

FLAT: A Federated Graph Attention Network for Secure Routing and Attack Detection in MANETs

Anuja Priyam*

Harcourt Butler Technical University, Department of Computer Science & Engineering, Kanpur, India

E-mail: anujapriyam@gmail.com

ORCID iD: <https://orcid.org/0009-0006-7863-7230>

*Corresponding Author

Anita Yadav

Harcourt Butler Technical University, Department of Computer Science & Engineering, Kanpur, India

E-mail: ayadav@hbtu.ac.in

ORCID iD: <https://orcid.org/0000-0003-0404-0213>

Received: June 11, 2026; Revised: June 23, 2026; Accepted: July 2, 2026; Published: August 8, 2026

Abstract: Mobile Ad Hoc Networks (MANETs) are autonomous wireless networks that do not rely on fixed infrastructure for communication among the mobile nodes. The mobile nodes communicate with each other with no centralized control, and routes are not established in advance. This makes MANETs vulnerable to routing attacks such as Black Hole and Sybil attacks, which divert or drop packets and thereby degrade network performance. This paper presents a novel hybrid approach combining Federated Learning (FL) and Graph Attention Network (GAT), which is termed FLAT. GAT dynamically assigns importance to neighbouring nodes, allowing the model to capture the mobility, link instability, and heterogeneous node behaviour inherent in MANETs. FL, in turn, enables routing and attack patterns to be learned in a decentralized manner without sharing raw data. The approach was evaluated in NS-3.36 on networks ranging from 10 to 150 nodes, under Black Hole and Sybil attacks with 20% of the nodes acting maliciously, and was compared against AODV, SAODV, AOMDV, and the optimization-based Dolphin Cat Optimizer, using PDR, PLR, throughput, and end-to-end delay as evaluation metrics. It is found that the PDR of FLAT is approximately 15.1% higher than AODV and 8.2% higher than the Dolphin Cat Optimizer, while its PLR is reduced by about 77% relative to AODV, 75% relative to SAODV, and 50% relative to the Dolphin Cat Optimizer. With respect to throughput performance, FLAT shows a performance level of about 4.4, 2.8, 2.3, and 2.1 times higher than AODV, SAODV, AOMDV, and Dolphin Cat Optimizer respectively. The above findings clearly show that FLAT outperforms current techniques with regard to all measured parameters.

Index Terms: Routing, Attack Detection, MANETs, Federated Learning, Graph Attention Networks, Secure Routing

1. Introduction

MANETs are wireless networks that have no fixed infrastructure, where there is routing between various mobile nodes using packet switching in the absence of any fixed infrastructure [1, 2, 20]. In such networks, there are different values associated with each of the mobile nodes as far as their pause time, communication range and speed are concerned. The topology of such MANETs along with their lack of central infrastructure makes them vulnerable to routing attacks. For example, the Black Hole attack involves a particular node making false claims about it having the shortest path, and hence the packets received at it are dropped by it. Similarly, Sybil attacks involve the generation of multiple identities by a single node. As the networks keep on developing in terms of nature and complexity, their routing and security processes also do not adapt themselves properly to the changes.

This paper aims at making use of GAT and FL in MANETs routing and security framework. The standard methodologies for learning from graphs are referred to as graph attention networks, or GATs. GATs have demonstrated the capability to learn relationships in graph-structured data and are considered ideally suited for capturing the dynamic and interconnected behaviors of MANETs. GATs have been used quite effectively across diverse applications, including computational neuroscience [3] and quantum chemistry [4]. Also, with graph processing algorithms and the

use of Graphical Processing Units (GPUs), some tasks can be performed in less time and to greater effect [5]. The success of these efforts has supported the broader effort of applying GATs techniques in networking technology [6, 7] and there is even a worldwide challenge to apply GATs to solve problems in networking [8]. Alternatively, FL will allow the joint distributed training of a model at various nodes and safeguard the data privacy of those edge nodes which is inherent to distributed networks.

These are the contributions of this paper:

- FLAT is a unified framework that integrates Federated Learning with a Graph Attention Network so that a single model performs both routing optimization and attack detection in MANETs, without sharing raw node data.
- It introduces a graph-construction and feature-engineering scheme in which each node builds a local graph from its neighbours and the GAT predicts link reliability for routing while simultaneously flagging malicious nodes.
- The proposed methodology incorporates detection of two distinct attack types - Black Hole and Sybil. It quantifies detection performance alongside routing quality.
- FLAT is evaluated in NS-3.36 across network sizes from 10 to 150 nodes and compared against AODV, SAODV, AOMDV and Dolphin Cat Optimizer using PDR, PLR, throughput, and end-to-end delay.

2. Literature Review

This section reviews existing approaches to secure and efficient routing in MANETs, focusing on federated learning, graph neural networks, and reinforcement-learning-based protocols. A variety of routing protocols have been introduced to optimize energy efficiency in MANETs [19].

FL decreases resource overhead and preserves privacy by allowing nodes to collaboratively train a model without sharing their data. The research outlined in [16] is a trust-aware routing framework that combines FL and multiobjective optimization methods that improve the reliability of data transmission by identifying and removing malicious nodes, ultimately increasing security in the network. FL can affect PDR, PLR, and throughput across performance measures. In PDR and PLR Improvement FL frameworks such as HIDE-MAN, FL decreases PLR with malicious node identification, such as blackhole attackers that attempt to drop packets. This improvement in node security ensures better PDR. During throughput optimization, training decentralized intrusion detection models, FL prevents DoS attacks and congestion, thereby maintaining stable throughput in MANETs [11].

The function of GNNs in routing optimization for next-generation communication networks, such as MANETs, is examined in a research study [17]. It further classifies GNN-based routing algorithms into three different categories including reinforcement learning-based dynamic routing algorithms, supervised learning-based modeling of networks and routing, and supervised learning-based routing optimization. Literature Review on Deep Learning for MANET Routing [18] makes use of GNNs in solving the problems faced by wireless networks, especially routing problems within MANETs. This literature review shows the potential of GNNs in improving the effectiveness and flexibility of routing in the dynamic environment of the network. GNNs have been very useful in dynamically analyzing the graph structure of the data.

In routing and PDR optimization tasks, GNNs predict the optimal routes by accounting for link quality, node mobility, and the overall traffic load in the network. GNNs also avoid congested or potentially malicious nodes which translate to increase PDR [10, 14]. GNNs mitigate congestion by learning traffic patterns and routing data as needed which maintains acceptable throughput even in high mobility scenarios [14]. GNNs also assist to discover weak or malicious links in real time which can reduce packet losses caused by routes that are compromised or very unstable [11, 10]. GNNs are excellent for modeling trust relationships, which makes them ideal for Sybil attack detection. GNNs can model nodes and how they interact, such that anomalous behaviors such as one node impersonating many identities can be identified [11, 10].

Performance metrics, such as PDR, PLR, and throughput, are highly influenced by potential security threats. Blackhole Attacks have drastic impacts on both PDR and throughput by discarding every packet routed through malicious nodes. In the case of FL and using GNNs for dynamic trust modelling, PLR can be significantly reduced [11, 10]. Sybil Attacks introduce fake identities into the routes which can also create congestion and ultimately reduce throughput. FL models combine aggregate trust metrics to detect these weak nodes, and GNNs improve node authentication to produce high PDR and stable throughput rates [14, 10].

Reinforcement Learning (RL) approaches (Q-Learning, DQN) respond dynamically to the changing nature of MANET topologies, while simultaneously improving the KPIs. RL-based routing performs route learning to determine the optimal paths that ultimately minimize packet losses from route failures, and therefore increases PDR [12, 15]. Similarly, the protocols successfully handle the traffic load across and through the network to optimize throughput of the network (congestion/ resource-sensitive) [14].

Protocols such as QoSRR and FRESH enhance workflows for QoS purposes by modifying the route discovery

mechanism of AODV to account for link quality and link energy levels, which improves PDR values, and maintains throughput even in low power networks [13].

Protocols improved Energy-Efficient PLR: For PLR purposes, the protocols restrict the search for better possible routes to the most stable routes with respect to predicted link energy costs, and they only use more energy than they need [15, 13]. Researchers [9] extract the proposed quality of the path, while accounting for the energy and end to end delay measures in situations when overhead control is restricted and scalability is possible.

Across these studies, two broad directions emerge. The first comprises federated and trust-based intrusion-detection methods, such as HIDE-MAN and FL-based trust frameworks [11,16], which preserve privacy and detect malicious nodes effectively but treat attack detection separately from routing and rely on classical optimization rather than graph-structured learning. The second comprises GNN-based routing methods [17,18], which exploit the graph structure of the network to optimize routes and link quality but are typically trained centrally — exposing raw node data and do not jointly defend against attacks such as Black Hole and Sybil. Consequently, no existing approach unifies privacy-preserving federated training, graph-attention-based routing, and multi-attack detection within a single model. FLAT is designed to close this gap by coupling federated learning with a graph attention network so that routing optimization and attack detection are performed jointly and without sharing raw data.

From this review, the following gaps remain:

- Existing FL-based intrusion-detection methods detect malicious nodes but do not feed the detection outcome into the routing decision, so secure routing and attack detection are optimized in isolation [11,16].
- GNN-based routing methods improve route selection using graph structure but are predominantly centralized, do not preserve node-level data privacy, and do not defend against identity-based attacks such as Sybil [17,18].
- No prior work jointly addresses routing optimization and detection of multiple attack types (Black Hole and Sybil) within a single privacy-preserving, graph-attention-based federated framework evaluated across varying network sizes and attack scenarios.

The combination of FL and GATs provides an innovative solution to the issues involved in the process of routing within MANETs. The combination utilizes the power of FL along with that of GATs for improving both the security as well as the effectiveness of routing operations. Some areas of research gaps identified in this study are concerned with detection of attacks along with evaluation of performance parameters like PDR, throughput, PLR, etc.

3. Graph Attention Network (GAT)

A GAT is a neural network architecture tailored for graph-structured data that utilizes masked self-attention to determine the relative importance of neighboring nodes. Given a graph $G = (V, E)$ with node features $h_i \in R^F$, GAT first applies a linear transformation:

$$Z_i = Wh_i \quad (1)$$

where $W \in (R^{F' \times F})$ is a learnable weight matrix. The attention coefficient between node i and its neighbor j is computed as:

$$e_{ij} = \text{Leaky ReLU}(a^T [Z_i \| Z_j]) \quad (2)$$

where $\|$ stands for concatenation, a is a learnable attention vector and h_i and h_j are the input feature vectors of nodes i and j . Softmax is used to normalize these coefficients across neighbors.

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})} \quad (3)$$

Next, a weighted aggregation of the updated node representation is produced.

$$h'_i = \sigma \left(\sum_{j \in N(i)} a_{ij} Z_j \right) \quad (4)$$

where σ is a non-linear activation function. In the context of MANET routing attack detection, GAT has shown its strength in capturing the dynamic topological relationships and emphasizing the influence of suspicious nodes.

Various types of figures and tables are defined below related to the graph structures and representations underlying the GAT framework.

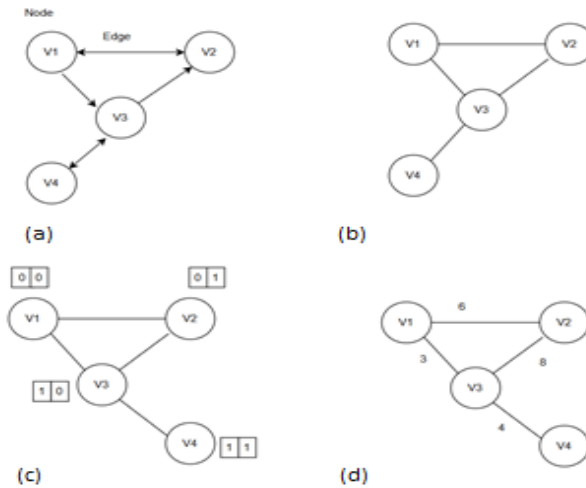


Fig. 1. (a) Directed Graph (b) Undirected Graph (c) Node Feature Graph (d) Edge Feature Graph

Table 1. Various Types of Matrices

Node Feature Matrix		Edge Feature Matrix				Adjacency Matrix				Degree Matrix			
0	0	0	6	3	0	0	1	1	0	2	0	0	0
0	1	6	0	8	0	1	0	1	0	0	2	0	0
1	0	3	8	0	4	1	1	0	1	0	0	3	0
1	1	0	0	4	0	0	0	1	0	0	0	0	1

Figure 1 illustrates the basic graph types used in this work: (a) a directed graph, (b) an undirected graph, (c) a node-feature graph, and (d) an edge-feature graph. A directed graph assigns each edge an orientation, expressing a one-way relationship between nodes, while an undirected graph has orientation-free edges that represent mutual connections; the neighborhood of a node is the set of nodes directly linked to it. Node features are the properties attached to each node — in Figure 1(c), binary values (0/1) mark whether a feature is present — whereas edge features describe the links between nodes, shown in Figure 1(d) as the distances between connected nodes. These attributes are organized into the matrices summarized in Table 1: The adjacency matrix documents which nodes are related, the diagonal degree matrix saves each node's degree, or the number of edges incident to it, and node- and edge-feature matrices contain node and edge properties.

4. Federated Learning

Several nodes or devices use this distributed machine learning technique to develop a common model without exchanging local data. Rather, the model is trained locally on each device, and only the model updates (such as gradients or weights) are shared with peers or a central server. This technique is especially useful in systems like MANET, where the nodes are decentralized and mostly mobile, and their resources are scant. It ensures that raw data never leave the nodes, which is essential for preserving the privacy of sensitive information in MANETs, such as user data or location information. FL empowers the nodes in a MANET to jointly construct a strong machine learning model by utilizing the distributed data that would otherwise be inaccessible to the individual nodes.

In Fig. 2. above, nodes independently train models with their respective data. Nodes then transmit updated models to the central server rather than raw data. The server then aggregates the updates to enhance the global model. The nodes then receive the revised global model. Continue till you reach convergence.

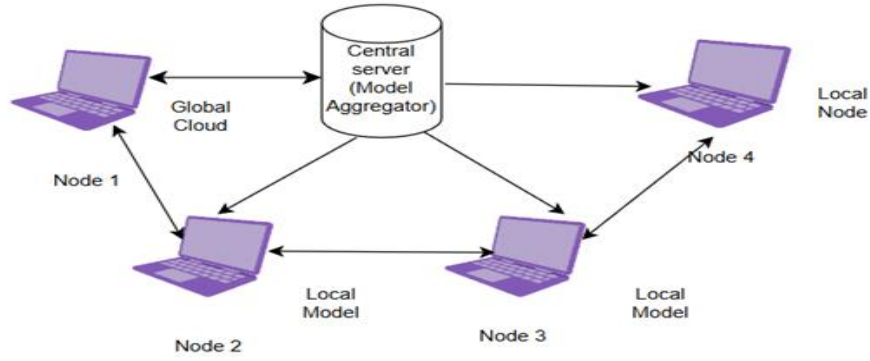


Fig. 2. Federated Learning Architecture

5. Federated Learning-Based Graph Attention Network (FLAT)

As illustrated in Fig 3, the FLAT framework comprises a set of mobile nodes forming the MANET together with a central federated aggregation server. Each node constructs a local graph $G = (V, E)$ from its one-hop neighbours and trains a local GAT model on its own private traffic by minimizing the binary cross-entropy loss given in Eq 5, so that raw data never leaves the node and only the local model parameters θ_k are transmitted to the server. The server combines the parameters received from all participating nodes through Federated Averaging (Eq 6) to form an improved global model Θ , which is broadcast back to every node for the next round; this exchange repeats over T communication rounds until convergence. The converged global model is then deployed for two tasks: secure routing, in which the predicted link-reliability score $LR = \delta(h_i W h_j)$ is used to select the most reliable, lowest-cost path, and attack detection, in which a node is flagged as malicious (Black Hole or Sybil) when its predicted probability $\hat{y} \geq t$.

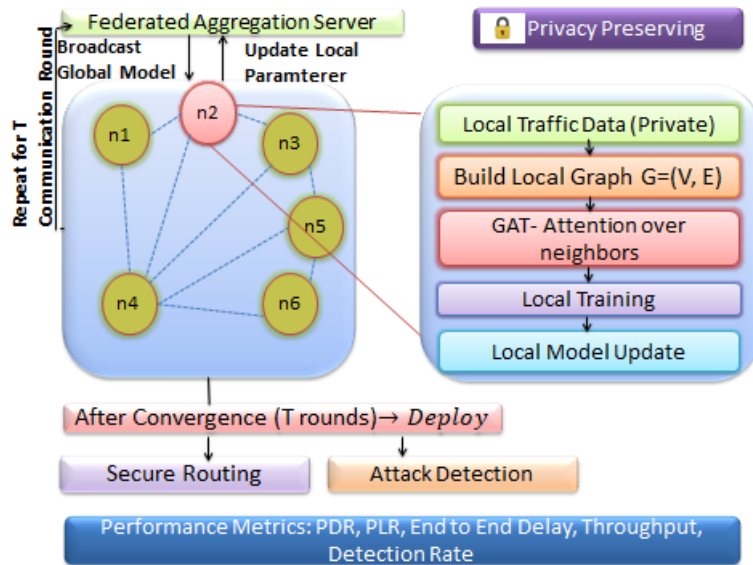


Fig. 3. FLAT System Architecture and Federated Data Flow

FL allows decentralized devices to train local models independently while maintaining data privacy by keeping the raw data on each device. Under this approach, GATs play a role in aggregating the updates of the local models in order to improve the global model. As a result, there are reduced risks of privacy breaches associated with centralized data storage, and increased scalability as the workload is distributed on different nodes – a fundamental benefit for resource-constrained MANETs. The proposed federated framework will allow the GAT to learn the representations of the nodes and links from each node's graph. At the same time, federated learning will aggregate the local models in order to share the knowledge of malicious behavior without passing the raw data [19].

Such a decentralized approach guarantees the protection of privacy and security, which are very important factors in MANETs. In comparison with traditional intrusion detection systems, federated learning with GATs will enable collaborative learning in MANETs protecting their local data. In this framework, each node uses its own labeled traffic to train a local GAT model and transmits only the model parameters to the central server, which compiles the

parameters received from all participating nodes to improve the global GAT model. The local training objective minimized at each node is the supervised binary cross-entropy classification loss given in Eq 5.

$$L = -\left(\frac{1}{N}\right) \sum [y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)] \quad (5)$$

where N is the number of training samples at the node, $y_i \in \{0,1\}$ is the true label (malicious = 1, normal = 0), and $\hat{y}_i$ is the maliciousness probability predicted by the GAT. After local training, each node transmits its GAT parameters to the central server, which aggregates the updates from all nodes to refine the global model using the Federated Averaging (FedAvg) formulation given in Eq 6.

$$\theta = \sum_{k=1}^n w_k \theta_k \quad (6)$$

Where θ denotes the global model parameters, θ_k the local parameters from node k , and $w_k = \frac{n_k}{n}$ the aggregation weight proportional to node k 's local dataset size n_k (with $n = \sum n_k$). When all nodes contribute equally, this reduces to the simple average $\theta = \frac{1}{N} \sum \theta_k$ shown in Fig. 3. This ensures that the knowledge learned by each local model contributes effectively to the training of the global model. Each node sends its update to the central server, where these contributions are aggregated to update the global model. Once new attack pattern information is integrated into the global model, the updated parameters are shared back with the local models. By sharing the information, the local models can immediately detect and block malicious packets, hence preventing and mitigating cyber-attacks within the MANET environment.

6. Methodology

The proposed FLAT methodology operationalizes the architecture of Fig 3 as the step-by-step procedure summarized in Fig 4. After the mobile nodes are identified and the initial GAT model is distributed, each node builds its local graph $G = (V, E)$, extracts and Min-Max-normalizes its features, and trains the GAT locally; the resulting parameters are then aggregated by Federated Averaging and the updated global model is redistributed in each round until convergence. The converged model is used to predict link reliability for routing-path selection and to detect Black Hole and Sybil attacks, after which the routing metrics (PDR, PLR, throughput, end-to-end delay) and the detection rate are computed. The graph construction strategy, feature definitions, routing-decision mapping, and detection thresholding underlying these steps are detailed in Sections 6.2–6.5.

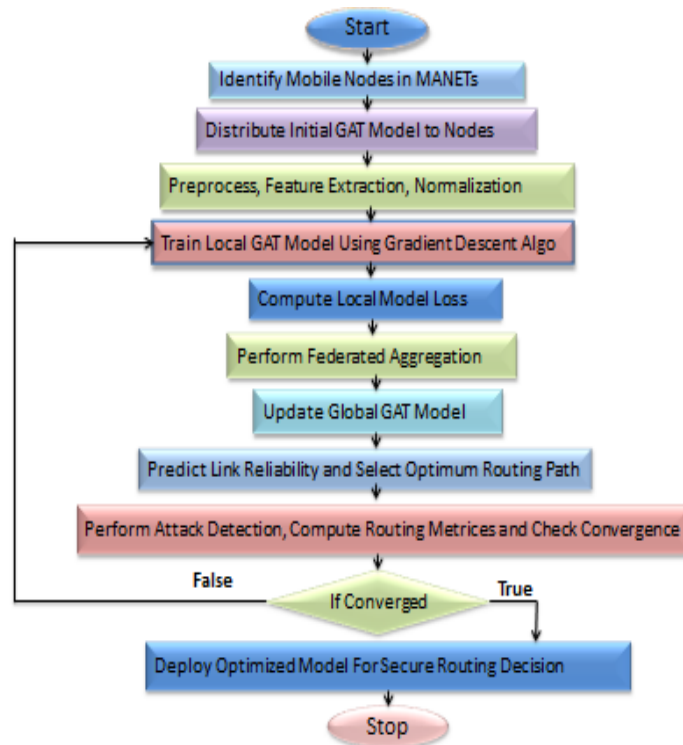


Fig. 4. Workflow Architecture of FLAT

6.1. Algorithms Steps

Input

V (V1, V2,..., Vn): Numbers of nodes in MANETs

E (E1, E2,..., En): Numbers of Edges that links between nodes

G (V, E): Dynamic graph representation of the MANETs with vertices V and Edges E

D: Decentralized data set on each node for training the GAT Model.

T: Total number of communication rounds in Federated Learning.

η : Learning rate for local model optimization

L: Loss function for GAT Model

M: Model parameters of the GAT for routing

Output

Predicted routing paths, Attack detection rate and Routing metrics

START

Initialize G (V, E), local GAT Model

while network is operational

Collect and Preprocess Data

do

GAT training

Extract features from decentralized data set D between nodes V

Normalize using Min-Max Scaling

$$Y' = \frac{Y - Y_{min}}{Y_{max} - Y_{min}}$$

Train Mn using Dataset Dn via Gradient Descent

$$\mathbf{g}_i^{t+1} = \mathbf{g}_i^t - \eta \nabla_{\mathbf{g}_i} L_i(\mathbf{g}_i)$$

Federated Aggregation

Send Model weight $\mathbf{g}_i$ to the server

Do Federated Averaging

$$\mathbf{g}^{t+1} = \frac{1}{N} \sum_{i=1}^N \mathbf{g}_i^t$$

Update global model $\mathbf{M}_{global}$ with $\mathbf{g}^{t+1}$ and forward it back to all nodes

Routing Prediction using GAT Model

For each Source to Destination pair of nodes

Use $\mathbf{M}_{global}$ to predict link reliability LR for each edge

$$LR = \delta(\mathbf{h}_i' W \mathbf{h}_j')$$

Where $\mathbf{h}_i'$ and $\mathbf{h}_j'$ are node embedding, W is learnable weight matrix and δ is the sigmoid activation function

Choose the best route based on the least amount of delay and the highest reliability.

Attack Detectionre detailed in below S

$$Detection\ Rate = \frac{B}{a}$$

$$Black\ Hole\ Detection\ Rate_B = \frac{Detected\ Black\ Holes}{Total\ Number\ of\ Black\ Holes}$$

$$Sybil\ Attack\ Detection\ Rate_S = \frac{Detected\ Sybil\ Nodes}{Total\ Number\ of\ Sybil\ Nodes}$$

Compute

$$PDR = \frac{Packet\ Delivered\ Successfully}{Total\ Packet\ Sent} * 100$$

$$\begin{aligned}
 PLR &= 1 - PDR \\
 \text{Throughput} &= \frac{\text{Data Delivered in bits}}{\text{Total Time in seconds}} \\
 &\text{End to End Delay}
 \end{aligned}$$

Repeat all above steps if result is unsatisfactory

Stop
End

6.2. Graph Construction Strategy

At each communication round t , the MANET is represented as a dynamic graph $G = (V, E)$, where V is the set of mobile nodes ($|V| = N$) and E_t is the set of active links at round t . An undirected edge $(i, j) \in E_t$ exists if and only if nodes i and j are mutually within transmission range R , i.e. their Euclidean distance $d_{ij} \leq R$, as obtained from the periodic neighbor-discovery (HELLO) messages collected in NS-3.36. The adjacency matrix A_t is rebuilt from each node's current neighbor table, with a self-loop ($A_{ii} = 1$) added so every node also attends to its own state. Because nodes move under the random-waypoint model, E_t and A_t are recomputed every round, allowing the GAT to capture the continuously changing topology and link instability of the MANET rather than a static snapshot.

6.3. Feature Definitions

Each node i is described by a local feature vector $h_i \in R^F$, extracted on-device from NS-3.36 Flow Monitor and the node's own counters, so that no raw data leaves the node. The node features are: residual energy, buffer/queue occupancy, node speed, node degree, packet-forwarding ratio (packets forwarded $\div$ packets received — the primary signal for Black Hole behaviour), control/identity-advertisement rate (a signal for Sybil behaviour), and a running trust score over a sliding window. Each edge (i, j) carries a feature vector e_{ij} comprising the normalized link distance d_{ij} , a link-quality estimate (SNR or ETX), and link duration. All features are scaled to $[0, 1]$ using the Min-Max normalization defined in the algorithm, giving a node-feature dimension $F = 7$ and an edge-feature dimension of 3.

6.4. Routing-Decision Mapping

The GAT (Eqs. 1–4) transforms the input features into updated node embeddings h'_i . For every active link, a link-reliability score is predicted as a display equation:

$$LR_{ij} = \delta(h'_i W'_i), LR_{ij} \in (0, 1) \quad (7)$$

where W is a learnable weight matrix and δ is the sigmoid activation; LR_{ij} is the probability that link (i, j) delivers packets successfully. Each edge is assigned a cost

$$w_{ij} = -\log(LR_{ij}) \quad (8)$$

and for each source–destination pair the route is selected as:

$$P^* = \arg \min_p \sum \{(i, j) \in P\} w_{ij} \quad (9)$$

which is equivalent to maximizing the end-to-end product of link reliabilities subject to the delay constraint, and is solved with a shortest-path search over the reliability-derived weights. This is the explicit mapping from GAT output to the routing table.

6.5. Attack Detection and Thresholding

In parallel, a classification head on the same GAT produces a maliciousness probability $\hat{y}_i \in (0, 1)$ for each node, trained with the supervised binary cross-entropy loss (Eq. 5; $y_i = 1$ for malicious, 0 for normal). A node is flagged as malicious when:

$$\hat{y}_i \geq \tau \quad (10)$$

with detection threshold $\tau = 0.5$, selected on a validation split to balance detection rate against false positives. Black Hole nodes are identified by the combination of an abnormally low packet-forwarding ratio and high $\hat{y}_i$, while Sybil nodes are flagged through the GAT's modelling of inconsistent identity/neighbour patterns. The detection rate is then computed as $\frac{B}{a}$ (Eq. 9), where a is the total number of injected malicious nodes and B the number correctly flagged.

7. Results And Discussion

These findings highlight the strength of utilizing FL in conjunction with GAT in solving the problems of routing and security in MANETs. With the use of the distributed FL algorithm, scalability can be achieved, while the capability to model dynamic and decentralized networks through the implementation of GAT is possible. Table 3 presents the training loss per round of FLAT for every node configuration. On the other hand, the corresponding PDR, PLR, E2E delay, and throughput values are provided in the figures shown below. One hundred fifty (150) nodes have been introduced to a simulated environment and were tested in the simulator implemented using the NS-3.36 network simulator based on the simulation parameters of Table 2. The network traces together with the values of PDR, PLR, E2E delay, and throughput were extracted from NS-3.36 Flow Monitor, whereas the GAT-based model is developed using PyTorch Geometric.

Table 2. Simulation Parameters of NS3.36

Parameter	Value
Simulator	NS-3.36
Simulation Area	700 × 800 m
Propagation Model	Two-Ray Ground
MAC Type	IEEE 802.11
Antenna	Omni-directional
Mobility Model	Random Waypoint
Number of Nodes	10–150
Node Speed	0–20 m/s
Pause Time	2 s
Traffic Type	CBR over UDP
Packet Size	512 bytes
Simulation Time	300 s
Attack Types	Black Hole, Sybil
Malicious Node Ratio	20%
Baseline Protocols	AODV, SAODV, AOMDV, Dolphin Cat Optimizer [21]
ML Framework	Python, PyTorch Geometric
Hardware Platform	NVIDIA T4 GPU, 16 GB RAM
Dataset Partition	70% Training / 15% Validation / 15% Testing
Communication Rounds (T)	5
Local Epochs	5
Batch Size	32
Learning Rate (η)	0.001
GAT Layers / Attention Heads	2 Layers / 8 Attention Heads
Aggregation Method	FedAvg

But the performance of the algorithm starts to decline in loss decrease starting from round 5, probably because of feature saturation and learning stabilization. This problem can be solved in table 3 through dynamic changing of the learning rate, more data variety, or use of regularization.

Table 3. Average Training Loss of FLAT per Communication Round

No. of Nodes	Round 1	Round 2	Round 3	Round 4	Round 5
20	0.0873	0.0865	0.0861	0.0871	0.0864
40	0.1916	0.1895	0.1873	0.1776	0.1865
60	0.1963	0.1937	0.1849	0.1922	0.1916
80	0.0934	0.0917	0.0921	0.0915	0.0931
100	0.0733	0.0669	0.0532	0.0614	0.0616

Different numbers of nodes used to assess the suggested architecture. Simulation results have been compared to AODV, SAODV, AOMDV and the Dolphin Cat Optimizer [21]. The results are shown as graphs in Figs 5, 6, 7, 8, 9 and 10.

7.1. Packet Loss Ratio

This ratio measures the ratio between packets that do not arrive at the destination node and the total number of packets transmitted; the smaller the value, the more reliable the route selection process becomes. Based on Fig. 5. and Table 4, it is clear that FLAT yields the smallest values of this parameter among all compared approaches in each node density and goes down from 10.7% (at 30 nodes) to 5.2% (at 150 nodes), which represents a reduction of 77%, 75%, and 50% compared to AODV, SAODV, and the Dolphin Cat Optimizer, correspondingly. This is due to different approaches used in the routing path selection process. AODV and SAODV are reactive protocols and discover paths mostly based on the shortest hop count; due to movement, these routes break or go through unreliable edges in most cases, so their losses remain constant (about 28–33%) and do not decrease when network size increases. The FLAT algorithm has the ability to achieve both: it employs a graph attention method that assigns weights based on link quality, residual energy, mobility, and forwarding characteristics and predicts a reliability level for each link, thereby making sure that packets take the most reliable routes instead of just the shortest ones, and also has an attack detection system that can detect and evade Black Hole nodes. Since the federated GAT system is always trained based on the latest network conditions, FLAT is able to adjust itself with the changing topology structure, which is why the packet loss rate in FLAT is lower than any other algorithm.

$$\text{Packet Loss Ratio} = \frac{\text{Total Packets Sent} - \text{Total Packets Received}}{\text{Total Packets Sent}} \quad (11)$$

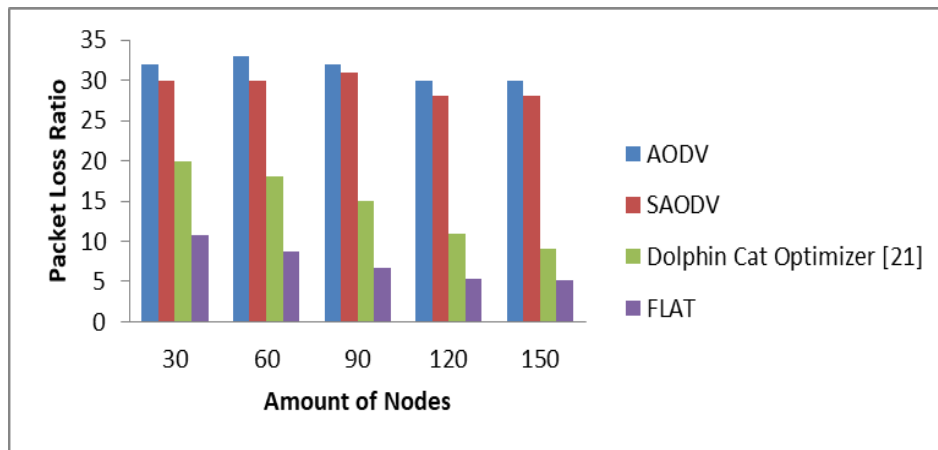


Fig. 5. Packet Loss Ratio vs Amount of Nodes

Table 4. PLR vs Amount of Nodes

No. of Nodes	AODV	SAODV	Dolphin Cat Optimizer [21]	FLAT
30	32	30	20	10.73
60	33	30	18	8.79
90	32	31	15	6.69
120	30	28	11	5.35
150	30	28	9	5.18

7.2. Packet Delivery Ratio

FLAT achieves a greater Packet Delivery Ratio since it employs routing through reliable links based on their expected reliability instead of using only the number of hops. On average, it achieves a delivery ratio of 91.5% against 84.6% in the case of Dolphin Cat Optimizer and 79.5% of AODV, which is approximately 8.2% and 15.1% better than those of Dolphin Cat Optimizer and AODV, respectively. Since the graph attention scheme gives preference to neighboring nodes with reliable links, enough energy, and stable forwarding, it makes the packet routes less vulnerable to disruption caused by mobility. In addition, FLAT performs an attack aware detection to avoid Black Hole attacks, i.e., nodes that would intercept any forwarded packets. Since the federated GAT scheme works by constantly updating itself with the present network conditions, it is able to adapt in real-time. This is the reason behind the higher PDR and its slower deterioration under poor conditions.

PDR performance vs time is shown in Fig. 6. and Table 5. It can be calculated using eq 12.

$$PDR = \frac{\text{Packet Delivered Successfully}}{\text{Total Packet Sent}} * 100 \quad (12)$$

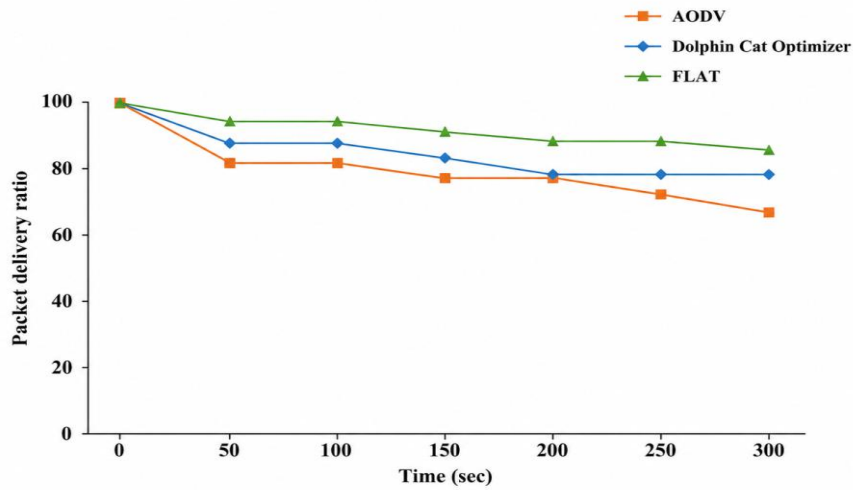


Fig. 6. PDR vs Time

Table 5. PDR vs Time

Time (sec)	AODV	Dolphin Cat Optimizer [21]	FLAT
0	100	100	100
50	81	88	95
100	80	87	94
150	78	83	91
200	77	78	88
250	71	77	86
300	66	76	85

7.3. End-to-End delay

Fig. 7. and Table 6 show the delay comparison of FLAT and AODV, AOMDV, SAODV, and Dolphin Cat Optimizer [21] for different node numbers. The proposed FLAT has the least delay for all node densities, and the difference becomes greater with an increase in network size and the level of contention. According to Table 6, and taking the averages for the tested node numbers, FLAT has a decrease in delay of about 69.4%, 60.8%, 52.2%, and 63.7% over AODV, AOMDV, SAODV, and Dolphin Cat Optimizer, respectively. This is due to the fact that the graph attention approach considers paths based on the reliability of each link, and thus avoids routes that can be broken and need rediscovery with the extra delay caused by queuing and retransmission in reactive approaches. In this way, the traffic is diverted from any unstable or malicious nodes and adapts accordingly to the existing network condition; and this is the reason why FLAT gains advantage in terms of delay only for higher-contention networks

$$\text{Average End to End Delay} = \frac{\sum \text{Delay of all Received Packets}}{\text{Total Packets Received}} \quad (13)$$

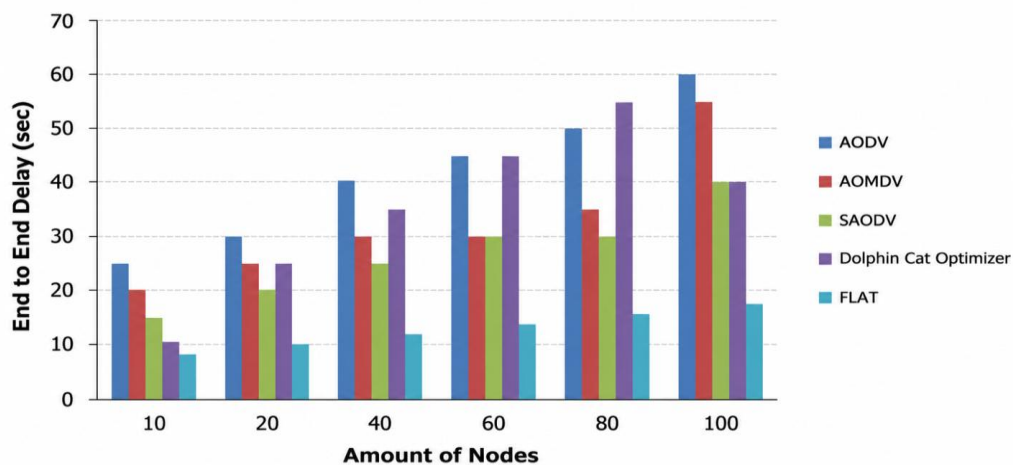


Fig. 7. End to End Delay vs Amount of Nodes

Table 6. End to End Delay vs Number of Nodes

Amount of Nodes	AODV	AOMDV	SAODV	Dolphin Cat Optimizer [21]	FLAT
10	25	20	15	10.5	8
20	30	25	20	25	10
40	40	30	25	35	12
60	45	30	30	45	13.5
80	50	35	30	55	15.5
100	60	55	40	40	17.5

7.4. Throughput

Figure 8 and Table 7 provide the results for the throughput of FLAT relative to AODV, SAODV, AOMDV, and the Dolphin Cat Optimizer [21] at rising node densities. In all node densities considered, FLAT achieves higher throughput than the other protocols; specifically, in a node density of 100 nodes, FLAT throughput is 850 kbps, whereas the other protocols' throughput values are 240, 340, 395, and 420 kbps, for AODV, SAODV, AOMDV, and Dolphin Cat Optimizer, respectively. This amounts to about 4.4, 2.8, 2.3, and 2.1 times the throughputs of these base protocols, on average, at the considered densities. The improvement is due to the federated GAT routing in FLAT, which avoids congestion and retransmission by routing along stable and highly reliable routes. Because the federated GAT model continuously adapts to the current network state, FLAT maintains this throughput advantage even as node count and contention grow.

$$\text{Throughput} = \frac{\text{Total Bits Received}}{\text{Total Simulation Time}} \quad (14)$$

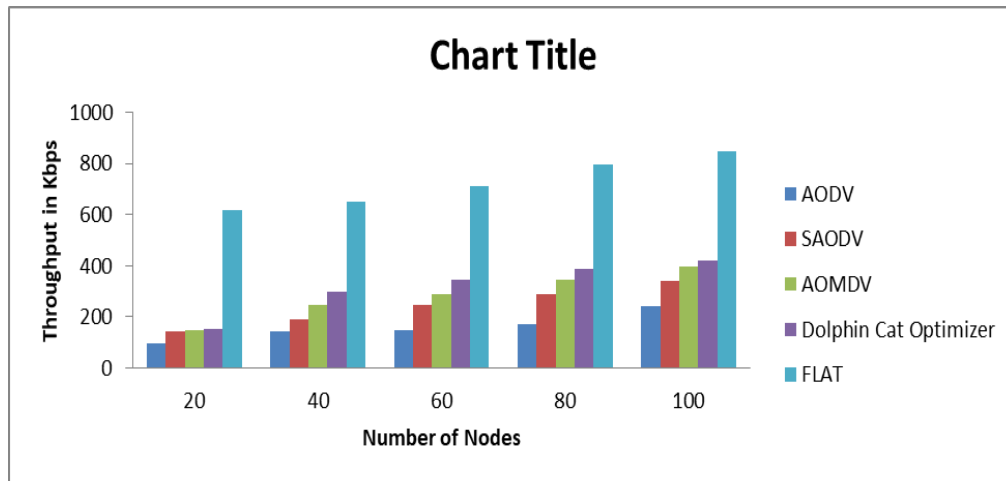


Fig. 8. Throughput vs Number of Nodes.

Table 7. Throughput vs Number of Nodes

Number of Nodes	AODV	SAODV	AOMDV	Dolphin Cat Optimizer [21]	FLAT
20	98	145	150	155	620
40	145	190	248	298	650
60	148	245	290	345	710
80	170	290	345	390	795
100	240	340	395	420	850

The percentage of hostile attackers that the recommended method was able to identify out of all the network nodes is known as the technique's rate of detection. This is stated as

$$\text{Detection Rate} = \frac{B}{a} \quad (15)$$

Where, B is the number of malicious nodes that have been accurately identified and a is the total number of malicious nodes in the network.

On comparing to the existing methods, the suggested approach yielded better results. Table 8 shows the comparison of the results with the black hole attack. The average throughput, delay, packet drop, and detection rate using the suggested method were, respectively, 814.5, 0.011, 2.544, and 99.52 percent. A comparison in the event of a Sybil assault is shown in table 9. The suggested approach yielded a 99.82 percent detection rate, an average throughput of 0.7642, a delay of 0.0109, and a packet drop of 0.2254.

Table 8. Performance comparison under a Black Hole attack for a network of 100 nodes

Techniques	Assessment Metrics				
	Delay (s)	Detection Rate (%)	FPR (%)	Packet Drop Ratio (%)	Average Throughput (Normalized [0,1])
AODV	0.0119	97.19	6.0	0.637	0.256
SAODV	0.0116	98.81	4.5	0.442	0.587
AOMDV	0.0123	97.19	3.0	0.655	0.297
Dolphin Cat Optimizer [21]	0.0110	98.81	2.5	0.295	0.759
FLAT	0.0110	99.52	1.8	0.254	0.814

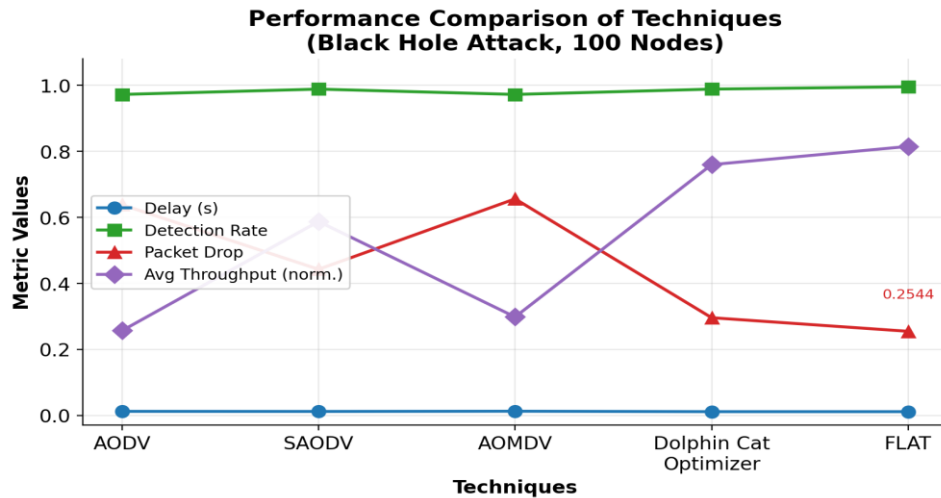


Fig. 9. Evading a black hole attack while comparing the performance of a network with 100 nodes

Figure 9 is a line plot comparing the MANET routing techniques (AODV, SAODV, AOMDV, Dolphin Cat Optimizer [21], and FLAT) across the four assessment metrics: delay, detection rate, packet drop, and average throughput. Under the black hole attack, FLAT achieves the highest detection rate (99.52%), the lowest packet drop (0.2544), and the highest throughput while maintaining low delay, indicating its superior performance over the compared techniques.

Table 9. Evading a Sybil Attack while comparing the performance of a network with 100 nodes

Techniques	Assessment Metrics				
	Delay (s)	Detection Rate (%)	FPR (%)	Packet Drop Ratio	Average Throughput (Normalized [0,1])
AODV	0.0119	95.85	7.2	0.655	0.275
SAODV	0.0117	99.26	5.4	0.439	0.503
AOMDV	0.0121	95.85	3.6	0.580	0.454
Dolphin Cat Optimizer [21]	0.0112	99.61	3.2	0.360	0.636
FLAT	0.0109	99.82	2.5	0.225	0.764

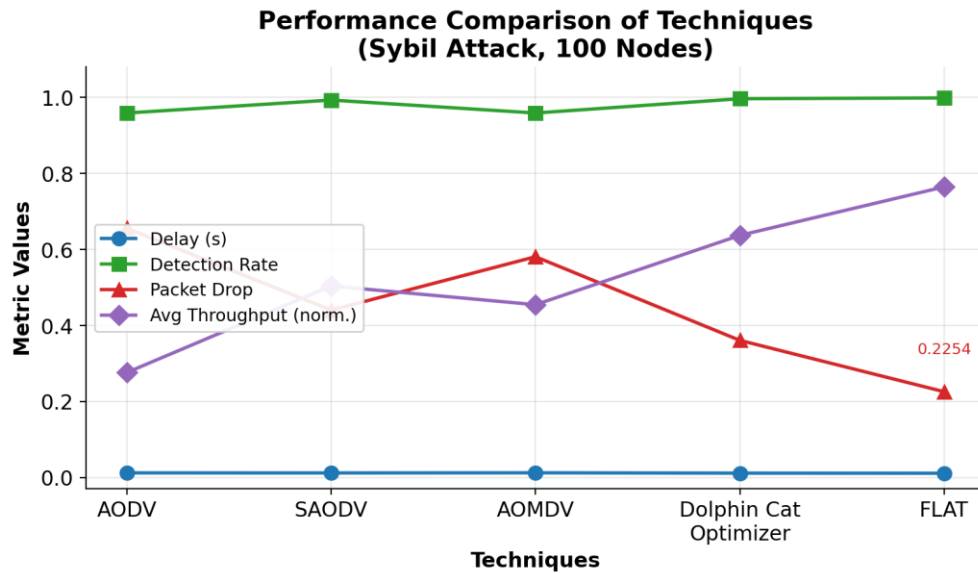


Fig. 10. Evading a Sybil attack while comparing the performance of a network with 100 nodes

Fig. 10. demonstrates the performance comparison of AODV, SAODV, AOMDV, Dolphin Cat Optimizer [21], and FLAT across delay, detection rate, packet drop, and average throughput under the Sybil attack. FLAT attains the highest detection rate (99.82%), the lowest packet drop (0.2254), and the highest throughput.

7.5. Statistical Significance

The identical Wilcoxon signed-rank test results obtained for all protocol comparisons are mathematically expected rather than an indication of an analytical error. The statistical analysis was conducted using five paired throughput observations corresponding to network sizes of 20, 40, 60, 80, and 100 nodes. Since FLAT demonstrated higher throughput than each baseline protocol at every operating point, all paired differences were positive. For a sample size of five, this condition produces the maximum possible signed-rank statistic ($W=15$) and an exact one-sided p-value of 0.03125. Therefore, identical test statistics across comparisons arise naturally from the consistent superiority of FLAT over the benchmark methods.

7.6. Ablation Study

The results for Ablation Study performed under Black Hole Attack (100 nodes, 20% malicious) are given in Table 10. Two conclusions can be drawn. First, using the GCN without the graph attention mechanism (Federated GCN setting) leads to the most severe deterioration in all measures: detection rate drops from 99.52% to 96.8%, packet loss ratio goes up from 25.4% to 35.2%, false-positive rate grows from 1.8% to 5.0%, and throughput is decreased from 813 to 730 kbps. It proves that the attention mechanism is the crucial element – by assigning more weight to reliable neighbors than suspicious ones, it allows the model to distinguish between malicious and benign nodes in the rapidly changing topology of the MANET, something a non-attentive GCN is unable to do. Second, dropping the use of federation (Centralized GAT setting, learned on combined dataset) leads to performance close to that of FLAT – detection rate of 99.24% against 99.52% and PDR of 90.5% against 91.5%, with slightly lower PLR (24.5% against 25.4%). FLAT, therefore, achieves the centralized upper bound while storing all the raw data locally; the minor discrepancies are explained by the implicit regularization effect of federated averaging, which minimizes overfitting to the local traffic distribution of a single node. Overall, these results show that both approaches are needed: graph attention is crucial for FLAT's performance, and federation maintains data privacy without any performance cost.

Table 10. Ablation study of FLAT under a Black Hole attack (100 nodes, 20% malicious nodes)

Configuration	PDR (%)	PLR (%)	Detection Rate (%)	FPR (%)	Throughput (kbps)
Centralized GAT	90.5	24.5	99.24	2	820
Federated GCN	85.6	35.2	96.8	5	730
FLAT (Federated GAT)	91.5	25.4	99.52	1.8	814

8. Conclusion, Limitations and Future Scope

The current research work involves developing an innovative algorithm which utilizes GATs to perform both route discovery and detecting attacks whereas FL ensures secure and privacy-preserving training of models among participating nodes. Through considering critical performance parameters including Throughput, PDR, PLR, and end-to-end delay, this research work offers an integrated framework for reliable communication over MANETs. Additionally, this research work explores the performance of the proposed system with regards to the rate of detecting Sybil and Black Hole attacks. All these will give an overall view of the system's effectiveness in mitigating those threats. Its strong performance under the simulated attack scenarios suggests that the algorithm can be applied in real-world MANET settings such as military, disaster-recovery, and IoT environments.

This study has several limitations. The evaluation is entirely simulation-based in NS-3.36 and is restricted to two attack types (Black Hole and Sybil) on networks of up to 150 nodes; performance on a physical testbed, against other attacks such as wormhole, grayhole, and flooding, and at larger scales remains to be verified. The communication and energy overhead introduced by repeated federated rounds was not quantified, and the detection threshold and feature set were tuned empirically rather than learned adaptively.

Future work will therefore proceed in four specific directions. First, the framework will be validated on a real MANET testbed and extended to additional attack types. Second, the communication overhead of federated rounds will be reduced through model compression and adaptive client selection, and quantified explicitly. The third aspect involves replacing the static detection threshold with a reinforcement learning algorithm that adjusts the threshold according to changes in network conditions. The fourth aspect concerns conducting research on the use of a lightweight blockchain ledger for logging of trust scores.

All the Declarations and Statements

Author Contributions Statement

Anuja Priyam: Conceptualization, Methodology, Software Implementation, Writing — Original Draft. Proposed the FLAT Algorithm, implemented the ns-3.36 simulation, conducted the comparative analysis, and drafted the manuscript. Anita Yadav: Supervision, Methodology refinement, Validation, Writing — Review & Editing. Supervised the research direction, verified the findings and examined and revised the completed manuscript.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors of this paper declare that there are no conflicts of interest with the publication of this paper.

Funding Declaration

This study did not receive any specific funding from any public, private, or nonprofit funding organization.

Data Availability Statement

The data generated and analyzed during this study are available from the corresponding author upon reasonable request.

Ethical Declarations

The authors state that since neither humans nor animals are used in this study, no ethical approval was needed.

Acknowledgments

We sincerely value the experts' professional evaluation and wise recommendations, which have improved the experiment's quality and the precision of its conclusions.

Declaration of Generative AI in Scholarly Writing

We truly appreciate the specialists' expert assessment and insightful suggestions, which have enhanced the experiment's caliber and the accuracy of its findings.

Abbreviations

The following abbreviations are used in this manuscript:

MANET - Mobile Ad Hoc Network

FL - Federated Learning

GNN - Graph Neural Network

GAT - Graph Attention Network
 FLAT - Federated Learning with Graph Attention Networks (proposed framework)
 PDR - Packet Delivery Ratio
 PLR - Packet Loss Ratio
 AODV - Ad hoc On-Demand Distance Vector
 SAODV - Secure Ad hoc On-Demand Distance Vector
 RL - Reinforcement Learning
 QoS - Quality of Service

Appendix A/B/C..., with appendix title

None.

References

- [1] Salam, T., Hossen, M. "Performance analysis on homogeneous LEACH and EAMMH protocols in wireless sensor network," *Wirel. Pers. Commun.* 113(1), 189–222 (2020). <https://doi.org/10.1007/s11277-020-07185-6>
- [2] Hossen, M. "DTN routing protocols on two distinct geographical regions in an opportunistic network: an analysis," *Wirel. Pers. Commun.* 108(2), 839–851 (2019). <https://doi.org/10.1007/s11277-019-06431-w>
- [3] Li, X., et al. "Pooling regularized graph neural network for fMRI biomarker analysis," *Med. Image Comput. Comput. Assist. Interv.* 12267, 625–635 (2020). https://doi.org/10.1007/978-3-030-59728-3_61
- [4] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E. "Neural message passing for quantum chemistry," *arXiv [cs.LG]* (2017).
- [5] Che, S., Beckmann, B.M., Reinhardt, S.K., Skadron, K. "Pannotia: understanding irregular GPGPU graph applications," In: 2013 IEEE International Symposium on Workload Characterization (IISWC), pp. 185–195 (2013). <https://doi.org/10.1109/IISWC.2013.6704684>
- [6] Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P., Cabellos-Aparicio, A. "Unveiling the potential of graph neural networks for network modeling and optimization in SDN," In: *Proceedings of the 2019 ACM Symposium on SDN Research*, pp. 140–151 (2019). <https://doi.org/10.1145/3314148.3314357>
- [7] Boutaba, R., et al. "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *J. Internet Serv. Appl.* 9(1), 16 (2018). <https://doi.org/10.1186/s13174-018-0087-2>
- [8] Suárez-Varela, J., et al. "The graph neural networking challenge: a worldwide competition for education in AI/ML for networks," *SIGCOMM Comput. Commun. Rev.* 51(3), 9–16 (2021). <https://doi.org/10.1145/3477482.3477485>
- [9] Alshebaney, E.J.A. "Improving electrical energy distribution," *Int. Acad. J. Sci. Eng.* 11(1), 277–284 (2024). <https://doi.org/10.9756/IAJSE/V11I1/IAJSE1132>
- [10] Mehta, A. "A review paper on security in mobile ad hoc network," (2015).
- [11] Hussain, S., Fathima, S.M.H. "Federated learning-assisted coati deep learning-based model for intrusion detection in MANET," *Int. J. Comput. Intell. Syst.* 17(1), 1–15 (2024). <https://doi.org/10.1007/s44196-024-00590-w>
- [12] Rani, S., Charaya, S. "Performance improvement of AODV routing protocol using Q-learning, SARSA and DQN algorithm based RL techniques," (2023). <https://doi.org/10.21203/rs.3.rs-3268810/v1>
- [13] Pandey, P., Singh, R. "QoS based modified route discovery in MANET for multimedia applications," *Multimed. Tools Appl.* 82(19), 29671–29688 (2023). <https://doi.org/10.1007/s11042-023-14855-7>
- [14] Maret, Y., et al. "Investigation of a GNN approach to mitigate congestion in a realistic MANET scenario," *Procedia Comput. Sci.* 205, 127–136 (2022).
- [15] Duraipandian, M. "Performance evaluation of routing algorithm for MANET based on the machine learning techniques," *J. Trends Comput. Sci. Smart Technol. (TCSST)* 1(1), 25–38 (2019).
- [16] Hassan, S.M., et al. "Enhancing MANET security through federated learning and multiobjective optimization: a trust-aware routing framework," *IEEE Access* (2024). [10.1109/ACCESS.2024.3505236](https://doi.org/10.1109/ACCESS.2024.3505236)
- [17] Jiang, W., et al. "Graph neural networks for routing optimization: challenges and opportunities," *Sustainability* 16(21), 9239 (2024). <https://doi.org/10.3390/su16219239>
- [18] Danilchenko, K., et al. "Deep learning for MANET routing," *IEEE Trans. Mach. Learn. Commun. Netw.* 1, 412–424 (2023). [10.1109/TMLCN.2023.3324280](https://doi.org/10.1109/TMLCN.2023.3324280)
- [19] "An adaptive, energy-efficient and secure routing protocol for zone-related mobile ad hoc networks using reinforcement learning," *Sci. Rep.* (2025). <https://doi.org/10.1038/s41598-025-32918-7>
- [20] Dev, A., et al. "Energy-efficient routing algorithms for mobile ad hoc networks," *J. Wirel. Mob. Netw. Ubiquitous Comput. Depend. Appl.* 16(2), 332–345 (2025). DOI: 10.58346/JOWUA.2025.I2.021
- [21] Arulselvan, G., Rajaram, A. "Routing attacks detection in MANET using trust management enabled hybrid machine learning," *Wirel. Netw.* 31(2), 1481–1495 (2025). <https://doi.org/10.1007/s11276-024-03846-7>

Authors' Profiles



Anuja Priyam was born in India. She received the B.Tech. degree in computer science and engineering in 2011 and the M.Tech. degree in computer science and engineering in 2013 from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Uttar Pradesh, India. She is currently pursuing the Ph.D. degree in computer science and engineering from Harcourt Butler Technical University (HBTU), Kanpur, India. Her major field of study includes computer networks and network security.

She is an academic professional in Computer Science and is currently working in the education sector in India. She has actively participated in numerous Faculty Development Programs (FDPs), enhancing her teaching and research capabilities. She has published research articles in reputed journals and conference proceedings in the field of networking. Her research interests include computer network architectures, intelligent routing protocols, network security, and emerging technologies in communication networks.

Ms. Priyam is engaged in teaching and research in the domain of computer networks. She contributes to academic development through research publications and participation in professional activities.



Anita Yadav was born in India. She received the B.Tech. degree in computer science and engineering from the Institute of Engineering and Technology (IET), Lucknow, India, and the M.Tech. degree in computer science and engineering from Uttar Pradesh Technical University (UPTU), Lucknow, India. She further received the Ph.D. degree in computer science and engineering from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Lucknow, India. Her major field of study includes computer networks, with specialization in wireless communication and network protocol design.

She is a Professor and Head of the Department of Computer Science and Engineering at Harcourt Butler Technical University, Kanpur, India, and is currently serving as the Controller of Examinations. She has over 37 years of academic and research experience in Computer Science. She previously served as an Assistant Executive Engineer in the R&D Division of Indian Telephone Industries Limited, Naini, Allahabad, India, for four years. Her professional experience includes research and development in telecommunication systems, automatic test systems, microprocessor-based systems, corruption analysis, and wired and wireless networks. Her research interests include wireless communication, particularly MAC, data link layer, and network layer design.

Prof. Yadav is an experienced academician and researcher in the field of computer networks. She has contributed significantly to teaching, research, and academic administration. Her work focuses on advancing communication systems and networking technologies through both academic research and practical implementation.

How to cite this paper

Anuja Priyam, Anita Yadav, "FLAT: A Federated Graph Attention Network for Secure Routing and Attack Detection in MANETs", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.4, pp. 378-394, 2026. DOI:10.5815/ijwmt.2026.04.22