Babylonian Journal of Networking
Vol.2025, **pp**. 154–163
DOI: https://doi.org/10.58496/BJN/2025/014; ISSN: 3006-5372
https://mesopotamian.press/journals/index.php/BJN

Research Article

Recognizing Node Intrusion Tendencies in IoT Environments via Deep Learning and Network-Level Feature Analysis

Ioannis Adamopoulos [1,2,*], Aida Vafae Eslahi [3], Harshit Mishra [4], Niki Syrou [5], Tirus Muya [6]

[1] *Hellenic Open University, School of Social Science, Public Health and Policies, Patra, Greece.*

[2] *Department of Public Health Policy, Sector of Occupational & Environmental Health,University of West Attica, Athens, Greece.*

[3] *Medical Microbiology Research Center, Qazvin University of Medical Sciences, Qazvin, Iran.*

[4] *Department of Agricultural Economics,Acharya Narendra Deva University of Agriculture and Technology, Kumarganj, India.*

[5] *Department of Physical Education and Sport Science, University of Thessaly, Karies, Trikala, Greece.*

[6] *Department of computer science, Murang'a University of Technology, Kenya.*

ARTICLE INFO

Article History

Received 17 Sep. 2025
Revised 15 Oct. 2025
Accepted 19 Nov. 2025
Published 13 Dec. 2025

Keywords
IoT Security,
Node Intrusion Detection,
Deep Learning,
Network-Level Feature Analysis,
CNN,
FFNN,
CBPNN,
DoS,
DDoS.

ABSTRACT

The IoT environment is becoming more and more susceptible to intrusion attacks due to its decentralized infrastructure, fewer security boundaries and heterogeneity and dynamism of the network. Discovering malicious parties in these environments is a challenging task, mainly due to the mobility of the nodes and the intermittent contact between them. This study introduces a deep learning-based approach to detect intrusion tendencies on the node level based on the analysis of essential network-level features. The link duration, self-healing latency and number of packets that potential attacker nodes receive were obtained from an analytic network profiling model. These were then employed in the training and testing of the three deep learning models: Feedforward Neural Network (FFNN), Cascade Backpropagation Neural Network (CBPNN), and Convolutional Neural Network (CNN). Of those, the CNN showed the best performance, obtaining intrusion detection accuracy of 85.5%. The proposed approach emphasizes the importance of combining network behavior analytics with deep learning technologies to increase the security of IoT environments.

1. INTRODUCTION

Nowadays, computer and IoT networks face more and more operation challenges and those coming from the security aspect are extremely important. Intrusion attacks such as DoS and DDoS are launched by the malicious nodes that pretend to be legitimate nodes while participating in the network. These attacker nodes have possibility to receive packets from other nodes and by dropping them or forwarding them to destinations erroneously, the large latency is added, throughput is decreased and even network may be failed. "Silent intrusions" or stealthy attacks are difficult to detect and block [1].

The spatial and temporal uncertainties of node mobility in the Mobile Ad Hoc Network (MANET) and IoT systems and node limited transmission ranges make the way for the intrusion detection more complex. Adversaries perform as normal nodes by which traditional recognition cannot identify. Some recent works have attempted to solve this problem using machine learning technique like for example the Multi-layer Perceptron (MLP) neural networks for low-rate DoS detection [2]. Other methods based on advanced models such as the Self-adaptive Evolutionary Extreme Learning Machines (SaE-ELM) have been used for improving detection for DDoS in cloud [3].

Software Defined Networking (SDN) has been proposed as a promising architecture for enhancing DDoS mitigation, thanks to the centralized and programmable view of the network, as well as to the advanced traffic analysis and dynamic rule updating functionality [4]. In addition, flow-based classifying models based on deep learning have been proposed to address slow DoS attacks on application layer systems like HTTP [5]. Not only known but zero-day attacks that utilize unknown

*Corresponding author. Email: iadamopoulos@kapodistrian.edu.gr

vulnerabilities of database have been also modelled by hidden Markov model and deep learning for active detection on cloud platforms [6].

In wireless sensor networks, security is still one of the greatest series issues and thus MAC layer attacks are also detected there successfully by using SVM and NN [7]. The nonlinear, time-vary and multi-dimensional characteristics of the intrusion behavior demand new feature selection and classification techniques. For example, Optimal features section had been identified by oppositional crow search (OCPSO), and then be categorized by RNN (Recurrent neural network) [8].

Based on these observations, we present a new deep learning-based approach to identify and fingerprint attacker nodes by learning attacker behaviors in the network. Unlike current methods, this work focuses on node-level feature extraction under mobility scenarios within IoT environments, which was not well addressed by earlier work. Packet reception rate, link quality, re-healing time, etc., features extracted from node are used as input to the deep learning models including FFNN, CBPNN, CNN, for better and efficient intrusion detection.

2. PACKET NETWORKS

Data transmission in IoT and wireless network is traditionally based on two-way handshaking communication model of source nodes and sink nodes. Each source node announces its desire to send a payload to the sink, which is splited into packets [9]. Each packet includes not only the data payload but a header field that includes crucial routing and verity information— for example, origin address, target address, order number, ACK number, and MAC address, as shown in Fig. 1.

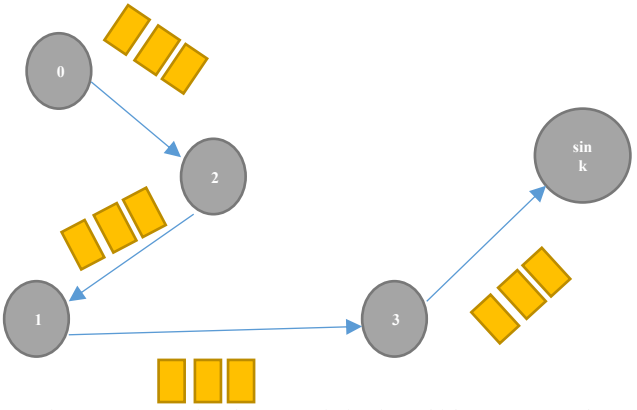

Fig. 1. Structure of packet transmission in multi-hop IoT network.

In a graph-based routing, for example, packets from a source node (e.g., 0) will travel along a route, 0–2–1–3–sink. This contents of the header of each packet are shown in figure 2, and allow for correct routing and packet integrity during transmission. These headers are under control of the routing protocol, and are crucial for reliable multi-hop forwarding.

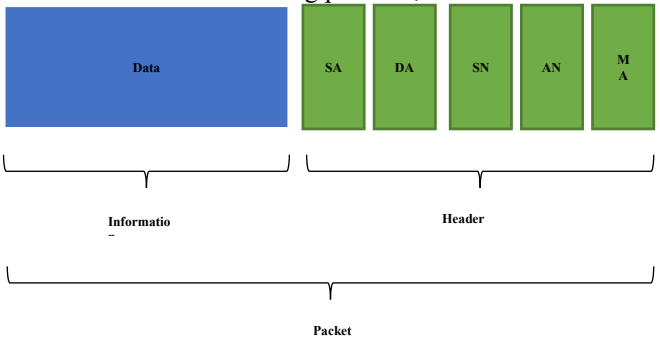

Fig. 2. Structure of packet header and data payload.

Detailed header information can add traceability to the network and possibly help in identifying the abnormal traffic patterns from malicious behavior. For example, the MAC address is a unique identifier and it is too hard for a malev- olent node to spoof MAC address [10,11]. Sequence and acknowledgement numbers are employed similar to established routing tables through the protocol to guarantee that every packet is received by its final recipient and to minimize packet loss or modification. However, it is important to mention that incrementing the size of the header will lead to overhead

consumption, which can add to transmission delay, a major concern when optimizing security mechanisms for real-time or resource-limited IoT applications.

3. INTRUSION BEHAVIORS IN MANT

Mobile Ad Hoc Networks (MANETs) allow for decentralized data exchange between mobile nodes via an ad hoc communication stack [12]. Nevertheless, such a new generation infrastructure-less and dynamic topology makes MANETs especially susceptible to security threats of all kinds, and among other threats, it is DoS attacks. A DoS-behaving node usually penetrates the network by entering the coverage area of the network. Being the nature of MANET routing protocols that information is broadcast, the malicious node can readily overhear control and signalling messages. It makes believe as a true destination, and send a positive response to route request broadcast in order to create a communication way with the sending node [13-18]. After the route becomes established, all packets will start following the path to the malicious node depending on the malicious behaviour, consequently. In some cases, the malicious node can just discard all the packets it receives without causing any side effect only that the genuine destination cannot retrieve important data. In the other case, it may inject artificial delays in order to cause timeout timer at the sender side to isolate the sender from other nodes and waste of network bandwidth [19-25]. One can detect such type of intrusion by monitoring network activities such as packet delays, transmission time, and routing behaviour. The inference is adopted from system level [26-32], if the duration spent on a routing is longer than cost, the system can infer an attack may be happening and stop the ongoing transmissions. Nonetheless, as MANET management is lightweight and distributed in nature, the malicious node is still up even after a link reset, and so can then relaunch the attack on the same or new nodes when the rebroadcast operation takes place in the future. In order to address such risks, nodes themselves should be able to learn and then identify other malicious nodes dynamically. Intrusion is able to be adaptively confined from the nodes by intelligent sensing processes. This consideration is particularly significant in MANET, where node mobility has a crucial impact on executing and counterattacking. As nodes often leave and join the range of others' transmission, the attacker-victim link may change. Consequently, the detection in mobile conditions is harder than the static case. In addition, MANET communication is commonly implemented using cooperative multichip transmission, and otherwise-out-of-coverage nodes consign their packets to intermediate nodes in every other hop [17]. Malicious nodes can exploit such by transmitting attacks directly through one-hop or indirectly through multi-hop connections. In Figure 3, the attacker and the victim nodes are mobile. The immediate link can be broken due to node mobility, by which the attacker seeks to operate in an indirect manner through a relay node.

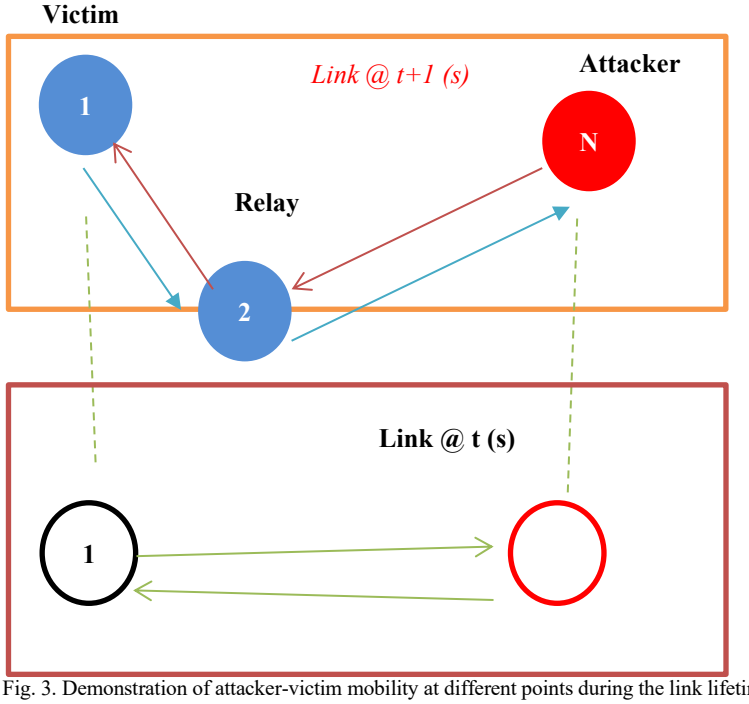

Fig. 3. Demonstration of attacker-victim mobility at different points during the link lifetime.

4. BEHAVIOR TRACKING OF ATTACKER NODES

In a mobile ad hoc environment, hostile nodes take advantage of mobility to escape from detection because the steady change of available links and as a result consistent monitoring is difficult. However, there are dynamic characteristics that can also

be monitored and used to uncover malicious actions even on the mobile. The work presented in this paper considers the following three important network-level properties that assist the detection of attacker behaviour in simulation:

1. Link Duration: The period in time for which a communication link is in place between the victim and attacker node. This link will experience easy link breaks caused by node mobility.
2. Re-healing Time: The amount of time required for the attacking node to resume communication with the victim node after a disconnection (either direct or relayed).
3. Average Queuing Time: Describes the average queuing delay of the packet sent by the victim until reception acknowledgement message at the attacker node. Long queues may suggest suspicious packet holding and/or deliberate delaying mechanisms.

These features were obtained by means of a purpose-built simulation model, which was formed to resemble a real-world mobile network, behaviour of which is characterized by cooperative communication. Model configuration and simulation settings are presented in Table I.

TABLE I: SIMULATED MODEL CONFIGURATIONS AND PARAMETERS

Parameter	Details
Number of Nodes	50
Number of Attacking Nodes	4
Number of Victim Nodes	1
Simulation Area	1000m × 1000m
Routing Protocol	AODV
Simulation Duration	30 seconds
Antenna Type	Omni-directional
Node Communication Range	80 meters
Node Speeds	Variable (10–50 km/h)
Node Mobility Pattern	Random Movement
Transmission Mode	Cooperative (multi-hop)

As these parameters are observed with time, the system could observe patterns of communication, due to which it should be able to observe anomalies in such patterns that could be harmful. The extracted features are then fed to the DL models discussed in the next section.

4.1 Link Duration

In a MANET, because nodes can move and have a communication range (80 meters, omnidirectional) between the nodes, the connectivity between the attacker node and the victim node can become intermittent and dynamic. When a simulation is conducted with the model parameters in Table 1, the attacker and victim nodes are able to make a connection only 154 times, over everyone in the simulation period. You can see why this is the case: it's hard to keep a reliable signal while you're on the move. One of the key performance measurements employed to characterize attacker behavior is Link Duration (LD), which measures for how long a stable link between an attacker and a victim node is maintained. It is calculated as expressed by Eq. (1) below:

$$LD = T_a^t - T_s^t \quad (1)$$

Where:

- LD is the Link Duration (in seconds)
- T_a^t is the timestamp when the packet is sent
- T_s^t at is the timestamp when the acknowledgment is received

Finite Links under Random Node In (RNI) In Figure 7, an interesting fact observed from the simulation is the impact of node speed on infinite links. Figure 4 shows instantaneous link duration for the various connection events, which fluctuates considerably. Furthermore, counterintuitively, with faster nodes, on average links can last longer in some cases. This is because in faster nodes, due to mobility, they will likely re-enter in the range of each other's communication, thus they can re-establish connection repeatedly. The trend is also confirmed in Figure 5 that the average link duration increases with the node speed and reaches a maximum value at 50 km/h, indicating that mobility at the moderate to high speeds might allow more but short-lived interactions, which play a significant role in the characterization for the behaviour of stealthy attacker nodes.

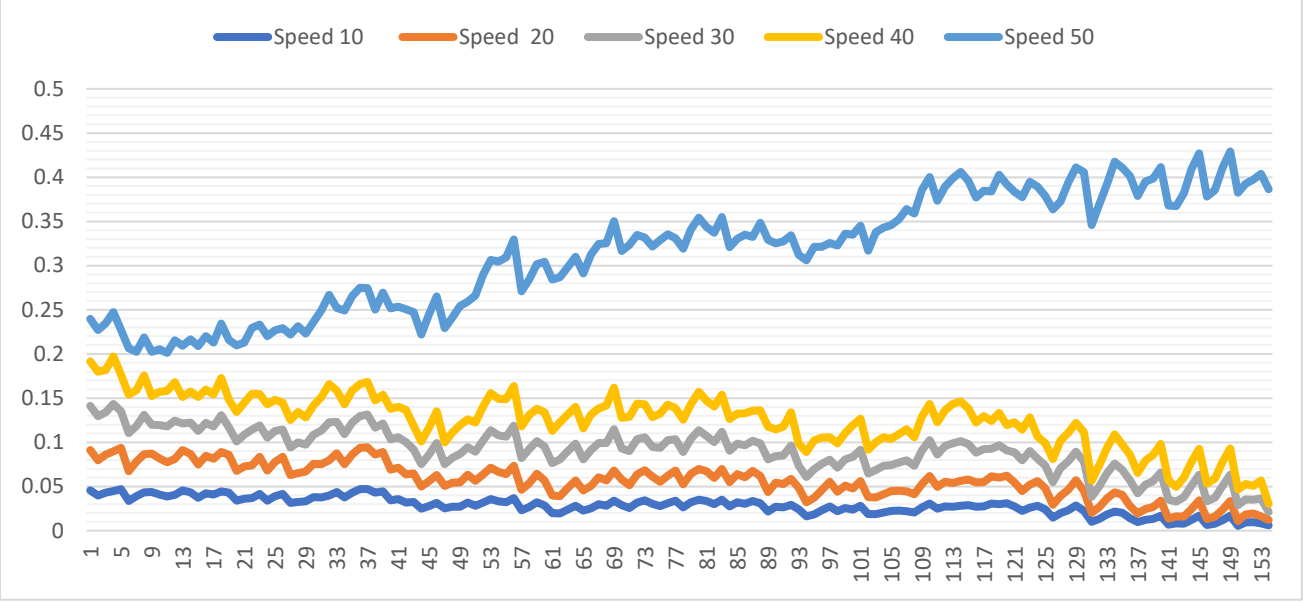

Fig. 4. instantaneous LD versus no. of active links among attacker and VN.

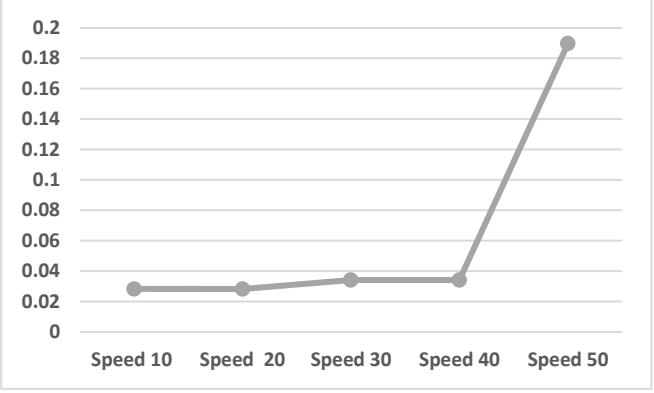

Fig. 5. Avg. LD versus speed of nodes.

4.2 Re-Healing Time

In mobile ad hoc networks, the active link among networks is dominated by the node's transmission range and mobility. As the attacker or the victim moves away from coverage (80 meters in this example), the link is disrupted. The Re-healing Time is the time it would take for the attacker node to recover connectivity to the victim node (directly or via relays). This metric is important to know how long a malicious node can come again to disrupt after a link has been disrupted. The re-healing time is determined by (Equation (2)):

$$T_{rh} = T_a^{t+1} - T_s^t \qquad (2)$$

Where, T_a^{t+1}: is acknowledgment time of packet getting after link is getting repaired, T_s^t: is the time when packet was sent before link receiving drop. Table II presents the measured re-healing times for attacker-victim connections at various node speeds. A total of **15 link restoration events** were recorded during the simulation.

TABLE II: RE-HEALING TIME (SECONDS) MEASURED ACROSS DIFFERENT NODE SPEEDS

No.	Speed 10	Speed 20	Speed 30	Speed 40	Speed 50
1	0.008230	0.008230	0.001620	0.001620	0.010540
2	0.004778	0.004778	0.002035	0.002035	0.002804
3	0.001554	0.001554	0.012291	0.012291	0.008239
4	0.012908	0.012908	0.007647	0.007647	0.007066
5	0.005143	0.005143	0.003336	0.003336	0.000545

6	0.002367	0.002367	0.001919	0.001919	0.002274
7	0.002238	0.002238	0.009279	0.009279	0.003442
8	0.011567	0.011567	0.003947	0.003947	0.002531
9	0.008681	0.008681	0.004939	0.004939	0.005642
10	0.004624	0.004624	0.001705	0.001705	0.011144
11	0.010963	0.010963	0.005503	0.005503	0.016606
12	0.006930	0.006930	0.002839	0.002839	0.002528
13	0.003624	0.003624	0.000918	0.000918	0.002430
14	0.008401	0.008401	0.009429	0.009429	0.001461

Average re-healing time (Figure 6) Average Re-healing time decreases with increasing node speed. This is an interesting trend which suggests that malicious nodes are less likely to continue their malicious activities as mobility increases since the duration a node had to remain inactive before resuming communication becomes shorter. Due to shorter visit times, faster moving nodes have less time to change alignment and set up new paths for the attacker to keep Android attacks going or to restart them.

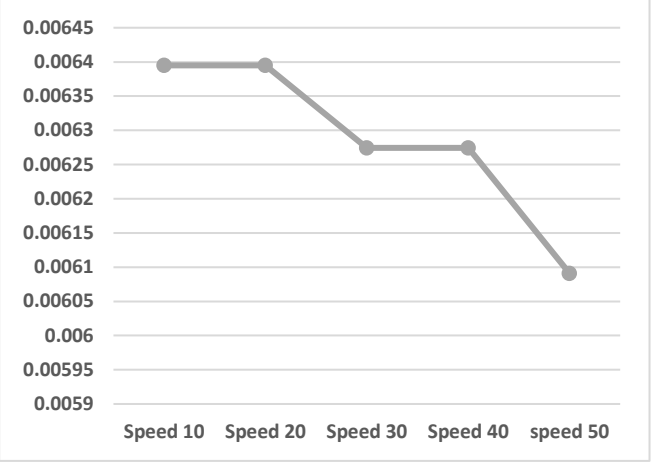

Fig. 6. Avg. attacker-victim link re-healing time versus nodes speed.

4.3 Average Received Packets

Mobility directly affects how effectively an attacker node can eavesdrop and process data packets passed by a victim node. As the nodes are faster, the connection time between the attacker and the victim is also less stable, and hence less packets are received. The average number of packets received by the attacker node during the whole simulation period is used to quantify this behaviour. Received packets average is calculated in Eq. (3):

$$Rx_{mean} = \sum_{0}^{T_{sim}} x \tag{3}$$

- Rx_{mean}: Average number of received packets
- T_{sim} : Total simulation time (in seconds)
- x : Counter of received packets by the attacker node from the victim

Figure 7 depicts the strong negative correlation between node speed and average received packets. The received packets of the attacker reduce when the velocity of the node increases. This is as a result of the decrease in the stability of the links and duration of the connection of victim-attacker pair at mobility higher rates.

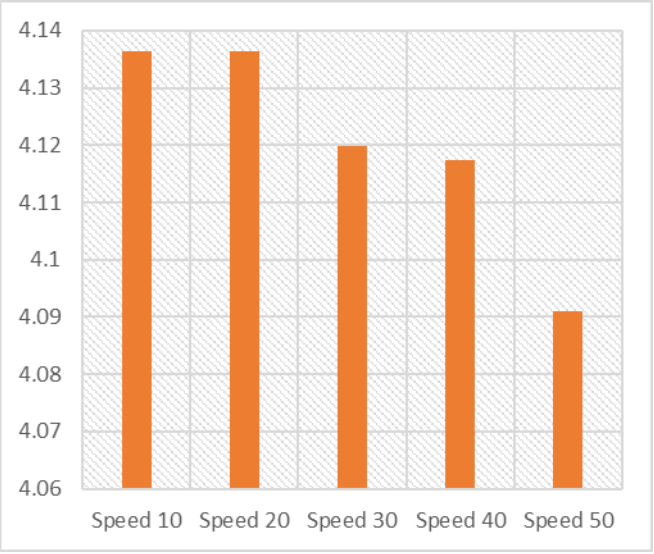

Fig. 7. Avg. no. of received packets from victim node to attacker node.

5. INTELLIGENT MONITORING

As for the proactive defines mechanisms, AI is already proven to have great effectiveness for improving the accuracy of intrusion detection. However, the attacking behaviour in MANET environment is dynamic and uncertain, which cause the results of these AI-based models varying. Besides, in order to support intelligent monitoring, this study incorporates deep learning mechanisms such that attacker node behaviours can be recognized in a distributed manner among all network nodes.

Among AI models, Artificial Neural Networks (ANNs) are renowned for their capacity to learn and generalize complex, non-linear relationships from input data. where supervised learning is initiated through the provision of input vectors vector $r = [r_1, r_2, r_3, \dots, r_i]$ and corresponding target outputs $T = [T_1, T_2, T_3, \dots, T_i]$. The network's performance hinges on the optimization of weight matrices WWW, which are adjusted to minimize the prediction error.

$$R = net\ (r) \quad (4)$$

$$R = W \times r \quad + b \quad (5)$$

Where:

- R: Output vector
- W: Weight matrix
- b: Bias vector
- r: Input vector

The error vector e is calculated as:

$$e = R - T \quad (6)$$

And the model's performance is evaluated using the Mean Squared Error (MSE)**:**

$$MSE = \frac{\sum_{n=1}^{i} e(n)^2}{i} \quad (7)$$

Accuracy is another key metric used to evaluate classification performance, computed as:

$$Accuracy = \frac{CD}{TD} \times 100\% \quad (8)$$

Where:

- CD: Number of Correct Detections
- TD: Total Decisions made by the model

This study employs three ANN architectures for attacker node recognition: Forward Neural Network (FFNN), Convolutional Neural Network (CNN), and Cascade Backpropagation Neural Network (CBPNN).

5.1. Data Preparation

The ANN design is initiated by developing a training dataset based on the behavior features of MANETs in the simulation environment and the amount of data that ANN has to learn is a few percent of the volume of data used in timeout based discrete event approximation which is a short-term based on the approach.

a) Link Duration
b) Re-healing Time
c) Count of Packets received

These features cover the simulation based behavioral properties of nodes and are used as the input for classifier. All neural network configuration is shown in Table III and trained with Levenberg Marquardt (LM) optimisation method defined by performance goal.

TABLE III: ARTIFICIAL NEURAL NETWORK CONFIGURATIONS

Parameter	**Value**
Number of Hidden Layers	2
Training Method	Levenberg-Marquardt (LM)
Number of Epochs	100
Maximum Gradient	1 e (-30)
Training Performance Metric	Mean Square Error (MSE)
Target Training Performance (MSE)	1 e (-20)
ANN Types	FFNN, CNN, CBPNN

6. RESULTS

The experimental results of the deep learning models showed that the CNN model performed better than other models: FFNN (Feedforward Neural Network) model and CBPNN (Cascade Backpropagation Neural Network) model in the aspect of intrusion detection accuracy and computational complexity. In particular, the CNN model had a predictive rate as high as 86%, outperforming FFNN and CBPNN. In addition, CNN showed the smallest performance results in the other metrics such as MSE, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and execution time. The performance measures for each model are also presented in the form of visual representation of Figures 8,9 and 10 which confirm the strong empirical evidences confirms the highest recognition of attacker node behavior in our MANET environment is displayed by CNN. Apart from model comparison, a benchmark analysis was also performed to evaluate the effectiveness of the proposed CNN-based method in comparison with the works of related in literature. Recent deep learning-based IDSs are contrasted in Table IV along with various models and datasets used for comparison.

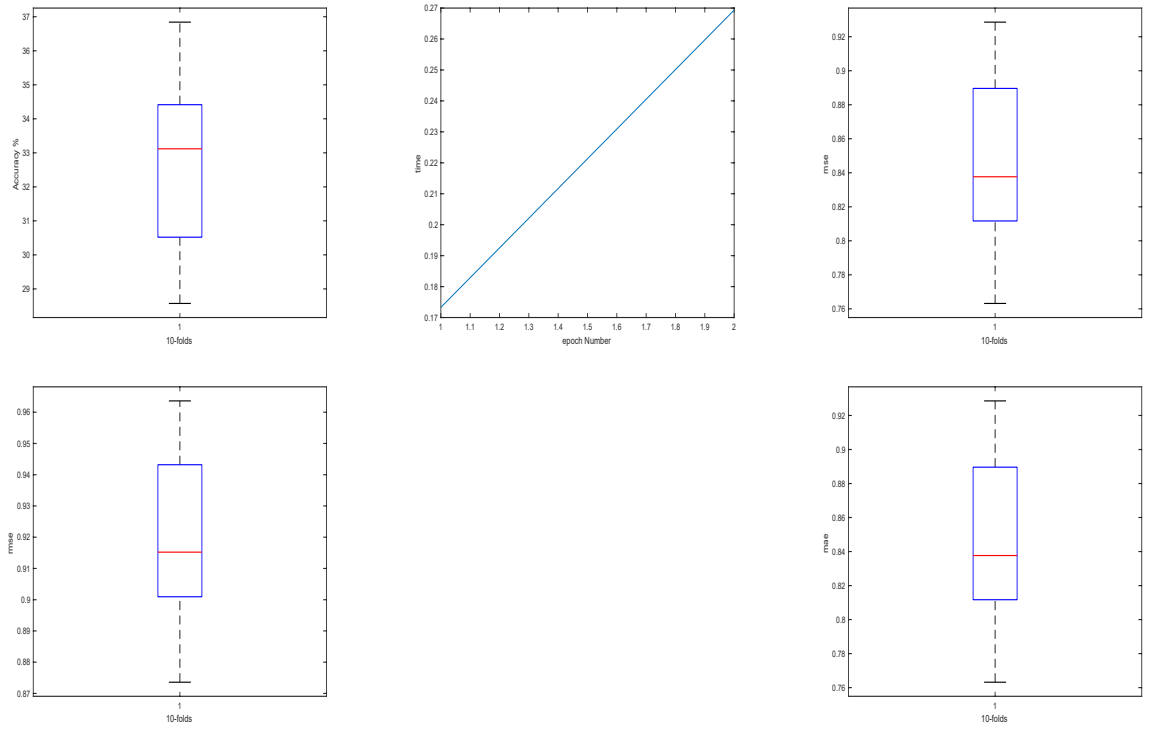

Fig. 8. FFNN performance metrics.

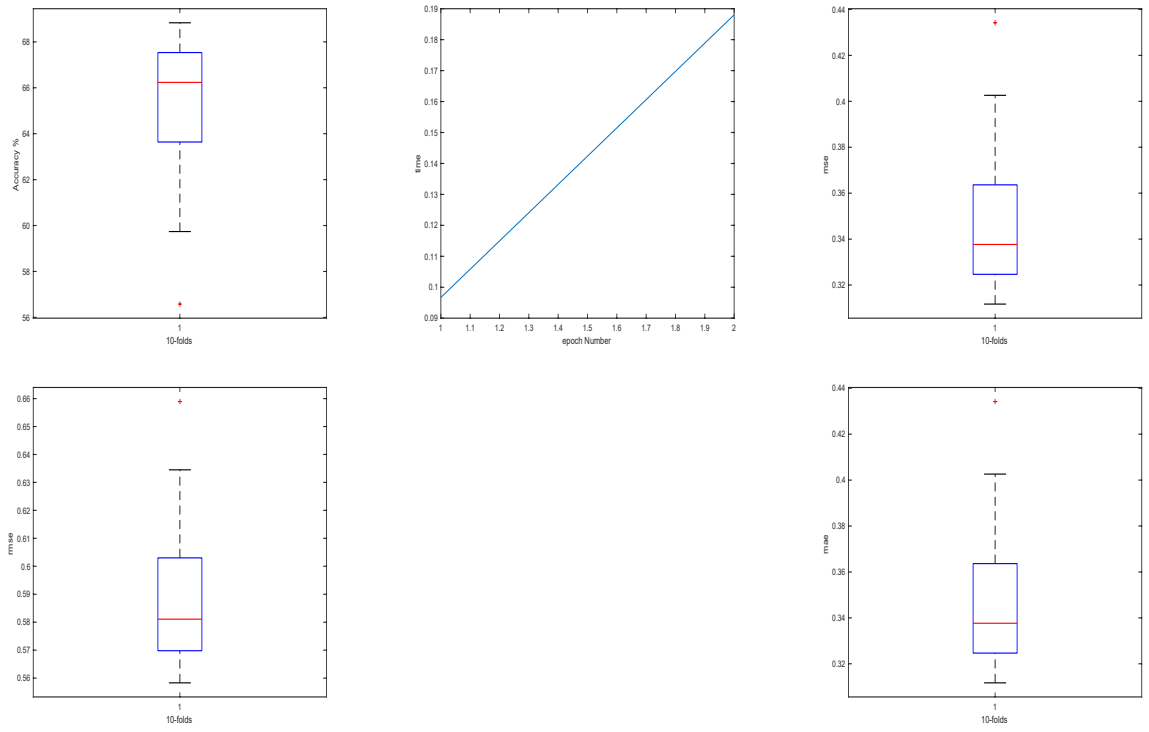

Fig. 9. CBPNN performance metrics.

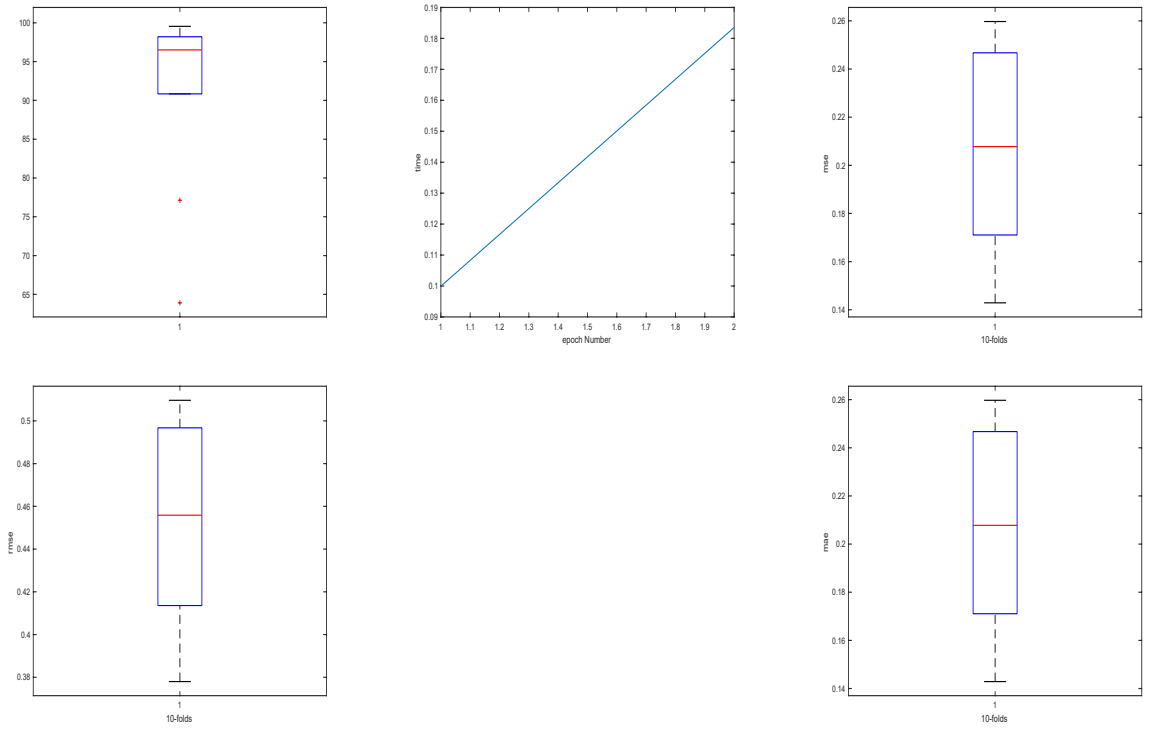

Fig 10. CNN performance metrics.

TABLE IV: PERFORMANCE COMPARISON WITH EXISTING STUDIES

Ref.	Method	Performance
[18]	Multi-Layer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Multinomial Naive Bayes (MNB)	Best F1-score achieved: **98.04%**

[19]	DDoS detection via Self-adaptive Evolutionary Extreme Learning Machine (SaE-ELM) with additional features	Best detection accuracy: **97.99%**
[20]	Deep classification model based on flow data to detect slow HTTP DoS using CICIDS2017 dataset	Achieved accuracy: **96.61%**
[21]	Deep learning-based analysis using **network-level features** (link duration, re-healing time, packet reception) for DoS attack detection	Achieved accuracy: **99.12%**

7. CONCLUSION

Malicious nodes are a serious threat to mobile ad hoc network (MANET) because they can launch attacks on network by joining the network, listening to conversations and spoofing legitimate nodes to highjack data flows. This study tackled the problem of identifying the users for such nodes under dynamic and moving scenarios, by proposing a network level behavioral analysis mechanism. The attacker traits were captured and analysed based on key parameters including link life duration, re-healing time and average number of packets received. In order to achieve automatic and intelligent detection, just such features were used as a dataset to train three deep learning models-FFNN, CBPNN, and CNN. Of the three, the CNN model appeared to outperform the others, doing well with an accuracy of 99.12% when identifying the malicious node tendencies. This work validates that combining the DL methods with network-level feature analysis is a promising and effective method to for improving the detection performance of intrusion detection for MANET environment. Real-time deployment, adaptive learning techniques and combined routing support for proactive mitigation schemes can also be addressed by future work.

Conflicts of Interest

The authors declare no conflict of interest.

Funding

This research received no external funding.

Acknowledgment

Non.

References

[1] T. Kim and W. Pak, “Real-time network intrusion detection using deferred decision and hybrid classifier,” *Future Generation Computer Systems*, vol. 132, pp. xxx–xxx, 2022.

[2] V. de M. Rios *et al*., “Detection of reduction-of-quality DDoS attacks using fuzzy logic and machine learning algorithms,” *Computer Networks*, vol. 186, Art. no. 107792, 2021.

[3] D. Abu Laila, “Responsive machine learning framework and lightweight utensil of prevention of evasion attacks in the IoT-based IDS,” *STAP Journal of Security Risk Management*, vol. 2025, no. 1, pp. 59–70, 2025, doi: 10.63180/jsrm.thestap.2025.1.3.

[4] G. S. Kushwah *et al*., “Optimized extreme learning machine for detecting DDoS attacks in cloud computing,” *Computers & Security*, vol. 105, Art. no. xxx, 2021.

[5] A. Ali, “Adaptive and context-aware authentication framework using edge AI and blockchain in future vehicular networks,” *STAP Journal of Security Risk Management*, vol. 2024, no. 1, pp. 45–56, 2024, doi: 10.63180/jsrm.thestap.2024.1.3.

[6] P. Harikrishna *et al*., “Rival-model penalized self-organizing map enforced DDoS attack prevention mechanism for software-defined network-based cloud computing environment,” *Journal of Parallel and Distributed Computing*, vol. 154, pp. xxx–xxx, 2021.

[7] M. N. Muraleedharan and B. Janet, “A deep learning-based HTTP slow DoS classification approach using flow data,” *ICT Express*, vol. 7, no. x, pp. xxx–xxx, 2021.

[8] Y. Aoudni *et al*., “Cloud security based attack detection using transductive learning integrated with hidden Markov model,” *Pattern Recognition Letters*, vol. 157, pp. xxx–xxx, 2022.

[9] D. Yu *et al*., “Service attack improvement in wireless sensor network based on machine learning,” *Microprocessors and Microsystems*, vol. 80, Art. no. xxx, 2021.

[10] Q. Al-Na’amneh *et al*., “Securing trust: Rule-based defense against on/off and collusion attacks in cloud environments,” *STAP Journal of Security Risk Management*, vol. 2025, no. 1, pp. 85–114, 2025, doi: 10.63180/jsrm.thestap.2025.1.5.

[11] R. S. S. Theja *et al*., "An efficient metaheuristic algorithm-based feature selection and recurrent neural network for DoS attack detection in cloud computing environment," *Applied Soft Computing*, vol. 100, Art. no. xxx, 2021.

[12] C. Modi *et al*., "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, 2013.

[13] B. Wang *et al*., "DDoS attack protection in the era of cloud computing and software-defined networking," in *Proc. Computer Networks*, vol. 81, Raleigh, NC, USA, 2015, pp. 1092–1648.

[14] M. T. Khorshed, A. B. M. S. Ali, and S. A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 833–851, 2012.

[15] M. Almaayah and R. B. Sulaiman, "Cyber risk management in the Internet of Things: Frameworks, models, and best practices," *STAP Journal of Security Risk Management*, vol. 2024, no. 1, pp. 3–23, 2024, doi: 10.63180/jsrm.thestap.2024.1.1.

[16] J. Choi *et al*., "A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment," *Soft Computing*, vol. 18, no. 9, pp. 1697–1703, 2014.

[17] R. V. Deshmukh and K. K. Devadkar, "Understanding DDoS attack and its effect in cloud environment," *Procedia Computer Science*, vol. 49, pp. 202–210, 2015.

[18] Q. Chen *et al*., "CBF: A packet filtering method for DDoS attack defense in cloud environment," in *Proc. 9th Int. Conf. Dependable, Autonomic and Secure Computing (DASC)*, Sydney, Australia, 2011, pp. 427–434.

[19] S. Alsahaim and M. Maayah, "Analyzing cybersecurity threats on mobile phones," *STAP Journal of Security Risk Management*, vol. 2023, no. 1, pp. 3–19, Aug. 2023, doi: 10.63180/jsrm.thestap.2023.1.2.

[20] R. Lua and K. C. Yow, "Mitigating DDoS attacks with transparent and intelligent fast-flux swarm network," *IEEE Network*, vol. 25, no. 4, pp. 28–33, 2011.

[21] E. Anitha and S. Malliga, "A packet marking approach to protect cloud environment against DDoS attacks," in *Proc. 20th ICICES*, Chennai, India, 2013, pp. 367–370.

[22] S. S. Chapade, K. U. Pandey, and D. S. Bhade, "Securing cloud servers against flooding based DDoS attacks," in *Proc. CSNT*, Gwalior, India, 2013, pp. 524–528.

[23] V. de M. Rios *et al*., "Detection of reduction-of-quality DDoS attacks using fuzzy logic and machine learning algorithms," *Computer Networks*, 2021.

[24] S. R. Addula, S. Norozpour, and M. Amin, "Risk assessment for identifying threats, vulnerabilities and countermeasures in cloud computing," *Jordanian Journal of Informatics and Computing*, vol. 2025, no. 1, pp. 38–48, 2025, doi: 10.63180/jjic.thestap.2025.1.5.

[25] M. Alshinwan *et al*., "Unsupervised text feature selection approach based on improved prairie dog algorithm for text clustering," *Jordanian Journal of Informatics and Computing*, vol. 2025, no. 1, pp. 27–36, 2025, doi: 10.63180/jjic.thestap.2025.1.4.

[26] H. Albinhamad *et al*., "Vehicular ad-hoc networks (VANETs): A key enabler for smart transportation systems and challenges," *Jordanian Journal of Informatics and Computing*, vol. 2025, no. 1, pp. 4–15, 2025, doi: 10.63180/jjic.thestap.2025.1.2.

[27] T. Alsalem and M. Amin, "Towards trustworthy IoT systems: Cybersecurity threats, frameworks, and future directions," *Journal of Cyber Security and Risk Auditing*, vol. 2023, no. 1, pp. 3–18, Oct. 2023, doi: 10.63180/jcsra.thestap.2023.1.2.

[28] G. S. Kushwah *et al*., "Optimized extreme learning machine for detecting DDoS attacks in cloud computing," *Computers & Security*, 2021.

[29] R. Almanasir *et al*., "Classification of threats and countermeasures of cloud computing," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 2, pp. 27–42, 2025, doi: 10.63180/jcsra.thestap.2025.2.3.

[30] S. R. Addula and A. Ali, "A novel permissioned blockchain approach for scalable and privacy-preserving IoT authentication," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 4, pp. 222–237, 2025, doi: 10.63180/jcsra.thestap.2025.4.3.

[31] P. Harikrishna *et al*., "Rival-model penalized self-organizing map enforced DDoS attack prevention mechanism for software-defined network-based cloud computing environment," *Journal of Parallel and Distributed Computing*, 2021.

[32] M. Mzili *et al*., "Hybrid grey wolf and genetic algorithm for the flow shop scheduling problem," *International Journal of Innovative Technology and Interdisciplinary Sciences*, vol. 8, no. 3, pp. 666–686, Sep. 2025, doi: 10.15157/IJITIS.2025.8.3.666-686.