

A Centralized Federated Framework for Black-Hole Attack Detection in Mobile Ad Hoc Networks using Gini Canberra Spearman-Based K-Means Clustering and Judy-Verkle Tree Verification

Shyamili P. V.

Department of Artificial Intelligence and Data Science, Sree Narayana Guru College of Engineering and Technology, Kannur, India

E-mail: shyamiliyamritha@gmail.com

ORCID iD: <https://orcid.org/0000-0002-2502-8937>

Anoop B. K.*

Kerala State Electronics Development Cooperation, Kerala, India

E-mail: dranoopbk@gmail.com

ORCID iD: <https://orcid.org/0000-0003-4288-5065>

*Corresponding Author

Received: May 28, 2026; Revised: June 17, 2026; Accepted: June 25, 2026; Published: August 8, 2026

Abstract: Black-hole (BH) attacks are among the most critical security threats in Mobile Ad Hoc Networks (MANETs) due to their decentralized and highly dynamic nature. Existing protection mechanisms primarily rely on threshold-based monitoring, trust evaluation, and route observation techniques, which often suffer from limited packet information, poor scalability, and inadequate attack verification capabilities. To address these limitations, this paper proposes a centralized federated framework for black-hole attack detection in MANETs. The proposed framework integrates Linear Scaling-based Shark Smell Optimization Algorithm (LS-SSOA) for Cluster Head (CH) selection, Gini Canberra Spearman-based K-Means Algorithm (GCS-KMA) for clustering, Round Log Sum BCrypt-based Message Authentication Code (RLBC-MAC) for secure route authorization, and Judy-Verkle Tree (JVT) for malicious path verification. A federated learning architecture consisting of a global server and local models is employed to continuously monitor network activities and improve attack detection efficiency. Experimental evaluation was conducted using MANET networks comprising 50–250 nodes in a Python-based simulation environment. The proposed framework achieved a detection accuracy of 98.96%, a false prediction rate of 2.06%, throughput of 6445 kbps, and reduced computational overhead compared with existing methods. The results demonstrate that the proposed centralized federated framework significantly improves network security, route reliability, and attack detection performance in MANET environments.

Index Terms: Linear Scaling based Shark Smell Optimization algorithm (LS-SSOA), Round Log sum of BCrypt based Message Authentication Code (RLBC-MAC), Gini Canberra Spearman based K-Means Algorithm (GCS-KMA), Judy Verkle Tree (JVT), Black-hole attack, Ad-hoc OnDemand Distance Vector (AODV), MANET

1. Introduction

MANETs are known for their unique characteristics like the nonexistence of centralized management, dynamic topology, and cooperative nature. However, it is highly challenging to build secure routing and protect nodes from cyber-attacks (Arunmozhi et al., 2023) (Moumen et al., 2023). BH attack or selective packet-dropping attack is the most severe of all attacks in MANET (Shafi et al., 2023) in which dropping the packets and publicizing themselves to route packets to pass via them are the malicious strategies adopted by adversarial nodes (Reddy et al., 2021). Since this causes important disruptions in communication for route discovery, ensuring the security of MANET is essential to maintain the integrity of network services (Farahani, 2021) (Hamdi et al., 2022). In the last few years, numerous studies concentrated on dealing

with BH and packet-dropping attacks. Certain approaches rely on determining the threshold of the number that shows the novelty of the path (Al-Shabi, 2022) (Boopathi&Seetha, 2021) and others rely on mobile agents (Bounouni et al., 2022). Some researchers find methods centered on eavesdropping and attaining trust betwixt nodes (Makani& Reddy, 2022). Nevertheless, when the network is decentralized, all these mechanisms become less effective to malicious nodes. Thus, through continuous monitoring of all endpoints, centralization in MANET could offer more security.

1.1. Problem Statement

Various limitations of prevailing methodologies are below:

- Owing to ignorance of monitoring any nodes' arrival or existence, the existing systems were sensitive to malicious activities.
- The predictability of actual BH attack nodes in MANET was lowered by considering minimum packet information.
- During packet transmission between the nodes, the existing system didn't have a proper verification technique.
- Frequent packet drops caused lower throughput in these networks.

Motivated by these limitations, the proposed model contributes the following:

- To introduce centralization for monitoring a node's arrival or existence.
- To add more packet information in predicting the actual attack node in MANET.
- To have proper authorization utilizing the RLBC-MAC algorithm.
- To improve the throughput of the network utilizing JVT-based attack detection.

The paper's remaining is organized as; related studies are demonstrated in Section 2, the proposed approach is detailed in Section 3, the experimental study is given in Section 4, and the conclusion is clarified in Section 5.

1.2. Research Gap and Contributions

Although several approaches have been proposed for black-hole attack detection in Mobile Ad Hoc Networks (MANETs), most existing methods rely on threshold-based route monitoring, trust evaluation mechanisms, mobile agents, or routing protocol modifications. These approaches are effective only under specific network conditions and often suffer from limitations such as inadequate monitoring of node entry and exit, insufficient packet-level verification, increased routing overhead, and poor scalability in highly dynamic network environments. Furthermore, recent studies have primarily focused on distributed detection mechanisms, while the advantages of centralized monitoring combined with federated intelligence remain largely unexplored. In addition, secure route authorization and efficient malicious path verification have received limited attention in existing MANET security frameworks.

To address these limitations, this work proposes a centralized federated framework for black-hole attack detection in MANETs. The proposed framework combines cluster-head optimization, intelligent clustering, secure route authentication, and malicious path verification to enhance network security and communication reliability.

The major contributions of this work are summarized as follows:

- A centralized federated monitoring framework is introduced to continuously supervise node participation and network activities in MANET environments.
- A Linear Scaling-Based Shark Smell Optimization Algorithm (LS-SSOA) is employed for optimal Cluster Head (CH) selection to improve cluster stability and communication efficiency.
- A Gini Canberra Spearman-Based K-Means Algorithm (GCS-KMA) is proposed for efficient node clustering while handling variations in node density and network size.
- A Round Log Sum BCrypt-Based Message Authentication Code (RLBC-MAC) mechanism is developed to provide secure route authorization and prevent unauthorized packet forwarding.
- A Judy-Verkle Tree (JVT)-based verification framework is designed to identify malicious routing paths with reduced computational overhead and improved scalability.
- Comprehensive experimental evaluation demonstrates the effectiveness of the proposed framework in terms of detection accuracy, throughput improvement, response time reduction, and attack mitigation capability.

2. Related Literature Survey

(Mankotia et al., 2023) presented a Dynamic Threshold-Ad-hoc On-Demand Distance Vector (DT-AODV) protocol in MANET. The dynamic threshold value identified the malicious node in route discovery. Even though DT-AODV had better performance, the dynamic threshold was not appropriate as the change in consecutive responses caused misleading.

(Ramachandran et al., 2022) enhanced the AODV to detect BH attacks in MANET. The timing and baiting strategies improved the AODV. The approach considerably outperformed previous techniques. Yet, owing to the facility to follow multiple paths, the energy consumption was higher.

(Vijayalakshmi et al., 2023) found the malicious packet-dropping attack in MANET. The model utilized a host-based detection system to observe node behavior and update the trust value for decision-making. An increased Packet Delivery Ratio (PDR) was accomplished by the system. The bandwidth consumption was augmented by the dynamic change in routing policy.

(Veeraiah et al., 2021) recommended a trust-centric secure energy-effectual navigation in MANET. The detection procedure was centered on the predefined threshold value and the Cat Slap Single-player Algorithm (C-SSA). Even though the method obtained a maximum detection rate, it ignored the path containing no malicious nodes, which may result in high routing overhead.

(Kancharakuntla & El-Ocla, 2022) presented an Enhanced Black-hole Resistance (EBR) protocol in MANET. The Time To Live (TTL) and Round Trip Time (RTT) mechanisms performed BH attacks' detection. As per the simulation outcomes, EBR behaved better compared to other protocols. The key limitation of the model was utilizing the minimum number of packet information.

3. Proposed Manet Attack Detection Model

To overcome the issue of detecting BH or selective packet-dropping attacks, this paper concentrated on centralization in MANET. In Figure 1, the proposed model's schematic diagram is depicted,

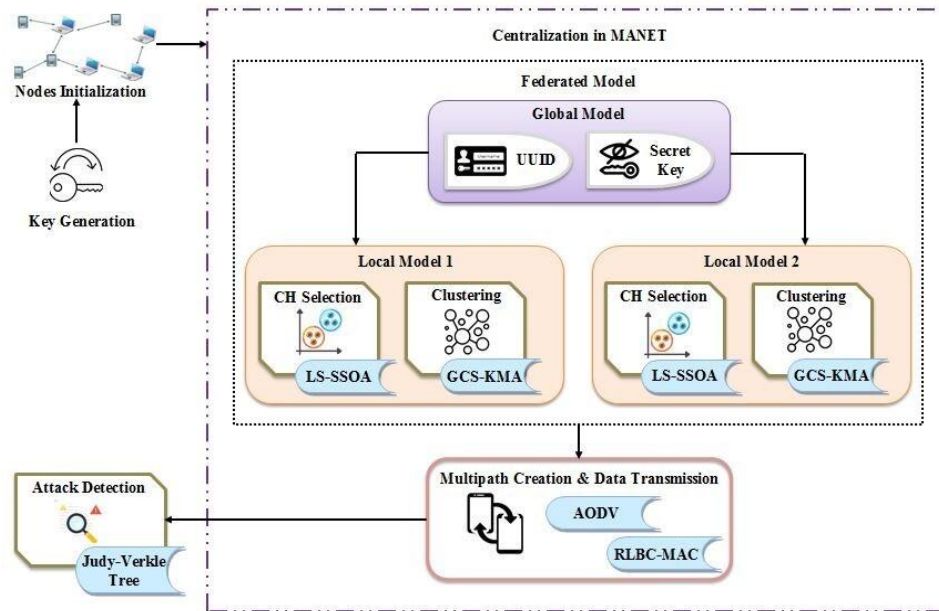


Fig. 1. Flow diagram of the proposed model

3.1 Nodes Initialization

To run the attack detection module, the set of mobile nodes ($M_{(n)}$) that create MANET is initialized along with their key generator(P).

$$M_{(n)} = \{M_{(1)}, M_{(2)}, \dots, M_{(N)}\} \quad (1)$$

$$P = \{P_{key(n)} | n \in M_{(n)}\} \quad (2)$$

Here, the public key of each node is signified as($P_{key(n)}$).

3.2. Federated Learning Framework

The proposed centralized architecture is implemented using a federated learning framework consisting of a Global Server (GS) and multiple Local Models (LMs). Each local model is responsible for monitoring the nodes belonging to its cluster and periodically updating the global server with local observations. Unlike conventional centralized approaches that require complete data sharing, the federated framework only exchanges model parameters, thereby reducing communication overhead and preserving local node information.

Initially, the Global Server initializes the global model parameters and distributes them to all local models. Each local model performs local training using the node information available within its cluster. After local training, the updated model parameters are transmitted to the Global Server for aggregation. The Global Server generates a new global model using the Federated Averaging (FedAvg) algorithm and redistributes the updated parameters to all participating clusters.

The aggregation process is mathematically expressed as:

$$W(t+1) = \sum_{i=1}^K (n_i / N) W_i(t)$$

where:

$W(t+1)$ = Updated global model,
 $W_i(t)$ = Local model parameters of cluster i ,
 n_i = Number of training samples in cluster i ,
 N = Total number of training samples,
 K = Total number of participating clusters.

This iterative aggregation process enables continuous learning of network behavior and improves the identification of malicious nodes while maintaining scalability in dynamic MANET environments.

3.2.1. CH Selection

By utilizing LS-SSOA, CH selection for the local models is performed. SSOA is selected because of its ability to effectively balance exploration and exploitation during optimization. However, the conventional inertia factor may reduce convergence speed during iterative optimization, since its velocity relies only on its prior velocity. Therefore, the linear scaling technique, which effectively speeds up the convergence by scaling the velocity as per higher odor intensity, is utilized for the velocity updation phase.

The LS-SSOA algorithm starts by producing the initial solutions (i.e., the number of nodes to select CH) and moving towards the strong odor with a velocity. The initial velocity vectors (ϖ_n) are,

$$\varpi_n = \{\varpi_1, \varpi_2, \varpi_3, \dots, \varpi_N\} \quad (3)$$

After that, the velocity by considering the speed limit ($\varpi_{(n,k)}$) is provided as,

$$\varpi_{(n,k)} = \min \left[\left| a_k \cdot h \partial \left(f_{ob(M_{(n)})} \right) \right| + b_{ls(k)} \cdot h \cdot \varpi_{(n,k-1)} \left| c_k \cdot \varpi_{(n,k-1)} \right| \right], k = 1, 2, \dots, K \quad (4)$$

$$b_{ls(k)} = \Omega * \varpi_{(n,k)} + \theta \quad (5)$$

Where, the random values are denoted as $(a_k, h) \in (0,1)$, the gradient to indicate direction is represented as $\partial \left(f_{ob(M_{(n)})} \right)$, the velocity limit ratio is notated as c_k , the inertia coefficient determined using the LS technique is specified as $b_{ls(k)}$, Ω indicates the multiplier to scale velocity, the offset is denoted as θ , and the previous velocity is given as $\varpi_{(n,k-1)}$.

The sharks' foraging behavior is explained by forwarding and rotational behaviors and the updating process. The shark has a new position ($M_{fnew(n,k)}$) in forward moving centered on its prior position ($M_{(n-1,k)}$) and ($\varpi_{(n,k)}$).

$$M_{fnew(n,k)} = M_{(n-1,k)} + \varpi_{(n,k)} \cdot \nabla t \quad (6)$$

Where, the time constant is represented as ∇t .

The rotating shark follows the local search strategy in rotational movement.

$$M_{rnew(n,k)} = M_{fnew(n,k)} + M_{fnew(n,k)} \cdot h \quad (7)$$

Here, the new position in the local search is specified as ($M_{rnew(n,k)}$).

Algorithm 1: LS-SSOA-based CH selection

Input: number of nodes $M_{(n)}$
Output: Selected CHs $M_{CH(opt)}$

Begin
Initialize input population $M_{(n)}$, velocity $\varpi_{(n)}$, number of iteration k_{max}
Evaluate fitness $fit(M_{(n)})$
For each node i do (k)
Calculate $M_{fnew(n)}$ using $\varpi_{(n,k)}$ and $M_{(n-1,k)}$
Obtain $M_{rnew(n)}$ using $M_{rnew(n)}$
Compute ($fit(M_{fnew(n,k)}, M_{rnew(n,k)})$)
Update $M_{CH(opt)}$
End for
Return $M_{CH(opt)}$
End

Likewise, by assuming a min-max problem as an objective function ($fit(M_{fnew(n,k)}, M_{rnew(n,k)})$), the optimal CHs ($M_{CH(opt)}$) with minimum distance ($dist(M_{fnew(n,k)}, M_{rnew(n,k)})$) and maximum energy ($\Im(M_{fnew(n,k)}, M_{rnew(n,k)})$) are selected.

$$M_{CH(opt)} = minmax\{fit(M_{fnew(n,k)}, M_{rnew(n,k)}) + fit(M_{fnew(n,k)}, M_{rnew(n,k)})\} \quad (8)$$

$$fit = dist(M_{fnew(n,k)}, M_{rnew(n,k)}) + \Im(M_{fnew(n,k)}, M_{rnew(n,k)}) \quad (9)$$

After selecting the CHs, the global model generates a UUID and a secret key to each CH to monitor outside node entry, rejoining existing nodes, etc.

3.2.2. Secret Key Generation

By utilizing the Diffie-Hellman algorithm, a secret key is generated for each CH. To compute the secret key, the algorithm selects an integer (x) and a prime number (ϕ).

$$\xi_{key} = (P_{key(CH,G)})^x \bmod \phi \quad (10)$$

Here, the secret key generated for each CH is notated as ξ_{key} .

3.2.3.0 Clustering

Next, by utilizing GCS-KMA, clustering is done. Owing to the K-Means algorithm's faster computation, it is chosen. However, it troubles to cluster data of varying sizes and densities. Thus, the Gini Canberra Spearman distance, which efficiently measures the strength of relationships when the data has different sizes, is utilized.

In clustering, the nodes are grouped under respective CHs ($\{M_{CH(c)}\}_{c=1,2,3,\dots,C}$) for forming (C) clusters ($\chi_{cl(c)}$) by decreasing the difference between ($M_{(n)}$) and ($M_{CH(opt)}$).

$$\chi_{cl(c)} = min \sum_{c=1}^C \sum_{n=1}^N \delta_{cn} \varphi(M_{CH(c)}, M_{(n)}) \quad (11)$$

$$\phi(M_{CH(c)}, M_{(n)}) = \sum_{n,c} \frac{|M_{CH(c)} - M_{(n)}|}{\sqrt{|M_{CH(c)} + M_{(n)}|}} \quad (12)$$

Where, the variable to determine whether the node belongs to the cluster or not is specified as δ_{cn} , and the GCS distance between ($M_{CH(c)}, M_{(n)}$) is indicated as $\varphi(M_{CH(c)}, M_{(n)})$.

Algorithm 2: GCS-KMA-based clustering

Input: number of nodes and CHs ($M_{CH(c)}, M_{(n)}$)

Output: Final clusters ($\chi_{cl(c)}$)

Begin

Initialize ($M_{CH(c)}, M_{(n)}$), number of clusters (C)

For each node i do ($M_{CH(c)}, M_{(n)}$)

Calculate distance to each node

$$\varphi(M_{CH(c)}, M_{(n)}) = \sum_{n,c} \frac{(|M_{CH(c)} - M_{(n)}|)}{\sqrt{|M_{CH(c)} + M_{(n)}|}}$$

End for

Assign the node to $M_{CH(c)}$ with minimum distance

End for

Return ($\chi_{cl(c)}$)

End

3.3. Multipath Creation and Data Transmission

Multiple paths between the nodes in a cluster are created after centralization using the AODV along with authorization to identify malicious nodes in MANET.

Step 1: If ($M_{s(n) \in \chi_{cl(c)}}$) is a node that requests a route to its destination ($M_{d(n) \in \chi_{cl(c)}}$), then it broadcasts a request ($\mathfrak{R}_{req}$) message to all its neighbors ($M_{ng(n) \in \chi_{cl(c)}}$).

Step 2: On receiving ($\mathfrak{R}_{req}$), the neighbor node sends a reply ($\mathfrak{R}_{rep}$) for the availability of a valid route to ($M_{d(n) \in \chi_{cl(c)}}$) or further forwards a ($\mathfrak{R}_{req}$) message in case the route is unavailable. This comprises two vital phases as follows,

Update Routing Table: This structure stores every candidate route in which each node updates the following fields of the routing table, such as Packet Sequence Number, Source ID, Destination ID, Hop Count, Path ID, Time Response, Lifetime, Reply Time, Previous Hop Address, Next Hop Address, and Signature Verified, which are determined from the control packets and adds a reverse route entry.

Authorization: Each node creates a hash code for authorization utilizing the RLBC-MAC technique and matches it with the hashcode of the message for its authorization. A lengthy hashcode is generated utilizing the log sum computation-based BCrypt function, which increases the complexity and time to be cracked by any malicious behavior, since using a shorter hash code can create a birthday attack.

The RLBC-MAC takes a message ($\mathfrak{R}_{req}$) comprising the Source Node ID, previous hop node ID, UUID of CHs, next hop node ID, and Destination Node ID, and the secret shared key ($\xi_{sec}()$) as input to the BCrypt hashing function (Φ_{hs}), which provides the final hashcode ($H_{\mathfrak{R}}$).

$$H_{\mathfrak{R}} = \Phi_{hs}(\tau * \varsigma * \sigma * (\mathfrak{R}_{req}, \xi_{sec}())()) \quad (13)$$

$$\sigma = \log(P_{key(sen)} \cdot P_{key(rec)}) \quad (14)$$

Where, the cost and salt parameters are signified as τ and ς , respectively, and the log sum of the sender and receiver public key ($P_{key(sen)}, P_{key(rec)}$) is exemplified as σ .

Afterward, the source node forwards ($\mathfrak{R}_{req}$) along with the hashcode ($H_{\mathfrak{R}(sen)}$), and the receiver re-calculates the receiver hashcode ($H_{\mathfrak{R}(rec)}$) for authorization.

$$M_{ng(n, val)} = \begin{cases} M_{ng(n, val)} & \text{if } (H_{\mathfrak{R}(sen)} = H_{\mathfrak{R}(rec)}) \\ M_{ng(n, mal)} & \text{Otherwise} \end{cases} \quad \xi_{key} = (P_{key(CH, G)})^x \bmod \phi \quad (15)$$

Here, the valid and malicious nodes are specified as ($M_{ng(n, val)}, M_{ng(n, mal)}$).

Step 4: This process continues till $(M_{d(n) \in \chi_{cl(C)}})$ receives $(\mathfrak{R}_{req})$ and generates $(\mathfrak{R}_{rep})$. In such a manner, multiple paths $\eta_{p(M_{s(n)}, M_{d(n)})}$ between $(M_{s(n)}, M_{d(n)})$ in each $(\chi_{cl(C)})$ are developed.

$$\eta_{p(M_{s(n)}, M_{d(n)})} = \left\{ \eta_{p(M_{s(n)}, M_{d(n)})} | (M_{s(n)}, M_{d(n)}) \in \chi_{cl(C)}, p = 1, 2, \dots, P \right\} \quad (16)$$

Subsequently, data transmission takes place in a way of multipath creation utilizing AODV in which authorization is to detect intermediate attacks that modify data transmitting between nodes.

3.4. Attack Detection

The route error($\mathfrak{R}_{err}$) message is sent back to all nodes to initiate malicious path discovery utilizing JVT for attack detection when the malicious node is detected. Verkle-tree is utilized for its scalability. Nevertheless, the Verkle tree's longer depth in node-by-node computation takes more time for generating proofs. Thus, by utilizing the Judy Array, the root hash is generated; also, the Judy array is faster and memory-efficient to process a large number of data.

If $(M_{s(n) \in \chi_{cl(C)}})$ is the root node, then it serves as the digest $(H_{\mathfrak{R}(dig)})$ for storing the hashes of all nodes in its path into Judy arrays (JA) as,

$$H_{\mathfrak{R}(dig)} = \{H(M_{ng(1)}), H(M_{ng(1)}), \dots, H(M_{ng(1)})\} \quad (17)$$

$$JA = \{J(H_{\mathfrak{R}(dis)}), D((H_{\mathfrak{R}(dis)}))\} \xi_{key} = (P_{key(CH,G)})^x \bmod \phi \quad (18)$$

Where, the levels and depth of the tree are signified as $J(H_{\mathfrak{R}(dis)}), D((H_{\mathfrak{R}(dis)}))$.

The branching factor that groups $(\eta_{p(M_{s(n)}, M_{d(n)})})$ into (K) subsets is selected while computing the Verkle tree over the paths. The vector commitment $(V_{(H,M)})$ and Verkle proof $(Z_{(p(val))})$ are computed for each $(\eta_{p(M_{s(n)}, M_{d(n)})})$ from the leaf to the Verkle root when the malicious node is detected.

The vector commitment hashes up the siblings of every node in the path from the leaf to the Verkle root. After the Verkle tree is computed, the malicious path is identified by checking the following:

$$Z_{(p(val))} = \begin{cases} \eta_{p(val)}, & \text{if } (H_{\mathfrak{R}(dig)} = V_{(H,M)}) \\ \eta_{p(mal)}, & \text{if } (H_{\mathfrak{R}(dig)} \neq V_{(H,M)}) \end{cases} \quad (19)$$

$$V_{(H,M)} = \{1, \lambda, V_1, V_2\} \quad (20)$$

Here, the valid and malicious paths are represented as $(\eta_{p(val)}, \eta_{p(mal)})$, and the stem to cover all the nodes to that root is denoted as λ . The attacks can be identified from the tree. Therefore, researchers can sense and avert the BH attacks from the mentioned network by the proposed routing mechanism.

4. Experimental Setup

The proposed MANET attack detection framework was implemented and evaluated using Python 3.10. The simulation environment consisted of Mobile Ad Hoc Networks containing 50 to 250 mobile nodes deployed within a dynamically changing network topology. The Random Waypoint Mobility Model was adopted to emulate realistic node movement patterns. The transmission range of each node was fixed at 250 m, while the packet size was maintained at 512 bytes throughout the simulation.

The simulation duration was set to 1000 seconds, and multiple communication sessions were executed to evaluate the robustness of the proposed framework under varying network conditions. Cluster formation was performed using the proposed GCS-KMA algorithm, while Cluster Head selection was carried out using LS-SSOA. Secure route establishment and authentication were implemented using the RLBC-MAC mechanism, and malicious path verification was performed using the Judy-Verkle Tree (JVT) structure.

The performance of the proposed framework was evaluated using the following metrics:

- Detection Accuracy (%)
- False Prediction Rate (FPR) (%)
- Detection Delay (ms)
- Throughput (kbps)

- Packet Delivery Ratio (PDR) (%)
- Packet Loss Ratio (PLR) (%)
- Response Time (ms)
- Energy Consumption (mJ)

The proposed methodology was compared against several existing approaches, including Verkle Tree, Merkle Tree, Radix Tree, Huffman Tree, K-Means, CLARA, PAM, FCM, and other state-of-the-art MANET black-hole attack detection frameworks reported in the literature.

4.1. Performance Analysis

In this, the proposed techniques' performances are analogized with the traditional methodologies.

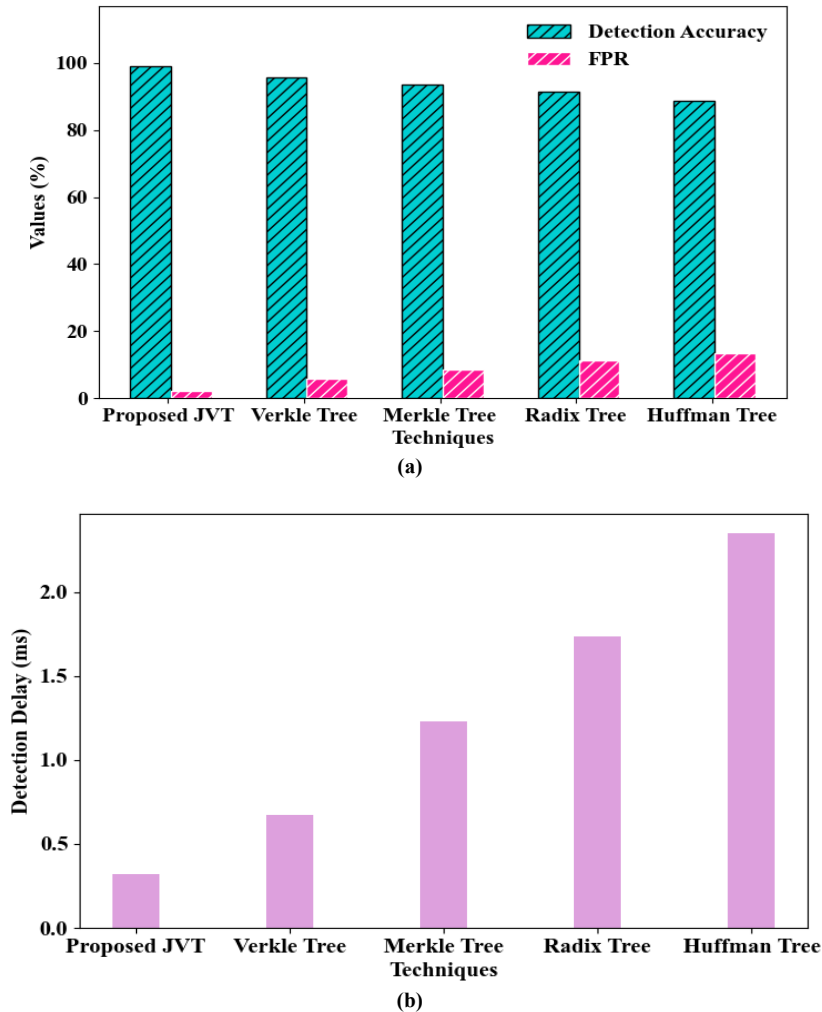


Fig. 2. (a) detection accuracy and False Prediction rate (FPR) (b) detection delay

The performance of the proposed JVT and prevailing Verkle Tree, Merkle Tree, Radix Tree, and Huffman Tree is investigated in Figure 2. The proposed method's detection accuracy and FPR are 98.96% and 2.06%, respectively. Moreover, the proposed approach detects an attack within 0.32ms. However, when compared to the proposed method, the existing methods' performance is poor. Hence, the centralized monitoring of network nodes and constructing a large number of data into a reduced-size tree need lesser computation effort when compared to checking node by node without considering whether it has a BH attack node or not.

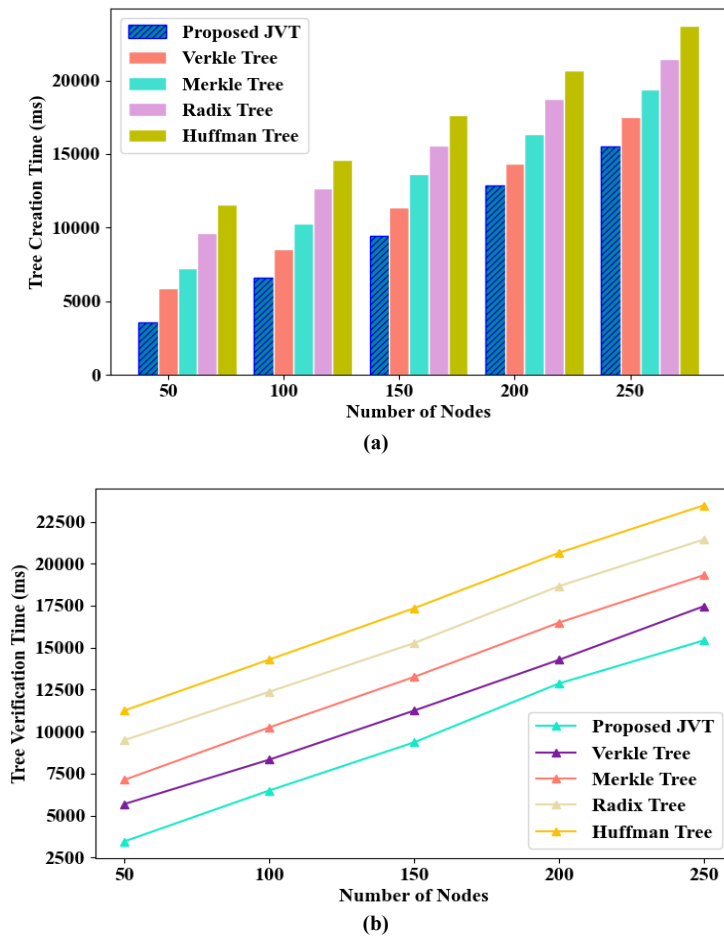


Fig. 3. (a) Tree creation time and (b) Tree verification time

In Figure 3, the tree creation time and verification time are evaluated based on the sensor nodes ranging from 50 to 250. For the tree creation and verification of 50 nodes, the proposed JVT takes 3547ms and 3458 ms, while the prevailing approaches take a larger time compared to the proposed technique. Thus, it can be said that the Judy arrays are faster and memory-effectual to scale up a large number of elements.

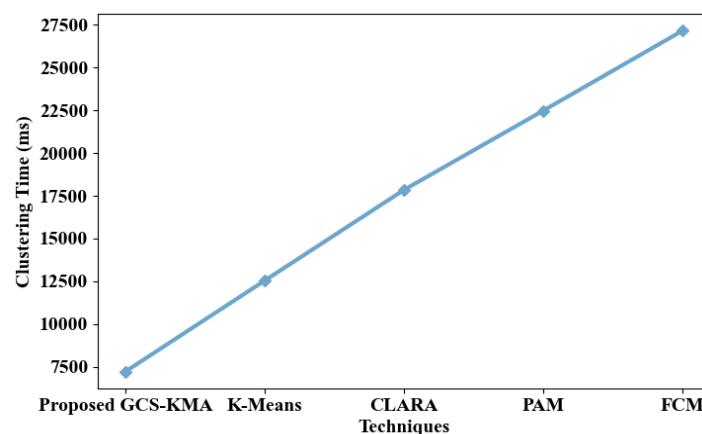
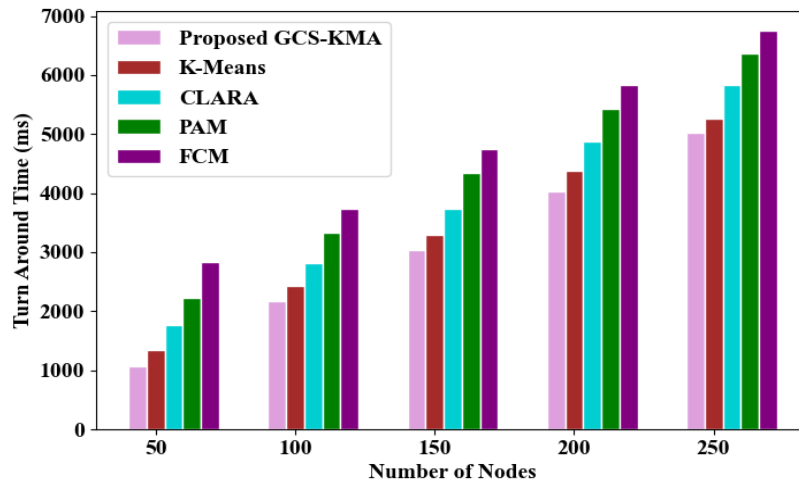
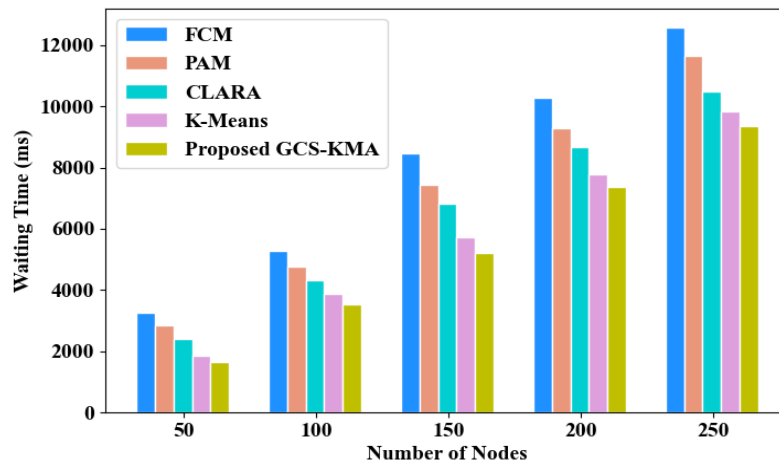


Fig. 4. Clustering Analysis

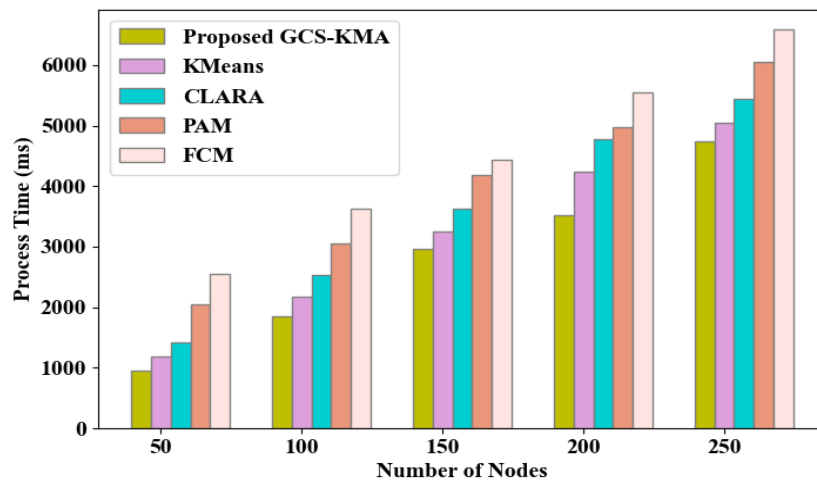
The clustering time of the proposed GCS-KMA and the existing Kmeans, Clustering LARge Applications (CLARA), Partition AroundMedoid (PAM), and Fuzzy C-Means (FCM) is displayed in Figure 4. 7235 ms is taken by the proposed GCS-KMA for clustering, which is faster when analogized to the prevailing methods.



(a)



(b)



(c)

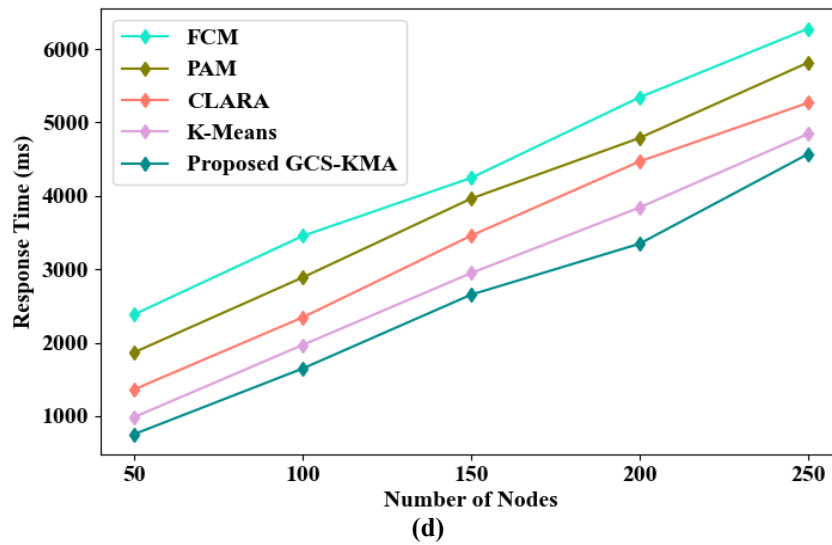


Fig. 5. Analysis of (a) Turn-Around Time, (b) Waiting Time, (c) Process Time, and (d) Response Time

In Figure 5, the result comparison of existing and proposed clustering methods is illustrated. The proposed technique consumes less time than other existing techniques for 250 nodes with a turn-around time of 5027 ms, waiting time of 9345 ms, process time of 4735 ms, and response time of 4570 ms.

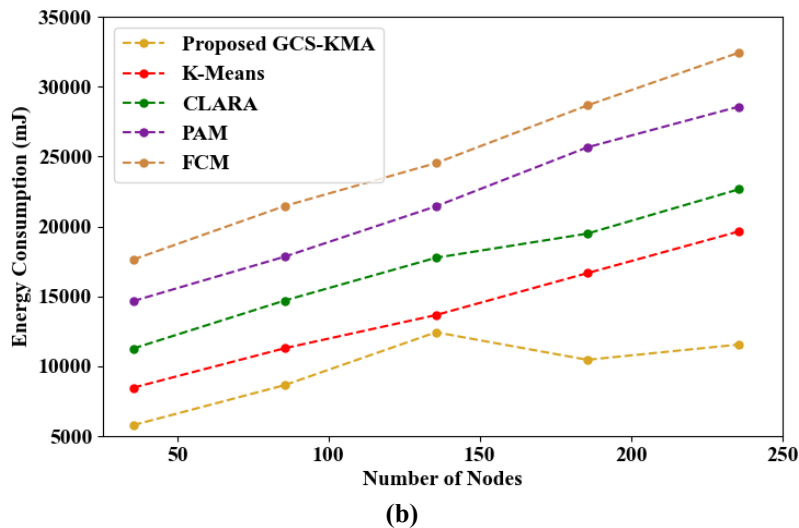
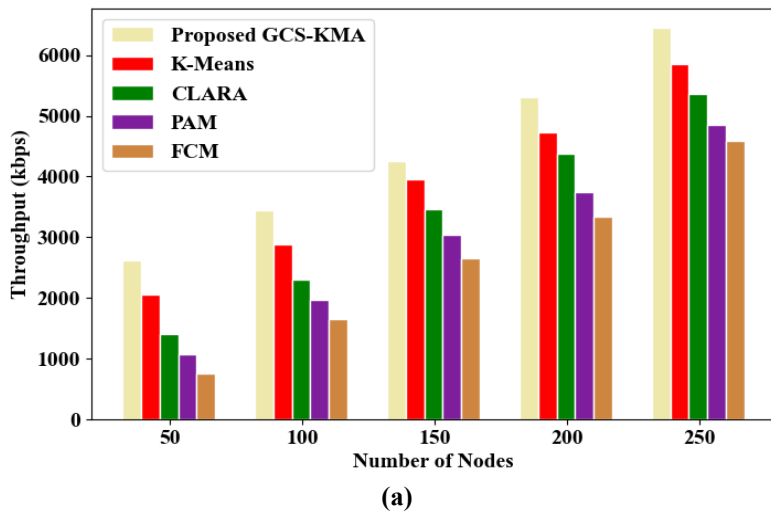


Fig. 6. (a) Throughput and (b) Energy consumption

The performance of the proposed and existing methods based on varying nodes from 50-250 is shown in Figure 6. When compared to the prevailing approaches, the total energy consumption of the proposed GCS-KMA (5784 mJ) is much lower; the throughput attained by the proposed model (6445 kbps) is higher than others. According to the outcomes, when using GCS-KMA, the network is highly efficient and consistent in grouping numerous data irrespective of their sizes and density.

Table 1. Performance analysis of proposed RLBC-MAC

Methods	Hashcode Generation Time (ms)
Proposed RLBC-MAC	487
BCrypt	865
Tiger	1258
SHA512	1687
MD5	2014

In Table 1, the performance of the proposed and conventional BCrypt, Tiger, Secure Hashing Algorithm (SHA-512), and Message Digest (MD-5) algorithms is illustrated. From Table 1, it is shown that the proposed RLBC-MAC takes less time (487 ms) for hashcode generation, which is faster than the existing methods. Thus, the proposed algorithm is more effective owing to the utilization of the RLBC method for decreasing the number of parameters to hash messages.

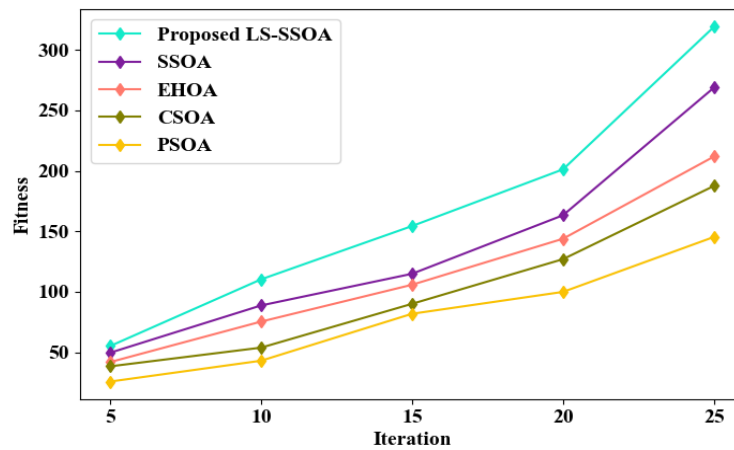


Fig. 7. Fitness Vs Iteration

The fitness of the proposed method and existing SSOA, Elephant Herding Optimization Algorithm (EHOA), Crow Search Optimization Algorithm (CSOA), and Particle Swarm Optimization Algorithm (PSOA) is compared in Figure 7. When compared to the prevailing approaches, the proposed technique attained a higher fitness level (318.95) for the 25th iteration.

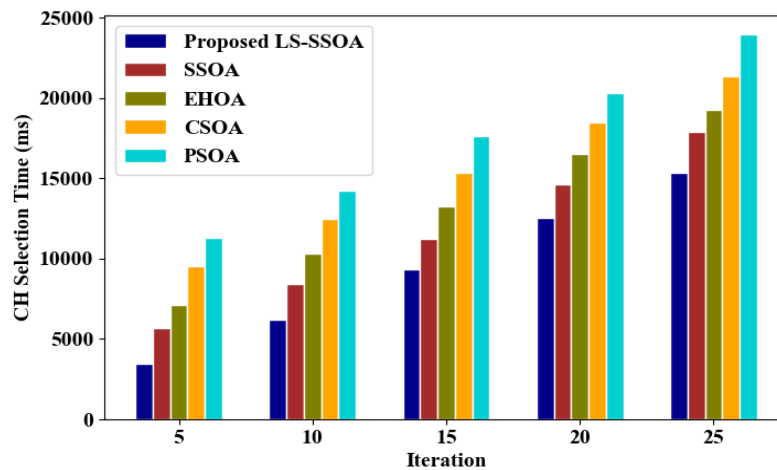


Fig. 8. Analysis of CH selection time

In Figure 8, the proposed method's CH selection time increases as the number of nodes increases from 5 to 25. In comparison, the proposed method is 2567ms faster compared to the prevailing approaches for selecting CH from 25 nodes. This shows that the optimal prediction nature of the proposed LS-SSOA is increased by the LS technique.

4.2. Comparative Analysis

In this section, the proposed model is compared with prevailing attack detection works in the literature.

Table 2. Comparative Analysis

Author Name	Method Used	Performance Metrics			
		Throughput (kbps)	Detection Rate (%)	PDR (%)	PLR (%)
proposed	GCS-KMA	6445	98.96	98.9	1.21
(Khalladi et al., 2021)	SplittedAdaptativeACKnowledgment	-	-	93.28	-
(Thebiga & Suji Pramila, 2020)	Correlation Coefficient-Based Approach	-	95.6	98.1	1.84
(Sivanesh&Dhulipala, 2021)	Accurate and Cognitive Intrusion Detection System	289	-	78	10
(Srilakshmi et al., 2021)	Genetic Algorithm with Hill Climbing	0.00085	91	89	-
(Srilakshmi et al., 2022)	Bacteria for Aging Optimization Algorithm (BFOA)	0.0007	83	-	-

It can be observed from Table 2 that the proposed methodology's performance is far better than the prevailing methodologies regarding throughput, detection rate, (PDR), and Packet Loss Ratio (PLR). The PDR and PLR are improved in the prevailing works; however, because of utilizing minimum packet information for attack detection, unauthorized data transmission, and improper monitoring of nodes' arrival and existence, they have a higher detection error. When the number of nodes and the hops between the nodes increase, it takes more time to detect the malicious node. This has been enhanced in the proposed approach by the centralized server that monitors the network nodes, more packet information to detect malicious nodes, and authorized data transmission, thus showing the proposed model's superiority.

5. Conclusion

In this paper, a centralized framework for detecting BH attacks in MANET is proposed by including clustering via GCS-KMA, authorized path discovery using RLBC-MAC, and attack detection using JVT. The proposed JVT and GCS-KMA methods are compared with the prevailing techniques for verifying the system's effectiveness. As per the experimental outcomes, a detection rate of 98.96% was attained by the proposed JVT. Conversely, the proposed method's throughput is improved by 599 kbps, and response time is reduced to 754 ms. Nevertheless, the major limitation of the proposed model is selecting a more efficient path for data transmission.

Recommendation for future work: For detecting diverse sorts of attacks, the proposed methodology will be extended in the future by incorporating optimal path selection.

All the Declarations and Statements

Author Contributions Statement

Shyamily P V: Conceptualization, Methodology, Investigation, Literature Survey, Validation, Writing – Original Draft Preparation, and Supervision.

Anoop B K: Software Implementation, Formal Analysis, Data Curation, Experimental Design, Visualization, Writing – Review and Editing, and Project Administration.

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

Conflict of Interest Statement

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Funding Declaration

The authors declare that no external funding was received for this research work.

Data Availability Statement

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

Ethical Declarations

This research does not involve human participants, human data, human tissues, or animals. Therefore, ethical approval and informed consent were not required.

Acknowledgments

The authors sincerely thank the reviewers and editors for their valuable comments, constructive suggestions, and professional evaluation, which significantly improved the quality, technical presentation, and clarity of this manuscript.

Declaration of Generative AI in Scholarly Writing

During the preparation of this manuscript, generative artificial intelligence tools were used only for language refinement, grammar improvement, formatting assistance, and manuscript editing. All technical content, methodology, experimental design, results, analysis, and conclusions were developed, verified, and approved by the authors. The authors take full responsibility for the accuracy, originality, and integrity of the final manuscript.

Abbreviations

The following abbreviations are used in this manuscript:

MANET – Mobile Ad Hoc Network

BH – Black Hole

LS-SSOA – Linear Scaling-Based Shark Smell Optimization Algorithm

GCS-KMA – Gini Canberra Spearman-Based K-Means Algorithm

RLBC-MAC – Round Log Sum BCrypt-Based Message Authentication Code

JVT – Judy-Verkle Tree

CH – Cluster Head

GS – Global Server

LM – Local Model

UUID – Universally Unique Identifier

AODV – Ad Hoc On-Demand Distance Vector

FPR – False Prediction Rate

PDR – Packet Delivery Ratio

PLR – Packet Loss Ratio.

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

None.

References

- [1] M. Al-Shabi, "Performance Ameliorations of AODV by Black Hole Attack Detection Utilizing IDSAODV as Well as Reverse AODV," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 14, no. 2, pp. 99–110, Aug. 2022, doi: <https://doi.org/10.17762/ijcnis.v14i2.5483>.
- [2] S. A. Arunmozhi, S. Rajeswari, and Y. Venkataramani, "Swarm Intelligence Based Routing with Black Hole Attack Detection in MANET," *Computer Systems Science and Engineering*, vol. 44, no. 3, pp. 2337–2347, 2023, doi: <https://doi.org/10.32604/csse.2023.024340>.
- [3] M. Boopathi and R. Seetha, "Accurate Detection of Multi-layer Packet Dropping Attacks Using Distributed Mobile Agents in MANET," *Journal of Physics: Conference Series*, vol. 1979, no. 1, p. 012040, Aug. 2021, doi: <https://doi.org/10.1088/1742-6596/1979/1/012040>.
- [4] M. Bounouni, L. Bouallouche-Medjkoune, A. Beraza, and A. Daoud, "Eliminating Selective Dropping Attack in Mobile Ad Hoc Network," *Wireless Personal Communications*, vol. 123, no. 4, pp. 3291–3308, Nov. 2021, doi: <https://doi.org/10.1007/s11277-021-09289-z>.
- [5] G. Farahani, "Black Hole Attack Detection Using K-Nearest Neighbor Algorithm and Reputation Calculation in Mobile Ad Hoc Networks," *Security and Communication Networks*, vol. 2021, pp. 1–15, Aug. 2021, doi: <https://doi.org/10.1155/2021/8814141>.
- [6] Mustafa Maad Hamdi *et al.*, "A study review on Gray and Black Hole in Mobile Ad Hoc Networks (MANETs)," *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Jun. 2022, doi: <https://doi.org/10.1109/hora55278.2022.9800011>.
- [7] D. Kancharakuntla and H. El-Ocla, "EBR: Routing Protocol to Detect Blackhole Attacks in Mobile Ad Hoc Networks," *Electronics*, vol. 11, no. 21, p. 3480, Oct. 2022, doi: <https://doi.org/10.3390/electronics11213480>.

- [8] R. Khalladi, M. Rebbah, and O. Smail, "A new efficient approach for detecting single and multiple black hole attacks," *The Journal of Supercomputing*, vol. 77, no. 7, pp. 7718–7736, Jan. 2021, doi: <https://doi.org/10.1007/s11227-020-03596-1>.
- [9] R. Makani and B. V. R. Reddy, "Trust-based-tuning of Bayesian-watchdog intrusion detection for fast and improved detection of black hole attacks in mobile ad hoc networks," *International Journal of Advanced Intelligence Paradigms*, vol. 21, no. 1/2, p. 53, 2022, doi: <https://doi.org/10.1504/ijaip.2022.121029>.
- [10] V. Mankotia, R. K. Sunkaria, and S. Gurung, "DT-AODV: a dynamic threshold protocol against black-hole attack in MANET," *Sādhana*, vol. 48, no. 4, Sep. 2023, doi: <https://doi.org/10.1007/s12046-023-02227-8>.
- [11] Idriss Moumen, Najat Rafalia, Jaafar Abouchabaka, and Youssef Chatoui, "AODV-based Defense Mechanism for Mitigating Blackhole Attacks in MANET," *E3S Web of Conferences*, vol. 412, pp. 01094–01094, Jan. 2023, doi: <https://doi.org/10.1051/e3sconf/202341201094>.
- [12] R. Ramesh *et al.*, "A Low-Latency and High-Throughput Multipath Technique to Overcome Black Hole Attack in Mobile Ad Hoc Network (MTBD)," *Security and Communication Networks*, vol. 2022, pp. 1–13, Aug. 2022, doi: <https://doi.org/10.1155/2022/8067447>.
- [13] B. Prabhakar Reddy, B. Bhaskar Reddy and B. Dhanajaya, "The AODV routing protocol with built-in security to counter black-hole attack in MANET," *Materials Today: Proceedings*, Aug. 2021, doi: <https://doi.org/10.1016/j.matpr.2021.08.039>.
- [14] S. Shafi, S. Mounika, and S. Velliangiri, "Machine Learning and Trust Based AODV Routing Protocol to Mitigate Flooding and Blackhole Attacks in MANET," *Procedia Computer Science*, vol. 218, pp. 2309–2318, 2023, doi: <https://doi.org/10.1016/j.procs.2023.01.206>.
- [15] S. Sivanesh and V. R. S. Dhulipala, "Accurate and Cognitive Intrusion Detection System (ACIDS): a Novel Black Hole Detection Mechanism in Mobile Ad Hoc Networks," *Mobile Networks and Applications*, vol. 26, no. 4, pp. 1696–1704, Jan. 2020, doi: <https://doi.org/10.1007/s11036-019-01505-2>.
- [16] 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, 2022, doi: <https://doi.org/10.1109/access.2022.3144679>.
- [17] U. Srilakshmi, N. Veeraiah, Y. Alotaibi, S. A. Alghamdi, O. I. Khalaf, and B. V. Subbayamma, "An Improved Hybrid Secure Multipath Routing Protocol for MANET," *IEEE Access*, vol. 9, pp. 163043–163053, 2021, doi: <https://doi.org/10.1109/ACCESS.2021.3133882>.
- [18] M. Thebiga and R. SujiPramila, "A New Mathematical and Correlation Coefficient Based Approach to Recognize and to Obstruct the Black Hole Attacks in Manets Using DSR Routing," *Wireless Personal Communications*, vol. 114, no. 2, pp. 975–993, Apr. 2020, doi: <https://doi.org/10.1007/s11277-020-07403-1>.
- [19] N. Veeraiah *et al.*, "Trust aware secure energy efficient hybrid Protocol for MANET," *IEEE Access*, pp. 1–1, 2021, doi: <https://doi.org/10.1109/access.2021.3108807>.
- [20] S. Vijayalakshmi and T. Anitha, "Hybrid Defense Mechanism Against Malicious Packet Dropping Attack for Manet Using Game Theory," *SSRN Electronic Journal*, 2022, doi: <https://doi.org/10.2139/ssrn.4205757>

Authors' Profiles



Shyamili P V is an Associate Professor in the Department of Artificial Intelligence and Data Science at Sree Narayana Guru College of Engineering and Technology, Kannur, Kerala, India. She holds a Ph.D. in Engineering and her research interests include Artificial Intelligence, Machine Learning, Mobile Ad Hoc Networks, Cyber Security, and Data Analytics. She has published research papers in national and international journals and actively contributes to research in intelligent communication and security systems.



Anoop B K is currently working as Manager (STSG) at Kerala State Electronics Development Corporation (KEL-TRON), Kerala, India. He holds a Ph.D. in Engineering and has extensive experience in academia, research, and industry. His research interests include Artificial Intelligence, Machine Learning, Embedded Systems, IoT, Cyber Security, Signal Processing, and Intelligent Sensing Systems. He has authored several research publications in reputed international journals and conferences.

How to cite this paper

Shyamili P. V., Anoop B. K., "A Centralized Federated Framework for Black-Hole Attack Detection in Mobile Ad Hoc Networks using Gini Canberra Spearman-Based K-Means Clustering and Judy-Verkle Tree Verification", *International Journal of Wireless and Microwave Technologies(IJWMT)*, Vol.16, No.4, pp. 363-377, 2026. DOI:10.5815/ijwmt.2026.04.21