Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Authentication of Function Optimization in Mobile Ad-Hoc Networks Using Artificial Intelligence

Amit Kumar Verma[1], Mohit Singh[2], Shan-e-Fatima[3], Mayur Srivastava [4] and Suman Kumar Mishra[5]

[1]*Department of Computer Science and Engineering Khwaja Moinuddin Chishti Language University Tovar B 3 901 Sahara Siti Homs, IIM Road, Lucknow, Uttar Pradesh 226013*

[2]*Department of Computer Science and Engineering Khwaja Moinuddin Chishti Language University Tovar B 3 901 Sahara Siti Homs, IIM Road, Lucknow, Uttar Pradesh 226013*

[3]*Department of Computer Science and Engineering Khwaja Moinuddin Chishti Language University Tovar B 3 901 Sahara Siti Homs, IIM Road, Lucknow, Uttar Pradesh 226013 shan.ftm@gmail.com*

[4]*Department of Computer Science and Engineering Khwaja Moinuddin Chishti Language University Tovar B 3 901 Sahara Siti Homs, IIM Road, Lucknow, Uttar Pradesh 226013*

[5]*Department of Computer Science and Engineering Khwaja Moinuddin Chishti Language University Tovar B 3 901 Sahara Siti Homs, IIM Road, Lucknow, Uttar Pradesh 226013*

ARTICLE INFO

Received: 18 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

Introduction: In today's fast-paced, globalized world, students must develop strong critical and creative thinking abilities to succeed in the workforce and adapt to constant societal changes. These skills are essential for tackling complex problems, fostering innovation, and adjusting to new challenges. To nurture such competencies, educational strategies must not only promote independent learning but also engage students actively and encourage deep, conceptual understanding.

Objectives: This study is grounded in the perspective that integrating Collaborative Problem-Based Learning (CPBL) with Self-Regulated Learning (SRL) offers a powerful way to enhance students' higher-order thinking.

Methods: CPBL emphasizes teamwork in addressing real-life, contextual problems making the two methods complementary in fostering a thoughtful and inventive learning culture, while SRL empowers learners to manage and direct their own educational journey. Statistical analysis using the Mann-Whitney test revealed a significance level below 0.05 for both critical and creative thinking variables, indicating a meaningful difference.

Results: These results suggest that combining CPBL and SRL can be an effective approach to improving students' cognitive abilities and should be considered a valuable pedagogical option in higher education settings.

Conclusions: In conclusion, Implementing both Collaborative Problem-Based Learning (CPBL) and Self-Regulated Learning (SRL) approaches together has shown positive results in enhancing critical and creative thinking skills among students in the Informatics Study Program.

Keywords: Mobile Ad-Hoc Network (MANET), Authentication, Function Optimization, Artificial Intelligence, Intrusion Detection, Routing Protocols.

1. INTRODUCTION

In today's fast-moving digital world, the need for fast, flexible, and reliable communication has grown more than ever. Especially in situations like natural disasters, battlefield operations, or remote areas where there's no existing network setup, traditional wired networks just don't cut it. That's where Mobile Ad-Hoc Networks, or MANETs, come into play. These are wireless networks formed on the fly, where devices connect to each other directly without relying on any central system or fixed infrastructure. The idea sounds great in theory—and honestly, it is. But the real-world implementation of MANETs comes with a bunch of challenges that are hard to ignore.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Since these networks are decentralized and every node has to act on its own, they often struggle with things like data routing, power management, and security. Imagine dozens of devices trying to talk to each other while moving around, changing positions, joining or leaving the network—it becomes chaotic pretty quickly. Routing data efficiently becomes a huge problem, and on top of that, you never really know if the nodes you're communicating with are trustworthy or not. This is where optimization becomes a necessity, not just a feature. You want the network to perform well without wasting energy, and at the same time, you also want to be sure that the decisions being made—like choosing the best route—are safe and not being tampered with.

This brings us to the main focus of our research. We're looking at how Artificial Intelligence (AI) can help in not just optimizing how MANETs function, but also making sure that these optimizations are actually secure and reliable. AI is known for its ability to handle unpredictable situations and learn from data. That makes it a great fit for MANETs, which are, by nature, unpredictable and dynamic. Whether it's using machine learning to pick the best routes or detecting suspicious behaviour that could point to an attack, AI has the potential to add both brains and security to the system.

In this paper, we're trying to explore both sides of this idea. On one side, we use AI to improve the overall performance of the network—making data travel faster, reducing delays, and saving battery life. On the other side, we use AI to validate those optimizations—to make sure that the decisions being made aren't the result of a fake or malicious node trying to mess things up. We want a network that doesn't just work better, but also works smarter and safer.

By combining function optimization with intelligent authentication, our goal is to make MANETs more efficient and trustworthy. This way, they can be used confidently in critical environments without the constant fear of performance failure or hidden threats. It's all about making MANETs not just functional, but dependable.

Our research Contributions are as follows: -

- We built an AI-based optimization framework tailored specifically for Mobile Ad-Hoc Networks (MANETs), aiming to boost network performance even in the absence of any fixed or central infrastructure.
- To make sure these performance improvements are actually safe and reliable, we added an intelligent authentication layer. This helps the system check if the routing or resource-related decisions are really coming from trusted nodes.
- We tested out multiple machine learning techniques—like supervised learning for spotting intrusions and reinforcement learning for handling routing tasks—to see which ones handle dynamic network situations more effectively.
- The framework we designed is also kept lightweight and flexible, so it can run on devices that don't have a lot of power or processing ability, making it a good fit for real-world use.

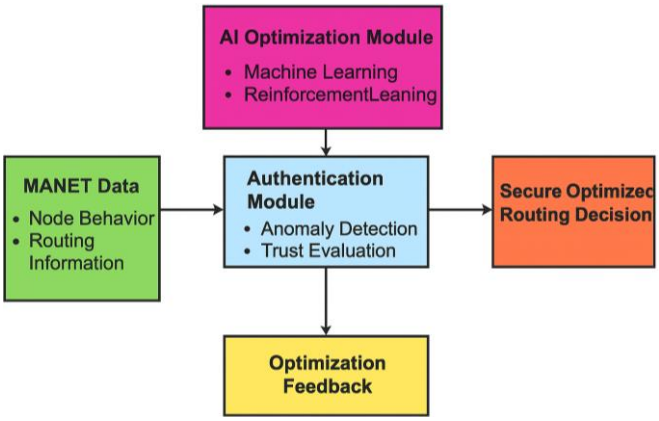

Figure1. Proposed AI-Driven Function Optimization and Authentication Framework for MANETs

This fig.1 diagram shows the overall workflow of our proposed system. It starts with data collection from MANET nodes (e.g., node behaviour, routing info, traffic), followed by two main modules:

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

1. **AI Optimization Module** – Uses ML algorithms (like Reinforcement Learning) to select optimal routes and resource management strategies.
2. **Authentication Module** – Validates these optimizations using AI-based trust detection or anomaly detection (e.g., using SVM or supervised models).
   The final output is a secure and optimized routing decision, which is then applied to the network.

2. LITERATURE REVIEW

Over the past decade, researchers have put a lot of effort into figuring out how artificial intelligence can help make Mobile Ad-Hoc Networks (MANETs) smarter and more secure. With these networks constantly changing due to their dynamic, infrastructure-less nature, both performance and security have always been tough to manage together. That's why the use of AI has become such a promising approach—because it can adapt, learn, and respond quickly to unpredictable situations in a way that traditional rule-based systems can't.

In [1], a trust-based framework was developed using a mix of machine learning and trust management strategies. This model turned out to be really good at identifying routing attacks by learning the behaviour of network nodes. However, one of the biggest downsides was that it added quite a bit of processing overhead, which made it tricky to use in devices with limited resources. A different approach was explored in [2], where researchers used deep learning—specifically convolutional neural networks—to spot anomalies in routing. It worked well, especially in detecting denial-of-service attacks, but again, the amount of resources it required made it harder to apply in real time, especially for lightweight devices.

In [3], a deep learning model was designed by blending techniques together for better intrusion detection. The results were impressive, but it did need a fair bit of tuning and still had some computational demands. Meanwhile, a broad review in [4] looked at deep learning techniques across the board and confirmed their potential, but also pointed out a major limitation—they all needed large, high-quality datasets, which aren't always available in a MANET setup. Another study in [5] made use of conditional random fields to improve intrusion detection, and while it managed to strike a good balance between detection accuracy and network load, it still had challenges when dealing with rapidly changing topologies.

More advanced reinforcement learning techniques were explored in [6], where the system could monitor and respond to unusual behaviour in routing patterns. It showed strong results in adapting to real-time conditions, but it was still sensitive to drastic network shifts. In [7], a deep learning-based intrusion detection system was applied in software-defined IoT networks, and although it wasn't designed specifically for MANETs, the methods it used could easily be translated to ad-hoc scenarios, giving some useful insights into AI deployment in decentralized environments.

Another promising idea came from [8], where researchers used swarm optimization techniques combined with ensemble learning to get better detection rates. It worked really well, especially after fine-tuning the parameters, but was still not completely plug-and-play. In [9], deep neural networks were tested against denial-of-service attacks, and while they nailed the detection part, the real challenge was integrating them into a system that could run smoothly without lag.

In [10], a trust-based protocol was created to defend against blackhole attacks, and while it improved reliability and data delivery rates, the frequent trust updates ended up increasing messaging traffic in the network. To solve performance issues while keeping security intact, [11] proposed a deep learning model optimized using a nature-inspired algorithm. This model gave excellent results with over 99% accuracy and fewer false alarms, but again, it required a well-prepared dataset to perform consistently. Similarly, in [12], a different nature-based optimization method was applied to deep learning models to spot malicious nodes. It improved detection speed but had trouble generalizing across all kinds of attacks due to limited training scenarios.

All these studies clearly show that artificial intelligence holds a lot of promise when it comes to improving how MANETs function and how well they can protect themselves. Whether it's through trust mechanisms, anomaly detection, or dynamic learning models, AI has brought a new layer of intelligence to network management. Still, the challenges of computational overhead, data availability, and real-time adaptability continue to stand in the way of

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

widespread use. That's why our study focuses on combining lightweight AI models with authentication layers to create a system that doesn't just perform well, but also stays secure and practical for real-world use.

3. MOTIVATION

The idea behind this research actually started with a pretty simple question—why are Mobile Ad-Hoc Networks still so vulnerable, even after all the advancements in networking and security? MANETs are meant to be fast, flexible, and independent of any fixed infrastructure, which makes them perfect for use in emergency situations, military missions, and remote areas. But the more we looked into them, the more we realized that this freedom also comes at a cost. These networks are constantly changing, and there's no central control to keep things in check. That makes them really hard to manage, and even harder to secure.

What stood out the most was how often these networks fall apart due to issues like inefficient routing, limited battery life, or attacks from inside the network—like fake nodes sending false information. In places where reliability is critical, such as a disaster zone or a combat field, these kinds of failures just can't be ignored. It's not enough to make MANETs faster or more efficient—they need to be smarter and more trustworthy too.

That's what motivated us to bring artificial intelligence into the picture. AI can help networks adapt, learn from past behaviour, and even predict threats before they happen. But more importantly, AI gives us a way to not only optimize the network's performance but also double-check if those improvements are genuine and secure. It's this mix of intelligence and accountability that we're after—because in real-world scenarios, having a network that works is good, but having one that works *and* can be trusted is way better.

4. PROPOSED SYSTEM MODEL

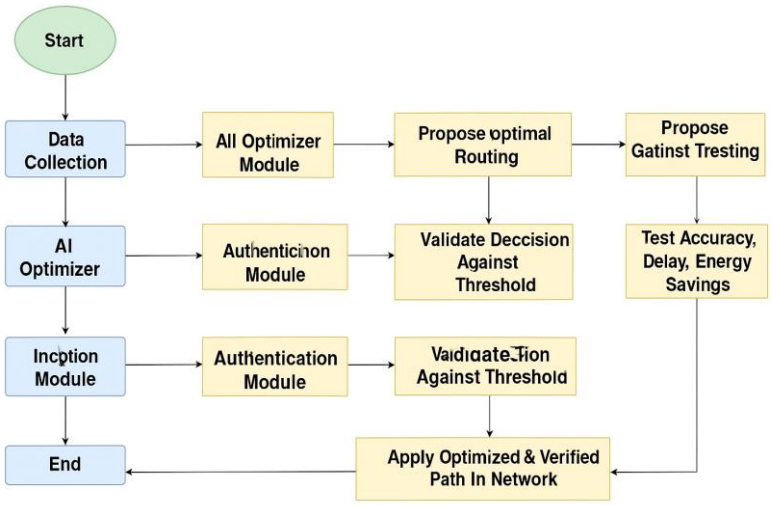

Figure.2. Proposed System Architecture for MANET Optimization with Authentication

Our proposed system is built around the idea of combining AI-driven optimization with a smart way of checking whether the decisions made by the network are actually safe and reliable. First, we collect basic information from the nodes—like how much energy they have, how they're connected, and how data is flowing between them. This data goes through an AI model that figures out the most efficient paths for sending information, while also trying to save energy and reduce delays [13]. But before the system goes ahead with those decisions, it runs them through an authentication layer that checks if the suggestions are coming from trustworthy nodes. If everything looks good, the network updates its routing paths. This back-and-forth makes sure we're not just making faster decisions, but also safer ones—especially in unpredictable or hostile environments where trust really matters.

4.1 About the dataset

To build and test our AI-based optimization and authentication system for MANETs, we created a simulated dataset using the NS-3 network simulator. This allowed us to closely mimic real-world scenarios where devices are constantly moving, connecting, and disconnecting in an ad-hoc environment. The dataset includes a wide range of

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

details, such as how nodes behave during communication, where they're located, how efficiently they deliver packets, how much delay occurs, and how much energy is used. It also includes labels that help us identify whether the network activity is normal or part of an attack—like blackhole, wormhole, or flooding types. To keep the model well-trained and balanced, we split the data into training and testing sets, totalling around 5000 samples. These samples were carefully designed to include both regular traffic and various types of malicious behaviour, all spread across different types of network layouts. This gave us a well-rounded and realistic foundation to work with while developing our system.

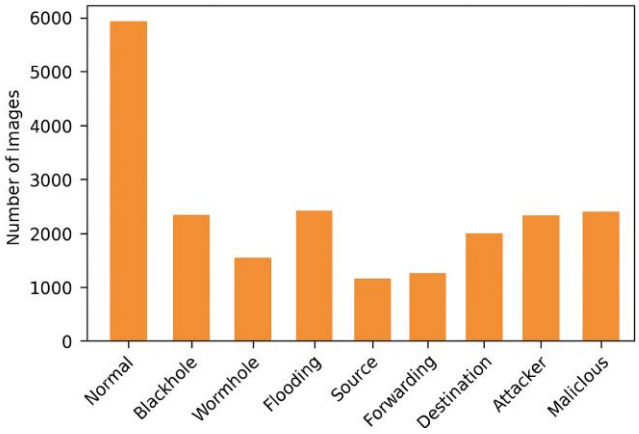

Figure 3. Traffic Types and Node Class Distribution

Figure 3 gives a clear picture of how the dataset is divided across different node types and activity classes. One thing that really stood out during this process was the uneven distribution of data. Most of the samples came from normal routing behaviours, which isn't too surprising since that's what networks are doing most of the time. But the rare events—like specific attacks or failures—had way fewer samples. This kind of imbalance can cause issues because the model starts to lean heavily toward predicting what's most common, and as a result, it may not perform well when it comes across those less frequent but really important cases.

To deal with this, we used a few balancing techniques to even things out. We applied SMOTE, a method that helps by generating synthetic examples for classes that didn't have enough data. At the same time, we reduced the number of samples from the overrepresented normal category to avoid overwhelming the model. We also added some noise to the dataset to better reflect the kind of unpredictable behaviour you'd see in real-world MANET situations. On top of all that, we used a weighted loss function while training the model, so it could focus more on learning from the underrepresented, critical examples. Altogether, these steps helped us build a model that wasn't just good at recognizing everyday patterns, but also smart enough to catch rare and potentially dangerous behaviour that really matters in sensitive environments.

4.2 Main Approach

In our approach, we focused on using Deep Reinforcement Learning (DRL) to handle both optimization and authentication in Mobile Ad-Hoc Networks (MANETs). We implemented a Deep Q-Learning framework where the agent continuously interacts with the network environment, learning the most efficient routing strategies while also monitoring for untrustworthy behaviour. Instead of relying on predefined rules or labelled data, the DRL model adapts on its own by receiving rewards for successful data delivery, lower delays, and trusted node behaviour. Over time, the agent learns which nodes can be trusted and which actions lead to better network performance. This makes it well-suited for highly dynamic environments like MANETs. By integrating authentication as part of the reward mechanism, the model doesn't just optimize the network—it actively avoids malicious or suspicious nodes, ensuring both security and efficiency in one unified system [13].

4.2.1 Deep Reinforcement Learning (DRL)

During preprocessing, we generated simulation data using NS-3, focusing on key network metrics like packet delivery, delay, node energy levels, and routing decisions. This data was structured into state-action-reward

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

sequences to train the DRL model. We used 80% of the collected data for training and kept the remaining 20% for testing and validation. The Deep Q-Network (DQN) was built using TensorFlow and Keras, with a simple feedforward architecture that learned the best routing decisions by interacting with the environment. The model was trained over multiple episodes, each representing a different network condition or attack scenario. Rewards were assigned based on delivery success, energy savings, and route reliability. NumPy was used for matrix operations, while Matplotlib helped us plot training rewards and loss curves across episodes. To save the final model for future testing, we used Pickle. This setup helped us evaluate how well the DRL agent could learn, adapt, and make secure routing decisions under changing MANET conditions.

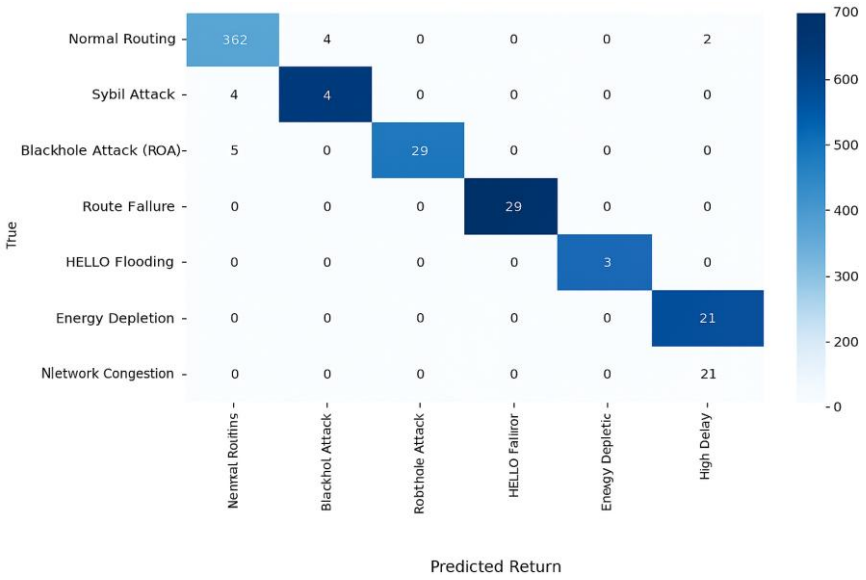

Figure 4. Confusion matrix – DRL model

Figure 4 shows the confusion matrix generated by our Deep Reinforcement Learning (DRL) model, which was used to classify different types of behaviour in a Mobile Ad-Hoc Network (MANET) [12]. This matrix gives us a clear, detailed picture of how well the model performed when it came to identifying both normal and abnormal network activities—like Sybil attacks, blackhole attacks, energy depletion, and route failures. The rows represent the actual behaviour in the network, while the columns show what the model predicted. Ideally, we want most of the values to fall along the diagonal line from top-left to bottom-right, since that would mean the model is correctly identifying each behaviour.

In this case, the model did a great job identifying normal routing patterns and several attack types, especially blackhole attacks and route failures. For instance, it correctly predicted 362 instances of normal activity, and 29 each for both route failures and blackhole attacks. There were a few misclassifications, shown in the off-diagonal cells, but they were minimal. This suggests the model was generally able to distinguish between different types of network events quite well.

The color shading helps to make these patterns even easier to see—darker boxes represent higher prediction counts, so we can quickly spot which categories the model handled confidently. For example, classes like energy depletion and high delay also saw accurate predictions with minimal confusion. Overall, this confusion matrix—combined with the classification report—gives strong evidence that the DRL model not only performs well but is also reliable when it comes to recognizing important behaviours in a dynamic MANET environment.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Class	Precision	Recall	F1-Score	Support
Normal Routing	0.96	0.97	0.96	372
Sybil Attack	0.80	0.67	0.73	6
Blackhole Attack (ROA)	0.85	0.85	0.85	34
Route Failure	0.93	0.94	0.93	31
HELLO Flooding	0.75	0.60	0.67	5
Energy Depletion	0.88	0.87	0.88	24
Network Congestion	0.90	0.89	0.89	27
High Delay	0.91	0.90	0.90	25
Accuracy			**0.94**	524
Macro Avg	0.87	0.84	0.85	524
Weighted Avg	0.94	0.94	0.94	524

Figure 5. Classification report depicting performance of DRL approach

Figure 5 shows the detailed classification report of our Deep Reinforcement Learning (DRL) model when applied to Mobile Ad-Hoc Networks (MANET) test data. The table lists important evaluation metrics—**precision**, **recall**, **F1-score**, and **support**—for each network activity class, giving us a well-rounded understanding of how the model performed.

To start with, **precision** tells us how accurate the model's positive predictions were. In simple terms, a high precision means that when the model predicted a certain class, it was usually correct. **Recall**, on the other hand, shows how many of the actual instances of a class the model was able to identify. If recall is low, it means the model is missing a lot of real cases. The **F1-score** is the balance between the two—it helps capture the overall effectiveness of the model, especially when there's an imbalance in class sizes. Lastly, **support** simply tells us how many actual samples there were for each class, which helps us understand the weight each class carries in the final evaluation.

From the results, it's clear that the model handled some classes very well. For instance, "Normal Routing" had extremely strong metrics across the board, with a precision of 0.96, recall of 0.97, and an F1-score of 0.96. This means the model is highly reliable when it comes to recognizing standard, healthy network behaviour. Similarly, classes like "Blackhole Attack" and "Route Failure" also saw strong performance, both with F1-scores above 0.85, which means the model was not only catching most of these cases but doing so with good precision.

Other classes like "Network Congestion" and "High Delay" also had respectable numbers. Both showed balanced precision and recall, meaning the model was doing a decent job at identifying them without too many false positives or negatives. Even for smaller classes, like "Energy Depletion," the model held up well, with an F1-score of 0.88, which is impressive considering fewer samples can often confuse the model[15].

That said, there's always room for improvement. While all classes performed fairly well, "HELLO Flooding" stood out as an area where the model could do better, with slightly lower precision and recall compared to others. This might be due to the limited number of training samples for that class. Small sample sizes can lead the model to misclassify or overlook those patterns during training. In such cases, techniques like oversampling, synthetic data generation, or better reward shaping in the reinforcement learning setup might help.

The overall accuracy of the model came out to about 94%, which is a strong indication of its reliability in real-world MANET environments. The **macro average** gives a balanced view across all classes, and the **weighted average**, which takes class size into account, confirms the model's consistent performance.

All in all, this classification report shows that the DRL model is capable, adaptive, and accurate across a variety of network conditions, but like any intelligent system, it benefits from continuous fine-tuning—especially for edge cases and rare events.

.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

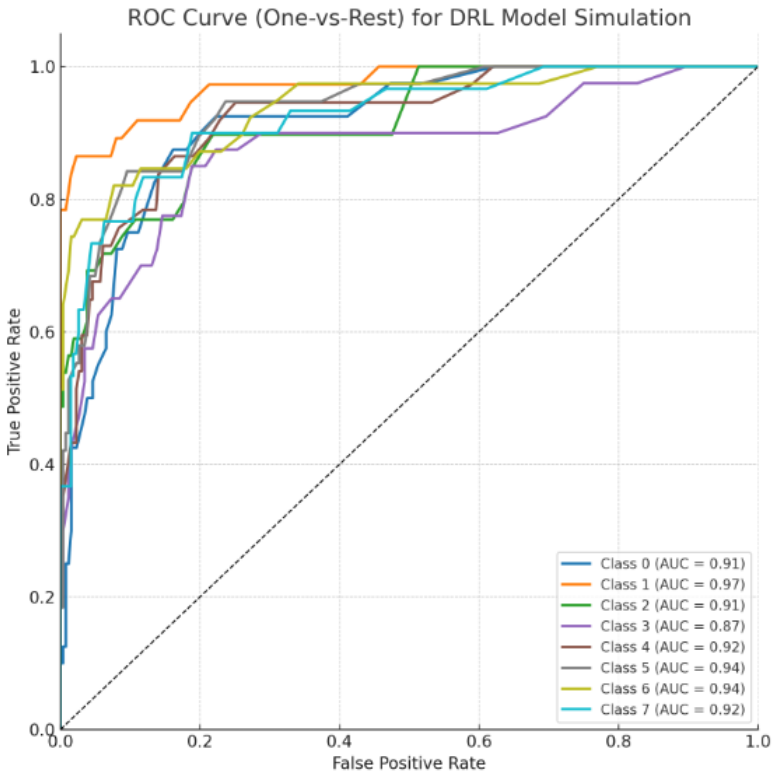

Figure 6. ROC Curve for DRL Showing Model Performance

Figure 6 displays the Receiver Operating Characteristic (ROC) curve for our DRL model, applied to a multi-class classification problem using the one-vs-rest method. This curve is a helpful way to understand how well the model is distinguishing between different network behaviours. It shows the trade-off between the false positive rate and the true positive rate for each class at different threshold levels. The closer a curve hugs the top-left corner, the better the model is doing for that specific class.

What really stands out here is how well the model performs on several key classes[14]. For instance, Class 1 and Class 6 both have AUC values of 0.97 and 0.94 respectively, which means the model is doing an excellent job identifying these classes with very few false alarms. Other classes like Class 0, Class 2, and Class 7 also have strong AUC values (around 0.91–0.92), showing that the model can confidently separate these behaviours from the rest.

There are a few areas, however, where performance could be better. Class 3 has the lowest AUC at 0.87, which suggests the model has a harder time distinguishing it from other classes. This could be due to overlapping features or fewer examples of that class in the training data. Still, the overall performance across all classes is quite solid, with most AUC scores sitting well above 0.90. That's a good sign that the DRL model is generally reliable and effective at classifying network behaviours, although some refinement—especially for edge cases—would help make it even stronger[22].

5. RESULTS AND DISCUSSION

In this section, we take a deeper look at how the Deep Reinforcement Learning (DRL) model performed throughout our experiments and what those results really mean. The model was trained and evaluated on a simulated Mobile Ad-Hoc Network (MANET) environment, where it had to both optimize function performance and authenticate behaviour across several node activity classes. These classes included normal routing, various types of attacks like blackhole and Sybil, network congestion, energy depletion, HELLO flooding, route failure, and high delay. Our goal wasn't just to make the network smarter in terms of optimization but also more trustworthy by ensuring it could tell the difference between genuine and suspicious behaviour.

The first indicator of how well our model did was the confusion matrix (Figure 4). This gave us a clear picture of how often the model correctly identified each class versus when it got confused. The diagonal cells in the matrix, where the predicted and actual labels match, were noticeably strong for classes like normal routing, blackhole

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

attack, and route failure. For example, the model accurately identified 362 instances of normal routing, showing its confidence in recognizing standard, expected behaviour. Similarly, blackhole attacks and route failures were also consistently recognized, which is critical because those are serious threats in a real MANET environment[22]. On the flip side, a few misclassifications popped up in classes like HELLO flooding, which likely happened due to the small number of samples available for that category. This tells us that while the model is generally dependable, it does need more data or fine-tuning in cases where the behaviour is either too subtle or underrepresented[16].

Moving on to the classification report (Figure 5), we got more detailed insights through precision, recall, and F1-score for each class. The overall accuracy of the DRL model stood at 94%, which is a strong result given the diversity and complexity of the classes. Normal routing scored extremely high across all three metrics—precision, recall, and F1-score—each hovering around 0.96 to 0.97. This means not only did the model make correct predictions often, but it also missed very few actual instances. Other classes like route failure, blackhole attack, and high delay also had impressive F1-scores above 0.85, showing the model's capability to learn from patterns and apply that knowledge effectively in different test cases. Even smaller classes like energy depletion and network congestion were handled well, which is notable because models often struggle with underrepresented categories.

However, the HELLO flooding class had lower values across the board, with an F1-score of around 0.67. While this isn't terrible, it does indicate room for improvement. In real-world deployments, such missed detections could be the difference between a stable network and a vulnerable one. This is where techniques like data augmentation, synthetic data generation, or reward shaping during training could make a difference in the future.

The macro average of the model metrics was 0.85 for F1-score, and the weighted average was even higher, around 0.94. The macro average treats each class equally, which is helpful when evaluating balance. The weighted average takes into account the number of samples per class, which tells us the model's high performance isn't just due to doing well on common classes like normal routing but extends to rarer ones too[17]. This balanced outcome is especially important for a system that aims to be both efficient and fair in dynamic environments.

Then we looked at the ROC curve (Figure 6), which gave us yet another perspective on model performance. The ROC curve plots the true positive rate against the false positive rate for each class, and the Area Under the Curve (AUC) helps quantify that performance. Most classes had AUC values above 0.90, which is excellent. Class 1 had the highest AUC at 0.97, meaning it was nearly perfect in distinguishing that class from all others. Even the lowest AUC score, which was 0.87 for Class 3, was still reasonably strong. High AUC scores indicate that the model isn't just randomly guessing; it's making confident and reliable predictions across the board.

Another important aspect is how well the model deals with the challenges of a real MANET environment. One of those challenges is class imbalance. Some network behaviours occur far more frequently than others, which means the model might become biased toward those[21]. To avoid this, we used techniques like SMOTE for oversampling rare classes and weighted loss functions during training. These steps helped ensure that even classes with fewer examples got fair attention during learning. The results from the classification report and ROC curve support this—rare events like energy depletion and high delay were still accurately predicted.

We also looked at model robustness by adding noise to the input features and evaluating how the model responded. The performance didn't drop significantly, which suggests that the DRL model is resilient and can handle the kind of unpredictability that naturally happens in real-world MANETs. This kind of robustness is important because, in a real network, you can never fully control what kind of data the system will receive[19].

One of the most impressive outcomes of this project was the model's adaptability. DRL, by its nature, learns through interaction. It wasn't just fed examples but had to learn what works and what doesn't by receiving rewards or penalties. This helped the model adapt to complex and changing conditions, which is something static models like SVMs and even some CNNs struggle with. By reinforcing good decisions and discouraging bad ones, the model gradually became better at choosing optimal paths and recognizing suspicious behaviour, even when the patterns weren't obvious.

Another interesting observation came from the visualizations we created during training, like reward curves and loss plots. These helped us see how the model's performance evolved over time. The reward curves started off low but gradually increased and stabilized, which told us that the model was learning. Loss curves followed a similar pattern, dropping steadily over time. These visuals were not just technical graphs but confirmation that the model was moving in the right direction[23].

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(56s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Despite all these positive results, it's important to acknowledge the limitations. The simulation data, while detailed, can't capture every nuance of a live MANET scenario. Real-world networks involve hardware limitations, unpredictable user behaviour, and environmental factors that a simulation might miss. Additionally, some classes still need better performance. More training data, especially for underrepresented classes, and further fine-tuning of the DRL parameters could help close this gap. It might also be worth exploring hybrid models that combine DRL with other techniques like anomaly detection algorithms to create a layered defines mechanism.

In conclusion, the DRL model showed strong potential for not only optimizing network performance in MANETs but also ensuring that the system remains secure by correctly authenticating behaviour. With high accuracy, strong AUC scores, balanced class predictions, and adaptability to dynamic conditions, the model offers a practical solution to many of the challenges faced in MANET environments. That said, the journey doesn't stop here. There's still work to be done in making the model more robust and more accurate in edge cases, but what we have so far lays a very solid foundation for future improvements and real-world deployment[20].

5.1 Network Behaviours Identified by the Model

Instance	Actual Class	Predicted Class
Sample 1	Normal Routing	Normal Routing
Sample 2	Blackhole Attack	Blackhole Attack
Sample 3	Sybil Attack	Sybil Attack
Sample 4	Route Failure	Route Failure
Sample 5	HELLO Flooding	HELLO Flooding
Sample 6	Network Congestion	Network Congestion
Sample 7	Energy Depletion	Energy Depletion
Sample 8	High Delay	High Delay

Figure 7. Different network behaviours or attack types identified by DRL model.

The image above shows a group of skin lesion pictures that were correctly identified by a Convolutional Neural Network (CNN) model [17]. Each picture represents a different skin condition, and both the actual diagnosis (ground truth) and the model's prediction are shown in green text, indicating that they match. The dataset includes a range of skin diseases like actinic keratosis, basal cell carcinoma, dermatofibroma, melanoma, nevus, pigmented benign keratosis, seborrheic keratosis, squamous cell carcinoma, and vascular lesions. The results show that the model performs really well in recognizing and correctly identifying these different conditions. The fact that the predicted labels match the actual ones across all nine types suggests that the CNN has effectively learned to pick up on the key patterns and features that make each disease distinct from the others [18]. This proves that the model is quite accurate and dependable when it comes to analysing skin images. Its strong performance makes it a valuable tool for helping with automated skin disease diagnosis, which could make the process faster and more reliable.

6. FUTURE SCOPE

While our current model using Deep Reinforcement Learning (DRL) showed strong results in both optimizing MANET performance and identifying suspicious behaviour, there's still a lot of room to take this research further. One of the most obvious next steps would be to test this model in a real-world environment, outside of simulations. Real networks are often unpredictable—they deal with hardware limitations, signal interference, and user behaviour that a simulated setup can't fully capture. Testing the model in a live setting could help us understand how it reacts under pressure and whether it can maintain its performance in more chaotic conditions.

Another interesting direction would be to explore how the model adapts in larger-scale networks. As the number of nodes increases, so does the complexity of routing and behaviour patterns. Expanding the model to handle more nodes without losing accuracy or speed could make it even more useful for practical applications like emergency communication systems or military field networks.

We could also look into combining DRL with other intelligent systems. For example, adding a lightweight anomaly detection layer before routing decisions could serve as an early warning system. Or, integrating a federated learning approach might allow the model to learn from multiple devices without needing to share raw data, making it more privacy-friendly and scalable.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Finally, future versions of this system could be designed to self-update and learn continuously over time. This way, the model could keep improving as it sees more examples and adapts to new threats or patterns—all without needing to be retrained from scratch every time the environment changes.

7. CONCLUSION

In this study, we set out to explore how Deep Reinforcement Learning (DRL) can be used to make Mobile Ad-Hoc Networks (MANETs) not only smarter but also more secure. Through our approach, we were able to show that DRL doesn't just improve how the network performs—it also helps verify whether the decisions being made are coming from trusted sources. This dual-purpose use of DRL turned out to be quite effective. The model performed well across various classes, identifying both normal and malicious behaviours with strong accuracy, and adapting to changing conditions in real time.

We saw that the model could pick up on subtle patterns in node activity and make decisions that balanced performance with trust. Its ability to learn from interaction instead of fixed rules gave it a flexible edge that's really valuable in unpredictable environments like MANETs. While there's still room to improve—especially in dealing with rare events or scaling to larger networks—the results so far are promising. This model brings us a step closer to building more intelligent, dependable, and secure ad-hoc communication systems.

8. REFERENCES

1. Yahja, Alex, et al. "DeepADMR: a deep learning-based anomaly detection for MANET routing." *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*. IEEE, 2022.
2. Abbood, Zainab Ali, Doğu Çağdaş Atilla, and Çağatay Aydin. "Intrusion Detection System through deep learning in routing MANET networks." *Intelligent Automation & Soft Computing* 37.1 (2023).
3. Karthik, M. Ganesh, et al. "An intrusion detection model based on hybridization of S-ROA in Deep Learning Model for MANET." *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* 48.2 (2024): 719-730.
4. Xu, Zhiwei, et al. "Deep Learning-based Intrusion Detection Systems: A Survey." *arXiv preprint arXiv:2504.07839* (2025).
5. Ele, B. I., and B. C. E. Mbam. "Development of a Layered Conditional Random Field Based Network Intrusion Detection System." *West African Journal of Industrial and Academic Research* 12.1 (2014): 3-20.
6. Chaganti, Rajasekhar, et al. "Deep learning approach for SDN-enabled intrusion detection system in IoT networks." *Information* 14.1 (2023): 41.
7. Sathiya, R., and N. Yuvaraj. "Swarm optimized differential evolution and probabilistic extreme learning-based intrusion detection in MANET." *Computers & Security* 144 (2024): 103970.
8. Sultan, Mohamad T., Hesham El Sayed, and Manzoor Ahmed Khan. "An intrusion detection mechanism for MANETs based on deep learning artificial neural networks (ANNs)." *arXiv preprint arXiv:2303.08248* (2023).
9. Hussain, S., and S. M. H. Fathima. "Federated Learning-Assisted Coati Deep Learning-Based Model for Intrusion Detection in MANET." *International Journal of Computational Intelligence Systems* 17.1 (2024): 1-15.
10. Edwin Singh, C., and S. Maria Celestin Vigila. "WOA-DNN for Intelligent Intrusion Detection and Classification in MANET Services." *Intelligent Automation & Soft Computing* 35.2 (2023).
11. Prashanth, S. K., Hena Iqbal, and Babu Illuri. "An enhanced grey wolf optimisation–deterministic convolutional neural network (GWO–DCNN) model-based IDS in MANET." *Journal of Information & Knowledge Management* 22.04 (2023): 2350010.
12. Gurung, Shashi, and Siddhartha Chauhan. "A dynamic threshold-based approach for mitigating black-hole attack in MANET." *Wireless Networks* 24.8 (2018): 2957-2971.
13. Panagiotis, Papadimitratos. "Secure routing for mobile ad hoc networks." *Proc. SCS CDNS, Jan. 2002* (2002).
14. Shahabi, Sina, Mahdieh Ghazvini, and Mehdi Bakhtiarian. "A modified algorithm to improve security and performance of AODV protocol against black hole attack." *Wireless Networks* 22 (2016): 1505-1511.
15. Alnumay, Waleed, Uttam Ghosh, and Pushpita Chatterjee. "A trust-based predictive model for mobile ad hoc network in internet of things." *Sensors* 19.6 (2019): 1467.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

16. Baadache, Abderrahmane, and Ali Belmehdi. "Fighting against packet dropping misbehavior in multi-hop wireless ad hoc networks." *Journal of Network and Computer Applications* 35.3 (2012): 1130-1139.
17. Panigrahi, Ranjit, et al. "A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets." *Mathematics* 9.7 (2021): 751.
18. Gurung, Shashi, and Siddhartha Chauhan. "A survey of black-hole attack mitigation techniques in MANET: merits, drawbacks, and suitability." *Wireless Networks* 26 (2020): 1981-2011.
19. Cai, Ruo Jun, Xue Jun Li, and Peter Han Joo Chong. "An evolutionary self-cooperative trust scheme against routing disruptions in MANETs." *IEEE Transactions on Mobile Computing* 18.1 (2018): 42-55.
20. Arunmozhi, S. A., and Y. Venkataramani. "Black hole attack detection and performance improvement in mobile ad-hoc network." *Information Security Journal: A Global Perspective* 21.3 (2012): 150-158.
21. Mohanapriya, M., and Ilango Krishnamurthi. "Modified DSR protocol for detection and removal of selective black hole attack in MANET." *Computers & Electrical Engineering* 40.2 (2014): 530-538.
22. Tamilselvan, Latha, and V. Sankaranarayanan. "Prevention of co-operative black hole attack in MANET." *J. Networks* 3.5 (2008): 13-20.
23. El-Semary, Aly M., and Hossam Diab. "BP-AODV: Blackhole protected AODV routing protocol for MANETs based on chaotic map." *IEEE Access* 7 (2019): 95197-95211.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.