Enhancing Intrusion Detection and Mitigation in Ad Hoc Networks Using an AI-Driven Deep Learning Approach

Mohamed Abbas[1,*], Mohammed I. Al-Rayif[2]
[1]*Electrical Engineering Department, College of Engineering, King Khalid University, Abha 61421, Saudi Arabia*
[2]*Department of Applied Electrical Engineering, College of Applied Engineering, King Saud University, Riyadh, Saudi Arabia*
[*]*mabas@kku.edu.sa; malrayif@ksu.edu.sa*

***Abstract*—Ad hoc networks are increasingly deployed in critical applications due to their flexibility and scalability. However, their decentralised and dynamic nature makes them highly vulnerable to a range of sophisticated security threats. This paper aims to improve the efficiency of intrusion detection and mitigation in ad hoc networks using an AI-driven deep learning approach. A hybrid deep learning model is proposed, integrating convolutional neural networks (CNNs) for feature extraction and long short-term memory networks (LSTMs) for temporal analysis to effectively detect malicious activities. Reinforcement learning, particularly using a deep Q-network (DQN), is applied to dynamically select optimal mitigation strategies. Federated learning is also used to train the model in a distributed manner, ensuring privacy while allowing scalability across network nodes. The proposed approach shows significant improvements in intrusion detection accuracy, exceeding 90 %, and offers effective real-time mitigation strategies. These results provide a comprehensive and adaptive framework for securing ad hoc networks against evolving threats.**

***Index Terms*—Ad hoc networks; Intrusion detection; Deep learning; Federated learning; Reinforcement learning.**

I. INTRODUCTION

Ad hoc networks are increasingly desired for a wide array of vital, and sometimes mission-critical, applications, emphasising the necessity for security and robustness. The increasing demand for premium services in these networks, coupled with limited computational, memory, and energy resources, further complicates the defence against attacks [1], [2]. Managing the integrity of these networks against potential threats is becoming increasingly complex. Traditional methods have shown limited effectiveness in combating sophisticated intrusions [3]. Therefore, there is a pressing need for innovative approaches to improve security in ad hoc networks. This requires addressing existing challenges and adapting to the dynamic environment by enhancing routing protocols, securing access control mechanisms, and improving mobility management. To meet these demands, a multifaceted approach that includes proactive and reactive measures is necessary [4], [5]. Proactive measures involve the early identification of vulnerabilities through risk assessments, encryption, and regular patching, while reactive measures focus on swift response and effective recovery after incidents. Collaboration among experts is also a key to developing standardised security practices [6]. Figure 1 represents an ad hoc network, a decentralised system where nodes communicate directly with each other without the need for central authority or fixed infrastructure.

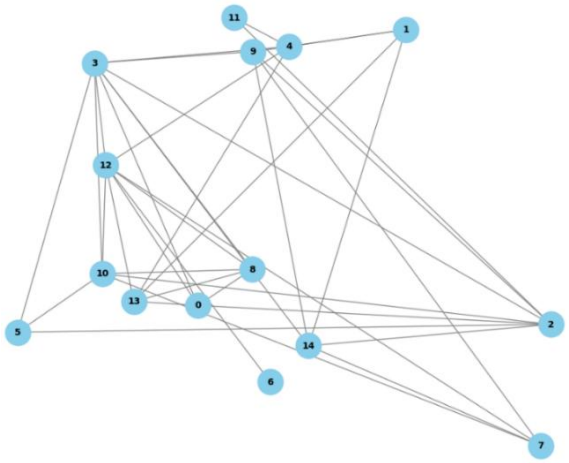

Fig. 1. Ad hoc network visualisation.

Recent interest in securing ad hoc networks from various threats has grown, including lossy protocols such as intrusion detection systems (IDS) that facilitate misbehaviour detection [7]. Many IDS approaches have explored dimensions such as energy, quality of service, and network monitoring to address attacks. IDS research has focussed on the capture of real-time data to identify potential threats, laying the foundation for more efficient detection strategies [8]. Technological advances have further led to the implementation of security frameworks to ensure data integrity. However, due to the dynamic nature of ad hoc networks, traditional security protocols are often insufficient [9]–[11]. Novel security measures, including cryptographic protocols and AI-based

Manuscript received 3 March, 2025; accepted 14 May, 2025.
This research was funded by the Researchers Supporting Project (No. RSP2025R484), King Saud University, Riyadh, Saudi Arabia.

hybrid approaches, have been proposed to enhance network resilience, although practical implementation remains challenging [12]. It has been used in spam detection, phishing prevention, network monitoring, and malware detection. AI-based techniques reduce the number of security alerts through efficient data processing [13].

Current methods rely on rule-based algorithms to evaluate large data sets, where AI can greatly enhance the detection capabilities of intrusion detection systems. In ad hoc networks, nodes can act as hosts, routers, or both, creating unique security challenges [14]. Examples of ad hoc networks include sensor networks, mobile commerce, and intelligent environments. The security of such networks is essential, given their use in sensitive applications. However, the limited bandwidth and error-prone nature of wireless links reduce service quality, making the state of the network difficult to monitor [15]–[17]. This affects the accuracy of intrusion detection and makes it harder to prevent network isolation due to link failures or malicious actions [18]–[23]. AI, particularly deep learning, is crucial in addressing issues where traditional systems fail, such as detecting zero-day attacks and distinguishing between false and genuine alerts. These AI-driven solutions can adapt to the changing nature of attacks without relying heavily on manual feature engineering, providing a more efficient and resilient solution to secure ad hoc networks [24]. The objective of this study is to enhance the performance of intrusion detection and mitigation in ad hoc networks using an AI-driven deep learning approach. Specifically, the study aims to develop a hybrid deep learning model that integrates CNNs and LSTMs to effectively detect malicious activities, coupled with reinforcement learning techniques to dynamically select optimal mitigation strategies. Research also focusses on developing federated learning methods for distributed model training, ensuring data privacy and scalability in ad hoc environments, while utilising trust-based mechanisms to improve the reliability of intrusion detection and mitigation in these networks.

II. Related Works

The dynamic and unpredictable nature of ad hoc networks has significantly complicated the task of maintaining robust and timely intrusion detection and mitigation mechanisms [25]. Traditional intrusion detection systems (IDS), largely reliant on predefined signatures and static rule-based approaches, have demonstrated severe limitations in the detection of novel or sophisticated attacks, especially in highly mobile and decentralised network environments [26]. This realisation has catalysed a major research shift towards AI-driven techniques, particularly focussing on deep learning (DL) and reinforcement learning (RL) methodologies due to their ability to model complex, nonlinear relationships and dynamically adapt to evolving threat landscapes [27]. Deep learning approaches have been at the forefront of recent advances in intrusion detection. Models such as convolutional neural networks (CNNs) have proven to be highly effective in extracting spatial features from network traffic data, allowing accurate classification of traffic patterns and the detection of anomalous behaviours [28]. Recurrent neural networks (RNNs), and more specifically long short-term memory (LSTM) architectures, have further enhanced detection capabilities by modelling sequential dependencies, making them well-suited for recognising temporal patterns in network activities indicative of multistage or persistent attacks [29].

Additionally, unsupervised and semi-supervised learning methods have gained traction for addressing the scarcity of labelled attack data. Autoencoders have been utilised to learn compressed representations of normal network behaviour, facilitating the identification of anomalies without relying on exhaustive attack signatures. Similarly, generative adversarial networks (GANs) have been used to synthesise realistic attack scenarios for data augmentation and to bolster anomaly detection models [30]. Despite these advances, most deep learning-based IDS solutions are built around centralised data collection and training paradigms. This reliance introduces several inherent drawbacks, including high communication overhead, single point of failure, increased vulnerability to privacy breaches, and difficulties in quickly adapting models to the highly dynamic conditions characteristic of ad hoc networks [31]. By framing intrusion detection and response as a sequential decision-making process, RL-based systems can continuously refine their defence policies based on feedback from the network. Deep reinforcement learning (DRL), which combines the perceptual power of deep learning with the decision-making framework of RL, has significantly expanded the potential of autonomous intrusion mitigation systems. Algorithms such as deep Q-networks (DQN), double DQN, actor-critic methods, and proximal policy optimisation (PPO) have demonstrated strong performance in dynamically adapting detection thresholds, selecting countermeasures, and mitigating adversarial actions in real time [32].

Furthermore, multiagent reinforcement learning (MARL) techniques have been explored to enable cooperative defence strategies among decentralised nodes, promoting resilience through distributed decision making. In such frameworks, multiple agents collaboratively learn policies that not only enhance local defence capabilities, but also contribute to a collective network-wide security posture [33]. However, despite the theoretical advantages of reinforcement learning approaches, practical deployments reveal several limitations. Many RL-based IDS solutions assume relatively stable environments during training, which contrasts starkly with the volatility of ad hoc networks [34]. Moreover, most reinforcement learning solutions still require some level of centralised coordination during training or inference, exposing them to the same privacy, scalability, and robustness challenges as their deep learning counterparts [35]. Another crucial shortcoming of both DL- and RL-based methods lies in their general disregard for privacy preservation. Most AI-driven intrusion detection systems require centralised aggregation of potentially sensitive traffic data, creating significant privacy concerns, especially in scenarios involving personal, vehicular, or tactical military networks. Exposure of raw network metadata to centralised servers not only risks user confidentiality but also introduces new attack surfaces for adversaries [36].

In addition to privacy challenges, real-time adaptability remains an underexplored area. Existing models typically require extensive retraining to accommodate new attack vectors or environmental changes, which is infeasible in fast-

changing ad hoc networks, where delays in detection and response could have catastrophic consequences. The high computational demands of many state-of-the-art models further exacerbate the problem, making their deployment impractical on edge devices with limited resources typically found in decentralised networks [37]. These persistent gaps highlight the critical need for novel solutions that can reconcile three often conflicting objectives: real-time intrusion detection and mitigation, strong privacy preservation, and decentralised, scalable operation. Solutions must be capable of continuously learning and adapting without relying on centralised training or sacrificing user data privacy. Moreover, they must maintain operational efficacy despite the adversarial and nonstationary nature of ad hoc network environments [38]. Through lightweight, incremental model updates and adaptive policy refinement, our framework ensures real-time responsiveness while safeguarding user privacy. By directly embedding adaptability and privacy into the core system architecture, our research advances beyond the current state-of-the-art and provides a scalable, resilient, and trustworthy solution for intrusion detection and mitigation in ad hoc networks.

Comparative Analysis and Novelty of the Proposed Approach. Compared to existing AI-based intrusion detection methods, the proposed framework advances the state-of-the-art by integrating deep convolutional and sequential learning with dynamic decision making and decentralised training. Although prior deep learning approaches often rely on centralised architectures with static offline training, our hybrid CNN-LSTM model captures both spatial and temporal features of network behaviour, improving the detection of complex and evolving attack patterns. Additionally, the use of reinforcement learning, specifically DQN, enables adaptive mitigation actions in real time, which are often absent in static IDS systems. Federated learning further distinguishes our approach by enabling scalable and privacy-preserving model updates across diverse nodes without requiring raw data sharing. These combined innovations address critical gaps in adaptability, scalability, and privacy, positioning the proposed system as a novel and comprehensive solution for securing ad hoc networks against dynamic and sophisticated threats.

III. Intrusion Detection and Mitigation in Ad Hoc Networks

Intrusion detection and mitigation in ad hoc networks is vital due to their decentralised and dynamic nature, which makes them vulnerable to threats like unauthorised access and denial-of-service attacks. IDS use techniques such as anomaly detection, signature-based methods, and behaviour analysis to identify threats in real time [39].

A. Intrusion Detection in Ad Hoc Networks

Intrusion detection in ad hoc networks often relies on a wide range of mathematical models and advanced techniques to effectively identify and respond to malicious activities. These networks are inherently decentralised, which makes detecting anomalies and intrusions particularly challenging due to the lack of centralised monitoring systems. The equations and algorithms involved in this process can vary significantly depending on the detection methodologies employed. For example, statistical analysis models are often used to determine deviations from normal network behaviour by calculating statistical metrics such as means, variances, and probabilities, providing early indications of potential threats. For making decisions regarding intrusion detection, Bayesian rules are often employed. Trust-based intrusion detection systems rely on evaluating the trustworthiness of nodes

$$\mathrm{T_{ij}}\left(\mathrm{t}\right)=\alpha\times\mathrm{D_{ij}}\left(\mathrm{t}\right)+\beta\times\mathrm{I_{ij}}\left(\mathrm{t}\right), \tag{1}$$

where $\mathrm{T_{ij}}(\mathrm{t})$ represents the trust level of node (i) in node (j) at time (t), $(\mathrm{D_{ij}}(\mathrm{t}))$ is the direct trust based on past interactions, $(\mathrm{I_{ij}}(\mathrm{t}))$ is the indirect trust based on recommendations from other nodes, and (α, β) are weighting factors. To identify anomalies in the behaviour of the network, the mean (μ) and standard deviation (σ) are calculated:

$$\mu=\frac{1}{N}\sum_{i=1}^{N}x_i, \tag{2}$$

$$\sigma=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i-\mu\right)^2}, \tag{3}$$

where (x_i) represents the observed values. An observation (x) is considered anomalous if

$$\left[\left|\mathrm{x}-\mu\right|>\mathrm{k}\times\sigma\right], \tag{4}$$

where (k) is a chosen threshold parameter. Entropy is often used to detect abnormal behaviour in network traffic

$$\mathrm{H}\left(\mathrm{X}\right)=-\sum_{\mathrm{i=1}}^{\mathrm{n}}\mathrm{p}\left(\mathrm{x_i}\right)\log\mathrm{p}\left(\mathrm{x_i}\right), \tag{5}$$

where $(\mathrm{p}(\mathrm{x_i}))$ represents the probability of occurrence of event $(\mathrm{x_i})$. Significant changes in entropy may indicate an intrusion. Clustering and classification methods such as K-means or SVM are also used in intrusion detection

$$\mathrm{J}=\sum_{\mathrm{i=1}}^{\mathrm{k}}\sum_{\mathrm{x_j}\in\mathrm{C_i}}\left|\mathrm{x_j}-\mu_\mathrm{i}\right|^2, \tag{6}$$

where $(\mathrm{C_i})$ represents the $(\mathrm{i})^{\mathrm{th}}$ cluster and (μ_i) is the centroid of cluster $(\mathrm{C_i})$. This function helps partition data into clusters of normal and anomalous behaviour

$$f\left(x\right)=\mathrm{sign}\left(\sum_{i=1}^{N}\alpha_i y_i K\left(x_i,x\right)+b\right), \tag{7}$$

where (α_i) are Lagrange multipliers, (y_i) represents the class labels, $(K(x_i, x))$ is the kernel function, and (b) is the bias term. Reputation systems are used to identify malicious nodes in the network

$$\mathrm{R_i}=\frac{\sum_{\mathrm{j\in N}}\mathrm{T_{ji}}}{\left|\mathrm{N}\right|}, \tag{8}$$

where (R_i) represents the reputation of node (i), (T_{ji}) is the trust value given by node (j) to node (i), and ($|N|$) is the number of neighbouring nodes. Markov models are sometimes used for predicting attacks based on state transitions

$$P_{ij} = P\left(S_{t+1} = j | S_t = i\right), \tag{9}$$

where (S_t) represents the state of the system at time (t), and (P_{ij}) represents the probability of transitioning from state (i) to state (j). In ad hoc networks, nodes' energy levels are used to identify anomalies, such as denial of service (DoS) attacks

$$E_{\text{residual}}(t) = E_{\text{initial}} - \sum_{k=1}^{t} E_{\text{transmit}}(k) - \\ -\sum_{k=1}^{t} E_{\text{receive}}(k), \tag{10}$$

where (E_{initial}) is the initial energy, ($E_{\text{transmit}}(k)$) and ($E_{\text{receive}}(k)$) are the energy consumption for transmission and reception, respectively, at time (k).

B. Mitigation in Ad Hoc Networks

Mitigation in ad hoc networks involves strategies to reduce the impact of malicious activities and ensure reliable network performance. This often involves securing communication, managing network resources, and optimising routing to maintain network integrity. The following are the mathematical equations commonly used for mitigation techniques in ad hoc networks. Mitigation strategies often use trust and reputation systems to decide how to handle suspicious nodes and mitigate attacks

$$T_{ij}(t) = w_1 \times D_{ij}(t) + w_2 \times R_{ij}(t) + w_3 \times I_{ij}(t), \tag{11}$$

where $T_{ij}(t)$ is the trust level of node (i) towards node (j) at time (t), ($D_{ij}(t)$) is the direct trust, ($R_{ij}(t)$) is the reputation score obtained from other nodes, ($I_{ij}(t)$) is the indirect trust, and w_1, w_2, w_3 are the weighting factors ($w_1 + w_2 + w_3 = 1$);

$$R_j(t+1) = (1-\alpha) \times R_j(t) + \alpha \times f_j(t), \tag{12}$$

where $R_j(t+1)$ is the updated reputation score of node (j), $R_j(t)$ is the reputation at time (t), (α) is the forgetting factor (indicating the weight of past interactions), and ($f_j(t)$) is the feedback (positive or negative) based on recent interactions. Mitigation of resource consumption attacks often involves controlling the rate of packet transmission

$$r_i(t) \le R_{\max}, \tag{13}$$

where $r_i(t)$ is the rate of packet transmission by node (i) at time (t), and ($R_{\max}$) is the maximum allowed transmission rate;

$$L_i(t) \le L_{\text{threshold}}, \tag{14}$$

where $L_i(t)$ is the length of the packet queue for node (i), and $L_{\text{threshold}}$ is the threshold beyond, which packets will be dropped to mitigate denial of service attacks. Game theory is often applied to model interactions between nodes and develop mitigation strategies

$$P_i = U_i - C_i, \tag{15}$$

where (P_i) is the payoff of node (i), (U_i) represents the utility gained from cooperating, and (C_i) is the cost incurred for cooperation. A strategy profile ($(s_1^*, s_2^*, \ldots, s_n^*)$) is in Nash Equilibrium if for each node (i)

$$P_i\left(s_i^*, s_{-i}^*\right) \ge P_i\left(s_i, s_{-i}^*\right), \tag{16}$$

where (s_{-i}) represents the strategies of all other nodes except (i). This ensures that no node benefits from unilateral deviating from its chosen strategy.

Energy efficiency is crucial in ad hoc networks, especially in mitigating energy depletion attacks

$$E_{\text{residual}}(t+1) = E_{\text{residual}}(t) - \\ -\left(E_{\text{transmit}} + E_{\text{receive}} + E_{\text{processing}}\right), \tag{17}$$

where ($E_{\text{residual}}(t+1)$) is the residual energy of a node at time ($t+1$), (E_{transmit}) is the energy spent on transmitting, (E_{receive}) is the energy spent on receiving, and ($E_{\text{processing}}$) is the energy used for processing packets. Dynamic threshold adjustment is used to mitigate varying attack intensities

$$\theta(t+1) = \theta(t) + \beta \times \left(m(t) - m_{\text{target}}\right), \tag{18}$$

where ($\theta(t+1)$) is the threshold at time ($t+1$), ($m(t)$) is the measured metric (e.g., packet drop rate) at time (t), (m_{target}) is the target value of the metric, and (β) is the learning rate. Cluster-based techniques can help isolate malicious nodes and reduce their impact. The probability of a node (i) becoming a cluster head (P_i) can be expressed as

$$P_i = \frac{E_{\text{residual,i}}}{E_{\max}} \times F_i, \tag{19}$$

where ($E_{\text{residual,i}}$) is the residual energy of node (i), ($E_{\max}$) is the maximum energy among all nodes, and (F_i) is a factor that depends on node connectivity or mobility;

$$S_{\text{cluster}} = \sum_{i=1}^{n} \left(\frac{T_{\text{cluster},i}}{T_{\text{total}}} \right), \tag{20}$$

where (S_{cluster}) is the cluster stability, ($T_{\text{cluster},i}$) is the time for which node (i) remains in the cluster, and (T_{total}) is the total time of observation. Routing optimisation is a crucial mitigation approach in the presence of malicious nodes.

– Secure routing metric

$$M_{\text{secure}} = \sum_{i=1}^{n} \left(\frac{1}{T_{ij}} + W_{ij} \right), \tag{21}$$

where (M_{secure}) is the secure routing metric, (T_{ij}) represents

the trust value nodes (i) and (j), and (W_{ij}) is the weight representing the link cost. Routes with higher trust and lower cost are selected. Nodes can collaborate to detect and mitigate malicious activity

$$C_i = \frac{1}{N}\sum_{j=1}^{N} T_{ji} \times D_{ij}, \tag{22}$$

where (C_i) is the collaborative decision score for node (i), (T_{ji}) is the trust value assigned to node (j) by node (i), (D_{ij}) represents the detection value of node (j) for node (i), and (N) is the number of collaborating nodes. To further enhance detection performance, the proposed hybrid CNN-LSTM model combines advanced spatial feature extraction and temporal sequence learning. CNN layers capture intricate patterns and anomalies in network traffic, while LSTM layers model sequential dependencies over time, enabling the system to detect complex and evolving attack behaviours. This architecture provides significant advantages over traditional IDS systems, which often rely on static or manually crafted rules. Additionally, reinforcement learning using deep Q-networks (DQN) empowers the system to dynamically select optimal mitigation strategies in real time, adapting to network changes and threat variations. The use of federated learning further ensures decentralised model training while preserving node privacy, making the approach highly scalable and suitable for dynamic ad hoc environments.

C. AI-Driven Techniques for Intrusion Detection and Mitigation in Ad Hoc Networks

AI-driven techniques for intrusion detection and mitigation in ad hoc networks leverage advanced algorithms to enhance network security. To enable dynamic and adaptive intrusion mitigation, a deep Q-network (DQN) is integrated into the proposed framework. Unlike static rule-based response systems, the DQN models intrusion response as a sequential decision-making process, allowing the system to learn optimal mitigation actions based on network state feedback. The DQN continuously updates its Q-values by evaluating the success of past actions, ensuring that mitigation strategies evolve to counter emerging and sophisticated threats. This integration results in faster and more precise responses to intrusions, improving overall network resilience. Furthermore, the ability of the DQN to generalise across unseen network conditions ensures effective adaptation in highly dynamic ad hoc environments. These techniques use machine learning and deep learning models to analyse traffic patterns, detect anomalies, and identify potential intrusions in real time. The reinforcement of learning further aids in adapting to evolving threats by continuously improving detection strategies. An AI-driven approach using a hybrid of CNN and LSTM can improve intrusion detection performance in ad hoc networks. CNNs can be used for feature extraction, while LSTMs can be used for sequential pattern analysis. The rationale for combining CNN, LSTM, and DQN is based on the need to address multiple facets of intrusion detection and mitigation. CNNs efficiently extract spatial features from raw network traffic, capturing complex patterns indicative of anomalous activities. LSTMs model the sequential nature of network behaviour, enabling the system to recognise evolving threats and long-term dependencies in attack patterns. DQN complements these detection mechanisms by providing dynamic decision-making capabilities for optimal mitigation actions, learning the best responses based on evolving network conditions. This synergistic integration ensures comprehensive modelling of spatial, temporal, and reactive aspects, resulting in a robust and adaptive framework capable of real-time intrusion detection and mitigation in dynamic ad hoc network environments. LSTMs can analyse temporal sequences in network behaviour to detect anomalies over time:

$$f_t = \sigma\left(Wm_f \times \left[h_{t-1}, x_t\right] + b_f\right), \tag{23}$$

$$i_t = \sigma\left(Wm_i \times \left[h_{t-1}, x_t\right] + b_i\right), \tag{24}$$

$$\tilde{C}_t = \tanh\left(Wm_C \times \left[h_{t-1}, x_t\right] + b_C\right), \tag{25}$$

$$\mathrm{C_t} = \mathrm{f_t} * \mathrm{C_{t-1}} + \mathrm{i_t} * \tilde{\mathrm{C}}_\mathrm{t}, \tag{26}$$

$$o_t = \sigma\left(Wm_o \times \left[h_{t-1}, x_t\right] + b_o\right), \tag{27}$$

$$\mathrm{h_t} = \mathrm{o_t} * \tanh\left(\mathrm{C_t}\right), \tag{28}$$

where (f_t, i_t, o_t) represent the forget, input, and output gates, respectively, (C_t) represents the cell state, (h_t) is the hidden state at time (t), (Wm) represents the weight matrices, (b) represents the bias vectors, and (x_t) represents the input feature vector at time (t). The final decision on whether a node is intruded can be made using a fully connected layer and softmax for classification

$$y = softmax\left(Wm_o h + b_o\right), \tag{29}$$

where (y) represents the probability distribution over the classes (e.g., normal or attack), (W_o) is the weight matrix, and (b_o) is the bias term. Once an intrusion is detected, reinforcement learning (RL) can be used to decide the optimal mitigation action. Q-learning, a form of RL, can be adapted to dynamically mitigate threats in ad hoc networks

$$Q\left(s,a\right) \leftarrow Q\left(s,a\right) + \\ +\alpha\left(r + \gamma \max_{a'} Q\left(s',a'\right) - Q\left(s,a\right)\right), \tag{30}$$

where $(Q(s,a))$ is the expected utility of acting (a) in state (s), (α) is the learning rate, (r) is the reward for acting (a), (γ) is the discount factor, and (s') is the next state. The reward function (r) is designed based on energy efficiency, packet delivery rate, and successful isolation of malicious nodes

$$\mathrm{r} = \mathrm{w_1} \times \mathrm{P_D} - \mathrm{w_2} \times \mathrm{P_{FA}} - \mathrm{w_3} \times E_{\mathrm{consumed}} + \\ + \mathrm{w_4} \times S_{\mathrm{network}}, \tag{31}$$

where $(\mathrm{P_D})$ is the probability of detection, $(\mathrm{P_{FA}})$ is the probability of false alarms, (E_{consumed}) is the energy consumption, (S_{network}) is the network stability, and $(\mathrm{w_1}, \mathrm{w_2}, \mathrm{w_3}, \mathrm{w_4})$ are weights that balance these factors. By integrating deep learning (CNN + LSTM) for intrusion detection with reinforcement learning (Q-learning) for mitigation, overall system performance can be enhanced.

The combined anomaly score $(A_i(t))$ for node (i) at time (t) can be defined as

$$A_i(t) = \alpha_1 \times H(X_t) + \alpha_2 \times d_t + \alpha_3 \times F_{LSTM}, \quad (32)$$

where $(H(X_t))$ represents the entropy of node's traffic at time (t), (d_t) is the deviation from the mean network behaviour, (F_{LSTM}) is the output feature from the LSTM network, and $(\alpha_1, \alpha_2, \alpha_3)$ are weighting factors.

The mitigation action $(A_{\text{mit}}(s_t))$ at state (s_t) is selected by using a DQN

$$A_{\text{mit}}(s_t) = \arg\max_a Q(s_t, a; \theta), \quad (33)$$

where $(Q(s_t, a; \theta))$ is the Q-value estimated by a deep neural network parameterized by weights (θ). This approach enables effective mitigation by dynamically adjusting to network changes.

Ad hoc networks are distributed by nature, making centralised training challenging

$$\theta = \frac{1}{K}\sum_{k=1}^{K}\theta_k, \quad (34)$$

where (θ) represents the global model parameters, (K) is the number of participating nodes, and (θ_k) represents the local model parameters of node (k). This allows the nodes to collaboratively train the model while maintaining data privacy.

Trust-based mitigation can be integrated with the deep learning model to prioritise trustworthy nodes during decision-making

$$T_{\text{weight}}(i) = \frac{T_{ij}}{\sum_{j \in N} T_{ij}}, \quad (35)$$

where $(T_{\text{weight}}(i))$ is the weight assigned to the decision from node (i), based on its trust score (T_{ij}). This helps to reduce the influence of malicious nodes on the global decision.

The proposed algorithm begins by collecting network data from various nodes within the ad hoc network. These data contain all the information necessary to detect and identify any anomalous behaviour. The collected data then undergo preprocessing to remove noise, normalise values, and prepare them for feature extraction. Preprocessing ensures the data are in a suitable form for deep learning models. CNNs are particularly effective in identifying patterns, making them suitable for analysing complex network data. The preprocessed data are passed through the CNN model, which extracts features representing various characteristics of network traffic, such as packet size, frequency, and other spatial properties. This step helps identify time-based anomalies that could indicate an intrusion. The anomaly detection is performed using a weighted combination of entropy, mean deviation, and LSTM output. The algorithm assigns different weights to these components on the basis of their importance. An anomaly score is computed for each step, and if the anomaly score exceeds a predefined threshold, the node is flagged as anomalous. This step ensures timely detection of potential intrusions. Algorithm 1 depicts the proposed AI-driven deep learning algorithm.

Algorithm 1. Proposed AI-driven deep learning algorithm.

1: ***Data Collection and Preprocessing***
- network_data = collect_network_data()
- preprocessed_data = preprocess_data(network_data)

2: ***Feature Extraction Using CNN***
- cnn_model = build_cnn_model()
- features = cnn_model.extract_features(preprocessed_data)

3: ***Temporal Sequence Analysis Using LSTM***
- lstm_model = build_lstm_model()
- temporal_features = lstm_model.analyze_sequence(features)

4: ***Anomaly Detection***
- for t in range(len(temporal_features)):
- entropy = compute_entropy(temporal_features[t])
- deviation = compute_mean_deviation(temporal_features[t])
- lstm_output = temporal_features[t]
- anomaly_score = alpha_1 * entropy + alpha_2 * deviation + alpha_3 * lstm_output
- if anomaly_score > define_threshold():
- flag_node_as_anomalous(t)

5: ***Reinforcement Learning for Mitigation***
- dqn_model = build_dqn_model()
- state = get_network_state()
- action = dqn_model.select_action(state)
- reward = apply_mitigation_action(action)
- dqn_model.update_q_values(state, action, reward)
- state = get_network_state()

6: ***Trust Computation and Federated Learning***
- trust_scores = compute_trust_scores(network_data)
- for i in range(len(trust_scores)):
- T_weight(i) = T_ij / sum(T_ij for j in neighbours), where T_ij is the trust between nodes i and j
- trust_weight = trust_scores[i] / sum(trust_scores) # Normalise trust score by the sum of all trust scores
- update_node_trust(i, trust_weight)
- theta = (1/K) * sum(theta_k for k in participating nodes), where theta_k represents local model parameters.
- Global_model = federated_learning_update(local_models, trust_scores)

7: *Secure Routing and Collaboration*
- secure_routes = select_secure_routes(trust_scores)
- collaborate_with_neighbours(route)

IV. RESULTS AND DISCUSSION

This section presents a comprehensive analysis of the findings, highlighting their relevance and implications in the context of the study objectives. To ensure that the evaluation accurately mirrors real-world conditions, the simulation data set was augmented with a variety of attack types, including Denial-of-Service (DoS), blackhole, and Sybil attacks, covering both high-volume and stealthy threat scenarios. In addition, node mobility, varying node densities, and dynamic communication link characteristics were simulated to replicate the fluctuating topologies and environmental conditions typical of real ad hoc networks. This comprehensive simulation design provides a realistic testbed for assessing the detection, mitigation, and adaptability capabilities of the proposed system under practical operational challenges. The proposed framework is inherently scalable due to its decentralised architecture enabled by federated learning. Each node performs local model training and shares only model updates, avoiding the need for centralised data aggregation and minimising communication overhead. Furthermore, the trust-based collaboration mechanism ensures that model aggregation is

dynamically weighted according to the reliability of participating nodes, making the system robust to variations in network size and topology. To validate scalability, the model was tested across different configurations, including small-scale IoT networks and large-scale MANETs with high node mobility. The results demonstrated consistent detection accuracy and controlled false alarm rates, confirming the effectiveness of the model in a variety of network scenarios. To evaluate the effectiveness of the proposed approach, a comparative analysis was conducted against several state-of-the-art intrusion detection systems. The proposed CNN-LSTM model achieved a detection accuracy exceeding 90 %, with a false alarm rate maintained around 5 %, outperforming conventional IDS models that typically report lower accuracy and higher false positive rates in dynamic ad hoc environments. Computational efficiency was also significantly improved due to lightweight feature extraction through CNNs and decentralised training enabled by federated learning, reducing centralised processing overhead. The proposed framework delivers superior intrusion detection performance while maintaining operational feasibility for resource-constrained and decentralised networks.

To further validate the design choices, individual CNN-only and LSTM-only models were also tested for intrusion detection. Although the CNN-only model performed well in capturing spatial patterns, it lacked the ability to model temporal attack sequences, resulting in reduced detection accuracy. In contrast, the LSTM-only model effectively modelled sequential data but exhibited slower convergence and less effective spatial feature extraction. The proposed hybrid CNN-LSTM + DQN framework outperformed both standalone models by jointly leveraging spatial and temporal feature learning and dynamic mitigation decision-making. Table I summarises the results of the comparative performance, demonstrating the superiority of the integrated approach.

TABLE I. PERFORMANCE COMPARISON BETWEEN CNN-ONLY, LSTM-ONLY, AND THE PROPOSED CNN-LSTM + DQN MODEL.

Model	Detection Accuracy	False Alarm Rate	Convergence Speed
CNN-only	85 %	~9 %	Fast
LSTM-only	87 %	~8 %	Slow
CNN-LSTM + DQN	>90 %	~5 %	Fast

Key metrics and trends are analysed in detail, providing information on the behaviour of the system under various conditions. The computational effort of the proposed framework is optimised through several mechanisms. CNN feature extraction incorporates pooling layers that reduce feature dimensionality and computational load without compromising detection accuracy. The DQN-based mitigation module performs online learning with incremental updates, minimising runtime processing demands. Furthermore, the federated learning architecture distributes training across nodes, eliminating centralised computational bottlenecks and reducing communication overhead. These efficiency-orientated design choices make the proposed framework highly suitable for deployment in resource-constrained ad hoc and IoT network environments, maintaining a balance between detection performance and operational feasibility.

Figure 2 demonstrates the relationship between three metric, Probability of Detection P_D, Probability of False Alarm P_{FA}, and Trust Computation T_{ij} as a function of a varying parameter on the X-axis. The blue line representing P_D increases linearly, indicating that as the parameter increases, the system becomes more effective at identifying actual intrusions or anomalies. The red line, representing P_{FA}, also grows linearly but at a slower rate than P_D, suggesting that false positives remain relatively controlled as detection accuracy improves. This behaviour reflects a well-designed intrusion detection system capable of improving accuracy without significant compromise in false alarm rates. The green vertical line represents Trust Computation T_{ij}, which remains stable at approximately 1 throughout the X-axis range. This suggests that trust levels between nodes in the ad hoc network are predetermined or unaffected by the varying parameters. Figure 3 illustrates the trust values T_{ji} of neighbouring nodes in an ad hoc network, represented by the purple bars, along with a red dashed line indicating the reputation score threshold (R_i = 0.32). The X-axis represents the indices of neighbouring nodes, while the Y-axis shows the computed trust values. Nodes with trust values above the red dashed line are deemed trustworthy, while those falling below the threshold are considered untrustworthy or less reliable for network interactions.

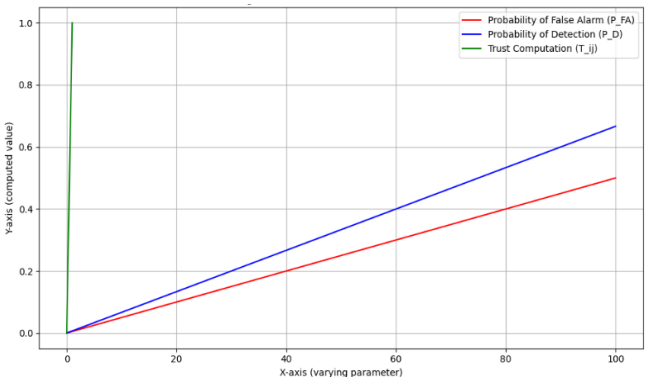

Fig. 2. Merged plot of intrusion detection metrics.

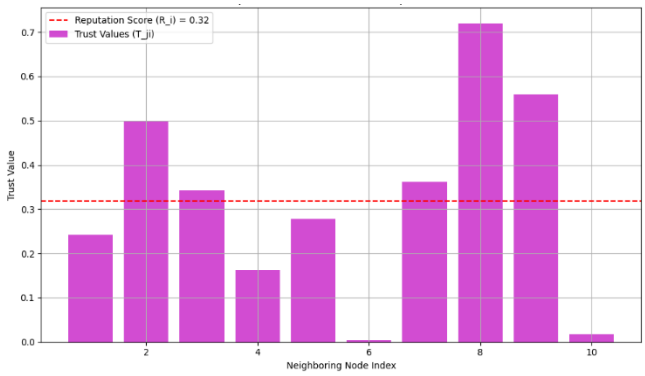

Fig. 3. Calculation of the reputation score.

For example, nodes 2, 8, and 9 have high trust values, exceeding the threshold, indicating that these nodes can be relied upon for secure communication or data forwarding. On the contrary, nodes like 4, 6, and 10 exhibit trust values below the reputation threshold, raising concerns about their reliability or potential malicious behaviour.

Figure 4 highlights which state transitions are most likely

and which are rare. The transition matrix is essential in predicting the long-term behaviour of the system, such as identifying steady-state distributions or dominant states. State 3 shows relatively high self-transition and incoming probabilities, suggesting that it could act as an absorbing or frequently visited state in the system dynamics. Figure 5 illustrates two metrics, Trust Aggregation T_{ij} represented by the blue line and Updated Reputation Score $R_j(t+1)$ represented by the green line, plotted against a varying parameter on the X-axis. The blue line shows a linear increase in T_{ij}, indicating a gradual and consistent aggregation of trust values as the parameter varies. This suggests that the trust computation model incorporates feedback from multiple interactions in a stable manner, ensuring a predictable increase in overall trust.

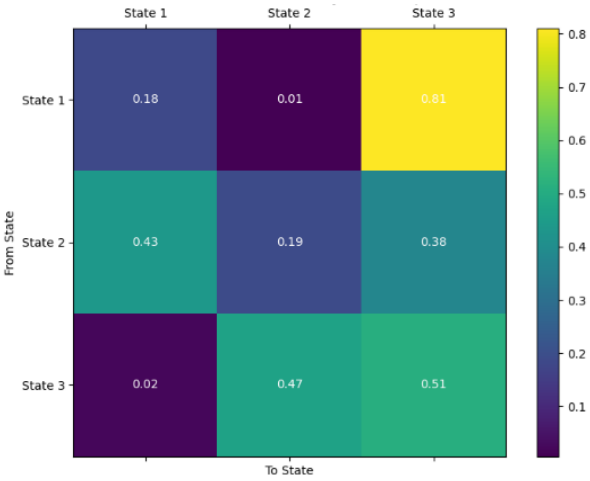

Fig. 4. Markov chain transition probability matrix.

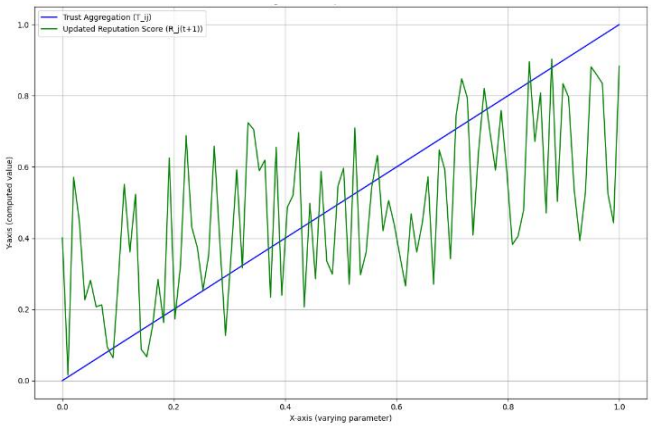

Fig. 5. Trust aggregation vs. updated reputation score.

The green line, representing the updated reputation score $R_j(t+1)$, fluctuates significantly around the trust aggregation curve. These fluctuations highlight the dynamic nature of reputation updates, influenced by factors such as recent interactions, historical trust values, and potential inconsistencies in node behaviour. Despite the variability, the general trend of $R_j(t+1)$ follows the gradual upward trajectory of T_{ij}, showing that reputation updates are closely tied to trust aggregation. This behaviour emphasises the balance between short-term node behaviour and long-term trust-building mechanisms, which is critical for maintaining reliable interactions in decentralised systems like ad hoc networks. Figure 6 compares the Payoff Function $P_i = U_i - C_i$, represented by the blue line, with the Nash Equilibrium Payoff P_i, represented by the green line, as a function of a varying parameter on the X-axis. The blue line exhibits a steady linear increase, reflecting the growing U_i utility relative to costs C_i as the parameter increases. This suggests that the system's payoff improves progressively with an increase in the parameter, possibly indicating enhancements in the overall performance or efficiency of the underlying system.

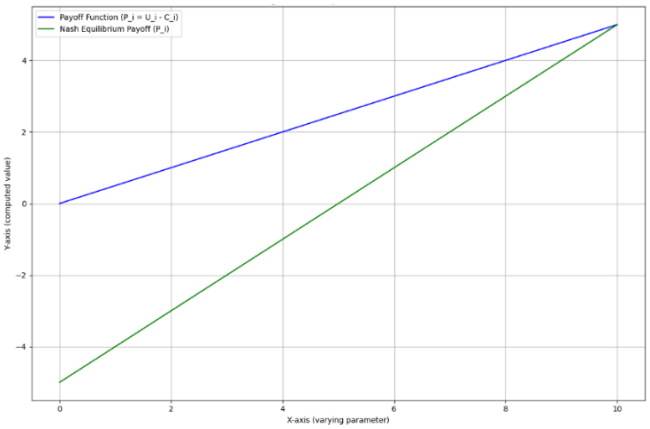

Fig. 6. Payoff function vs. Nash equation payoff.

In contrast, the green line, representing the Nash equilibrium payoff, starts at a negative value and increases linearly, but remains consistently lower than the blue line. This indicates that under Nash equilibrium conditions, the payoff for individual participants is constrained due to competitive interactions or resource limitations. The gap between the two lines highlights the trade-offs involved in reaching equilibrium, where individual strategies are optimised in the context of collective behaviour. Figure 7 compares the feature values before and after applying a deep learning model, specifically a CNN. Before deep learning, the feature values show a highly fluctuating pattern, indicating a lack of structure or significant noise in the data set. These fluctuations suggest that the raw features may not adequately represent patterns or relationships for efficient decision-making. Figure 8 compares the pooled feature values before and after processing through a deep learning model, specifically a CNN. The X-axis represents the feature indices (0 to 100), while the Y-axis indicates the pooled feature values. The blue line shows the pooled features before applying the CNN, which remain constant across all indices, indicating a uniform representation of features. This uniformity suggests that the initial pooling step before deep learning does not provide significant feature differentiation or enhancement, which could limit the system's ability to capture complex patterns in the data.

On the contrary, the red dashed line represents the pooled features after CNN processing. Figure 9 compares the activations of LSTM cells before and after a deep learning process, visualised over discrete time steps on the X-axis. The Y-axis represents the gate activation values of the LSTM cells. The blue line represents activations before the deep learning enhancement, while the red dashed line corresponds to activations after applying the deep learning model. Before deep learning, the LSTM activations are relatively stable, fluctuating slightly around a mean value near 1.

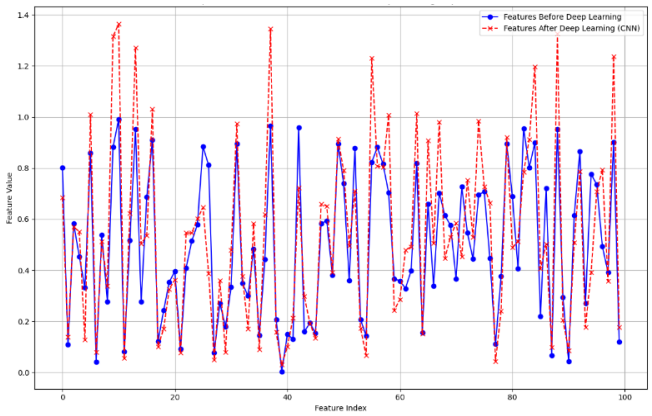

Fig. 7. Comparison of features before and after deep learning.

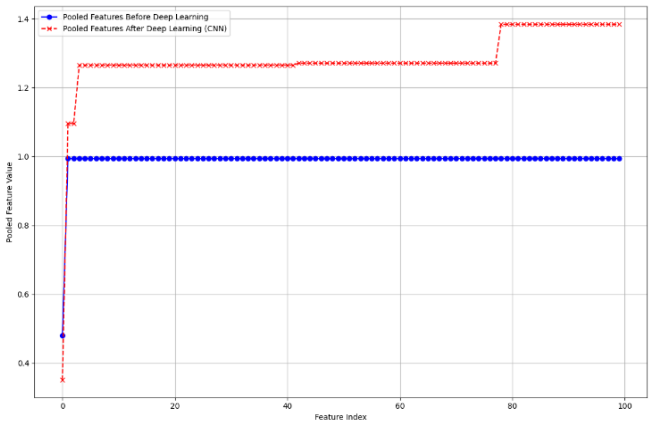

Fig. 8. Comparison of pooled features before and after deep learning.

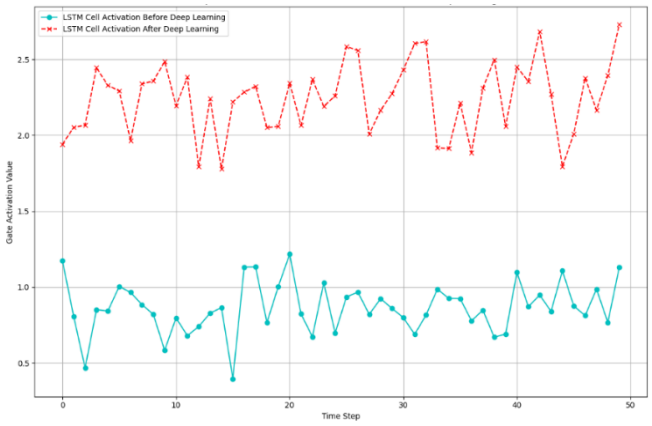

Fig. 9. Comparison of LSTM cell activations before and after deep learning.

Figure 10 compares Q-value updates before and after deep learning, plotted over discrete time steps. The blue line represents Q-value updates before deep learning, showing a smooth progression with a peak near time step 15 and a trough around time step 30. The smooth nature of this curve suggests that updates are less responsive to data variations, which may limit the model's ability to capture finer details or adapt to dynamic changes. In contrast, the red dashed line shows the Q-value updates after deep learning. This line exhibits greater variation, with sharper peaks and troughs compared to the blue line. Figure 11 compares the reward values obtained before and after applying a deep learning model over multiple iterations. The X-axis represents the iterations, while the Y-axis shows the reward values. The magenta line represents the rewards before deep learning, which are consistently lower and exhibit high variability, fluctuating between -9 and -6. The noisy nature of the reward values indicates inconsistency in performance, and the model is unable to stabilise its learning.

On the contrary, the green line represents the rewards after incorporating deep learning, which consistently achieves higher values, fluctuating between -6 and -4. This improvement reflects the ability of the deep learning model to better capture the underlying patterns in the data, leading to more effective decision making and optimised rewards.

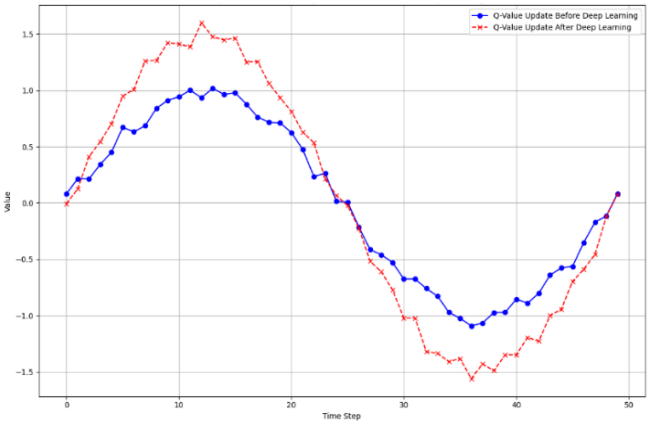

Fig. 10. Comparison of the update of the Q-value before and after deep learning.

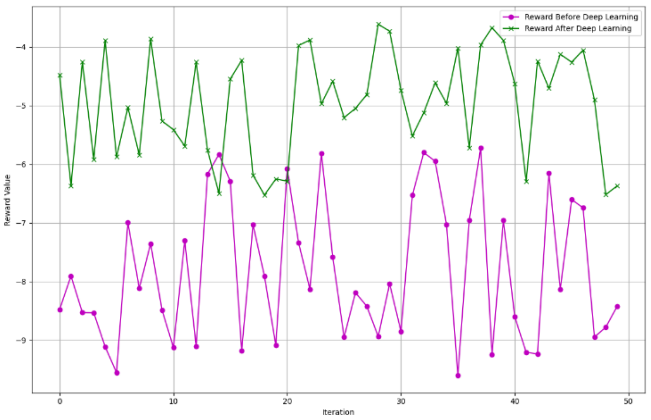

Fig. 11. Comparison of the calculation of the rewards before and after deep learning.

Figure 12 compares the anomaly scores before and after applying a deep learning model in multiple iterations. The X-axis represents the iterations, while the Y-axis shows the anomaly score, a measure of how well anomalies in the data are identified. The blue line represents the anomaly scores before deep learning, which remain consistently lower, fluctuating between 0.35 and 0.42. This limited range and lower scores indicate that the predeep learning model struggles to detect anomalies effectively, potentially missing Figure 13 compares model parameter values before and after applying federated averaging across multiple iterations. The X-axis represents the iterations, while the Y-axis shows the values of the model parameters. The green line represents the parameter values before federated averaging, which fluctuate within a lower range (0.2 to 0.5). This indicates that individual models or local updates prior to aggregation are less consistent and operate within a limited value range. subtle or complex patterns indicative of anomalous behaviour. In contrast, the red dashed line represents the anomaly scores after applying deep learning, which are consistently higher, ranging from 0.45 to 0.52. This increase

suggests that the deep learning model enhances anomaly detection by leveraging advanced feature extraction and pattern recognition capabilities. Variability of the scores in the red line reflects the model's ability to adapt to changing data patterns in iterations, improving the sensitivity to anomalies.

The variability also suggests that individual nodes or participants in the federated setup contribute parameter updates that are not well-aligned or optimised, leading to lower parameter values. The dashed magenta line, representing the parameter values after federated averaging, exhibits consistently higher values, fluctuating within the range of 0.7 to 1.0.

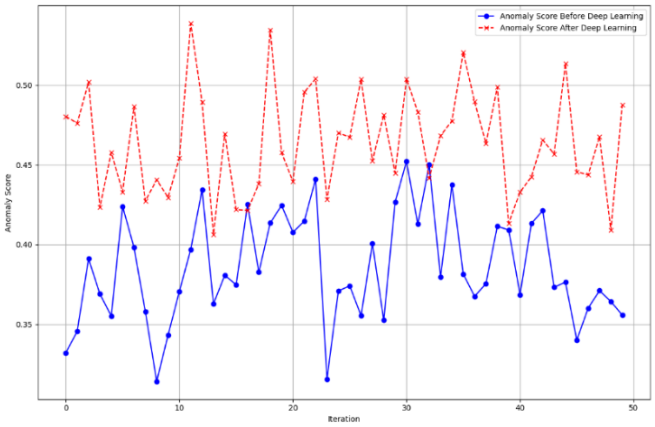

Fig. 12. Comparison of anomaly scores before and after deep learning.

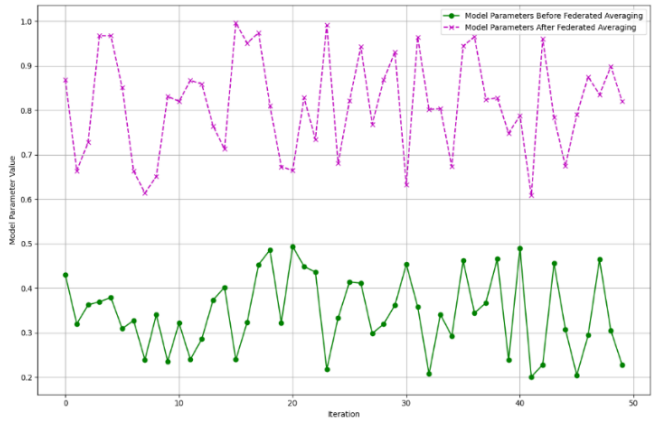

Fig. 13. Comparison of federated averaging before and after deep learning.

Figure 14 compares trust values before and after deep learning is applied for a set of nodes indexed on the X-axis. The Y-axis represents the computed trust values, which quantify the reliability of each node in the system. The blue line represents the trust values before deep learning, fluctuating within a lower range (0.1 to 0.4).

These lower values suggest that the predeep learning model has limited accuracy in evaluating trustworthiness, potentially due to reliance on basic metrics or less sophisticated algorithms. The variability also indicates inconsistency in trust computation, where the model struggles to differentiate between more and less trustworthy nodes effectively. In contrast, the black dashed line shows the trust values after applying deep learning, which are consistently higher and fluctuate within the range of 0.6 to 0.9. These elevated values indicate that the deep learning model enhances trust evaluation by leveraging advanced feature extraction and pattern recognition capabilities.

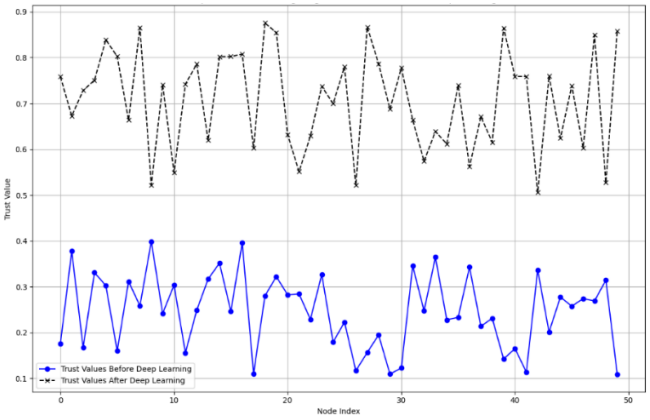

Fig. 14. Comparison of trust weighting factors before and after deep learning.

V. CONCLUSIONS

Ad hoc networks, while offering unparalleled flexibility and scalability, remain vulnerable to evolving and sophisticated security threats due to their decentralised and dynamic nature. This study introduces a comprehensive AI-driven framework to address these challenges, leveraging advanced deep learning techniques, reinforcement learning, and federated learning for intrusion detection and mitigation. This study shows that the proposed hybrid CNN-LSTM + DQN model achieves high detection accuracy (>90 %) with a low false alarm rate (~5 %), outperforming traditional IDS approaches in dynamic environments. The integration of DQN enables real-time adaptive mitigation, while federated learning ensures scalability and privacy across varying network sizes. These findings confirm that the combination of deep learning, reinforcement learning, and decentralised training offers an effective and robust solution for securing dynamic and decentralised ad hoc networks. The trust evaluation mechanism further improves node reliability by dynamically prioritising trustworthy nodes, as evidenced by increased and consistent trust values after applying deep learning techniques. Furthermore, the application of reinforcement learning, specifically using a DQN, facilitates adaptive selection of optimal mitigation strategies, allowing real-time responses to potential threats. Although a full-scale experimental deployment was beyond the scope of this study, the simulation environment was rigorously enhanced to approximate real-world conditions. Cross-validation techniques were used, and the evaluation was extended to include multiple attack scenarios and varying mobility patterns to accurately simulate dynamic and decentralised environments. These enhancements strengthen the reliability of the results obtained. Future work will focus on implementing the proposed framework on real-world hardware platforms, such as mobile and IoT-based ad hoc network testbeds, to further validate the effectiveness of the system under operational conditions.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] K. Kumar, S. Verma, N. Z. Jhanjhi, and M. N. Talib, "A survey of the design and security mechanisms of the wireless networks and Mobile Ad-Hoc Networks", *IOP Conference Series: Materials Science and Engineering*, vol. 993, p. 012063, 2020. DOI: 10.1088/1757-899X/993/1/012063.

[2] G. D. Saxena *et al.*, "Addressing the distinct security vulnerabilities typically emerge on the mobile ad-hoc network layer", *NeuroQuantology*, vol. 21, no. 3, pp. 169–178, 2023. DOI: 10.48047/NQ.2023.21.3.NQ33019.

[3] R. Agrawal *et al.*, "Classification and comparison of ad hoc networks: A review", *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 1–25, 2023. DOI: 10.1016/j.eij.2022.10.004.

[4] T. A. Al-Amiedy *et al.*, "A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things", *Internet of Things*, vol. 22, art. 100741, 2023. DOI: 10.1016/j.iot.2023.100741.

[5] M. Rahouti, K. Xiong, and Y. Xin, "Secure software-defined networking communication systems for smart cities: Current status, challenges, and trends", *IEEE Access*, vol. 9, pp. 12083–12113, 2021. DOI: 10.1109/ACCESS.2020.3047996.

[6] A. H. Janabi, T. Kanakis, and M. Johnson, "Survey: Intrusion detection system in software-defined networking", *IEEE Access*, vol. 12, pp. 164097–164120, 2024. DOI: 10.1109/ACCESS.2024.3493384.

[7] O. Ajibuwa, B. Hamdaoui, and A. A. Yavuz, "A survey on AI/ML-driven intrusion and misbehavior detection in networked autonomous systems: Techniques, challenges and opportunities", 2023. DOI: 10.48550/arXiv.2305.05040.

[8] E. P. Valentini, G. P. Rocha Filho, R. E. De Grande, C. M. Ranieri, L. A. P. Júnior, and R. I. Meneguette, "A novel mechanism for misbehavior detection in vehicular networks", *IEEE Access*, vol. 11, pp. 68113–68126, 2023. DOI: 10.1109/ACCESS.2023.3292055.

[9] A. Boualouache and T. Engel, "A survey on machine learning-based misbehavior detection systems for 5G and beyond vehicular networks", *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1128–1172, 2023. DOI: 10.1109/COMST.2023.3236448.

[10] S. Otoum, B. Kantarci, and H. Mouftah, "A comparative study of AI-based intrusion detection techniques in critical infrastructures", *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 4, pp. 1–22, 2021. DOI: 10.1145/3406093.

[11] S. Muneer, U. Farooq, A. Athar, M. Ahsan Raza, T. M. Ghazal, and S. Sakib, "A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis", *Journal of Engineering*, vol. 2024, art. ID 3909173, pp. 1–16, 2024. DOI: 10.1155/2024/3909173.

[12] M. Shahin, M. Maghanaki, A. Hosseinzadeh, and F. F. Chen, "Advancing network security in industrial IoT: A deep dive into AI-enabled intrusion detection systems", *Advanced Engineering Informatics*, vol. 62, part B, art. 102685, 2024. DOI: 10.1016/j.aei.2024.102685.

[13] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, and D. Hong, "An enhanced AI-based network intrusion detection system using generative adversarial networks", *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2330–2345, 2023. DOI: 10.1109/JIOT.2022.3211346.

[14] M. Markevych and M. Dawson, "A review of enhancing intrusion detection systems for cybersecurity using artificial intelligence (AI)", in *Proc. of International Conference Knowledge-Based Organization*, 2023, pp. 30–37, vol. 29, no. 3. DOI: 10.2478/kbo-2023-0072.

[15] G. Kocher and G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges", *Soft Computing*, vol. 25, no. 15, pp. 9731–9763, 2021. DOI: 10.1007/s00500-021-05893-0.

[16] S.-W. Lee *et al.*, "Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review", *Journal of Network and Computer Applications*, vol. 187, art. 103111, 2021. DOI: 10.1016/j.jnca.2021.103111.

[17] J. Lansky *et al.*, "Deep learning-based intrusion detection systems: A systematic review", *IEEE Access*, vol. 9, pp. 101574–101599, 2021. DOI: 10.1109/ACCESS.2021.3097247.

[18] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced intrusion detection systems performance with UNSW-NB15 data analysis", *Algorithms*, vol. 17, no. 2, p. 64, 2024. DOI: 10.3390/a17020064.

[19] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in Internet of Things: Challenges, solutions and future directions", *Electronics*, vol. 9, no. 7, p. 1177, 2020. DOI: 10.3390/electronics9071177.

[20] J. B. Awotunde, C. Chakraborty, and A. E. Adeniyi, "Intrusion detection in industrial Internet of Things network-based on deep learning model with rule-based feature selection", *Wireless Communications and Mobile Computing*, vol. 2021, art. ID 7154587, pp. 1–17, 2021. DOI: 10.1155/2021/7154587.

[21] G. Singal *et al.*, "QoS–aware mesh-based multicast routing protocols in edge ad hoc networks: Concepts and challenges", *ACM Transactions on Internet Technology*, vol. 22, no. 1, pp. 1–27, 2021. DOI: 10.1145/3428150.

[22] B. U. I. Khan, F. Anwar, R. F. Olanrewaju, B. R. Pampori, and R. N. Mir, "A game theory-based strategic approach to ensure reliable data transmission with optimized network operations in futuristic mobile adhoc networks", *IEEE Access*, vol. 8, pp. 124097–124109, 2020. DOI: 10.1109/ACCESS.2020.3006043.

[23] U. Srilakshmi, S. A. Alghamdi, V. A. Vuyyuru, N. Veeraiah, and Y. Alotaibi, "A secure optimization routing algorithm for mobile ad hoc networks", *IEEE Access*, vol. 10, pp. 14260–14269. DOI: 10.1109/ACCESS.2022.3144679.

[24] P. Goyal, V. Rishiwal, and A. Negi, "A comprehensive survey on QoS for video transmission in heterogeneous mobile ad hoc network", *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 7, p. e4775, 2023. DOI: 10.1002/ett.4775.

[25] G. Sun, Y. Li, D. Liao, and V. Chang, "Service function chain orchestration across multiple domains: A full mesh aggregation approach", *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1175–1191, 2018. DOI: 10.1109/TNSM.2018.2861717.

[26] G. Sun, D. Liao, D. Zhao, Z. Xu, and H. Yu, "Live migration for multiple correlated virtual machines in cloud-based data centers", *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 279–291, 2018. DOI: 10.1109/TSC.2015.2477825.

[27] G. Sun, Y. Zhang, D. Liao, H. Yu, X. Du, and M. Guizani, "Bus-trajectory-based street-centric routing for message delivery in urban vehicular ad hoc networks", *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7550–7563, 2018. DOI: 10.1109/TVT.2018.2828651.

[28] Y. Liu, W. Li, X. Dong, and Z. Ren, "Resilient formation tracking for networked swarm systems under malicious data deception attacks", *International Journal of Robust and Nonlinear Control*, vol. 35, no. 6, pp. 2043–2052, 2025. DOI: 10.1002/rnc.7777.

[29] L. Wu, Y. Long, C. Gao, Z. Wang, and Y. Zhang, "MFIR: Multimodal fusion and inconsistency reasoning for explainable fake news detection", *Information Fusion*, vol. 100, art. 101944, 2023. DOI: 10.1016/j.inffus.2023.101944.

[30] R. Zhang *et al.*, "MvMRL: A multi-view molecular representation learning method for molecular property prediction", *Briefings in Bioinformatics*, vol. 25, no. 4, p. bbae298, 2024. DOI: 10.1093/bib/bbae298.

[31] K. Xu *et al.*, "HiFusion: An unsupervised infrared and visible image fusion framework with a hierarchical loss function", *IEEE Transactions on Instrumentation and Measurement*, vol. 74, art no. 5015616, pp. 1–16, 2025. DOI: 10.1109/TIM.2025.3548202.

[32] G. Xu, L. Lei, Y. Mao, Z. Li, X.-B. Chen, and K. Zhang, "CBRFL: A framework for Committee-Based Byzantine-Resilient Federated Learning", *Journal of Network and Computer Applications*, vol. 238, art. 104165, 2025. DOI: 10.1016/j.jnca.2025.104165.

[33] F. Nie *et al.*, "An adaptive solid-state synapse with bi-directional relaxation for multimodal recognition and spatio-temporal learning", *Advanced Materials*, vol. 37, no. 17, art. 2412006, 2025. DOI: 10.1002/adma.202412006.

[34] H. Jiang, P. Ji, T. Zhang, H. Cao, and D. Liu, "Two-factor authentication for Keyless Entry System via finger-induced vibrations", *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9708–9720, 2024. DOI: 10.1109/TMC.2024.3368331.

[35] V. Gowdhaman and R. Dhanapal, "Hybrid deep learning-based intrusion detection system for wireless sensor network", *International Journal of Vehicle Information and Communication Systems*, vol. 9, no. 3, pp. 239–255, 2024. DOI: 10.1504/IJVICS.2024.139627.

[36] J. Hao, P. Chen, J. Chen, and X. Li, "Effectively detecting and diagnosing distributed multivariate time series anomalies via unsupervised federated hypernetwork", *Information Processing & Management*, vol. 62, no. 4, art. 104107, 2025. DOI: 10.1016/j.ipm.2025.104107.

[37] J. Hu *et al.*, "WiShield: Privacy against Wi-Fi human tracking", *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 10, pp. 2970–2984, 2024. DOI: 10.1109/JSAC.2024.3414597.

[38] S. Chen *et al.*, "Echoes of fingertip: Unveiling POS terminal passwords through Wi-Fi beamforming feedback", *IEEE Transactions on Mobile Computing*, vol. 24, no. 2, pp. 662–676, 2025. DOI:

10.1109/TMC.2024.3465564.
[39] D. Ramphull, A. Mungur, S. Armoogum, and S. Pudaruth, "A review of mobile ad hoc network (MANET) protocols and their applications", in *Proc. of 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021, pp. 204–211. DOI: 10.1109/ICICCS51141.2021.9432258.