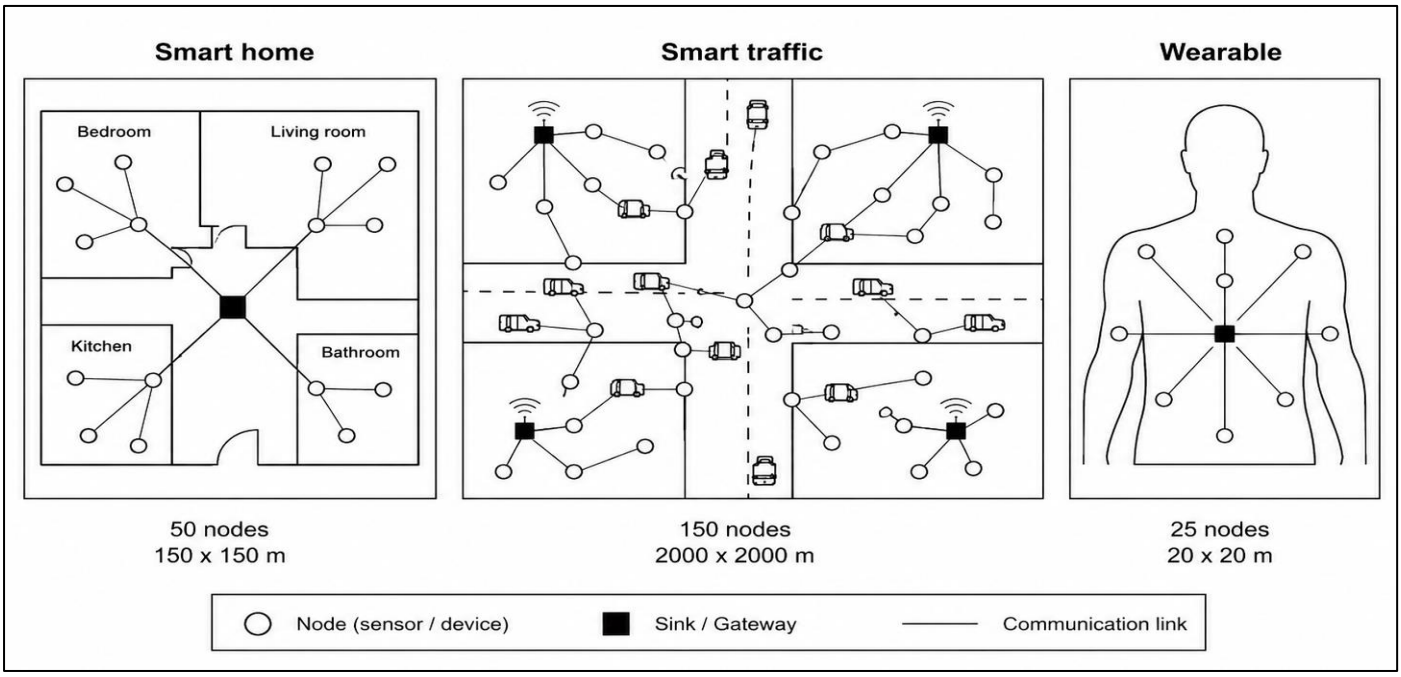



Fig 8 Example Network Layouts for the three Deployment Scenarios

➢ *Baseline Routing Methods*

Five methods were compared. The first two were simplified RPL-like and AODV-like baselines. The third used ACO without the DQN security process. The fourth used an AODV-like route method with a hand-coded traditional IDS. The fifth was the full ACO-DQN framework with evidence gating. The terms RPL and AODV in the results refer to these simplified simulator versions and not full standards-compliant implementations.

Table 6 Baseline routing methods compared in this study

| Method | Main role |
|---|---|
| RPL-like | Tree-oriented baseline for constrained routing |
| AODV-like | On-demand route-discovery baseline |
| ACO-only | Bio-inspired routing without the DQN security process |
| Traditional IDS + AODV-like | Threshold-based detection with separate routing |
| Full ACO-DQN | ACO routing with DQN-supported, evidence-gated security actions |

➢ *Attack Models*

- *Selective Forwarding*

The selective-forwarding model caused malicious nodes to drop a controlled share of packets while forwarding the rest. Partial forwarding made the attack less obvious than a complete blackhole. The model tested whether several behaviour features and repeated observations could identify the gradual loss.

- *Sinkhole*

The sinkhole model allowed a malicious node to advertise an attractive route condition so that traffic moved toward it. The routing effect was important because ACO can strengthen a path that appears useful. The security mechanism had to reduce the route influence before the false attraction caused continued packet loss.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

- *Wormhole*

The wormhole model showed a made up shortcut between nodes that work together. Once the shortcut existed, the path scores people saw could shift, and traffic might move away from the usual nearby nodes. This was a broad behaviour level description. It was not meant to match every timing detail or each packet flow you would see in a real wormhole tunnel.

- *Coordinated and Adaptive Attacks*

The coordinated condition combined malicious behaviours across more than one node. The adaptive condition changed behaviour to reduce the value of fixed thresholds. These scenarios tested whether the framework could respond when the attack was not represented by one constant pattern. They remained controlled models rather than a complete catalogue of adversarial strategies.

The evaluation used selective forwarding, sinkhole, wormhole, coordinated and adaptive attacks. The models were simplified behavioural representations designed for controlled comparison. They did not reproduce every message and timing rule found in real protocol implementations.

Table 7 Attack Models and Their Simulated Behaviour

| **Attack** | **Simulation behaviour** |
|---|---|
| Selective forwarding | Malicious nodes dropped part of the traffic they were expected to forward. |
| Sinkhole | A malicious node advertised attractive route information to draw traffic. |
| Wormhole | Two malicious locations created an artificial short route. |
| Coordinated | Several malicious nodes acted together. |
| Adaptive | Malicious behaviour changed to reduce the chance of triggering fixed checks. |

➢ *Experiment Design*

The main attack experiment used five routing methods, three deployment scenarios, five attacks and 30 independent seeds. This produced 2,250 attack runs. The clean-control experiment used the same five methods and three scenarios with 30 seeds, producing 450 runs. Additional experiments examined scalability, sensitivity, ablation and model quantisation.

Table 8 Experiment Design and Number of Runs Per Experiment

| **Experiment** | **Runs / observations** |
|---|---|
| Main attack grid | 2,250 |
| Clean controls | 450 |
| Scalability | 180 |
| Ablation | 192 |
| Sensitivity | 100 |
| Measured simulation runs | 3,152 |
| Illustrative transfer projection draws | 30 |

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

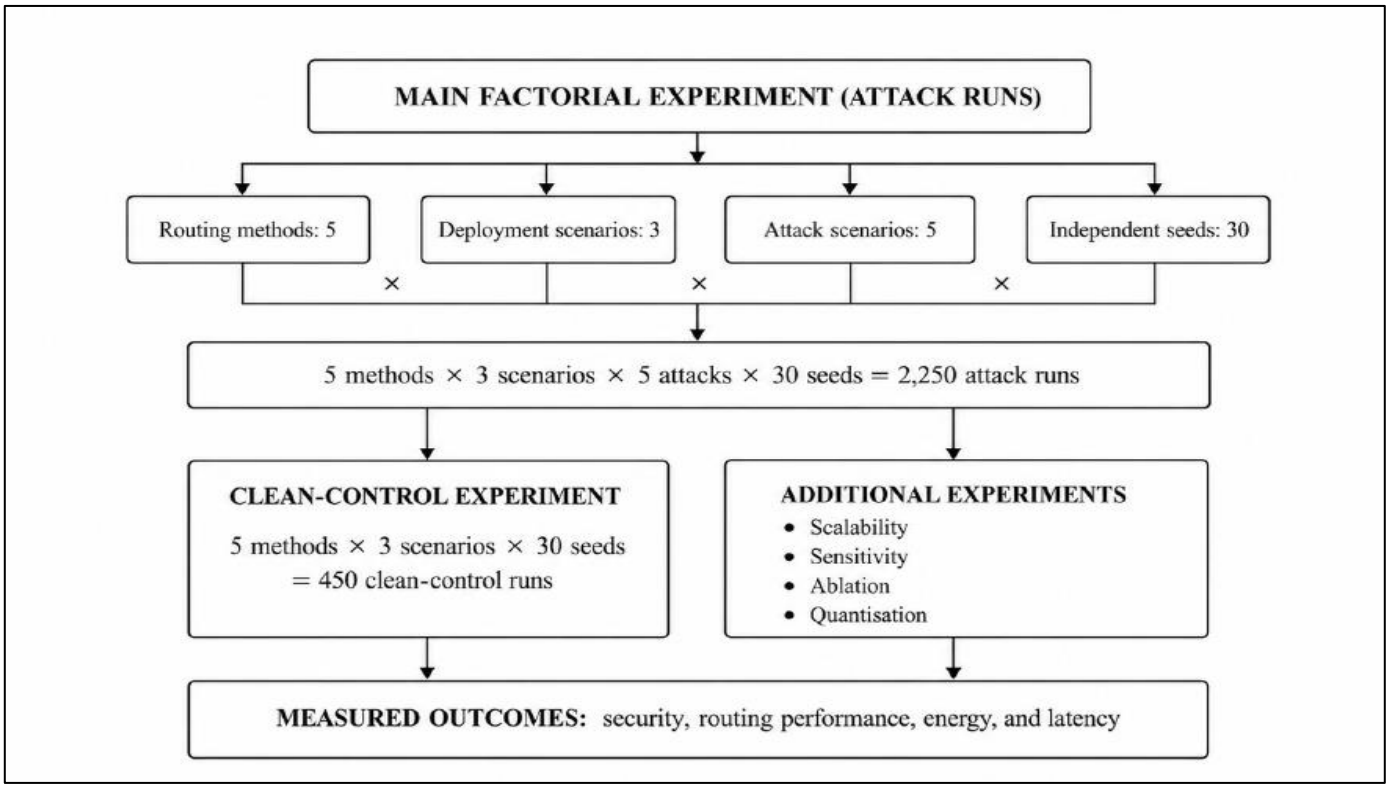



Fig 9 Experiment Design Matrix

➢ *Evaluation Measures*

Security performance was measured using attack-step detection, hard false-action rate, soft false-action rate and first-detection latency. Attack-step detection shows the share of attack control steps in which at least one malicious node was identified. It is not the same as node-level recall. The hard false-action rate counts stronger isolation actions against benign nodes. The soft false-action rate also includes pheromone-reduction actions.

Routing performance was measured using packet delivery ratio, end-to-end delay, energy per delivered packet, route length and related operational measures. Detection latency was measured in simulation steps. Any conversion to seconds depends on an explicit step-duration assumption.

The main measures were calculated as

$$Ndelivered\ PDR\ = , \tag{6}$$

$$Ngenerated\ Nattack\ steps\ detected\ Detection\ Rate\ = , \tag{7}$$

$$Nattack\ control\ steps\ Nstrong\ false\ actions\ Hard\ FAR\ = , \tag{8}$$

$$Nbenign\ control\ decisions\ Nsoft\ or\ strong\ false\ actions\ Soft\ FAR\ = , \tag{9}$$

$$Nbenign\ control\ decisions\ NX\ delivered\ 1\ D\ =\ Dp\ , \tag{10}$$

$$Ndelivered\ p = 1\ Etotal\ Epacket\ = . \tag{11}$$

$$Ndelivered$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

These definitions were applied consistently across the compared methods.

Table 9 Evaluation Measures and their Preferred Direction

| Measure | Meaning | Preferred direction |
|---|---|---|
| Attack-step detection rate | Share of attack control steps with at least one malicious node detected | Higher |
| Hard false-action rate | Strong action against benign nodes | Lower |
| Soft false-action rate | Pheromone reduction or stronger action against benign nodes | Lower |
| Packet delivery ratio | Delivered packets divided by generated packets | Higher |
| End-to-end delay | Mean delivery delay for successful packets | Lower |
| Energy per delivered packet | Energy used relative to successful delivery | Lower |

➢ *Data Analysis*

- *Descriptive and Inferential Analysis*

The first stage of analysis calculated the mean and variation of each measure across seeds. Results were grouped by method, scenario and attack so that overall values did not hide a weak condition. Clean-control results were analysed separately from attack results.

Omnibus tests were used to check whether the methods differed across repeated conditions. Pairwise comparisons then examined the proposed framework against each baseline. Corrected significance levels were used within the defined comparison families. Statistical significance was interpreted together with the size and direction of the difference, because a small difference can become statistically significant when many repeated observations are available.

The work did not treat the simulator trials as proof that the method will work the same everywhere. The findings only reflect what happened when we ran the setup with the specific settings. The attack model, the way the defenses were built, and the baseline code all mattered. Because of that, the write up avoids broad claims like “best in all IoT networks”. It sticks to wording tied to the test setup, such as “under the tested conditions”.

For the numbers, we used descriptive statistics and did overall checks, plus paired tests. We used matched random seeds so the runs stayed aligned across cases. The reading of the outcomes also stays tied to the simulator setup and its assumptions. Even when a result is statistically significant, that does not guarantee the same result in a real IoT deployment.

➢ *Ethical and Reproducibility Considerations*

The researchers ran tests on simulated data. No people took part, and there was no private or personal information used. They made it easier to repeat the work by using the same setup each time. They also used separate random seeds, and they stored the output files.

➢ *Chapter Summary*

This chapter described the complete experimental method. The proposed framework combined ACO routing, an offline-trained DQN and an evidence gate. It was tested in a controlled Python simulator against four simplified baselines across three IoT scenarios and five routing attacks. Chapter Four presents the results.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# CHAPTER FOUR
# RESULTS AND DISCUSSIONS

➢ *Introduction*

This chapter presents the measured results of the simulation experiments. The discussion follows the three research questions: the behaviour of the DQN-supported security process, the effect of security actions on ACO routing, and the performance of the full framework compared with the baselines.

➢ *DQN Training and Pre-Deployment Validation*

- *Interpretation of the Training Curve*

The training reward increased and then became more stable. This pattern shows that the network learned the reward structure used in the synthetic training process. It does not prove that the agent learned from real packet forwarding, because the training transitions were generated outside the live simulator.

Held-out validation showed that the model could separate the synthetic malicious and benign states used in the training design. The result supported deployment of the fixed policy in the simulator. The stronger test was therefore the main experiment, where the model received live observations produced by routing, mobility, sleeping nodes and attacks.

The training reward increased over the 260 training epochs and later became more stable. This shows that the model learned the reward pattern in the synthetic training distribution. It does not show that the model learned directly from live network episodes.

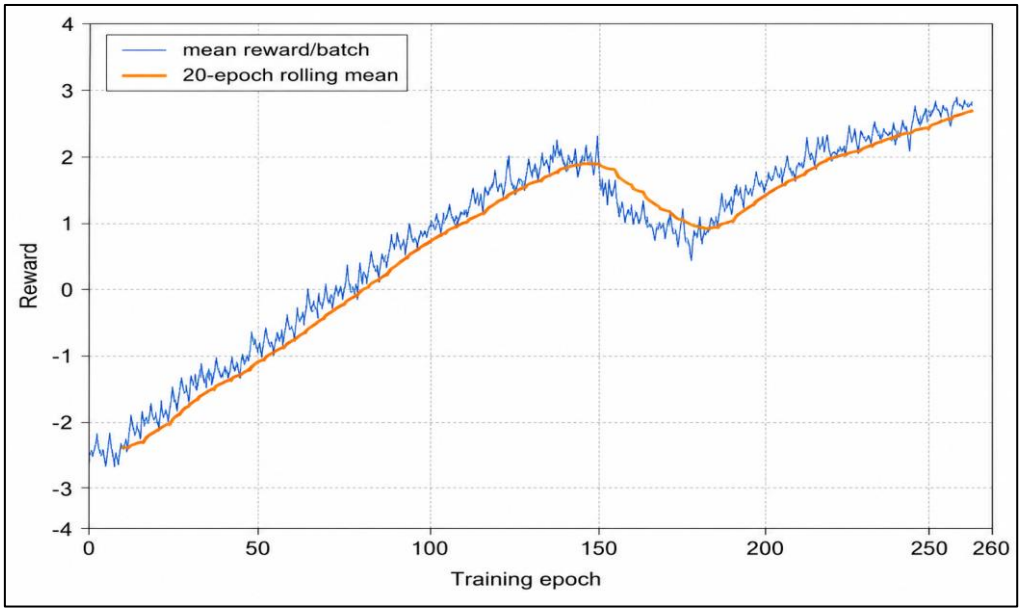



Fig 10 DQN Training Reward

Table 10 DQN validation Results on held-Out Synthetic States

| Validation measure | Value |
|---|---|
| Validation accuracy | 0.954 |
| Malicious-state response rate | 0.939 |
| Benign hard false-action rate | 0.014 |

These validation results were obtained on synthetic holdout states. The more important test was therefore the fixed model's behaviour inside the live simulator.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

➢ *Overall Security Performance*

- *Detection Performance*

The full framework achieved a mean attack-step detection rate of 0.847. The traditional IDS achieved 0.681. The difference shows that the combined DQN, evidence gate and routing feedback identified harmful behaviour more often during attack control steps in the tested scenarios.

The value is an attack-step presence measure. It records whether at least one malicious node was identified during a control step. It should not be read as the proportion of all malicious nodes detected. Node-level precision, recall and F1-score would provide a different view and remain useful for later work.

- *Hard and Soft False Actions*

Within the 2,250 attack-run dataset, the hard false-action rate of 0.033 shows that strong isolation actions against benign nodes were uncommon. This is a useful result because unnecessary isolation can reduce connectivity in a small network. The traditional IDS had a higher hard false-action rate of 0.126 under the study definition.

The soft false-action rate of 0.312 gives a more cautious interpretation. It includes pheromone reduction against benign nodes and is therefore much higher than the hard rate. The framework often reduced routing preference before it had enough evidence for isolation. This behaviour may protect the network, but it can also move traffic away from useful nodes. Both values are needed to understand the security cost.

Table 11 Security-Performance Comparison between the Traditional IDS and the full ACO-DQN Framework

| Protocol | Attack-step detection rate | Hard false-action rate | Soft false-action rate |
|---|---|---|---|
| Traditional IDS | 0.681 | 0.126 | Not separately reported |
| Full ACO-DQN | 0.847 | 0.033 | 0.312 |

The full framework detected malicious activity in a larger share of attack steps and applied fewer hard actions to benign nodes than the traditional IDS. However, the soft false-action rate was much higher than the hard rate. This means that the framework often applied cautious pheromone reduction to benign nodes even when it avoided isolation. Both rates are needed for a balanced interpretation.

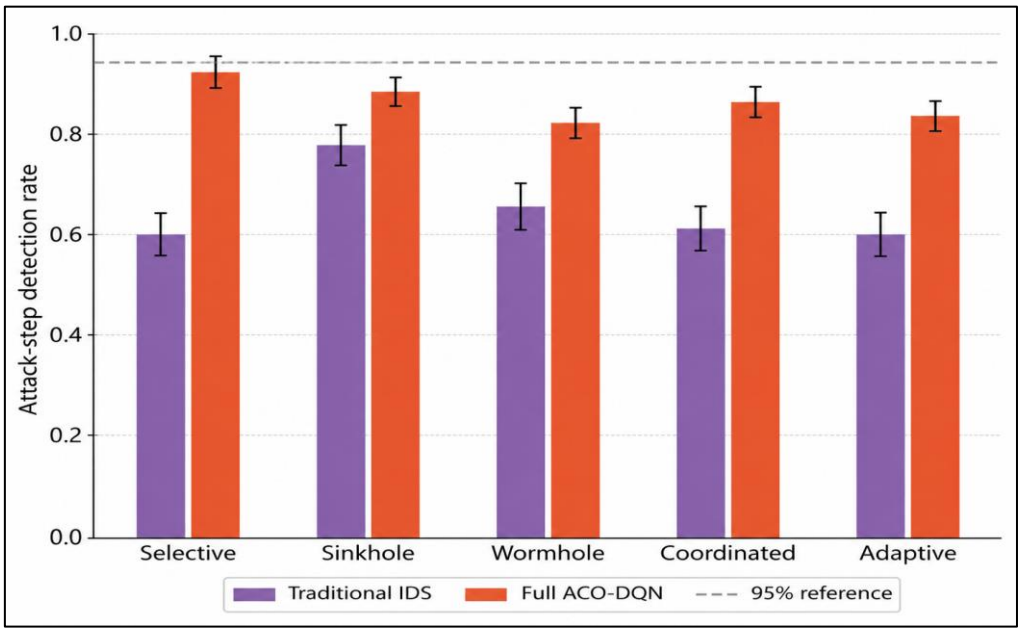



Fig 11 Detection Rate by Attack Type

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

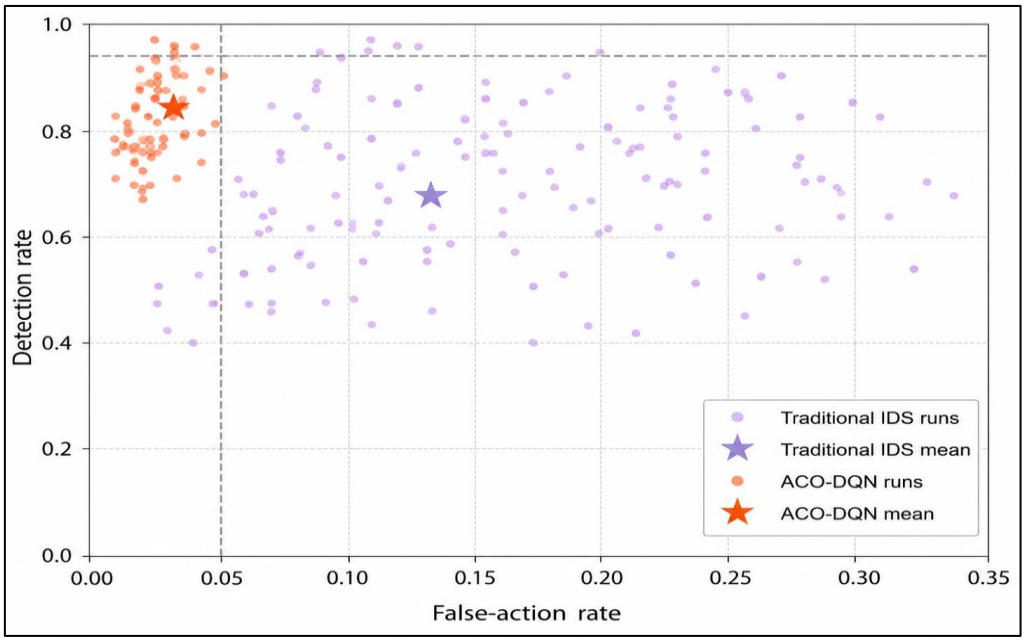



Fig 12 Detection and False-Action Trade-off

➢ *Inferential Statistical Results*

The inferential tests confirmed that the differences were not limited to descriptive means. The omnibus analysis found significant differences among the five routing methods in 36 of the 40 attack-metric combinations tested. The four non-significant cases were mainly network-lifetime comparisons, where the methods produced similar results under some attacks.

Matched-seed pairwise tests then compared the full ACO-DQN framework with the traditional IDS. After Bonferroni correction, the differences were significant for detection rate, hard false-action rate, packet delivery ratio and latency under all five attacks. Corrected p-values were below 0.001 in every comparison. The sinkhole detection comparison had the largest corrected value, p = 0.0009, and was still significant. The effect-size ranges are summarised in Table 12.

Table 12 Summary of Paired ACO-DQN and Traditional-IDS Comparisons

| Metric | Attacks significant | Cohen's dz range | Direction |
|---|---|---|---|
| Detection rate | 5 of 5 | 0.404 to 0.771 | Higher for ACO-DQN |
| Hard false-action rate | 5 of 5 | -1.083 to -1.034 | Lower for ACO-DQN |
| Packet delivery ratio | 5 of 5 | 1.342 to 2.277 | Higher for ACO-DQN |
| Latency | 5 of 5 | -0.713 to -0.679 | Lower for ACO-DQN |

These results support the direction of the descriptive findings. They remain limited to the matched simulation seeds, attack models and baseline implementations used in this study.

➢ *Overall Routing Performance*

- *Packet Delivery*

The proposed framework achieved a mean packet delivery ratio of 0.665, compared with 0.579 for the traditional IDS. The result suggests that placing the security response inside route selection helped traffic avoid harmful paths earlier. The value remained below perfect delivery because the scenarios also included mobility, sleep, weak links and attack effects.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

The simplified RPL-like and AODV-like methods produced lower delivery under the main attack conditions. Their results should not be interpreted as the performance of full standards-compliant implementations. They show how the route ideas behaved inside the common purpose-built simulator.

- *Delay and Energy*

Security improvement created a routing trade-off. Moving away from a suspicious route can increase hop count or use a less direct path. The reported delay and energy results therefore show whether the security gain was achieved at an unacceptable network cost.

The mean delay of the proposed method remained within the tested operating range, while energy per delivered packet stayed low relative to the value of successful delivery. These results support the feasibility of the route mechanism in simulation, but physical-device energy measurements are still required before deployment claims can be made.

Table 13 Routing-Performance Comparison Across the Compared Protocols

| Protocol | Packet delivery ratio | Mean delay (ms) | Energy per delivered packet |
|---|---|---|---|
| RPL-like | 0.503 | 32.987 | 0.0067 |
| AODV-like | 0.551 | 31.748 | 0.0065 |
| ACO-only | 0.587 | 30.931 | 0.0061 |
| Traditional IDS | 0.579 | 30.202 | 0.0065 |
| Full ACO-DQN | 0.665 | 29.480 | 0.0063 |

The full framework produced the highest average packet delivery ratio. Its mean delay was also slightly lower than the other methods. Energy per delivered packet was close to ACO-only and lower than the two conventional baselines. This suggests that the security integration improved delivery without creating a large energy penalty under the tested settings.

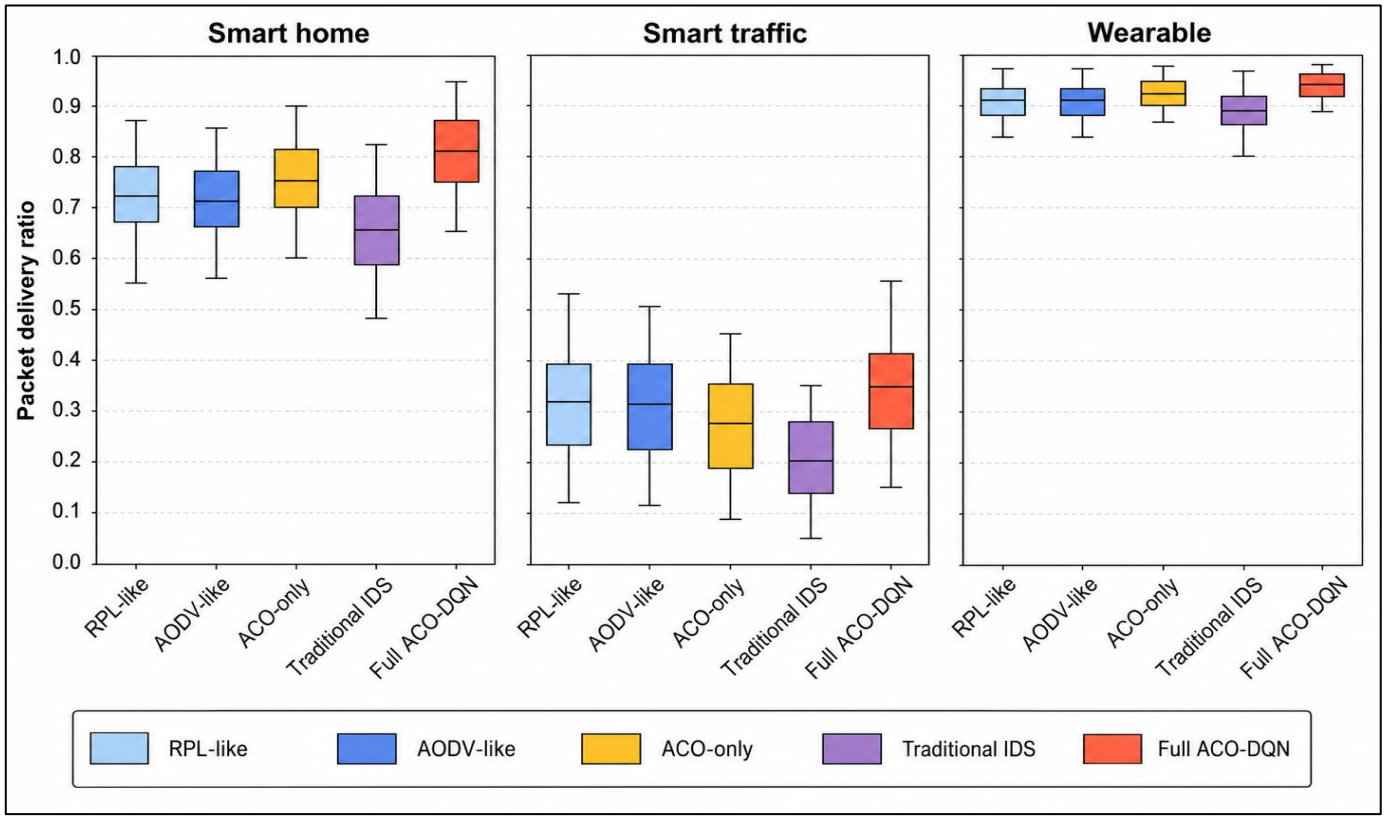



Fig 13 Packet Delivery Across Deployment Scenarios

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

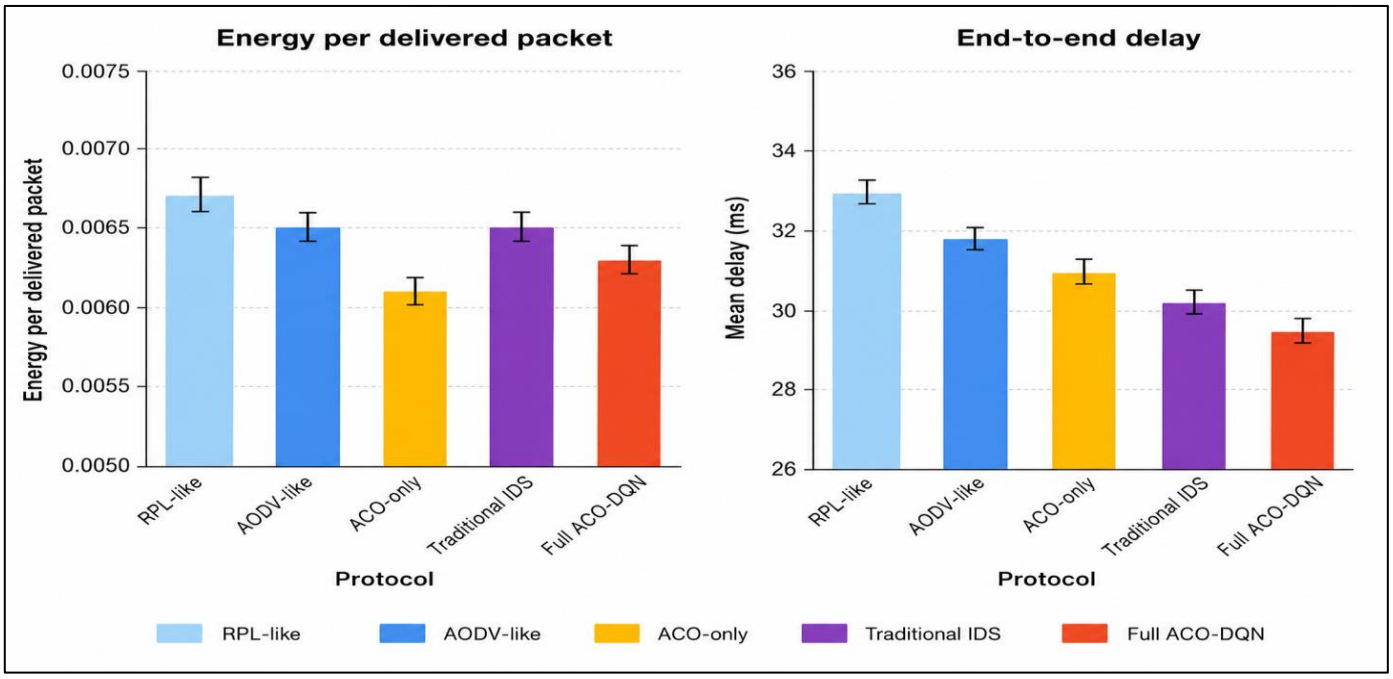



Fig 14 Energy use and end-to-end Delay

➢ *Performance by Attack Type*

Results were not the same for each attack. That was mostly because every attack affected different parts of what was measured. In the selective forwarding case, packet delivery shifted slowly over time. With the sinkhole, the main effect came from how traffic was pulled toward a target path. The wormhole led to a different view of how good a route looked. Coordinated and adaptive attacks also altered more than one signal at once. One overall mean therefore cannot describe every security condition.

The framework performed best when malicious behaviour produced consistent forwarding or route-advertisement evidence. More subtle conditions created a harder balance between early response and false action. The evidence gate reduced some aggressive DQN actions, which helped the hard false-action rate but could delay a strong response in ambiguous states.

The attack-level results support the use of several input features. A single packet-drop threshold would be weak against route-advertisement attacks, while route information alone would be weak against selective forwarding. Combining forwarding, route, energy and link features allowed the system to use a broader view of behaviour.

Performance changed across attacks. Selective forwarding and sinkhole behaviour produced clearer forwarding and route-advertisement evidence. Wormhole and adaptive attacks were more difficult because malicious behaviour could appear normal at the individual-node level or change over time. The full framework still performed better than the traditional IDS across the tested attack groups, but the result does not mean that every malicious node was found.

➢ *Clean-Control Results*

Clean controls were essential because the same network conditions that support IoT operation can look suspicious. Sleeping nodes, mobility and weak links may reduce forwarding without an attacker. A security method that reacts strongly to every loss would damage the network during normal use.

The clean hard false-action score stayed low. That pattern means the system usually did not pick harmless nodes as if they were bad. The soft score was higher. That also points to the fact that small pheromone cuts still happened even in clean conditions.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

From the clean-control runs, the value near 0.037 came out of the separate 450-run clean-only set. That is why it does not use the same base as the 0.033 number noted for benign nodes inside attack runs. These control results also help explain why a single headline false-positive number does not tell the whole story. In practice, people might allow short-lived route de-prioritization. They may not accept repeated node isolation. Treating the hard and soft parts as separate outcomes makes the results easier to use for routing rules.

The clean-control tests were meant to track what the system did when there was no attack. In those runs, the framework showed a hard false-action rate near 0.037. It also showed a soft false-action rate near 0.320. The gap suggests that most unnecessary events were the cautious pheromone reductions, not real isolation. That choice limited the chance of severe wrong actions. Even so, it could still reduce routing speed or quality, so more checking is needed on how this plays out over time.

➢ *Pheromone Behaviour*

The graphs of pheromone traces explain how the security outcome shaped routing. After a node got a penalty, its influence on routes dropped. After that, later ants were less likely to pick the same node. This is the core way the method went past using a separate intrusion detector.

Pheromone values did not drop for one cause only. Evaporation played a role, and route quality mattered too. So the figure reflects several things at once. It shows routing success, the normal loss from evaporation, and the parts of the security action that were confirmed. The study also compared the same path with the penalty and without it, so the pull of the penalty could be seen.

This shift did not last the same way in every run. If later actions worked well, the route updates could bring the node back into better standing. What happened depended on the action details and how the system was set up. This suggests the design can recover from a mild or short lived suspicion. A separate test on how long recovery takes was not part of this work.

The pheromone study indicates that the security step changed route preference. It did not just act like an external warning. When there was evidence of danger, the suspicious nodes and links got weaker routing influence. That point is the key contribution of the framework.

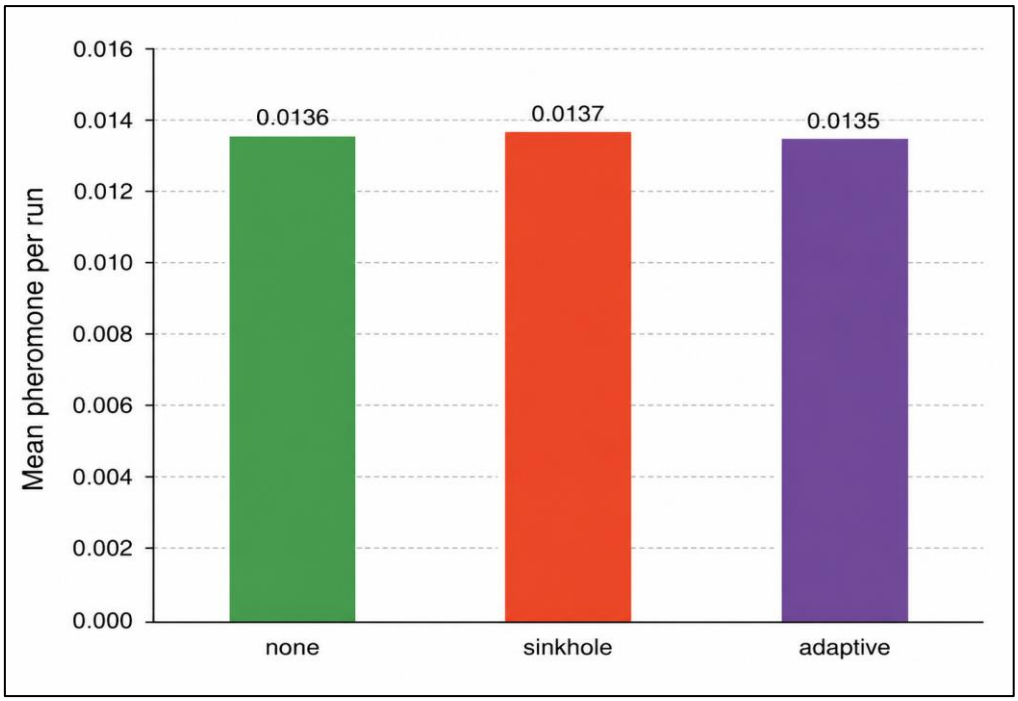



Fig 15 Mean Pheromone Behaviour Under Selected Conditions

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

➢ *Ablation Analysis*

The ablation analysis examined whether the full result required more than ordinary ACO or a separate detector. ACO-only showed the value of performance-based pheromone without the DQN security process. The traditional IDS comparison showed the value of detection that remained separate from ACO route preference.

The full framework performed better on the combined security and routing measures. This supports the integration idea, but the existing ablation does not isolate every component. A stronger future design should compare DQN without the evidence gate, the evidence gate without DQN, and the full DQN-gate combination under the same ACO route engine.

The conclusion is therefore limited. The study shows that the complete implemented framework outperformed the selected reduced baselines. It does not prove the exact share of improvement produced by the DQN and the evidence gate separately.

The available ablation compared ACO-only, traditional detector plus AODV-like routing, and the full ACO-DQN system. The full system produced the best combined result. However, this comparison does not isolate the independent contribution of the DQN and the evidence gate because several components change at the same time. The result supports the value of the full configuration, but it should not be treated as a complete causal ablation.

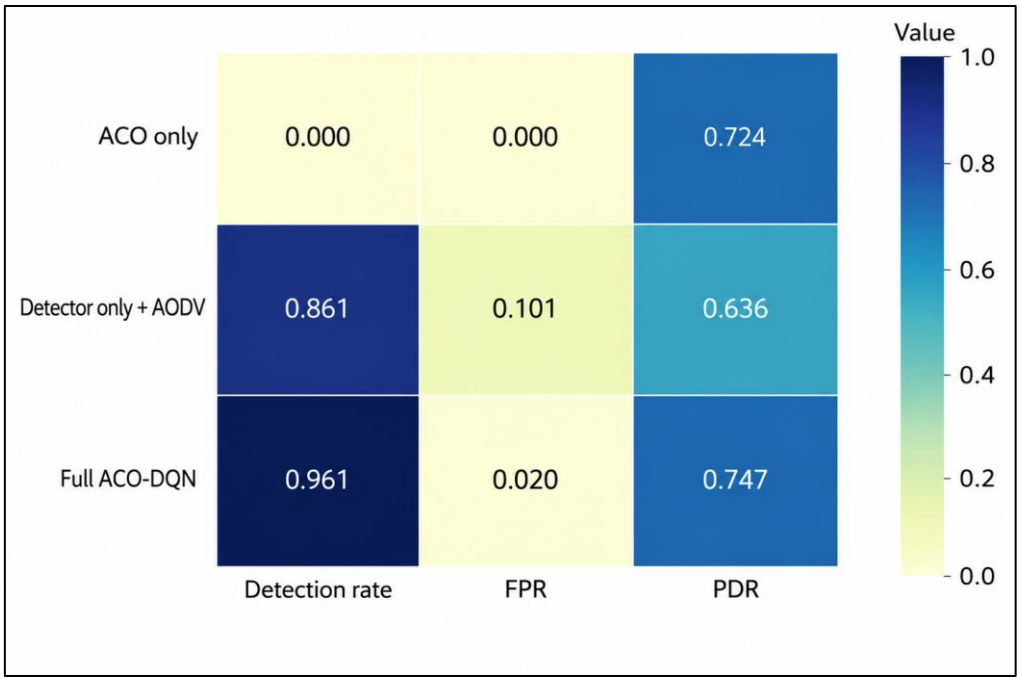



Fig 16 Ablation Results

➢ *Scalability*

The scalability experiment increased the number of nodes from 50 to 500. The proposed method remained operational as the network grew, which supports the use of local routing and a small action model. However, the area and radio settings meant that node density also changed. The experiment therefore measured a combination of larger scale and denser connectivity.

A constant-density experiment would provide a cleaner view of scale. In that design, the physical area would grow with the number of nodes so that each node retained a similar average number of neighbours. The

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

current result is still useful, but it should be described as tested scalability under the selected smart-traffic settings.

Computation at the security agent also matters. A gateway that evaluates more nodes must complete more inferences within the control interval. The simulator measured model-level inference on a Colab CPU, not complete gateway workload with radio communication and operating-system overhead.

The scalability experiment increased the node count from 50 to 500 in the smart-traffic setting. Detection remained high, while delay increased with network size. Because the physical area and communication range were held fixed, the experiment changed both node count and network density. It therefore measures scaling under densification rather than pure constant-density scalability.

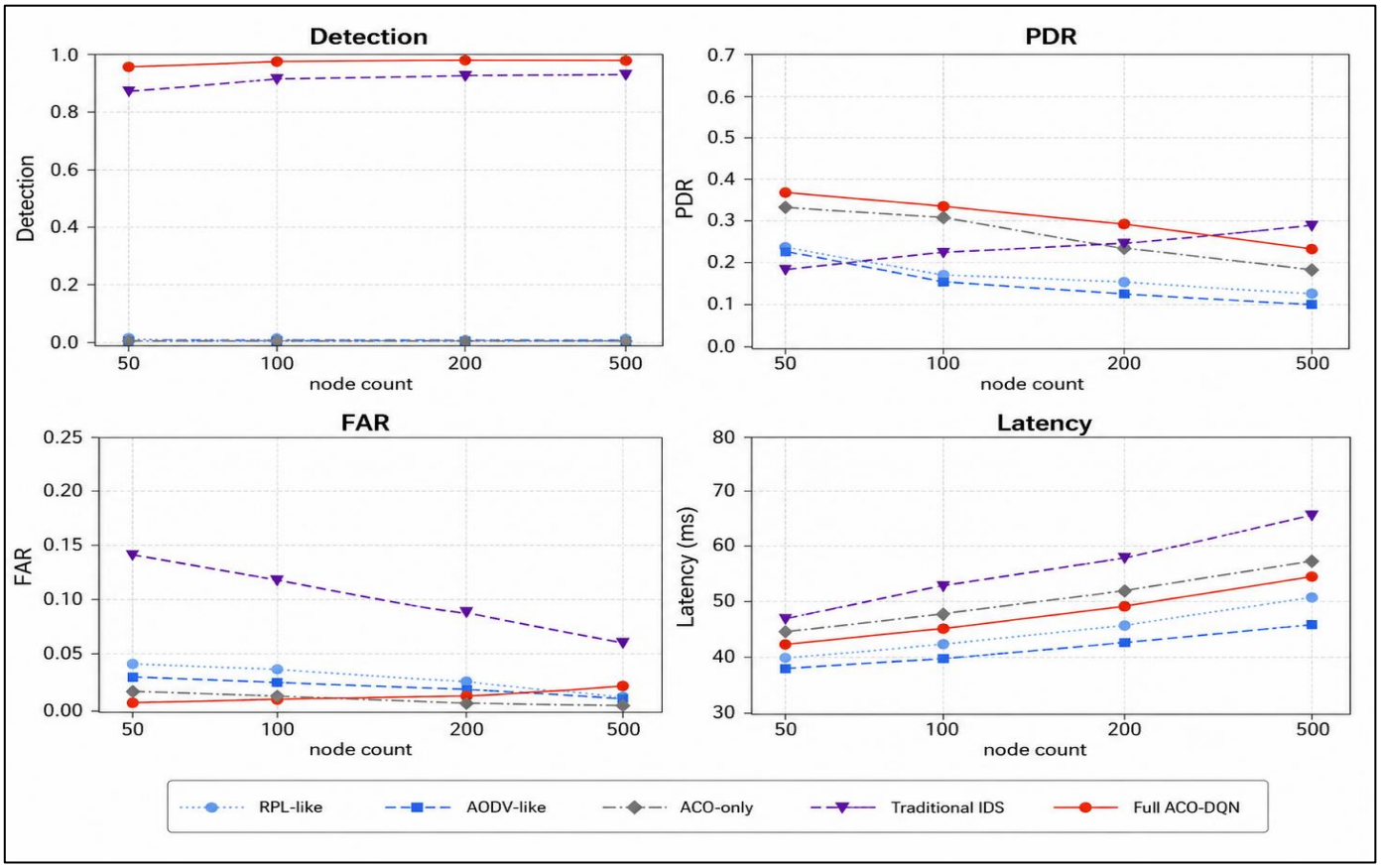



Fig 17 Scalability Results

➢ *Sensitivity Analysis*

The sensitivity study checked if the core results relied on a single security choice. When the cutoffs and the picked response values changed, the detection score, the false actions, and the routing path also changed. This points to the need to tune the system. It does not work well with one fixed set of values.

In general, stronger settings can speed up early action. At the same time, they may wrongly penalize normal nodes. Looser settings can help keep normal routes stable. But they may let an attack sway traffic for more time. The parameters used here were meant to fit the test cases. They were not claimed as a final setup for every IoT network.

For real use, you should set the parameters based on what you expect in your setup. That includes the device type, the traffic behavior, and what a failure would cost. A health-monitoring setup might choose a

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

different point than a home sensor setup. The cost of missing data and the cost of false isolation are not the same.

In the study, the test also shifted the security weight and other selected settings. When the security weight was set very low, the system reaction to harmful evidence dropped. When it was set very high, the system took more cautious steps. That could also limit routing options. So the chosen weight is a compromise for the simulator runs. It should not be treated as a single best value for all cases.

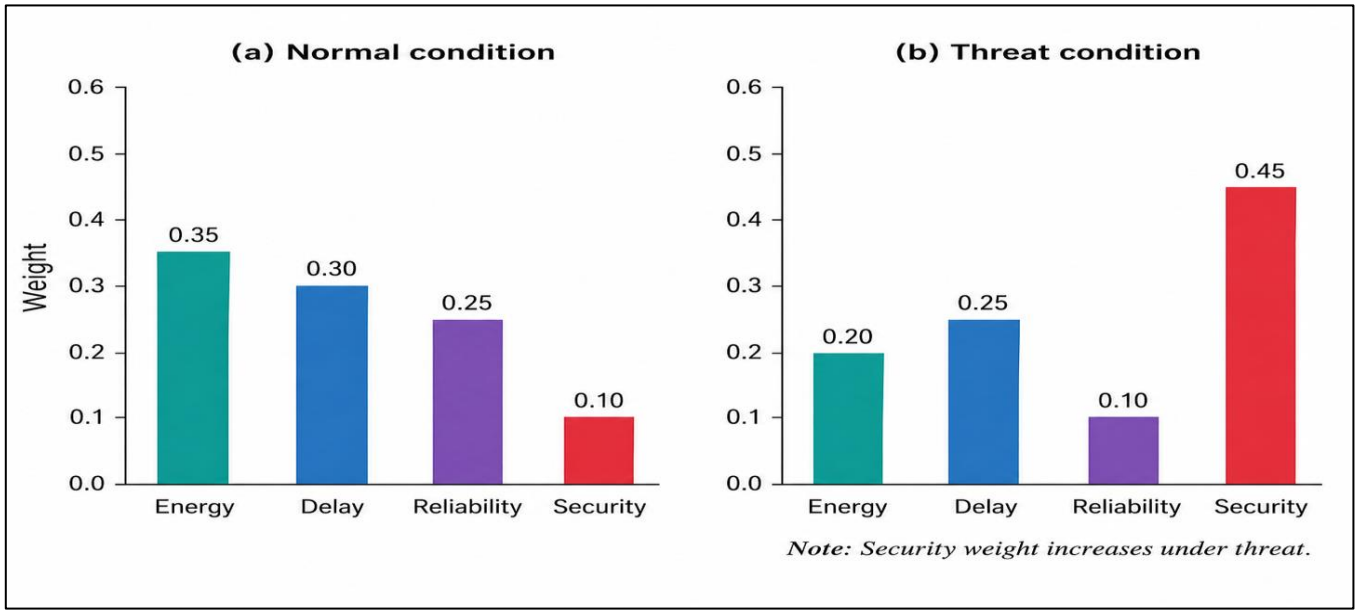



Fig 18 Sensitivity of Selected Security Settings

➢ *Model Quantisation*

Dynamic INT8 quantisation reduced the stored model size by about 60%. This supports a smaller deployment package and may reduce memory requirements. The quantised model remained small because the DQN architecture itself contained only two hidden layers.

INT8 inference was slower than FP32 in the Colab CPU measurement. Quantisation does not guarantee faster execution on every processor. Speed depends on hardware support, runtime libraries, batch size and model structure. The result therefore supports model-size reduction but not a general claim of faster inference.

The benchmark also measured the model operation rather than complete gateway memory and energy. Future work should include model loading, feature preparation, message transfer and repeated inference on the intended hardware.

Table 14 Quantisation Results for Model Size and CPU Inference Latency

| Model | File size (MB) | CPU inference latency (ms) |
|---|---|---|
| FP32 | 0.0323 | 0.071 |
| Dynamic INT8 | 0.0129 | 0.492 |

Dynamic INT8 quantisation reduced the model file size by about 60%. The quantised model was slower in the Colab CPU test, although both measurements were below one millisecond. These values show preliminary computational feasibility only. They do not confirm performance on a physical IoT gateway, and model file size is not the same as total runtime memory use.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

➢ *Transfer Projection*

The original transfer section was based on assumed convergence distributions rather than measured fine-tuning runs. It is therefore treated only as an illustrative projection and not as empirical evidence. No claim of demonstrated transfer learning is made in this thesis.

➢ *Answers to the Research Questions*

- *Answer to Research Question One*

The DQN evidence-gated controller used an offline-trained DQN to propose one of five graded actions from an 11-value live node-state vector. The evidence gate then checked the proposed action against forwarding and route-advertisement evidence before the response was allowed to affect routing. The validated response ranged from monitoring and watching to pheromone reduction, isolation and quarantine. During the main evaluation, the DQN weights remained fixed while live routing observations changed.

- *Answer to Research Question Two*

Validated security evidence was integrated directly into ACO route selection through pheromone control. The framework used the validated action to reduce, reinforce or remove pheromone influence for a node, while energy, delay and reliability remained part of the route heuristic. This allowed unsafe nodes to lose route preference before later routes were constructed and made security an operational input to route selection rather than a separate alert.

- *Answer to Research Question Three*

Under the tested attack conditions, the full framework achieved higher attack-step detection and packet delivery than the selected traditional-IDS baseline, with a lower hard false-action rate. The soft false-action result showed that cautious penalties were more common than hard isolation. The results support the framework within the purpose-built simulator and do not prove performance on every IoT network.

Research Question 1 asked how a DQN evidence-gated controller could be designed for selective forwarding detection in ACO. The study showed that a fixed DQN can use live node-state observations to propose graded security actions and that an evidence gate can validate those actions using current forwarding and route-advertisement evidence before routing is affected.

Research Question 2 asked how validated security evidence could be integrated directly into ACO routing. The study showed that the validated action can modulate pheromone reinforcement and evaporation, reduce the route preference of suspicious nodes and support isolation when the evidence is strong enough, thereby influencing later route choice.

Research Question 3 asked how the complete framework compared with the baselines. Under the tested simulator conditions, the full framework achieved higher attack-step detection and packet delivery, and a lower hard false-action rate, than the traditional

IDS. The soft false-action rate remained notable and shows the cost of cautious security actions.

➢ *Discussions in Relation to Existing Studies*

The findings agree with earlier work that shows the value of bringing security information closer to routing. Sharmin et al. (2023) put trust into the ACO pheromone. They saw better paths when attackers were present. In this work, we follow that broad theme. Still, we do not rely on a fixed trust value. Instead, we use a DQN with graded actions. We also add an evidence gate.

Our findings match earlier work on reinforcement learning for routing. Kandhoul and Dhurandher (2022) reported secure opportunistic routing using deep Q-learning. Wang and colleagues (2024) used scalable DRL

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

to improve RPL behavior. These studies show that learning can affect forwarding decisions. The present work adds a different route mechanism by translating the validated action into an ACO pheromone change.

Dubey et al. (2023) showed that reinforcement learning and ACO can improve IoT path selection. Their main focus was routing performance and energy. The current findings extend that direction by placing routing-layer security at the centre of the action. This comparison is important because the contribution is not the simple combination of ACO and learning, but the security-specific way in which the learning output changes pheromone.

Liu et al. (2023) remains the closest architectural comparison because learned security information entered ACO pheromone in a satellite network. The current study differs in the use of a DQN action space, an evidence gate and low-power IoT scenarios. The similar direction of the findings supports the wider argument that learned security evidence can improve ant-based route selection when it is used before later routes are chosen.

The hard and soft false-action results add a point that is less visible in many earlier studies. A security response has different costs depending on whether it only reduces route preference or fully isolates a node. Reporting both values showed that the framework was cautious more often than it was severe. This gives a fuller view than one false-positive measure.

The differences from earlier work must also be kept clear. The DQN did not learn online during the main experiments, the baselines were simplified, and the results came from a purpose-built simulator. The study therefore provides evidence for the implemented mechanism under controlled conditions. It does not replace standard-simulator or physical-device validation.

➢ *Chapter Summary*

The results back the core design goal. When security data updates pheromone levels and route choice, ACO routing improves. In the full framework, performance was higher than the baseline methods we compared it to. This showed up most clearly in the main security and delivery scores. There are limits to what we can claim. Our simulator is simplified. The DQN was trained offline with synthetic data. The evidence gate also constrains the setup. Metrics are based on our own definitions. We did not test on real hardware.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# CHAPTER FIVE
# SUMMARY, CONCLUSION AND RECOMMENDATIONS

➢ *Introduction*

This section finishes the report. It also highlights the main takeaway. Next, it states the key result. After that, it notes what the study does not fully address. It closes with suggestions for topics to look at in the future.

➢ *Summary of the Study*

The work focused on selective forwarding. In that case, a bad node may throw away some packets. Even so, the path can still look fine. The method put together ABRF with ACO routing. It also used a steady DQN security module. An evidence gate sat between the observations and the routing updates. The DQN set up the decision steps. It used the live node state as input. Then the evidence gate looked at the evidence at that moment. After that, it either approved the decision or did not. When the action was accepted, the system updated the pheromone values. It also shifted the route choice based on that update.

Testing covered three IoT scenarios, with selective forwarding as the main attack. Four additional attacks served as robustness tests. Evaluation used 2,250 attack runs, 450 clean controls and four simplified baselines for comparison.

➢ *Main Findings*

- *Security Finding*

The main security finding is that graded, evidence-checked actions improved attack response under the tested conditions. The framework detected malicious presence more often than the traditional IDS and applied fewer hard actions to benign nodes. The soft false-action value also showed that the system often used cautious route penalties instead of immediate isolation.

- *Routing Finding*

The main routing finding is that security information can improve packet delivery when it directly changes pheromone. The framework reduced the continued use of suspicious routes while keeping energy, delay and reliability in the route heuristic. The improvement came from the whole implemented process rather than from detection alone.

- *Deployment Finding*

The model was small and quantisation reduced its stored size. This supports further gateway-level testing. The Colab CPU result does not confirm deployment on a sensor node or a specific IoT gateway, and the quantised model was not faster in the measured environment.

First, the full framework achieved a mean attack-step detection rate of 0.847, compared with 0.681 for the traditional IDS. Second, its hard false-action rate was 0.033, compared with 0.126 for the traditional IDS. Third, it achieved a packet delivery ratio of 0.665, compared with 0.579 for the traditional IDS. The soft false-action rate was 0.312, which shows that cautious pheromone penalties were applied more often than hard isolation.

The findings also showed that the security action affected route preference through pheromone. This supports the central argument that security and routing should share the same route-quality decision process. Quantisation reduced the model file size, but the hardware evidence was limited to a Colab CPU benchmark.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

➢ *Contribution of the Study*

The first contribution is architectural. The study lays out a straightforward path from watching the system live to choosing an action with a DQN. It also describes how the evidence gets checked, how pheromone levels shift, and how the route is set later. Put together, this flow suggests that security can be built into ACO routing as a routine control, not as a separate kind of warning.

The second contribution is evaluative. The study reports attack-step detection, hard false actions, soft false actions and routing outcomes together. This makes the security cost visible and prevents a low isolation rate from hiding frequent soft penalties.

The third contribution is methodological honesty. The thesis separates offline DQN training from live simulation use, labels the baselines as simplified implementations and limits deployment conclusions to preliminary computational evidence. These limits make the findings easier to reproduce and defend.

The main contribution is a tested security-enabled ACO routing design in which a DQN-supported, evidence-gated action changes pheromone before later route choices are made. The study also provides a common simulation environment for comparing routing performance and security outcomes together.

The contribution is narrower than claiming that ACO and DRL have never been combined. Earlier studies have combined learning and ACO in other routing settings. The contribution of this work is the particular security-control path tested in the IoT simulator: live routing observations, a fixed DQN proposal, evidence validation, a graded response and direct pheromone influence.

➢ *Conclusion*

The study concludes that DQN-supported, evidence-gated routing can mitigate selective forwarding by changing ACO pheromone and route preference. Under the tested conditions, the framework improved attack-step detection and packet delivery and reduced hard false actions compared with the traditional IDS baseline.

The conclusion is limited to the simulator and experiment settings used in this research. The results do not prove standards-compliant RPL or AODV performance, end-to-end online reinforcement learning, transfer learning, or deployment readiness on physical IoT hardware.

➢ *Limitations*

- *Simulator and Baseline Limitations*

The purpose-built simulator represented the main network behaviours needed for the study, but it did not reproduce every radio, operating-system and protocol detail. The RPL-like and AODV-like baselines were simplified. Their results cannot be used as direct performance claims about full standard implementations.

- *Learning Limitations*

The DQN was trained using made up state transitions and rewards set by the researcher. During this setup, it did not pick up the real impact of each action on packet delivery, delay, and energy within the simulator. Also, the policy did not change while evaluation was running. For later work, the model should be trained on transitions created by the simulator and then offline and online learning should be compared.

- *Measurement Limitations*

A check for the attack step noted if it found at least one bad node in a control step. It did not track how many of all bad nodes were caught. The soft and hard false-action results helped the reading. Still, it would be better to report node-level precision and recall, plus the F1 score. Action-cost results would also make the evaluation more complete.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

- *Generalisation Limitations*

The scenarios and attacks were controlled abstractions. Different traffic, radio conditions, device faults or attacker strategies may change the results. The scalability test also changed density as node count increased. The conclusions should therefore remain within the tested settings.

- ✓ The simulator used simplified RPL-like and AODV-like behaviours rather than full protocol implementations.
- ✓ The DQN was trained on synthetic state transitions and predefined rewards, not on transition data produced by live simulator interaction.
- ✓ The evidence gate could strengthen or reduce the DQN action. The measured performance therefore belongs to the combined DQN-and-gate decision process.
- ✓ Attack-step detection measured whether at least one malicious node was detected at an attack control step. It did not measure node-level recall.
- ✓ The hard and soft false-action measures used different action thresholds. Both must be reported together.
- ✓ Detection latency was produced in simulation steps. A real-time value requires a defined duration for each step.
- ✓ The ablation did not fully separate ACO, DQN and evidence-gate effects.
- ✓ The scalability experiment also changed network density.
- ✓ The transfer section was illustrative and was not based on measured fine-tuning runs.
- ✓ The quantisation test used a Colab CPU and did not measure total gateway memory, energy use or real communication overhead.

➢ *Recommendations*

- *Recommendations for Model Development*

Future work should collect state transitions directly from the simulator or a testbed and define a reward that includes packet delivery, false isolation, delay and energy. This would connect learning more closely to the actual network outcome. The DQN-only, gate-only and full combinations should also be compared in a complete ablation.

- *Recommendations for Network Evaluation*

The framework should be tested in a standard network simulator or physical testbed with full RPL and AODV implementations. Constant-density scalability, recovery after a false penalty, mixed benign faults and longer attacks should be included. Node-level security measures should be added to the current attack-step results.

- *Recommendations for Deployment*

The quantised policy should be tested on the intended gateway hardware. The evaluation should measure total memory, feature-processing time, inference time, energy use and communication delay. A practical deployment should also define how security scores are distributed to nodes and how a quarantined node can be restored safely.

- ✓ Train the DQN on real simulator transitions using rewards based on packet delivery, malicious-route avoidance, false isolation, delay and energy.
- ✓ Run a full ablation with ACO-only, gate-only plus ACO, DQN-only plus ACO, DQN plus gate plus ACO, and the traditional IDS baseline.
- ✓ Provide precision, recall, and F1 at the node level, not only for attack-step detection.
- ✓ Set the simulation step time first, then report detection latency in seconds.
- ✓ Run tests for both fixed-area densification and constant-density scaling.
- ✓ Use real transfer tests, with recorded pretraining time and fine-tuning time.
- ✓ Check results on a standards-based network simulator or on a physical IoT test setup.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

✓ On the gateway hardware, track total memory, CPU load, power use, and communication overhead.
✓ Share the last code, configs, software setup, random seeds, and output hashes so others can rerun it.

➢ *Final Statement*

This thesis argues that secure routing can be viewed as one linked decision problem. It treats routing and detection as parts of the same process. The outcome looks good. Still, the best part is the direct, measurable connection it makes between security evidence and the ACO pheromone signal. Next, the work should check this connection using online training. It should also run stronger ablation tests. It should include evidence from real physical network setups.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

# REFERENCES


[1.] Alfriehat, N., Anbar, M., Aladaileh, M., Hasbullah, I. H., Shurbaji, T. A., Karuppayah, S. and Almomani, A. (2024). RPL-based attack detection approaches in IoT networks: Review and taxonomy. Artificial Intelligence Review, 57, 248.

[2.] Alansari, Z., Anuar, N. B., Kamsin, A. and Belgaum, M. R. (2023). RPLAD3: anomaly detection of blackhole, grayhole, and selective forwarding attacks in wireless sensor network-based Internet of Things. PeerJ Computer Science, 9, e1309.

[3.] Bala, K., Venu, D., Dudi, B. P., Nirmala, V., Ashok Kumar, K. and Manohara, M. (2026). Federated deep reinforcement Q-learning for secure and energy-efficient data routing in Internet of Things networks. Peer-to-Peer Networking and Applications, 19, 88.

[4.] Cevallos Moreno, J. F., Rizzardi, A., Sicari, S. and Coen-Porisini, A. (2023). Deep reinforcement learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges. Computer Networks, 236, 110016.

[5.] Dorigo, M. and Stützle, T. (2004). Ant Colony Optimization. Cambridge, MA: MIT Press.

[6.] Dubey, G. P., Stalin, S., Alqahtani, O., Alasiry, A., Sharma, M., Aleryani, A., Shukla, P. K. and Alouane, M. T.-H. (2023). Optimal path selection using reinforcement learning based ant colony optimization algorithm in IoT-based wireless sensor networks with 5G technology. Computer Communications, 212, 377-389.

[7.] Fu, T., Hao, S., Chen, Q., Yan, Z., Liu, H. and Rezaeipanah, A. (2024). An energy-aware secure routing scheme in Internet of Things networks via two-way trust evaluation. Pervasive and Mobile Computing, 105, 101995.

[8.] Han, H., Tang, J. and Jing, Z. (2024). Wireless sensor network routing optimisation based on improved ant colony algorithm in the Internet of Things. Heliyon, 10(1), e23577.

[9.] Jamshidi, S., Nikanjam, A., Wazed Nafi, K., Khomh, F. and Rasta, R. (2025). Application of deep reinforcement learning for intrusion detection in Internet of Things: A systematic review. Internet of Things, 31, 101531.

[10.] Kandhoul, N. and Dhurandher, S. K. (2022). Deep Q learning based secure routing approach for OppIoT networks. Internet of Things, 20, 100597.

[11.] Liu, Z., Rong, J., Jiang, Y. and Zhang, L. (2023). Satellite network security routing technology based on deep learning and trust management. Sensors, 23(20), 8474.

[12.] Mahjoub, C., Hamdi, M., Alkanhel, R. I., Mohamed, S. and Ejbali, R. (2024). An adversarial environment reinforcement learning-driven intrusion detection algorithm for Internet of Things. EURASIP Journal on Wireless Communications and Networking, 2024, 21.

[13.] Mnih, V., Kavukcuoglu, K., Silver, D. et al. (2015). Human-level control through deep reinforcement learning. Nature, 518, 529-533.

[14.] Omar, A. A. R. A., Soudan, B. and Altaweel, A. (2024). UOS_IOTSH_2024: A comprehensive network traffic dataset for sinkhole attacks in diverse RPL IoT networks. Data in Brief, 55, 110650.

[15.] Perkins, C., Belding-Royer, E. and Das, S. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561.

[16.] Pidy Pidy, C., Moskolaï Ngossaha, J., Tchappi, I. and Bowong Tsakou, J. (2026). CARAC-MDRL: Cluster-based ant-colony routing with adaptive cluster head and multi-agent deep reinforcement learning. Procedia Computer Science, 280, 163-170.

[17.] Sharmin, A., Olanrewaju, R. F., Khan, B. U. I., Anwar, F., Motakabber, S. M. A., Rosely, N. F. L. M. and Hashim, A. H. A. (2023). Secure IoT routing through manifold criterion trust evaluation using ant colony optimization. International Journal of Advanced Computer Science and Applications, 14(11), 131-143.

[18.] Tawfeek, M. A., Alrashdi, I., Alruwaili, M., Jamel, L., Elhady, G. F. and Elwahsh, H. (2025). Improving energy efficiency and routing reliability in wireless sensor networks using modified ant colony optimization. EURASIP Journal on Wireless Communications and Networking, 2025, 22.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct1627

[19.] Wang, Y., Li, Y., Lei, J. and Shang, F. (2024). Robust and energy-efficient RPL optimization algorithm with scalable deep reinforcement learning for IIoT. Computer Networks, 255, 110894.
[20.] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J. P. and Alexander, R. (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550.
[21.] Zaminkar, M. and Fotohi, R. (2020). SoS-RPL: Securing Internet of Things against sinkhole attack using RPL protocol-based node rating and ranking mechanism. Wireless Personal Communications, 114, 1287-1312.