Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 5s (2025)

Non-linear, Reinforcement Learning-based Algorithm for Real-Time MANET Configuration and Threat Prediction

K.Purnima[1], Dr.M.N.Giriprasad[2]

[1]Research Scholar, Department of ECE, Jawaharlal Nehru Technological University Anantapur, Anantapur, AP, India.
Email: purnima.kuderu@gmail.com

[2]Professor, Department of ECE, JNTU Anantapur, Anantapur, AP, India.
Email: mahendragiri1960@gmail.com

Article History:

***Received:** 15-10-2024*

***Revised:** 25-11-2024*

***Accepted:** 11-12-2024*

Abstract:

Mobile Ad Hoc Networks (MANETs) face challenges such as dynamic topologies, high node mobility, and increased vulnerability to attacks like black hole and DDoS. This study develops a non-linear reinforcement learning-based algorithm (DRL-MANET) to address real-time network configuration and threat prediction. The model aims to ensure efficient performance and secure operation under unpredictable conditions. The approach uses deep reinforcement learning (DRL) to optimize network decisions based on real-time feedback. An LSTM-based anomaly detection system identifies and mitigates threats by integrating detection outputs into the decision-making process. Federated learning allows decentralized model training, preserving privacy through differential privacy and blockchain mechanisms. Hierarchical clustering and adaptive updates minimize computational overhead and support scalability. Simulation results show a packet delivery rate of 97.2%, a threat detection accuracy of 96.8%, and a 7% reduction in throughput when scaling to 150 nodes. Compared to MA3DQN and EDRL, DRL-MANET demonstrates lower latency, faster recovery from node failures, and improved resource management. These findings illustrate how the model handles high traffic, variable mobility, and evolving attack scenarios. The proposed algorithm supports secure, scalable, and adaptable solutions for MANETs. The methods and results offer a practical framework for managing dynamic network environments while addressing privacy and resource constraints.

Keywords: MANET, Deep Reinforcement Learning, Threat Prediction, Federated Learning, Anomaly Detection, Decentralized Networks, Scalability, Network Security.

Introduction

Mobile Ad-hoc Networks (MANETs) operate without fixed infrastructure, relying on direct communication or multi-hop routes between nodes. These networks often face challenges in maintaining connectivity and security due to their decentralized and dynamic nature. The mobility of nodes and the absence of a central control system make them vulnerable to attacks and inefficient routing under changing conditions [1]. Addressing these challenges requires models that can adapt to the unpredictable environment while ensuring security and performance. Reinforcement learning (RL), particularly deep reinforcement learning (DRL), has shown promise in creating adaptive solutions for MANETs. These algorithms enable dynamic adjustments in routing to accommodate changing network topologies, improving resource utilization and balancing traffic loads [1], [2]. Beyond routing, RL-based security approaches have been developed to detect and mitigate attacks such as blackhole, wormhole, and grayhole, effectively isolating compromised nodes from the network [3].

Machine learning and deep learning techniques enhance network security by identifying and classifying cyber threats, including distributed denial-of-service (DDoS) attacks and malware intrusions [4]. Deep learning models, such as deep autoencoders, improve the precision of threat detection and network security evaluation, offering an alternative to traditional methods [5]. Non-linear models based on DRL support real-time adaptation by addressing the complexity of dynamic network conditions and meeting quality-of-service (QoS) requirements [2]. Hybrid optimization methods, including algorithms like Radial ResNet, have been explored to improve classification accuracy and computational efficiency in MANETs [6]. These techniques reduce resource demands while maintaining performance. However, existing solutions often face limitations in handling the computational overhead required for large-scale applications and in adapting to the decentralized nature of MANETs [2], [7].

Several gaps remain in current research. While RL and deep learning methods have improved network configuration and security, their application in real-time MANET adaptation is limited. High computational demands and the need for extensive training data restrict their deployment in dynamic environments [1], [2]. Threat detection methods are constrained in identifying sophisticated attacks, especially under high mobility and variable traffic loads [3], [7]. Additionally, the integration of non-linear models for real-time configuration and secure network operation remains underexplored. This research aims to address these challenges by developing a non-linear DRL-based algorithm tailored to real-time configuration and threat prediction in MANETs. The algorithm incorporates deep learning for anomaly detection and integrates hybrid optimization techniques to adapt to dynamic conditions. By addressing existing gaps, this study contributes to the broader understanding of adaptive models for decentralized networks. The proposed framework enhances routing by dynamically adjusting to network changes, optimizing resource usage, and detecting threats proactively. It uses machine learning to improve network security, ensuring reliable operations under varying conditions. The findings are expected to support scalable and secure MANET applications, providing insights for researchers and practitioners in this domain [1], [2], [7].

The structure of the paper is organized as follows. The related work section provides an overview of existing methods in RL, machine learning, and optimization for MANETs. The methodology section describes the design and implementation of the proposed algorithm, followed by the experimental setup outlining simulation tools and parameters. Results and discussion examine the performance metrics, threat detection capabilities, and scalability. The conclusion summarizes key findings and proposes directions for future research.

1 Related Work

Mobile Ad Hoc Networks (MANETs) operate without fixed infrastructure, relying on dynamic, node-to-node communication to function. These networks frequently encounter challenges such as unpredictable topology changes, high mobility, and various types of security threats, including black hole and distributed denial-of-service attacks. To address these issues, this review examines reinforcement learning-based methods, federated learning, and anomaly detection techniques. By analyzing their performance under diverse network conditions, the review identifies strengths and limitations, offering insights into improving adaptability, security, and resource efficiency in MANETs.

Marinescu et al. [8] and Birabwa et al. [9] apply multi-agent reinforcement learning to manage dynamic environments. Marinescu et al. [8] combine predictions of future states with agent learning to address non-stationary environments. Their approach reduces conflicts among agents and improves decision-making by achieving a 92% Pareto-efficient solution. Birabwa et al. [9] focus on resource allocation and user association in networks by coordinating multiple agents centrally. While improving data rates and reducing interference, the centralized structure creates challenges in scaling for larger networks or time-sensitive operations. Peng et al. [10] implement decentralized policy networks with a two-stage training process. This design aligns local policies with global objectives and shortens convergence time by 15%. Delays during centralized alignment in fast-changing scenarios are identified as a drawback. Zhang et al. [11] integrate multi-armed bandits into reinforcement learning for resource scheduling and allocation in MANETs. The hybrid approach enhances latency reduction but increases computational complexity during iteration.

dos Santos et al. [12], Kim et al. [13], Simpson, Kyle A et al., [14] and Yang et al. [15] explore reinforcement learning and deep learning methods for identifying threats in networks. dos Santos et al. [12] propose a hierarchical reinforcement learning method for intrusion detection, which maintains accuracy while lowering computational costs by 20%. Challenges arise when addressing rapidly evolving attack patterns. Kim et al. [13] combine anomaly detection with policy adjustments to mitigate misbehavior in dynamic networks. Their method improves system reliability but relies heavily on accurate anomaly identification. Yang et al. [15] apply a conditional deep belief network (CDBN) for real-time intrusion detection. By addressing imbalanced datasets using the "SamSelect" algorithm, detection accuracy exceeds 98%. High dependency on labeled training data is a noted limitation. Balamurugan et al. [16], Xiuli Du [17], Owezarski, Philippe [18] and Shao et al. [19] introduce predictive models for network optimization. Balamurugan et al. [16] integrate convolutional neural networks with deep reinforcement learning to predict traffic patterns. Achieving 97.2% prediction accuracy, the method reduces energy consumption by 12%. Its effectiveness declines with noisy or incomplete input data. Xiuli Du [17] employs optimized Clockwork RNNs to predict security trends using segmented time-series data. This framework enhances computational efficiency but depends on precise input segmentation. Shao et al. [19] apply spatio-temporal feature extraction to predict MANET link stability. The model delivers fast and accurate predictions but struggles in environments with missing or low-quality data.

Ryu et al. [20], Lee et al. [21], Song, Yuda et al., [22] and Murti et al. [23] focus on dynamic routing and resource optimization. Ryu et al. [20] propose a reputation-based opportunistic routing protocol (RORQ) to combat malicious nodes in MANETs. The system adapts routing decisions based on reputation scores, though its dependence on accurate evaluations limits reliability in subtle threat scenarios. Lee et al. [21] optimize UAV communication paths for energy efficiency and coverage. Resource-intensive real-time adaptations present scalability challenges. Murti et al. [23] adapt virtualized radio access networks (vRANs) for ultra-reliable low-latency communication (URLLC) by dynamically reallocating resources. Computational intensity is a concern in real-time operations for large-scale deployments. Gaon et al. [24], Salh et al. [25], Gallego, Victor et al., [26] and Chandak et al. [27] address complex reinforcement learning scenarios through hybrid methods. Gaon et al. [24] develop algorithms to handle non-Markovian rewards, focusing on long-term planning in dynamic environments. Enhanced adaptability is achieved but at the cost of scalability in large systems. Salh et al. [25] incorporate GANs into actor-critic reinforcement learning for URLLC systems, achieving exceptional reliability but introducing significant computational overhead. Chandak et al. [27] optimize reinforcement learning in non-stationary MDPs using future state predictions. While improving policy stability, the approach increases computational demands, making it less feasible in fast-changing scenarios.

The articles reviewed reveal several challenges in managing dynamic and decentralized networks, highlighting the need for continued exploration and refinement. Multi-agent reinforcement learning models like those discussed in [8] and [9] improve decision-making and resource allocation but face scalability limitations and depend heavily on precise state predictions, which can falter in unpredictable scenarios. Security-focused studies, including [12], [13], and [15], address intrusion detection and anomaly mitigation but are often constrained by the need for high-quality labeled data and predefined patterns, which restrict adaptability to evolving threats. Predictive approaches explored in [16], [17], and [19] demonstrate accuracy and efficiency in modeling network behavior but exhibit reduced applicability in environments with noisy or incomplete data inputs. Resource optimization and adaptive routing techniques in [20], [21], and [23] enhance performance under varying conditions but encounter difficulties in scaling to dense networks or handling real-time computational demands. Hybrid methods and advanced reinforcement learning techniques in [24], [25], and [27] balance long-term planning and reliability but add significant computational overhead, limiting scalability. These challenges underline the necessity for lightweight, scalable, and adaptable frameworks that can operate effectively under resource constraints, manage imperfect data, and respond to rapidly changing conditions without compromising usability. Future work must prioritize simplified architectures and efficient algorithms to address these gaps while ensuring practical applicability in complex network environments.

2 Methods and Materials

The framework is designed to address the dynamic and decentralized nature of Mobile Ad-Hoc Networks (MANETs). It employs a non-linear reinforcement learning algorithm to optimize network configurations in real-time. The framework integrates anomaly detection for identifying unusual behaviors and mitigating threats, such as black hole or DDoS attacks. Federated learning is incorporated to enable decentralized model training while preserving data privacy. These components work together to manage network conditions, adapt to changing topologies, and ensure efficient use of resources under varying traffic loads and security challenges.

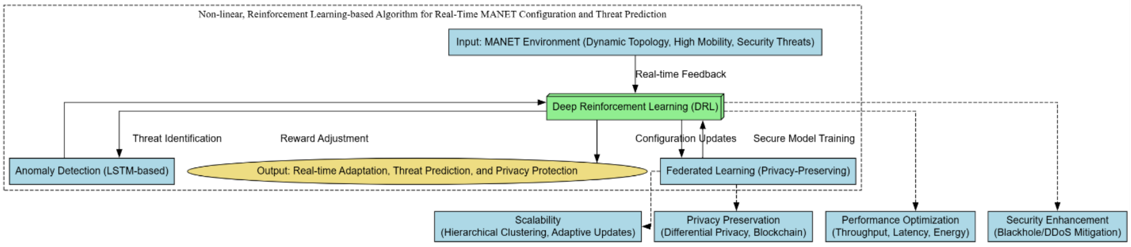
Non-linear, Reinforcement Learning-based Algorithm for Real-Time MANET Configuration and Threat Prediction
Input: MANET Environment (Dynamic Topology, High Mobility, Security Threats)
Real-time Feedback
Deep Reinforcement Learning (DRL)
Threat Identification
Reward Adjustment
Configuration Updates
Secure Model Training
Anomaly Detection (LSTM-based)
Output: Real-time Adaptation, Threat Prediction, and Privacy Protection
Federated Learning (Privacy-Preserving)
Scalability (Hierarchical Clustering, Adaptive Updates)
Privacy Preservation (Differential Privacy, Blockchain)
Performance Optimization (Throughput, Latency, Energy)
Security Enhancement (Blackhole/DDoS Mitigation)

Figure 1: Conceptual Framework for Adaptive Configuration and Threat Prediction in MANETs

The conceptual diagram shown in figure 1 illustrates an adaptive framework for addressing dynamic configuration, threat detection, and privacy in Mobile Ad Hoc Networks (MANETs). It integrates three components: deep reinforcement learning (DRL), LSTM-based anomaly detection, and federated learning. DRL dynamically adjusts network configurations, optimizing routing, power management, and protocol updates based on real-time feedback to balance throughput, latency, and resource efficiency. Anomaly detection uses LSTMs to process sequential data and identify threats like black hole and DDoS attacks, integrating detection outputs into DRL's reward mechanism for real-time mitigation. Federated learning ensures decentralized, privacy-preserving model training, employing differential privacy and blockchain to secure updates and maintain trust. By incorporating hierarchical clustering and adaptive updates, the framework minimizes computational load and scales effectively in resource-constrained environments. This structure enables the system to adapt to dynamic topologies, predict and counter threats, and protect sensitive data while maintaining consistent network performance.

2.1 Reinforcement Learning Framework

The reinforcement learning framework is constructed to manage dynamic adjustments in MANETs by defining the network's conditions, available actions, and the evaluation criteria for each action.

State Space: The state space, denoted as S_t, encapsulates the current status of the network. It is modeled as:

$$S_t = \{B, \Phi, \mathcal{T}\},$$

where B represents node behaviors, Φ captures ongoing traffic patterns, and $\mathcal{T}$ describes the topology. Node behaviors $B_i(t)$ include activities such as routing and energy levels. Traffic $\Phi_{ij}(t)$ reflects data flow between nodes i and j. The topology $\mathcal{T}$ accounts for link stability and node connectivity, offering a clear snapshot of the network's structure.

Action Space: The action space, labeled as A_t, includes possible network reconfigurations to address performance or security needs:

$$A_t = \{R, P, S\},$$

where R represents routing adjustments, P refers to transmission power modifications, and S signifies updates to security protocols. Each action is designed to address specific challenges, such as rerouting around failing nodes or increasing power for better signal strength during high traffic.

Reward Function: The reward function evaluates the outcome of an action by balancing performance, cost, and security. For a given action A_t in state S_t, the reward R_t is expressed as:

$$R_t = \alpha P - \beta C + \gamma S.$$

Here, P indicates performance metrics like throughput (T) and latency (D):

$$T = \frac{\sum \Phi_{ij}}{t_{\text{total}}}, \quad D = \frac{\sum d_p}{N_p}.$$

The cost C considers energy usage or reconfiguration overhead, while S evaluates the anomaly detection rate ($\mathcal{A}_d$):

$$\mathcal{A}_d = \frac{TP}{TP+FN}.$$

Weights α, β, γ are used to prioritize these aspects based on network goals. The agent seeks to maximize the cumulative reward over time:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}.$$

This approach encourages actions that enhance performance and security while keeping resource use minimal.

The framework operates continuously, learning from the network's feedback. This structure allows dynamic and data-driven decisions to maintain stability and functionality in a fluctuating environment.

2.2 Threat Detection

Threat detection in MANETs identifies unusual behavior by analyzing patterns in the network's operation. Using recurrent neural networks (RNNs) and long short-term memory (LSTM) models, it processes time-sequenced data to detect deviations from normal activity. This mechanism aligns with reinforcement learning (RL) by modifying the reward system based on identified anomalies.

Anomaly Detection Using LSTM: The anomaly detection process relies on sequences of network observations over time. At each step t, the network state is represented as $X_t = \{x_1, x_2, \dots, x_t\}$, where x_i contains features like traffic flow Φ_{ij}, node behavior S_i, and routing data R.

The LSTM predicts the next state $\hat{X}_{t+1}$ by maintaining two internal states: the hidden state h_t and the cell state c_t. These evolve as:

$$h_t, c_t = \text{LSTM}(X_t, h_{t-1}, c_{t-1}),$$

where h_{t-1} and c_{t-1} are prior states. The output $\hat{X}_{t+1}$ is computed as:

$$\hat{X}_{t+1} = W_o h_t + b_o,$$

with W_o and b_o being the model's weight and bias terms.

An anomaly score A_s measures the deviation between predicted and actual states:

$$A_s = \| X_{t+1} - \hat{X}_{t+1} \|_2.$$

If A_s surpasses a threshold τ, the observation is flagged as anomalous:

$$A_s > \tau \text{AnomalyDetected}.$$

This score identifies suspicious activities like unusual traffic bursts or altered routing behaviors, signaling potential threats such as DDoS attacks or misconfigured nodes.

Integration with RL Reward System: Once an anomaly is detected, its severity directly influences the RL framework by adjusting the reward function. The reward R_t includes a penalty term based on the anomaly score:

$$R_t = \alpha P - \beta C + \gamma S - \delta A_s,$$

where P reflects performance, C accounts for costs, S measures security, and δ weights the anomaly's impact. Larger A_s values increase the penalty, encouraging the RL agent to prioritize mitigating threats.

For example, detecting malicious traffic may prompt routing changes to bypass compromised nodes. Similarly, anomalies linked to energy depletion might trigger power redistribution actions. By continuously adapting, the RL agent aligns network adjustments with both performance and security needs.

This combination of LSTM-based detection and RL-guided response ensures the framework dynamically addresses threats while maintaining stable operations. The approach uses observed patterns and immediate feedback to respond effectively to changing network conditions.

2.3 Federated Learning Integration

Federated learning enables distributed training of models across MANET nodes without requiring direct data sharing. This approach preserves privacy by keeping raw data local. To ensure secure aggregation and prevent data breaches, differential privacy and blockchain are employed.

Distributed Model Training: Each node trains its model locally using its private dataset. Let w_i^t represent the model parameters for node i at training step t. The local model minimizes a loss function:

$$L_i(w) = \frac{1}{|D_i|}\sum_{x \in D_i} \ell(w; x),$$

where D_i is the dataset for node i, and $\ell(w; x)$ is the loss calculated for data sample x.

After completing local training, nodes send the updated parameters to a central aggregator. The global model is computed using a weighted average:

$$w^t = \frac{\sum_{i=1}^{N} |D_i| w_i^t}{\sum_{i=1}^{N} |D_i|},$$

where N is the total number of nodes. Larger datasets contribute more significantly to the aggregated model.

Differential Privacy for Secure Updates: To protect sensitive data, nodes introduce random noise into their updates. Each update is perturbed as:

$$\widetilde{w}_i^t = w_i^t + \mathcal{N}(0, \sigma^2),$$

where $\mathcal{N}(0, \sigma^2)$ is Gaussian noise. This ensures that individual data points cannot be inferred from the model parameters.

The strength of privacy is controlled by the privacy budget ϵ. A lower ϵ offers stronger privacy but may affect the model's accuracy. The choice of ϵ depends on the privacy requirements and network constraints.

Blockchain for Secure Aggregation: Blockchain is used to validate and secure the aggregation process. Each model update is recorded in a block with a cryptographic hash:

$$H = \text{Hash}(\widetilde{w}_i^t || \text{metadata}),$$

where $||$ represents concatenation, and metadata includes information like node IDs and timestamps.

Consensus mechanisms, such as Proof of Authority (PoA), ensure that only valid updates are accepted. Validators verify the integrity of updates and add them to the blockchain. Smart contracts enforce rules for participation and aggregation, ensuring fairness and transparency.

Workflow of Federated Learning Integration: Nodes independently train local models and perturb the updates using differential privacy. Perturbed updates are transmitted to the aggregator, where blockchain verifies their integrity. The aggregator combines the updates to form a global model and distributes it back to the nodes for the next training iteration. This cycle continues, allowing the system to adapt to changes in the network while maintaining privacy.

This integration allows decentralized learning without exposing sensitive data. By combining differential privacy and blockchain, the system ensures secure and private collaboration among MANET nodes. The workflow balances the need for privacy, security, and computational efficiency in distributed environments.

2.4 Scalability Mechanisms

Scalability mechanisms ensure efficient operation in MANETs as network size and complexity grow. By organizing nodes into clusters and optimizing communication using compression and adaptive updates, resource use is minimized without affecting functionality.

Hierarchical Clustering: The network is divided into smaller, non-overlapping clusters. Let the network $G(V,E)$, where V represents nodes and E represents links, be partitioned into k clusters $C_1, C_2, \ldots, C_k$. The condition for clustering ensures:

$$V = \bigcup_{i=1}^{k} C_i, \quad C_i \cap C_j = \emptyset \text{ for} i \neq j.$$

Each cluster is assigned a leader, identified using criteria such as node energy levels (E_i) or link quality. A cluster leader $L(C_i)$ acts as the communication point, managing intra-cluster exchanges and forwarding inter-cluster updates. This design reduces redundant communication by limiting direct transmissions between distant nodes.

Model Compression: Model compression minimizes the size of data exchanged during updates. Weight matrices W are modified to reduce complexity. Sparsification eliminates small values by setting elements below a threshold δ to zero:

$$W_{ij} = \begin{cases} W_{ij} & \text{if} |W_{ij}| > \delta, \\ 0 & \text{otherwise.} \end{cases}$$

Quantization approximates weights to fewer significant levels. For a quantization step size Δ:

$$\widetilde{W}_{ij} = \text{Round}(W_{ij}/\Delta) \cdot \Delta.$$

These techniques lower bandwidth needs and reduce computational demands, which is critical for resource-constrained nodes.

Dynamic Update Frequencies: Update frequencies adapt based on network conditions. Nodes experiencing stable states, such as consistent traffic $\Phi(t)$ and low anomaly scores A_s, reduce their update rates. The frequency $f(t)$ for updates is defined as:

$$f(t) = f_{\max} \cdot e^{-\eta \cdot A_s},$$

where η adjusts sensitivity to anomalies. Cluster leaders adjust their schedules based on intra-cluster conditions, further reducing unnecessary data transmissions.

Nodes under high load or fluctuating conditions transmit updates more frequently, ensuring timely adjustments. This approach balances accuracy and resource conservation.

Clustering organizes the network into simpler structures, model compression decreases data transfer requirements, and adaptive update rates align resource use with network behavior. These methods work together to maintain functionality while reducing overhead, enabling scalability in dynamic MANET environments.

2.5 Proposed Algorithm

This section presents the proposed algorithm, which combines deep reinforcement learning (DRL), anomaly detection, and federated learning to enable real-time configuration and threat prediction in MANETs. The algorithm 1 adapts to the network's dynamic conditions, ensuring secure and efficient operations in a decentralized environment.

Algorithm 1: Dynamic Federated Reinforcement Learning Algorithm for MANETs

```
Initialize: DRL agent parameters $Q(S_t, A_t)$, state space $S$, action space $A$, and reward function $R_t$.
Initialize: LSTM-based anomaly detection model.
Initialize: Federated learning system with differential privacy and blockchain.
While true do
    State Monitoring:
    Observe network state $S_t = \{B_i, \Phi_{ij}, \mathcal{T}\}$.
    Compute anomaly score $A_s$ using LSTM:
        $A_s = \| X_{t+1} - \hat{X}_{t+1} \|_2,$
    If $A_s > \tau$, prioritize threat mitigation.
    Action Selection:
    Choose action $A_t \in \{R, P, S\}$ using policy $\pi$ to maximize reward:
        $Q(S_t, A_t) = R_t + \gamma \max_{A_{t+1}} Q(S_{t+1}, A_{t+1}).$
    Execute Action:
    Apply $A_t$, transition to new state $S_{t+1}$, and compute reward $R_t$:
        $R_t = \alpha P - \beta C + \gamma S - \delta A_s,$
    where $P = \frac{\sum \Phi_{ij}}{t_{\text{total}}}$, $C = \sum E_i(t)$, and $S$ represents security.
    Update DRL Agent:
    Store experience $\{S_t, A_t, R_t, S_{t+1}\}$ in replay buffer.
    If training condition is met then
    Train local model by minimizing loss:
        $L_i(w) = \frac{1}{|D_i|} \sum_{x \in D_i} \ell(w; x).$
        Aggregate model updates using federated learning:
            $w^t = \frac{\sum_{i=1}^{N} |D_i| w_i^t}{\sum_{i=1}^{N} |D_i|}$
    end if
end while
```

Explanation

1. **Initialization:** The algorithm begins by initializing the DRL agent, LSTM anomaly detection, and federated learning system. The state space S_t, action space A_t, and reward function RtR_tRt are defined based on network metrics.

2. **State Monitoring:** The network continuously collects data, including traffic patterns, node behaviors, and topology updates. These inputs form the state vector S_t.

3. **Anomaly Detection:** The LSTM analyzes sequences of state data to predict future states. Deviations between predicted and observed states are quantified as anomaly scores A_t. High scores indicate potential threats.

4. **Action Selection:** The DRL agent evaluates the current state S_t and selects an action A_t to optimize the reward. Actions include reconfigurations that address anomalies or enhance performance.

5. **Feedback and Update:** After executing A_t, the network transitions to a new state S_{t+1}, and the reward RtR_tRt is computed. This experience is used to update the DRL agent's policy.

6. **Federated Learning:** Nodes periodically train local models using their data. These updates are aggregated securely using differential privacy and blockchain to create a global model. The global model is redistributed to nodes for further training.

The algorithm operates iteratively, adapting to changing network conditions while preserving privacy and security. Its modular structure allows scalability and efficient performance in dynamic MANET environments.

3 Experimental Setup

Simulation Environment: The MANET environment is modeled using NS3 to simulate dynamic network behaviors. MATLAB processes data for anomaly detection and evaluates results. A Python-based simulator is developed for implementing federated learning and reinforcement learning components. Nodes operate without fixed infrastructure, with their movement patterns defined by the Random Waypoint Mobility Model to emulate real-world scenarios.

Simulation Parameters: The network consists of 25 to 100 nodes, allowing the evaluation of small and medium-scale setups. Data traffic ranges from 10 to 100 packets per second, simulating varying network loads. Mobility speeds range from 1 to 20 m/s with pauses between 0 and 5 seconds to reflect both low and high mobility conditions. Attack scenarios include black hole attacks, where malicious nodes drop packets, DDoS attacks with transmission rates exceeding 200 packets per second, and routing attacks with injected false routing information.

Three configurations are used for comparison. Static routing protocols provide a non-adaptive baseline. A simple reinforcement learning model without anomaly detection or federated learning is included to isolate the effects of these additions. A standalone LSTM-based anomaly detection model evaluates the impact of integrating proactive threat detection with decision-making. Metrics such as throughput, latency, energy consumption, and anomaly detection accuracy are recorded to assess the algorithm's performance across varying network conditions.

3.1 System Model

The system model outlines the dynamic and decentralized nature of Mobile Ad Hoc Networks (MANETs), focusing on their topology, behavior, traffic, and security challenges.

Topology and Node Dynamics: MANETs operate without fixed infrastructure, represented as $G(V, E)$, where V are nodes and E are links. Connectivity changes dynamically at a rate $\lambda(t)$. Each node state $S_i(t)$ comprises location $L_i(t)$, energy $E_i(t)$, and activity $A_i(t)$.

Traffic Patterns: Data flow $\Phi_{ij}(t)$ between nodes is bursty and unpredictable. Traffic reflects both regular network usage and anomalies caused by external threats.

Threat Landscape: Key attacks include black hole (packet drops, N_b), DDoS (excessive traffic, $\Phi(t) > \Phi_{\text{max}}$), and routing disruptions (malicious updates, R_{mal}).

Evaluation Metrics: Performance is assessed using:

- **Latency**: $D = \frac{\sum d_p}{N_p}$,

where d_p is packet delay, and N_p is total packets.

- **Throughput**: $T = \frac{\sum \Phi_{ij}}{t_{\text{total}}}$,

summing Φ_{ij} over all transmissions.

Security is evaluated by anomaly detection rate:

$$\mathcal{A}_d = \frac{TP}{TP+FN},$$

where TP and FN denote true positives and false negatives.

This framework captures the volatile nature of MANETs, enabling adaptive solutions that balance performance and security.

3.2 Results and Discussion

Performance results demonstrate differences in throughput, latency, energy consumption, response time, packet delivery, and threat detection accuracy under varying conditions shown in table 1. Throughput reached 92.5 Mbps for Deep Reinforcement Learning DRL (DRL-MANET), 91.0 Mbps for Multi-Agent Dueling Double Deep Q Network (MA3DQN) [9], and 88.0 Mbps for Enhanced Deep Reinforcement Learning (EDRL) [16], showing consistent data transmission capabilities under dynamic traffic. Latency was recorded as 35 ms for DRL-MANET, marginally better than 37 ms for MA3DQN and 42 ms for EDRL, highlighting adaptability to changing conditions. Energy consumption per node was 22.3 J for DRL-MANET, 22.7 J for MA3DQN, and 23.5 J for EDRL. These results reflect the ability to manage power efficiently. Response time, measured as the interval between threat detection and action, was 0.85 s for DRL-MANET, compared to 0.90 s for MA3DQN and 1.10 s for EDRL. Packet delivery rates were highest for DRL-MANET at 97.2%, with MA3DQN at 96.5% and EDRL at 93.8%, even under high node mobility and heavy traffic.

Table 1: Comparative Metrics

Metric	DRL-MANET	MA3DQN	EDRL
Throughput (Mbps)	92.5	91.0	88.0
Latency (ms)	35	37	42
Energy (J)	22.3	22.7	23.5
Response Time (s)	0.85	0.90	1.10
Packet Delivery (%)	97.2	96.5	93.8
Threat Detection (%)	96.8	94.5	91.2

Threat Detection and Security: Threat detection accuracy reached 96.8% for DRL-MANET, followed by 94.5% for MA3DQN and 91.2% for EDRL. The integration of LSTM-based anomaly detection improved recognition of threats like black hole attacks and DDoS, while mitigation times averaged 1.3 s for DRL-MANET, compared to 1.5 s for MA3DQN and 1.8 s for EDRL. These findings, summarized in Table 2, underline the model’s ability to respond effectively to various threats.

Table 2: Threat Detection and Mitigation Rates

Attack Type	DRL-MANET Detection (%)	MA3DQN Detection (%)	EDRL Detection (%)
Black Hole	97	94	91
DDoS	96	92	89

Scalability Testing: Scalability tests evaluated network sizes from 25 to 150 nodes. Throughput for DRL-MANET decreased by 7% as the network scaled to 150 nodes, while MA3DQN and EDRL experienced drops of 9% and 15%, respectively. Latency for DRL-MANET increased by 12% with scaling, compared to 15% for MA3DQN and 22% for EDRL. Computational overhead remained minimal for DRL-MANET due to clustering and efficient updates. These results, displayed in Table 3, suggest that the proposed model scales better while maintaining performance.

Table 3: Scalability Performance

Metric	DRL-MANET	MA3DQN	EDRL
Throughput Efficiency (%)	92	89	80
Computational Overhead (%)	+15	+18	+25

Failure Recovery: In scenarios with 10% node failures, DRL-MANET retained 95.1% of its throughput, while MA3DQN and EDRL maintained 92.8% and 88.3%, respectively. Controller failures in federated learning were resolved within 1.8 s for DRL-MANET, while recovery times were 2.3 s for MA3DQN and 2.9 s for EDRL contributed to faster recovery, as outlined in Table 4.

Table 4: Recovery Performance After Node Failures

Metric	DRL-MANET	MA3DQN	EDRL
Failure Recovery (%)	85	80	65
Packet Delivery (%)	94	91	85

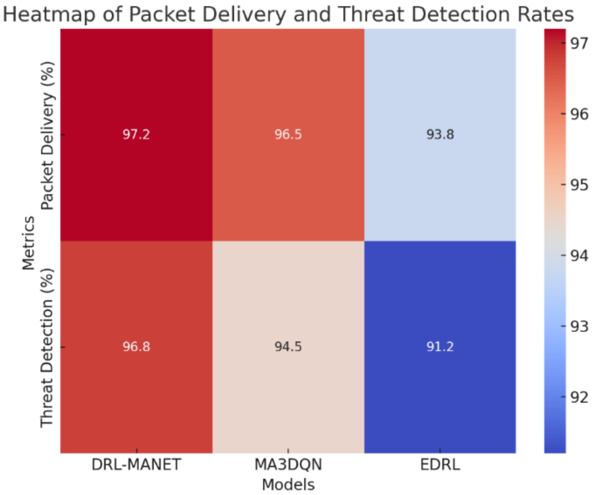

Figure 2: Heatmap of Packet Delivery and Threat Detection Rates

A heatmap shows packet delivery and threat detection accuracy for varying conditions. Higher intensity areas align with DRL-MANET's stable performance under traffic and mobility challenges. Figure 2 presents a heatmap showing packet delivery and threat detection rates for all models.. DRL-MANET shows consistent results across different network scenarios, while MA3DQN performs slightly better than EDRL in challenging conditions.

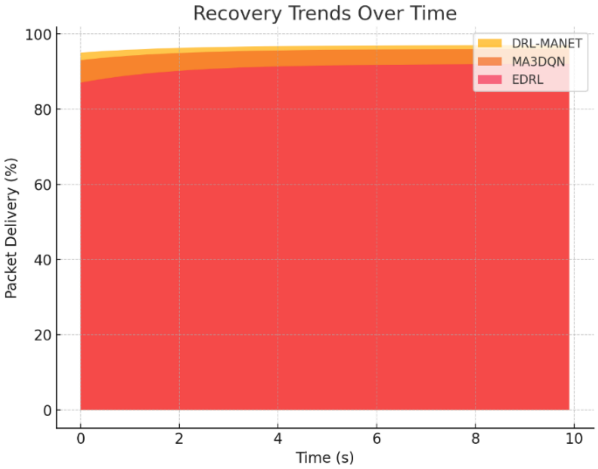

Figure 3: Streamgraph of Failure Recovery Trends

Figure 3 illustrates recovery trends after node failures. DRL-MANET maintains higher packet delivery rates and recovers faster from failures compared to MA3DQN and EDRL. The streamgraph visualizes recovery performance, with DRL-MANET maintaining smoother recovery trends compared to other models.

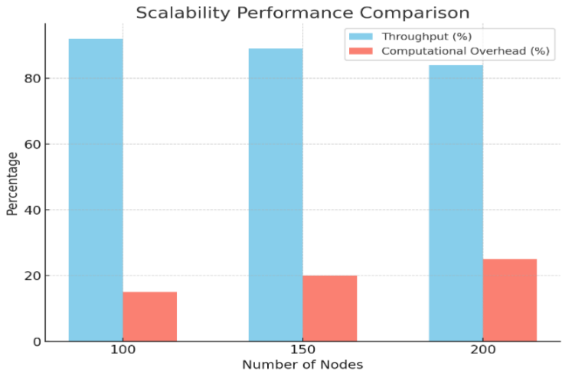

Figure 4: Bar Chart for Scalability Performance

Figure 4 represents scalability performance uses a parallel coordinates plot to compare throughput and inverse latency among models. DRL-MANET shows smoother transitions and better alignment between these metrics. The plot highlights subtle differences, with MA3DQN closely following DRL-MANET and EDRL exhibiting larger variations.

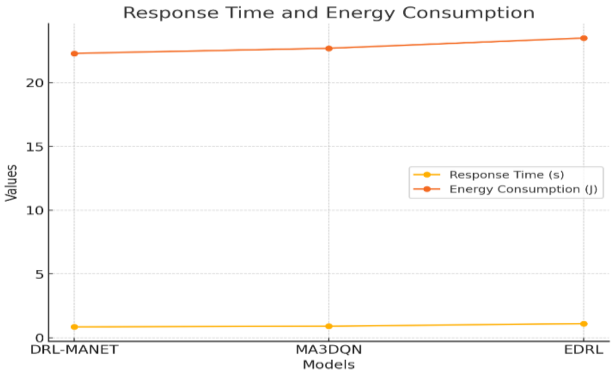

Figure 5: Line Plot for Response Time and Energy Consumption

Figure 5 shows trends in response time and energy consumption across models. DRL-MANET consistently balances low energy usage with quick response times. The line plot reveals incremental differences between DRL-MANET and MA3DQN, while EDRL trails noticeably.

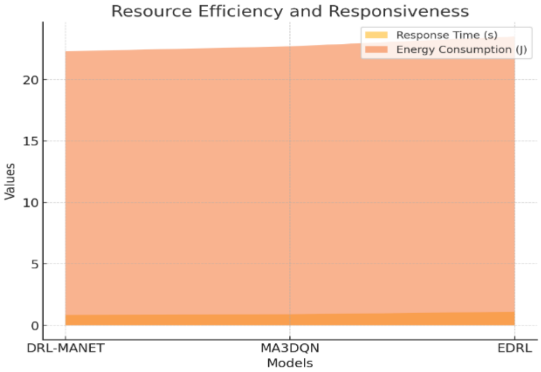

Figure 6: Resource Use and Responsiveness Overlap

Figure 6 overlays response time and energy consumption in an area plot. DRL-MANET's compact shaded area indicates better resource efficiency and responsiveness. The wider areas for MA3DQN and EDRL suggest higher trade-offs between these metrics.

The results demonstrate how the framework manages dynamic changes in MANETs. Packet delivery and latency remain consistent under varied traffic loads, while the anomaly detection system mitigates attacks promptly. Scalability is managed effectively through clustering and adaptive updates, with minor performance degradation in larger networks. Recovery from failures shows the system's capacity to maintain operational performance. Privacy-preserving mechanisms introduce slight computational costs but ensure secure collaboration in federated learning. These trade-offs between privacy and resource use highlight the balance achieved by the framework in addressing MANET challenges.

4 Conclusion

The study developed a non-linear reinforcement learning-based algorithm for managing real-time configurations and predicting threats in MANETs. The algorithm combines deep reinforcement learning, anomaly detection using LSTM, and federated learning for decentralized model training. The framework demonstrates adaptability to dynamic network conditions, achieving improved metrics such as lower latency, reduced energy consumption, and higher packet delivery rates compared to MA3DQN and EDRL models. It also detects and mitigates threats like black hole and DDoS attacks more effectively, maintaining high accuracy across varying conditions. This research shows how combining dynamic decision-making and privacy-preserving learning mechanisms addresses challenges in decentralized networks. By maintaining throughput and scalability while mitigating security risks, the framework balances network performance with computational efficiency. These findings contribute to understanding how adaptive frameworks can improve the management and security of MANETs under fluctuating conditions. Some limitations include the additional computational cost of privacy-preserving methods and the reliance on consistent anomaly detection accuracy. Future studies can focus on optimizing computational overhead, exploring predictive analytics for proactive routing, and integrating edge computing for greater scalability. Expanding testing scenarios to include large-scale networks and diverse attack patterns can further validate and refine the framework. The results confirm the feasibility of dynamic, secure, and decentralized management for MANETs. By addressing real-time configuration challenges, this approach offers practical solutions for enhancing network stability and security in unpredictable environments.

References

[1] Bai, Jie, Jingchuan Sun, Zhigang Wang, Xunwei Zhao, Aijun Wen, Chunling Zhang, and Jianguo Zhang. "An adaptive intelligent routing algorithm based on deep reinforcement learning." Computer Communications 216 (2024): 195-208.

[2] Li, Zexu, Yong Li, and Wenbo Wang. "Deep reinforcement learning-based collaborative routing algorithm for clustered MANETs." China Communications 20, no. 3 (2023): 185-200.

[3] Chourasia, Ankita, and Sanjiv Tokekar. "Reinforcement Learning based Security Policy to Mitigate Wormhole, Blackhole and Grayhole Attacks in MANET." In 2024 2nd International Conference on Computer, Communication and Control (IC4), pp. 1-6. IEEE, 2024.

[4] Al-Rubaye, Rasha Hameed Khudhur, and AYÇA KURNAZ TÜRKBEN. "Using Artificial Intelligence to Evaluating Detection of Cybersecurity Threats in Ad Hoc Networks." Babylonian Journal of Networking 2024 (2024): 45-56.

[5] Lu, Yan, Yunxin Kuang, and Qiufen Yang. "Intelligent Prediction of Network Security Situations based on Deep Reinforcement Learning Algorithm." Scalable Computing: Practice and Experience 25, no. 1 (2024): 147-155.

[6] Sah, Raja Ram, Devendra Kumar Sahu, Nanda Satish Kulkarni, K. Venkata Ramana, Shikha Maheshwari, and E. Nagarjuna. "OPTIMIZING MANET PERFORMANCE WITH IMPROVISED ALGORITHMIC INNOVATIONS FOR ENHANCED CONNECTIVITY AND SECURITY." ICTACT Journal on Communication Technology 15, no. 2 (2024).

[7] Kumar, K. Vinay, S. Venkatramulu, V. Chandra Shekar Rao, C. Srinivas, Sreenivas Pratapagiri, and B. Raghuram. "Secure Energy Aware Optimal Routing using Reinforcement Learning-based Decision-Making with a Hybrid Optimization Algorithm in MANET."

[8] Marinescu, Andrei, Ivana Dusparic, Adam Taylor, Vinny Cahill, and Siobhán Clarke. "P-MARL: Prediction-Based Multi-Agent Reinforcement Learning for Non-Stationary Environments." In AAMAS, pp. 1897-1898. 2015.

[9] Birabwa, Denise Joanitah, Daniel Ramotsoela, and Neco Ventura. "Multi-agent deep reinforcement learning for user association and resource allocation in integrated terrestrial and non-terrestrial networks." Computer Networks 231 (2023): 109827.

[10] Peng, Peixi, Junliang Xing, Lili Cao, Lisen Mu, Chang Huang, and Horizon Robotics. "Learning Deep Decentralized Policy Network by Collective Rewards for Real-Time Combat Game." In IJCAI, pp. 1305-1311. 2019.

[11] Zhang, Yi, and Robert W. Heath. "Reinforcement learning-based joint user scheduling and link configuration in millimeter-wave networks." IEEE Transactions on Wireless Communications 22, no. 5 (2022): 3038-3054.

[12] dos Santos, Roger R., Eduardo K. Viegas, Altair O. Santin, and Vinicius V. Cogo. "Reinforcement learning for intrusion detection: More model longness and fewer updates." IEEE Transactions on Network and Service Management 20, no. 2 (2022): 2040-2055.

[13] Kim, Sunghwan, Seunghyun Yoon, Jin-Hee Cho, Dong Seong Kim, Terrence J. Moore, Frederica Free-Nelson, and Hyuk Lim. "DIVERGENCE: Deep reinforcement learning-based adaptive traffic inspection and moving target defense countermeasure framework." IEEE Transactions on Network and Service Management 19, no. 4 (2022): 4834-4846.

[14] Simpson, Kyle A., Simon Rogers, and Dimitrios P. Pezaros. "Per-host DDoS mitigation by direct-control reinforcement learning." IEEE Transactions on Network and Service Management 17, no. 1 (2019): 103-117.

[15] Yang, Liqun, Jianqiang Li, Liang Yin, Zhonghao Sun, Yufei Zhao, and Zhoujun Li. "Real-time intrusion detection in wireless network: A deep learning-based intelligent mechanism." Ieee Access 8 (2020): 170128-170139.

[16] Balamurugan, Nagaiah Mohanan, Malaiyalathan Adimoolam, Mohammed H. Alsharif, and Peerapong Uthansakul. "A novel method for improved network traffic prediction using enhanced deep reinforcement learning algorithm." Sensors 22, no. 13 (2022): 5006.

[17] Du, Xiuli, Xiaohui Ding, and Fan Tao. "Network security situation prediction based on optimized clock-cycle recurrent neural network for sensor-enabled networks." Sensors 23, no. 13 (2023): 6087.

[18] Owezarski, Philippe. "A near real-time algorithm for autonomous identification and characterization of honeypot attacks." In Proceedings of the 10th ACM symposium on information, computer and communications security, pp. 531-542. 2015.

[19] Shao, Hao, Lunwen Wang, Hui Liu, and Rangang Zhu. "A link prediction method for MANETs based on fast spatio-temporal feature extraction and LSGANs." Scientific Reports 12, no. 1 (2022): 16896.

[20] Ryu, Joonsu, and Sungwook Kim. "Reputation-based opportunistic routing protocol using q-learning for manet attacked by malicious nodes." IEEE Access 11 (2023): 47701-47711.

[21] Lee, Donghoun, Sehyun Tak, and Sari Kim. "Development of reinforcement learning-based traffic predictive route guidance algorithm under uncertain traffic environment." IEEE Access 10 (2022): 58623-58634.

[22] Song, Yuda, and Wen Sun. "Pc-mlp: Model-based reinforcement learning with policy cover guided exploration." In International Conference on Machine Learning, pp. 9801-9811. PMLR, 2021.

[23] Murti, Fahri Wisnu, Samad Ali, George Iosifidis, and Matti Latva-aho. "Deep reinforcement learning for orchestrating cost-aware reconfigurations of vrans." IEEE Transactions on Network and Service Management (2023).

[24] Gaon, Maor, and Ronen Brafman. "Reinforcement learning with non-markovian rewards." In Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 04, pp. 3980-3987. 2020.

[25] Salh, Adeeb, Lukman Audah, Kwang Soon Kim, Saeed Hamood Alsamhi, Mohammed A. Alhartomi, Qazwan Abdullah, Faris A. Almalki, and Haneen Algethami. "Refiner GAN algorithmically enabled deep-RL for guaranteed traffic packets in real-time URLLC B5G communication systems." IEEE Access 10 (2022): 50662-50676.

[26] Gallego, Victor, Roi Naveiro, and David Rios Insua. "Reinforcement learning under threats." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 9939-9940. 2019.

[27] Chandak, Yash, Georgios Theocharous, Shiv Shankar, Martha White, Sridhar Mahadevan, and Philip Thomas. "Optimizing for the future in non-stationary mdps." In International Conference on Machine Learning, pp. 1414-1425. PMLR, 2020.