Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 6869307, 17 pages
https://doi.org/10.1155/2019/6869307

Research Article

FAPRP: A Machine Learning Approach to Flooding Attacks Prevention Routing Protocol in Mobile Ad Hoc Networks

Ngoc T. Luong[1,2] **Tu T. Vo**[1] **and Doan Hoang**[3]

[1]*Faculty of Information Technology, Hue University of Sciences, Hue University, Hue 530000, Vietnam*
[2]*Faculty of Mathematics and Informatics Teacher Education, Dong Thap University, Dong Thap 870000, Vietnam*
[3]*Faculty of Engineering and Information Technology, the University of Technology Sydney, Sydney 2007, Australia*

Correspondence should be addressed to Doan Hoang; doan.hoang@uts.edu.au

Received 6 July 2018; Revised 9 November 2018; Accepted 29 November 2018; Published 10 January 2019

Guest Editor: Jiageng Chen

Request route flooding attack is one of the main challenges in the security of Mobile Ad Hoc Networks (MANETs) as it is easy to initiate and difficult to prevent. A malicious node can launch an attack simply by sending an excessively high number of route request (RREQ) packets or useless data packets to nonexistent destinations. As a result, the network is rendered useless as all its resources are used up to serve this storm of RREQ packets and hence unable to perform its normal routing duty. Most existing research efforts on detecting such a flooding attack use the number of RREQs originated by a node per unit time as the threshold to classify an attacker. These algorithms work to some extent; however, they suffer high misdetection rate and reduce network performance. This paper proposes a new flooding attacks detection algorithm (FADA) for MANETs based on a machine learning approach. The algorithm relies on the route discovery history information of each node to capture similar characteristics and behaviors of nodes belonging to the same class to decide if a node is malicious. The paper also proposes a new flooding attacks prevention routing protocol (FAPRP) by extending the original AODV protocol and integrating FADA algorithm. The performance of the proposed solution is evaluated in terms of successful attack detection ratio, packet delivery ratio, and routing load both in normal and under RREQ attack scenarios using NS2 simulation. The simulation results show that the proposed FAPRP can detect over 99% of RREQ flooding attacks for all scenarios using route discovery frequency vector of sizes larger than 35 and performs better in terms of packet delivery ratio and routing load compared to existing solutions for RREQ flooding attacks.

1. Introduction

A Mobile Ad Hoc Network (MANET) [1] is a collection of wireless mobile devices (called nodes) that dynamically form an ad hoc network in situations such as disaster rescue, urgent conference, or military mission, without the support of a network infrastructure. The topology of the network may change frequently because nodes can join or leave the network at will. In a MANET, nodes coordinate among themselves to maintain the connections among them. Data transfer from a source node to a non-neighbor destination node is routed through intermediate nodes. A node can act as a host and a router at the same time. A network routing protocol in a MANET specifies how nodes in the network communicate with each other. It enables the nodes to discover and maintain the routes between any two of them. Many routing protocols have been developed for MANETs such as ad hoc on-demand distance vector (AODV) [2], dynamic destination sequenced distance vector (DSDV) [3], and zone routing protocol (ZRP) [4]. They are classified into three groups: proactive, reactive, and hybrid routing protocols. With proactive routing protocols, the routes between nodes need to be established before data packets can be sent. These protocols are suitable for fixed topology networks. In contrary, reactive routing protocols are suitable for dynamic topology networks as nodes only try to discover routes on demand. In complex network topologies, hybrid routing protocols are often used [5]. MANETs are thus essential in infrastructureless situations for communication; however, they suffer from various types of Denial of Service (DoS) attacks that deny user services or resources he/she would normally expect to receive. Disrupting routing services at the

network layer is an example of DoS [6, 7] where a malicious node (MN) tries to deplete resources of other nodes. Other types of DoS include Blackhole [8], Sinkhole [9], Grayhole [10], Whirlwind [11], Wormhole [12], and flooding attacks [13]. Flooding attack is a particular form of DoS attacks in MANETs where malicious nodes mimic legitimate nodes in all aspects except that they do route discoveries much more frequently with the purpose of exhausting the processing resources of other nodes. This type of attacks is simple to perform with on-demand routing protocols, typically as AODV [14]. Among HELLO, RREQ, and DATA flooding attacks, route request (RREQ) flooding attack is the most hazardous because it is easy to create a storm of request route packets and cause widespread damage. This paper focuses on the request route flooding attack.

Previous researches on RREQ flooding attacks mainly focus on detection algorithms that rely on the sending frequency of RREQ packets [13, 15–20]. Every node uses a fixed (or dynamic) threshold value to detect an attack. The threshold is calculated based on the number of RREQs originated by node per unit time. A node labels a neighbor node malicious if it receives more RREQs than the allowed threshold from its neighbors. These algorithms, however, have many weaknesses in dealing with the dynamics of MANETs. These include the following: (1) An algorithm with a fixed threshold is not flexible and is not able to cope with dynamic environments where optimal threshold values vary. (2) Even with dynamic threshold algorithms, where the threshold takes into account other factors such as network traffic, mobility speed, and frequency of malicious node attacks, misclassifications rates are still high. In high mobility environments, the connection state of network nodes changes very frequently; a node may not be able to capture accurate and adequate information to distill it to a single threshold. (3) A normal node may be mistaken for a malicious node even if it legitimately sends out a high number of route requests in response to a high priority event. Or (4) a malicious node may avoid the threshold detection mechanism simply by sending RREQ packets at a frequency just lower than the threshold value.

In this paper, we propose and investigate a different approach for detecting flooding attacks. Our solution relies on the route discovery history information of each node to classify a node as malicious or normal. The route discovery history of each node is represented by a route discovery frequency vector (RDFV). The route discovery histories reveal similar characteristics and behaviors of nodes belonging to the same class. This feature is exploited to differentiate abnormal behavior from a normal one. RDFV is defined as the feature vector for detecting malicious nodes in MANET environment. We propose a flooding attack detection algorithm to detect malicious node based on RDFV. We propose a novel flooding attacks prevention routing protocol by incorporating the FADA algorithm and extending the AODV protocol. We evaluate the performance of our solution in terms of successful detection ratio, packet delivery ratio, and routing load both in normal and under RREQ attack scenarios using NS2 simulation. The simulation results showed that our approach can detect over 99% of RREQ flooding attacks, had better packet delivery ratio and routing load compared to existing solutions for RREQ flooding attacks, and introduced negligible overhead relative to AODV for normal scenarios. The main contributions of the paper are as follows:

(1) It introduced a new route discovery history measure, the vector of route discovery frequency, to capture the behavior of MANET nodes.

(2) It proposed a flooding attack detection algorithm, a k-nearest neighbors-based machine learning algorithm, using RDFV dataset to detect malicious nodes.

(3) It proposed a flooding attack prevention routing protocol by integrating FADA into the original AODV protocol.

(4) It evaluated the effectiveness and the performance of the proposed solution for high-speed mobility MANETs under RREQ flooding attacks.

The remainder of this paper is structured as follows: Section 2 presents a review of the related work on detection of flooding attacks. Section 3 presents our solution and a novel flooding attacks prevention routing protocol by improving AODV protocol using FADA. Section 4 presents the results of evaluating the performance of the proposed solution relative to existing solutions. Section 5 concludes the paper.

2. Related Works

2.1. Overview of AODV. AODV is a popular reactive routing protocol in which a node only initiates the process for finding a path to the destination if it wants to send data. Basically, when the source node (N_S) wants to communicate with the destination node (N_D), without an already discovered route to the destination, N_S starts a route discovery process by broadcasting a route request (RREQ) packet containing the destination address. The nodes that receive the packet will in turn broadcast it. When N_D receives the packet, it will send a route reply (RREP) packet back to source node. Once a route has been discovered, HELLO and RERR packets can be used to maintain the status of the route.

Figure 1 describes the route discovery process of AODV; source node (N_7) discovers route to destination node (N_{11}) by broadcasting an RREQ to its neighbor nodes. When a node receives the RREQ packet for the first time, it broadcasts the packet and sets up a reverse path to the source. If the node receives the same RREQ subsequently, it simply drops the packet. When N_{11} gets a RREQ, it unicasts a RREP packet to the source node through the established reverse $\{N_{11} \longrightarrow N_{10} \longrightarrow N_9 \longrightarrow N_7\}$. When N_7 gets a RREP, it establishes successfully a new path to N_{11} with 3 hops routing cost and adds the new entry to its routing table.

2.2. Flooding Attacks on AODV. Flooding attack is a form of DoS attacks in which malicious nodes broadcast false packets in the network to exhaust the resources and disrupt the network operation. Depending on the type of packet used to flood the network, flooding attack can be categorized into three categories, RREQ, DATA, and HELLO flooding attack.

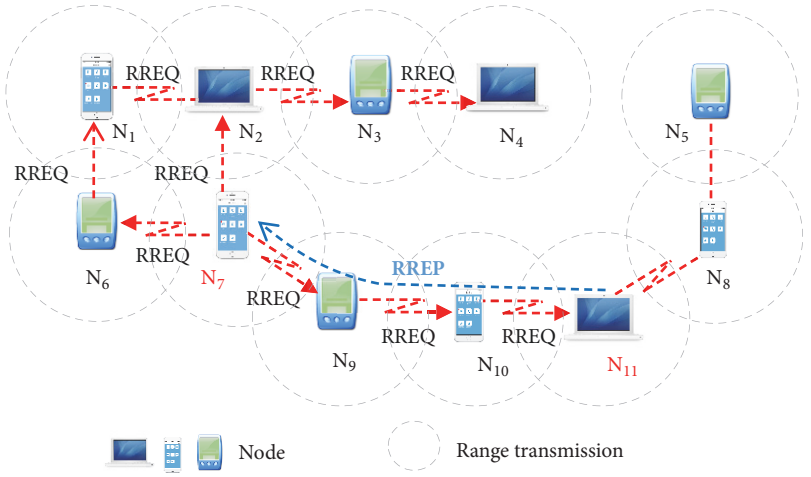


Figure 1: Description of route discovery process of AODV in the MANET.

In RREQ flooding attack, a malicious node continuously and excessively broadcasts fake RREQ packets, which causes a broadcast storm and floods. The RREQ flooding attack is considered most harmful in MANET because it can ruin the route discovery process by exhausting the channel bandwidths and the processing resources of affected nodes. In DATA flooding attack, a malicious node can excessively broadcast data packets to any nodes in the network. This type of attacks has more impact on the nodes participating in the data routing to the destinations. In HELLO flooding attack, nodes periodically broadcast HELLO packets to announce their existence to their neighbors. A malicious node abuses this feature to broadcast HELLO packets excessively and forces its neighbors to spend their resources on processing unnecessary packets. This type is only detrimental to the neighbors of a malicious node. Figure 2 shows the behavior of malicious nodes (M) in a MANET for these types of attacks.

2.3. Review on Related Research. This section summarizes related work on threshold-based, machine learning-based, hash function-based, and digital-signature-based approaches in detecting and preventing flooding attacks in MANETs. Table 1 summarizes these methods and their drawbacks.

2.3.1. On Fixed Threshold-Based Approach. Solutions are simple with a fixed threshold for mitigating the impact of RREQ flooding attacks. However, with a static threshold, these methods are not suitable for dynamic environments where nodes are highly mobile and frequently broadcast route request packets. In [15], Gada used three fixed thresholds: RREQ_ACCEPT_LIMIT, RREQ_BLACKLIST_LIMIT, and RATE_RATELIMIT. The default value of RATE-RATELIMIT is 10. If the rate of receiving request packets is greater than RREQ_ACCEPT_LIMIT but less than RREQ_BLACKLIST_LIMIT, packets are simply dropped and not processed. If it is greater than RREQ_BLACKLIST_LIMIT, the source is declared as a malicious node. The weakness of this solution is that it may lead to blacklisting of normal nodes false positive [16] and cause excessive end-to-end delay by dropping legitimate request packets once the RREQ_ACCEPT_LIMIT threshold is crossed.

In [16], Song et al. proposed a simple technique using an Effective Filtering Scheme (EFS) to detect malicious nodes. This solution uses two limit values: RATE_LIMIT and BLACKLIST_LIMIT. If the detected RREQ rate is higher than the RATE_LIMIT and the BLACKLIST_LIMIT, the malicious node is declared and it will be put into the black list. If the rate of RREQs originated by a node is between the RATE_LIMIT and the BLACKLIST_LIMIT, the RREQ packet is added to a "delay queue" waiting to be processed. Here the authors set the RATE_LIMIT threshold to 5 and set the BLACKLIST_LIMIT up to 10.

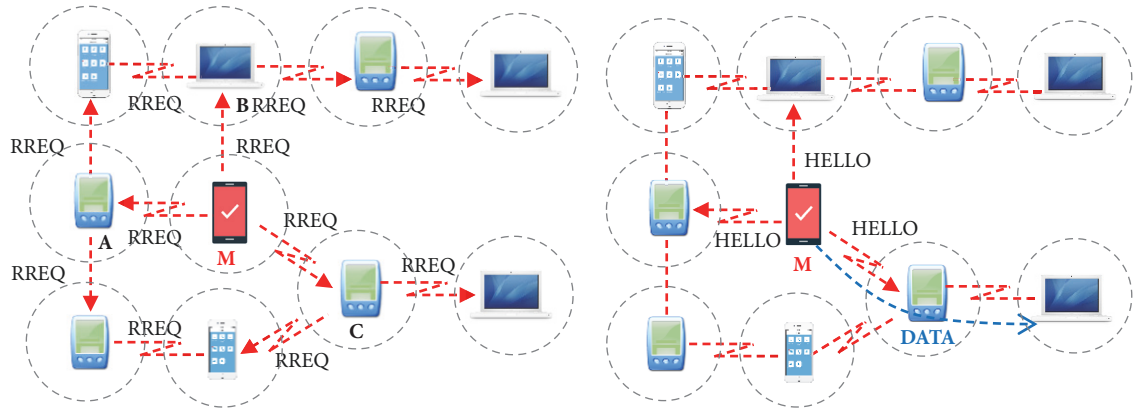
In [13, 17], the authors developed flooding attack prevention (FAP) that prevents RREQ and DATA flooding attacks in MANETs. They argued that the priority of a node is adversely proportional to its broadcast frequency of RREQ. Hence, nodes that generate a high frequency of route requests will have a low priority and may be removed out of the routing process. It is suggested that a node should not originate more than 10 RREQ packets per second and, hence, the threshold of FAP is set at 15 for a good margin.

2.3.2. On Dynamic Threshold-Based Approach. Solutions with dynamic thresholds are more flexible as they can cope with the dynamic environment of MANETs. In [18], Mohammad proposed an improved protocol called B-AODV. In this method, each node employs a balance index (BI) for acceptance or rejection of RREQ packets. If the RREQ rate is higher than the BI value, a malicious node is defined and the RREQ packet is dropped. The results showed that B-AODV is resilience against RREQ flooding attacks. The main drawback of B-AODV is that it may drop legitimate request packets of the node moving at high speed as the number of request packets may be higher than the balance index value [19]. Also, the method does not have a confirmation mechanism which can identify the node properly as a malicious node.

In [19], Gurung proposed a new mechanism called Mitigating Flooding Attack Mechanism. The mechanism is

Table 1: Summary of drawbacks of related works for detecting flooding attacks.

Ref	Name	Year	Method	Drawback
[15]	Proposed-AODV	2004	Fixed threshold	It uses static threshold value which is not suitable for high mobility environment.
[13]	FAP	2005		Malicious node can pass the security mechanism by transmitting RREQ packets at a frequency lower than the threshold.
[16]	EFS	2006		
[18]	B-AODV	2016	Dynamic threshold	It can drop valid request packets of the node moving with high mobility speed if the number of request packets is greater than BI value. Malicious node can pass the security mechanism by transmitting RREQ packets at a frequency lower than the threshold.
[19]	F-IDS	2017	Dynamic threshold	Performance varies. Using new control packets (ALERT) will increase communication overhead and limit the performance when operating in network environment without attacks. Malicious node can pass the security mechanism by transmitting RREQ packets at a frequency lower than the threshold.
[20]	SMA_2AODV	2017	Dynamic threshold	Malicious node can pass the security mechanism by transmitting the RREQ packets at a frequency lower than the threshold.
[21]	SVMT	2013	SVM	The proposed algorithm uses fixed threshold to detect malicious nodes.
[22]	kNN-AODV	2014	kNN	The algorithm for building training data sets was not presented or justified.



(a) RREQ flooding (b) HELLO and DATA flooding

Figure 2: Description of flooding attacks in the MANET.

based on a dynamic threshold and consists of three phases. It deploys special Flooding Intrusion Detection System (F-IDS) nodes to detect and prevent flooding attack. The F-IDS nodes are set in the promiscuous mode to monitor the behavior of nodes in the network. The proposed mechanism has several features: (1) it uses a dynamic threshold; (2) it has a confirmation mechanism in which the special F-IDS node confirms the node as a malicious node by sending a dummy reply packet and waits for the data packets; and (3) it has a recovery mechanism that allows the node to participate in the network after the expiry of the blocking time period. However, the use of several F-IDS nodes to monitor their neighbors and to communicate among them limits the performance of the overall network, especially when the network is not under attack.

In [20], Tu introduced security mobile agents (SMA) to detect flooding attacks. An improved protocol, SMA_2AODV, is proposed by integrating these SMAs into the discovery route process of the AODV protocol. During the training period, SMA agents are used to collect information for determining the minimal time-slot (the minimum time-slot for successfully discovering a path from a source node to a destination node) of the system (TS_{min}). After the training phase, node N_i checks the security of the RREQ packet received from source node N_j before broadcasting it to the neighbors. If route discovery time-slot is smaller than the minimal time-slot of the system ($T < TS_{min}$), a flooding attack is said to have occurred with N_j as the attacker. N_i then adds N_j to its black list. All RREQ packets of nodes in the black list will be dropped. The drawback of this method is

Table 2: Description of symbols.

Variable	Description
t_i	Route discovery time i^{th}
T_i	Inter-route discovery time i^{th}
V_{Ns}	Vector of route discovery frequency of N_S node
m	Size of vector of route discovery frequency
k	Cutoff value for kNN algorithm

that TS_{min} is only valid if no malicious node exists during the training period.

2.3.3. On Machine Learning Approach. In [21], Patel proposed the use of support vector machine (SVM) algorithm for detecting and preventing flooding attacks. The behavior of every node is collected and passes to the support vector machine to decide if a node is malicious based on a threshold limit.

In [22], Wenchao proposed a new intrusion detection system based on k-nearest neighbors (kNN) classification algorithm in wireless sensor network to separate abnormal nodes from normal nodes by observing their behaviors. An m-dimensional vector is used to represent nodes and their behaviors such as the number of routing messages that can be sent over a period of time, the number of nodes with different destinations in the sending routing packets, and the number of nodes with the same source node in the receiving routing packets. The paper shows that the system achieves high detection accuracy, but it does not provide justifications or the algorithm for building training datasets.

3. The Proposed FAPRP Solution

This section we present our algorithms and routing protocol for detecting flooding attacks in MANETs. First, we define a feature vector that represents the behavior of a node based on its history of route discovery: the route discovery frequency vector. Second, we describe an algorithm for obtaining the training dataset which describes the normal behavior and the abnormal behavior of nodes for normal/malicious classification. Third, we present our flooding attack detection algorithm, and finally we present our proposed AODV-based flooding attacks prevention routing protocol. Table 2 defines symbols used in the paper.

3.1. Route Discovery Frequency Vector. In order to detect RREQ flooding attacks with kNN, the crucial problem is the selection of a feature vector that maximizes the separation of the normal and the malicious data classes and produces highly reliable classification. The selected features should be able to succinctly capture the inherent behavior of a node performing RREQ requests and the time-related network activities through their historical data records in order to differentiate "normal" from "malicious" behavior. We propose a route discovery frequency vector as the feature vector for this purpose. To quantify this vector, we define the following terms.

Definition 1. Route discovery time (t_i) is the duration from the time a node first broadcasts a route discovery packet to the time it receives the corresponding route response. Assuming that node N_i receives the i^{th} RREQ packet from the source node *Ns* at time s_i and N_i receives the route response packet at time e_i, the route discovery time (t_i) is defined by

$$t_i = e_i - s_i. \tag{1}$$

Definition 2. Inter-route discovery time (T_i) is the duration from the end of a route discovery to the beginning of the next route discovery. Assuming that the node N_i receives the $i+1^{th}$ RREQ packet from the source node *Ns* at time s_{i+1}, the inter-route discovery time (T_i) is defined by (2).

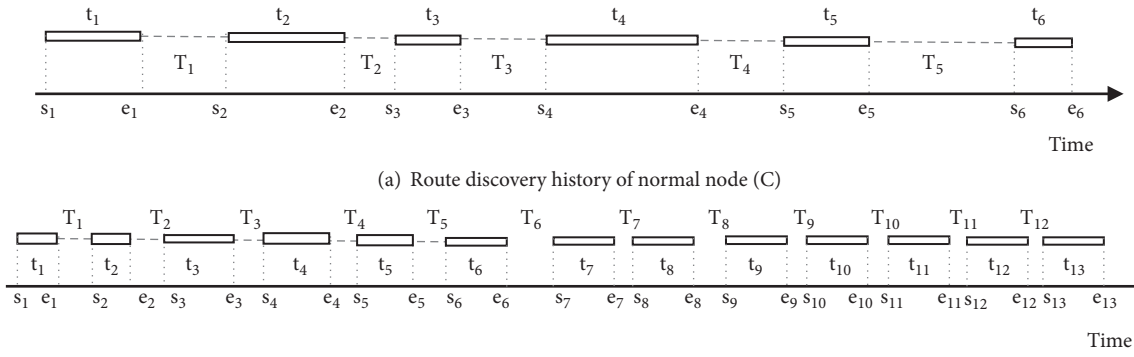
$$T_i = s_{i+1} - e_i \tag{2}$$

In AODV routing protocol, route discovery frequency of a node depends on how frequent the node has to find a path to the required destination. All normal nodes have route discovery frequencies within a range, but malicious nodes have higher route discovery frequencies as their aim is to flood the network. Consider Figure 2(a); it shows three normal nodes, A, B, C, and one malicious node, M. Figure 3(a) shows the route discovery history of the normal node (C) as recorded by the normal node (A). Figure 3(b) shows route discovery history of the malicious node (M) that is also recorded by the normal node (A). The figures show that node C sent 6 RREQ packets and node M sent 13 RREQ packets over roughly the same duration.

We use a m-dimensional vector V_{Ni} (a_1, a_2, a_3, …, a_m) to represent route discovery history of node N_i, where m is the size of the vector and a_i is the i^{th} inter-route discovery time.

Example 1. Route discovery history of the malicious node shown in Figure 3(b) is represented by the route discovery frequency vector V_M ($T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}, T_{11}, T_{12}$) of size 12.

Figure 4 shows typical vectors of size 40 of the route discovery frequency of normal and malicious nodes, by NS2 simulation. It can be seen that the inter-route discovery time values for all normal nodes (N_1 to N_5) are generally larger (> 1 sec) than those for malicious nodes (M_1 to M_5) as they have low route discovery frequencies. However, there are cases where the malicious inter-route discovery times (T_i) are indistinguishable from the normal ones. One reason for this is the mobility of nodes in the environment; a recording node may not receive RREQ packets from a malicious node until some later time. Other reason for the overlapping



(a) Route discovery history of normal node (C)

(b) Route discovery history of malicious node (M)

Figure 3: Route discovery history recorded at normal node (A).

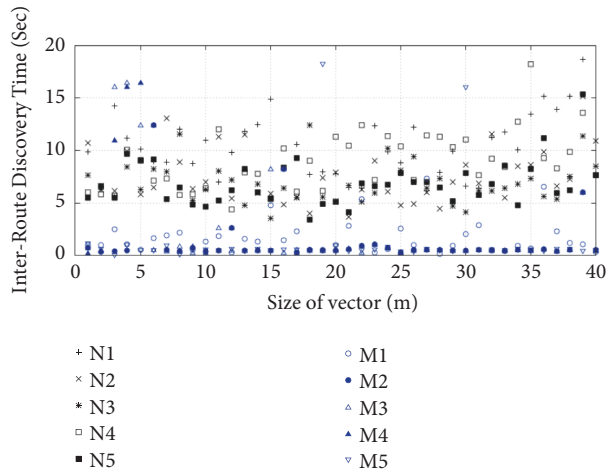


Figure 4: An example describes vectors of route discovery frequency of 5 normal nodes (N_1 to N_5) and 5 malicious nodes (M_1 to M_5).

region is when a malicious node floods the network at a frequency close to the rate at which a normal node can generate RREQs. As demonstrated in Section 4, our proposed algorithm successfully recognizes these abnormal cases based on route discovery frequency feature.

3.2. Algorithm for Obtaining a Training Dataset. We use NS2 [23] version 2.35 to build a training dataset of NVC (normal) and MVC (malicious) vector classes. The simulation scenario is set up with 100 normal nodes and 1 malicious node, operated in the area of 2000m x 2000m. Normal nodes move under random waypoint model with maximum speeds 0m/s, 10m/s, 20m/s, 30m/s, and 40m/s scenarios; a malicious node is positioned at the center (1000m x 1000m) as shown in Figure 5. Other simulation parameters include AODV routing protocol, 50 UDP connections, and constant bit rate (CBR) traffic type; the first data source commences at time 0, other data sources commence at 5 seconds apart after the first, and the malicious node, respectively, floods f packets every second (f may take on different values: 2, 5, 10, 50, and 100).

The training process proceeds as follows.

Step 1. Select the dimension or size (m) of the feature vectors.

Step 2. Set the frequency of flooding to 2 initially ($f = 2$ per second).

Step 3. For each of the mobile scenarios (0m/s, 10m/s, 20m/s, 30m/s, and 40m/s), simulate the MANET as follows. Each node records the inter-route time of a source node (T_i) on receiving a RREQ from the source node. Add T_i to the malicious history frequency vector if the source is malicious; otherwise it is added to the normal history frequency vector. At the end of this step for each scenario, two sets of vectors are established:

(i) 100 malicious vectors: $V_M^j(T_1^M, T_2^M, T_3^M, ..., T_m^M); \forall j = \overline{1..100}$;

(ii) 100 normal vectors: $\overline{V^j}$ $(\sum_{i=1}^{50} T_1^i/50, \sum_{i=1}^{50} T_2^i/50, \sum_{i=1}^{50} T_3^i/50, ..., \sum_{i=1}^{50} T_m^i/50)$; $\forall j = \overline{1..100}$.

Step 4. At the end of Step 3 for all 5 scenarios, 100 average vectors for MVC and 100 vectors for NVC are obtained for this particular flooding frequency (f=2).

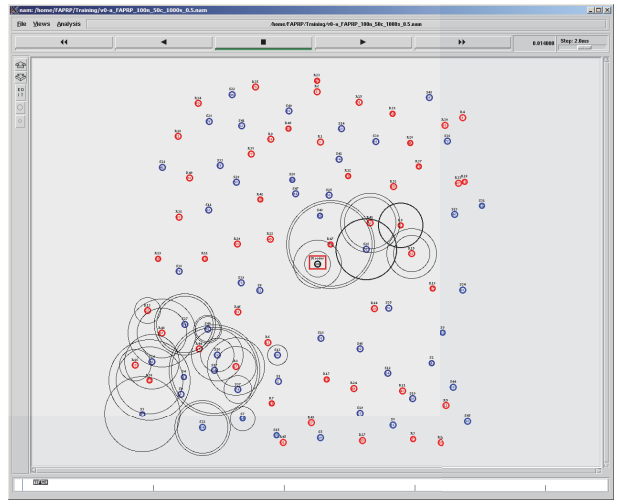


Figure 5: Static network topology simulation for training, 50 UDPs connections and malicious node positioned at the square in the center.

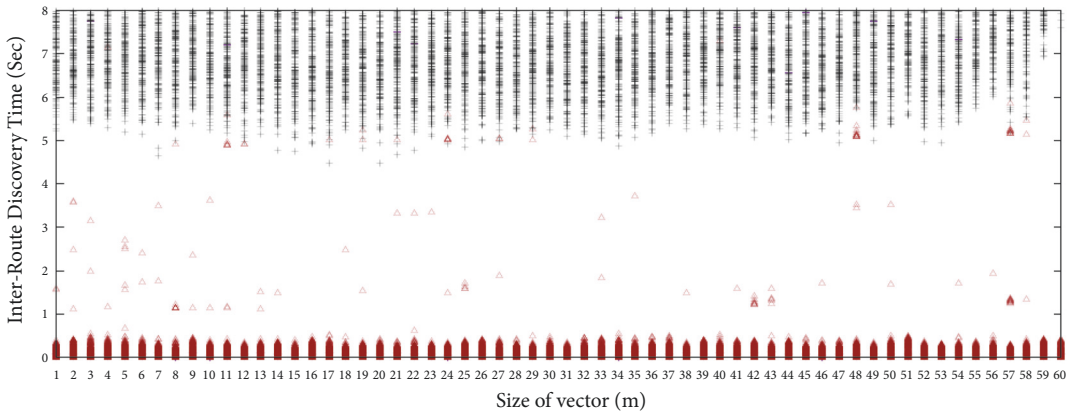


Figure 6: Two vectors class, black (+) for NVC and red (Δ) for MVC.

Step 5. The algorithm continues to establish MVC vectors and NVC vectors for other flooding frequencies (f = 5, 10, 50 and 100).

As a result of the training process, a training dataset with MVC and NVC vectors is shown in Figure 6. The training dataset is used to classify an unknown sample vector V (in the next section). In Figure 6, each vector is of size 60. It can be seen that there is an overlap between the two classes due to node mobility as well as the closeness of the rate of generation of RREQ packets of malicious and normal nodes.

3.3. Flooding Attack Detection Algorithm (FADA). All normal nodes collect route discovery information of source nodes in the network. On receiving a RREQ packet, a node employs the route discovery frequency vector (V_{Ns}) and uses a machine learning algorithm to determine if the source node is normal or malicious. The kNN-Classifier based on kNN [24] algorithm is utilized to classify the two classes based on the route discovery frequency vectors for NVC or MVC. The kNN algorithm is theoretically mature with low complexity that is widely used for data mining. The main idea is that if most of its k-nearest neighbors belong to a class, the sample belongs to the same class. In kNN, the nearest neighbor refers to the distance between two samples, and various distance metrics can be used based on the feature vector that represents the samples. One of the most popular choices is the Euclidean in (3) to calculate the distance between V_1 and V_2. Algorithm 1 describes our algorithm for recognizing malicious nodes.

$$d(V_1, V_2) = \sqrt{\sum_{i=1}^{m} (V_1[i] - V_2[i])^2} \quad (3)$$

3.4. FAPRP: A Novel Flooding Attacks Prevention Routing Protocol. In the original AODV protocol, as intermediate nodes accept all RREQ route discovery packets from any source nodes, hackers may exploit this vulnerability to perform RREQ flooding attacks. We propose the flooding attacks prevention routing protocol by introducing the flooding attacks detection algorithm into the route request phase of the AODV protocol as described in Figure 7. Similar to AODV, path discovery is entirely on-demand for FAPRP. When a

```
Input: Two class NVC and MVC, vector of route discovery frequency (V_Ns)
Output: True if V_Ns in NVC, else return False
Begin
    MAX_VECTOR = 500;
    Double Array disMVC [MAX_VECTOR], disNVC [MAX_VECTOR];
    For int vt = 1 to MAX_VECTOR do {
       disMVC[vt] = Euclidean (V_Ns, MVC.Vectors[vt]);
       disNVC[vt] = Euclidean (V_Ns, NVC.Vectors[vt]);
    }
    Sort (disMVC and disNVC, ASC); // ascending sort
    int k1 = k2 = 0;
    While (k1 + k2 < k) {
       if (disNVC[k1] < disMVC[k2]) k1++;
       else k2++;
    }
    Return (k1 > k2);
End
```

Algorithm 1: Flooding attack detection algorithm using kNN.

source node needs to send data packets to a destination node to which it has no available route, N_S broadcasts a RREQ packet to its neighbors. The intermediate node (N_i) receiving a RREQ packet from a preceding node (N_j) checks security as follows.

First, duplicate RREQ packets received by a node are dropped, similar to the AODV protocol. N_i may receive multiple RREQ packets coming from its neighboring nodes, but it only handles the first RREQ packet using the two parameters broadcast_id and src_add (source address) in the RREQ packet.

Second, unlike AODV routing protocol, N_i adds the information (s_i and e_i) to the route discovery history (RDH) of the source node. Each intermediate node stores the route discovery counter of all source nodes. If the value of the Counters[N_S] equals x, the source node N_S has initiated route discovery x times to this point. If the route history is full, N_i shifts all elements of RDH one position to the left and adds the new element (s_i, e_i) to the rightmost position.

In MANET, a source node sends and receives packets through its neighbor nodes. If all neighbor nodes of the source node reject packets, it will be isolated and cannot communicate with the other nodes in its network [13]. For this reason, in FAPRP routing protocol, only the source node's neighbor nodes deploy FADA algorithm to detect RREQ flooding attack. N_i uses the source node address and the preceding node address to determine if it is a neighbor of the source N_S. On receiving RREQ packets, the protocol works as follows.

Step A. If N_i is a neighbor of the source node N_S:

(i) N_i measures all T_i values in V_{Ns} using RDH of the source node.

(ii) If the route discovery frequency vector of source node (V_{Ns}) is not full, N_i ignores the security check and go to **Step B**.

(iii) Else, N_i uses FADA to classify N_S using its feature vector V_{Ns}.

(a) If V_{Ns} is in MVC, the source node is classified malicious, the RREQ packet is dropped, and the algorithm terminates.

(b) Else, go to **Step B**.

Step B. If N_i is not a neighbor of N_S, it executes other commands similar to AODV as follows:

(i) N_i saves broadcast_id and src_add values into its cache and adds a reverse route to source node into its routing table.

(ii) If N_i is destination or has a route toward the destination, it unicasts a RREP packet back to its neighbor from which it received the RREQ packet (N_j); otherwise, it rebroadcasts the RREQ packet.

When the destination node gets a RREQ, it updates the time instance e_i in the RDH of source node and unicasts a RREP packet to the source node through the reverse route. In the AODV protocol, there is no order information for the route response in the RREP packet. Therefore, N_i assumes that the RREP packet received is the response to the last route discovery. Thus, once the intermediate node receives an RREP packet, it updates e_i in the RDH of source node; that is, it sets i=Counters[N_S]. It increases the hop count field by 1 before forwarding the RREP packet back to the source node.

Example 2. Figure 8 describes how an intermediate node (N_i) handles the RREQ and RREP packets. First, on receiving RREQ packet at time p_1, N_i increases Counters[N_S] to 1 (Counters[N_S]=1) and records $s_1=e_1=p_1$. Second, on receiving the RREP packet at time p_2, N_i updates $e_1=p_2$. Next, at time p_3, N_i receives the RREQ packet, increases the Counters[N_S] by 1 (Counters[N_S]=2), and records $s_2=e_2=p_3$. Similarly, at time

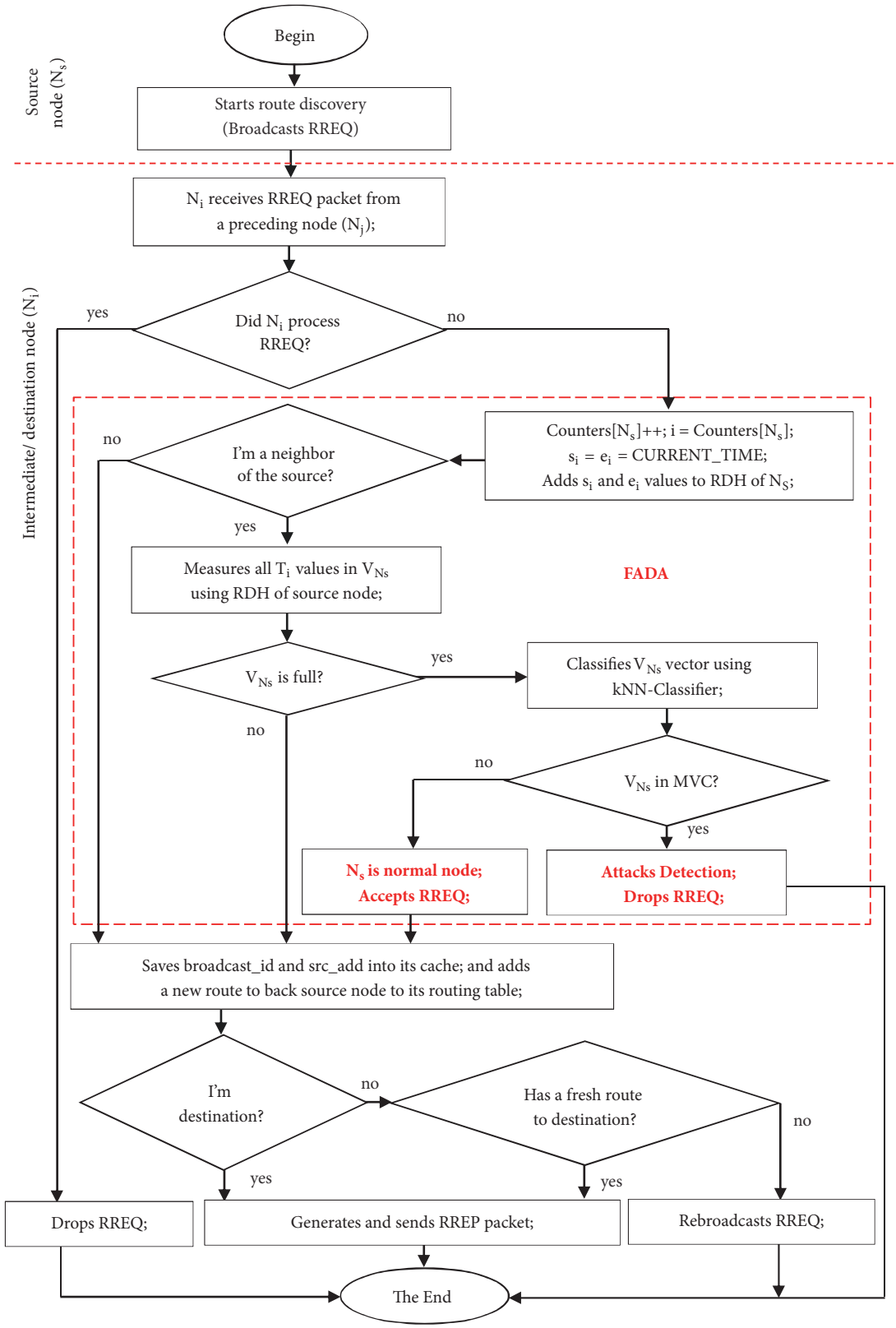


Figure 7: Request route process of FAPRP routing protocol.

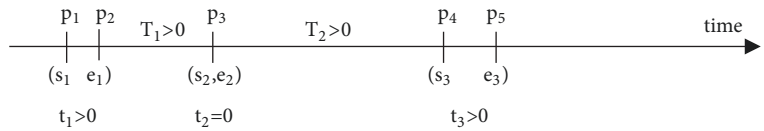


Figure 8: Route discovery history of the source node and 1 destination node.

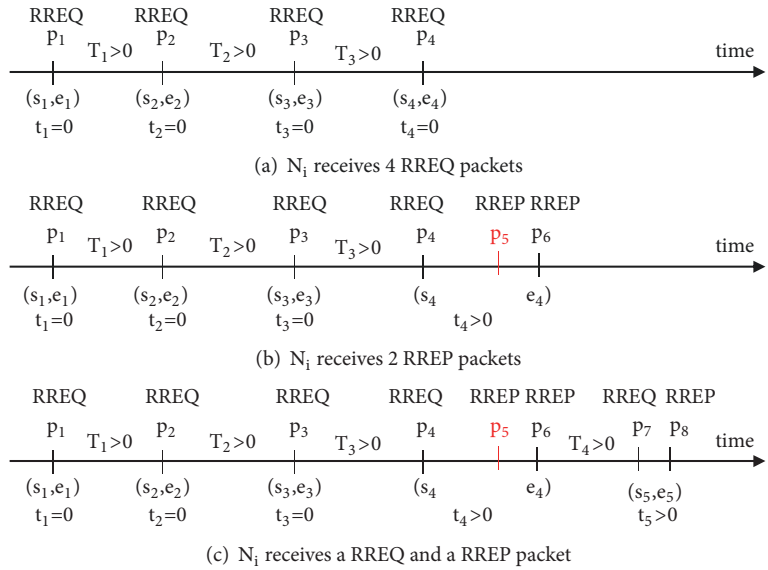


Figure 9: Route discovery history of a source and 3 destination nodes.

p_4, on receiving the next RREQ packet, Counters[N_S] is set to 3 and s_3=e_3=p_4. Finally, N_i records RREP at time p_5 and updates e_3=p_5. Because p_1<p_2 <p_3<p_4<p_5, $T_i > 0$, $\forall i = 1..2...$.

3.5. Discussion. RREQs may originate from the same N_S to many destination nodes (N_{D1}, N_{D2}, N_{Dn}). In this case, FADA only keeps the counter for N_S regardless of the destinations. This case is of interest because in detecting a malicious node, FADA only wants to see how often that node generates RREQ and does not care about the destinations.

Example 3. Using a network topology with n nodes, consisting of one source node N_S and three destination nodes N_{D1}, N_{D2}, and N_{D3}. Assume that N_S made route discovery seven times to three destination nodes N_{D1}, N_{D2}, and N_{D3}. Because of the mobile and noisy environment, 3 RREQ packets were lost, and N_i received only 4 RREQ packets at p_1, p_2, p_3, and p_4, respectively. The value of Counters[N_S] at N_i was then 4, which meant that as far as N_i was concerned, N_S has route discovered 4 times up to that point. Figure 9(a) shows the RDH of the N_S source node as recorded in N_i.

After p_4, N_i receives two RREP response packets to the source at p_5 and p_6. When receiving RREP at time p_5, N_i updates e_4=p_5, and N_i continues to update e_4=p_6 when receiving RREP packet at p_6. Figure 9(b) shows the RDH of the N_S source node after receiving two RREP packets.

Finally, N_i receives another RREQ packet from the N_S at time p_7 and a RREP packet at time p_8. On receiving this last RREQ, N_i increases Counters[N_S] by 1 (Counters[N_S]=5) and sets s_5=e_5=p_7, and on receiving the last RREP packet, N_i updates e_5=p_8. Figure 9(c) shows the RDH of the N_S source node at p_8.

Thus, based on the RDH of the source node, N_i can compute all T_i in V_{Ns} and use kNN-Classifier to decide if the source node is normal or malicious. In addition, all T_i values are larger than zero and it does not depend on the order of RREQ packets and the number of destination nodes.

4. Performance Evaluation by Simulation

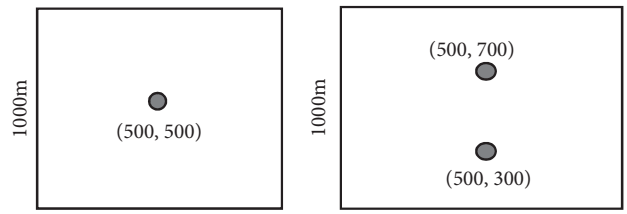
In this section, we use NS2 [23] version 2.35 to evaluate the impact of RREQ flooding attacks on AODV and the proposed FAPRP protocol.

4.1. Simulation Settings. Similar to [13], our simulation scenarios cover a 1000 meter by 1000 meter flat space, accommodating 50 normal mobile nodes. We consider 2 scenarios: one with a malicious positioned at the center (Figure 10(a)) and the other with two malicious nodes positioned as shown in Figure 10(b). Each malicious node may flood the network at the rate of 10 or 20 packets per second.

The random waypoint [25] model is utilized as the mobility model. The minimum node speed for the simulations is 1 m/s while the maximum is 30m/s. In each simulation scenario, 20 sources transmit data at a constant bit rate (CBR). Each source transmits 512-byte data packets at the rate of 2 packets/second. The first source emits data at time 0, and

Table 3: Simulation parameters.

Parameters	Setting
Simulation area	1000 x 1000 (m^2)
Simulation time	500 (second)
Number of normal nodes	50 (nodes)
Node transmission range (R)	250 (m)
Number of malicious nodes	1, 2 (nodes)
Attacks frequency	10, 20 (packet/second)
Maximum speeds	1..10, 1..20 and 1..30 (m/s)
Transport protocol	UDP
Traffic type	CBR (constant bit rate)
Number of traffic	20
Data rate	2 (packet/second)
Packet size	512 (bytes)
Queue type	FIFO (DropTail)
Routing protocols	AODV, B-AODV [18], FAPRP
Size of vector (m)	10, 15, 20, 25, 30, 35, 40 and 60
Cutoff value (k)	10, 15, 20, 25, 30, 35, 40, 45 and 50
Distance type	Euclid



(a) 1 node (b) 2 nodes

Figure 10: Malicious nodes location.

the following sources transmit data at 10 seconds apart. All parameters are described in Table 3.

We evaluate the original AODV, the B-AODV, and the FAPRP and compare their performance with and without RREQ flooding attacks in terms of attacks detection ratio, packet delivery ratio, end-to-end delay, and routing load metrics [18, 26].

(i) Attacks detection ratio (ADR) is calculated using (4). AT is the number of RREQ packets that are accepted true; the packets come from normal nodes. AF is the number of RREQ packets that are accepted false; the packets come from malicious nodes. DT is the number of RREQ packets that are dropped true; the packets come from malicious nodes. DF is the number of RREQ packets that are dropped false; the packets come from normal nodes.

$$ADR = \frac{AT + DT}{AT + AF + DT + DF} * 100\% \quad (4)$$

(ii) Packet delivery ratio (PDR) is the ratio of the received packets by the destination nodes to the packets sent by the source nodes (5), where n is number of data packets that are received by destination nodes and m is number of data packets that are sent by source nodes.

$$PDR = \frac{\sum_{i=1}^{n} DATA_i^{recieved}}{\sum_{j=1}^{m} DATA_j^{sent}} * 100\% \quad (5)$$

(iii) End-to-end delay (ETE) is the average delay between the sending time of a data packet by the CBR source and its reception at the corresponding CBR receiver (6), where $Delay_{DATA}^{i}$ is the delay time for sending i^{th} data packet to its destination successfully and n is number of data packets that are received by destination nodes.

$$ETE = \frac{\sum_{i=1}^{n} Delay_{DATA}^{i}}{n} \quad (6)$$

(iv) Routing load (RL) is the ratio of the overhead control packets sent (or forwarded) to successfully deliver data packets (7), where n is number of data

Table 4: AODV performances under flooding attacks.

Level	Number of MN	PDR (%)			RL (pkt)			ETE (Sec)		
		10m/s	20m/s	30m/s	10m/s	20m/s	30m/s	10m/s	20m/s	30m/s
0pkt/s	0	86.26	84.68	82.10	4.92	5.72	7.02	0.506	0.574	0.627
10pkt/s	1	72.63	68.68	64.13	25.45	28.61	30.61	1.032	1.232	1.304
	2	26.40	23.42	17.95	158.96	196.42	263.39	3.188	3.049	3.333
20pkt/s	1	28.75	25.57	19.63	140.55	171.48	228.57	3.292	3.013	3.059
	2	12.06	11.23	8.78	524.18	587.18	898.82	3.668	2.952	4.973
Standard deviation values										
0pkt/s	0	3.09	2.22	1.77	0.91	0.88	0.85	0.14	0.10	0.11
10pkt/s	1	3.92	7.43	2.10	1.86	6.10	1.46	0.19	0.35	0.06
	2	2.31	5.45	3.38	23.26	59.29	56.84	0.65	0.62	0.68
20pkt/s	1	2.69	6.25	3.80	21.13	45.05	44.26	0.32	0.37	0.65
	2	1.25	1.91	3.77	57.13	90.70	474.89	1.33	0.82	1.50

packets that are received by destination nodes and g *is* number of overhead control packets that are sent or forwarded. Routing discovery packets include legitimate RREQ, fake RREQ, RREP, HELLO, and RERR packets.

$$RL = \frac{\sum_{j=1}^{g} CONTROL\_PACKET_j^{overhead}}{\sum_{i=1}^{n} DATA_i^{recieved}} \quad (7)$$

4.2. Simulation Results

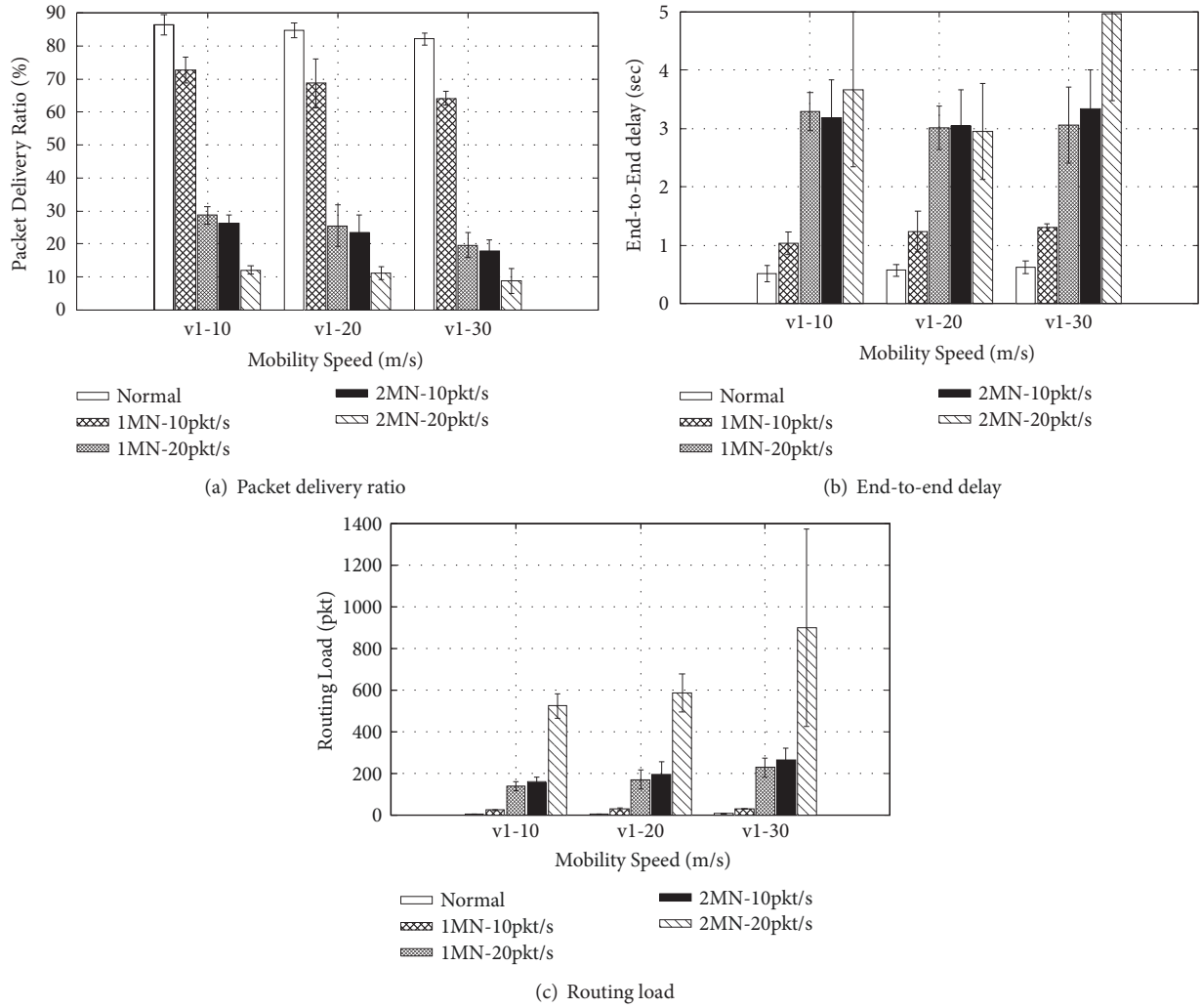
4.2.1. Effects of Flooding Attacks on the Original AODV Protocol. In this section we evaluate the performance of the AODV protocol with and without RREQ flooding attacks. We simulate 75 scenarios to evaluate the impact on the performance of AODV in terms of the above 4 defined metrics under various conditions including node mobility speeds, flooding frequencies, and malicious nodes. The main purpose of an RREQ flooding attack is to inject a large number of fake RREQ packets into the network making it less efficient in delivering legitimate packets. This effect is equivalent to handling excessive overhead packets causing a decrease in the network's packet delivery ratio, an increase in the average end-to-end packet delay, and an increase in the network's routing load. The simulation average results are shown in Table 4.

Figure 11 shows that the packet delivery ratio decreases, the routing load increases, and the end-to-end delay increases when the intruder floods attacking packets. Figure 11(a) shows that without flooding attack, the AODV packet delivery ratio is above 82.10% (1.77% standard deviation) and most packets reach their destination nodes. However, the packet delivery ratio reduced drastically to 12.06% (1.25% standard deviation) when the intruder uses 2 malicious nodes and floods 20 packets every second. Figure 11(b) shows that the average end-to-end delay increases as the flooding attack frequency increases. When the attacker uses 1 malicious node and broadcasts 10 RREQ packets every second, the average end-to-end delay changes from 0.506s before the attack to 1.032s after the attack for the 10m/s scenario. When the 2 malicious nodes broadcast 20 RREQ packets every second, the average end-to-end delay changes from 0.627s before the attack to 4.973s after the attack for the 30m/s scenario. Figure 11(c) shows that the routing load increases as the flooding attack frequency increases. When the attacker uses 1 malicious node and broadcasts 10 RREQ packets every second, the routing load changes from 4.92pkt before the attack to 25.45pkt after the attack for the 10m/s scenario. When the 2 malicious nodes broadcast 20 RREQ packets every second, the routing load changes from 7.02pkt before the attack to 898.82pkt after the attack for the 30m/s scenario.

4.2.2. Flooding Attacks Detection Performance of FAPRP. In this section we evaluate the malicious node detection performance of the proposed solution. Malicious node detection ratio is defined in (4). 216 scenarios are simulated: RDFV of size 10, 15, 20, 25, 30, 35, 40, and 60; the cutoff values of k for the kNN are set at 10, 15, 20, 25, 30, 35, 40, 45, and 50. Nodes move in a Random Way Point pattern with a specified maximum speed of 10m/s, 20m/s, and 30m/s. 20 source-destination UDP connections are set up among nodes. The intruder uses 2 malicious nodes and floods 20 packets every second.

The results in Figure 12 show that by making use of the route discovery history feature vector and the kNN machine data mining algorithm, our method achieves much higher malicious nodes detection ratios than those of existing algorithms and lower mistaken rates. The complexity of the overall detection algorithm is proportional to the size of the route discovery frequency vector. We see that the detection rate of FAPRP is above 99.0% and the mistaken rate is below 1.0% for all scenarios using RDFV vector sizes larger than 35. Figure 12(d) shows that the average of the maximum successful detection rate of FAPRP is above 99.77% when the cutoff value is 25 and RDFV vector size is 60. In brief, the proposed solution is effective in detecting the RREQ flooding attacks.

4.2.3. Performance Evaluation of AODV, B-AODV, and FAPRP. In this section we simulate 135 scenarios to evaluate the performance of the AODV, B-AODV, and FAPRP protocols under RREQ flooding attacks. The cutoff value



(a) Packet delivery ratio

(b) End-to-end delay

(c) Routing load

Figure 11: AODV performance under RREQ flooding attacks.

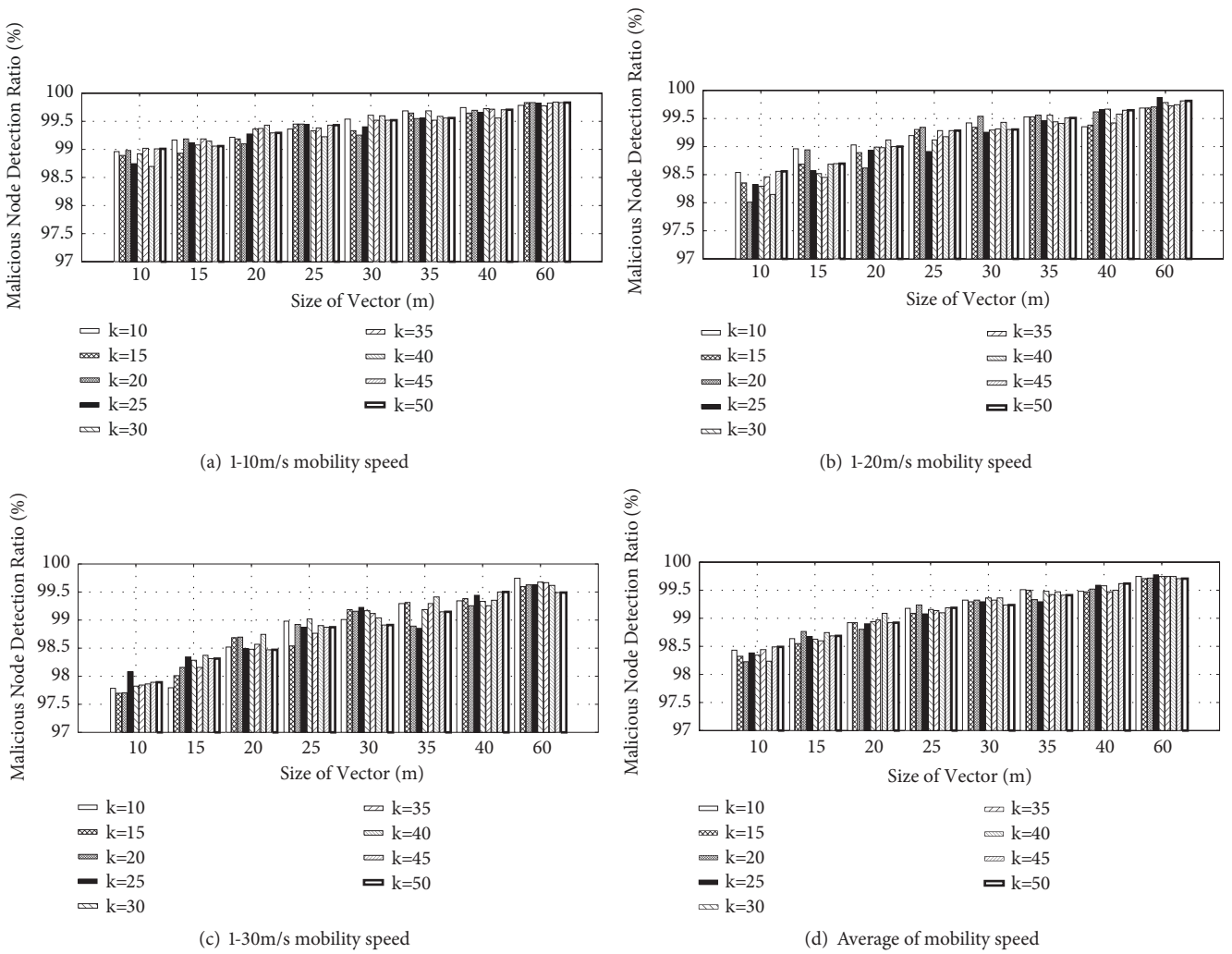
(k) is 25 and vector size (m) is 60. All nodes move in a Random Way Point pattern with specified maximum speeds of 10m/s, 20m/s, and 30m/s. Each of 2 malicious nodes floods 20 packets every second. 20 pairs of communicating nodes are set up among source nodes. The simulation average results are shown in Table 5.

(a) Packet Delivery Ratio. The results in Figure 13(a) show that the average packet delivery ratio for mobility speed by AODV is about 84.35% (1.86% standard deviation) in the absence of a malicious node. When there is one malicious node, the packet delivery ratio is about 24.65% (2.18% standard deviation) and 10.69% for two malicious nodes (0.9% standard deviation). This is due to RREQ flooding of the fake route request packets by the malicious node, resulting in a high consumption of bandwidth and buffer overloads at intermediate nodes with fake RREQs. For B-AODV in normal scenarios, the average packet delivery ratio is about 58.68% (3.16% standard deviation). In flooding scenarios, B-AODV average packet delivery ratio is above 59.32% when the intruder uses one or two malicious nodes. When our proposed solution is deployed, the packet delivery ratio for normal scenarios and high mobility speed is about 83.08% (2.47% standard deviation). Under flooding scenarios, FAPRP packet delivery ratio is above 82.06% when the intruder uses one or two malicious nodes, 2.73% maximum standard deviation. In brief, our solution is more efficient compared to AODV and B-AODV under normal network operation scenarios and more effective in handling RREQ flooding attacks with higher correct detection rates.

(b) End-to-End Delay. The results in Figure 13(b) show that with AODV, the average end-to-end delay is about 0.569s under normal scenarios. The end-to-end delays are about 3.121s and 3.864s for one and two malicious nodes, respectively. This high end-to-end delay is caused by the broadcasting of selective fake route request packets by the malicious nodes. For B-AODV under normal scenarios, the average end-to-end delay is about 1.091s. Under flooding scenarios, B-AODV end-to-end delay is about 1.056s with one malicious node and 1.145s with two malicious nodes. This is caused by the failure of B-AODV in detecting and preventing flooding

Table 5: AODV, B-AODV, and FAPRP performances.

				10m/s					
		PDR (%)			**RL (pkt)**			**ETE (sec)**	
MN	AODV	BAODV	FAPRP	AODV	BAODV	FAPRP	AODV	BAODV	FAPRP
0	86.26	59.89	84.73	4.92	3.11	4.69	0.506	0.790	0.526
1	28.75	55.01	83.94	140.55	4.13	6.05	3.292	0.865	0.566
2	12.06	59.30	83.80	524.18	5.98	7.34	3.668	0.921	0.598
					20m/s				
0	84.68	58.20	83.77	5.72	3.42	5.54	0.574	1.142	0.639
1	25.57	56.61	83.41	171.48	4.60	6.87	3.013	1.120	0.626
2	11.23	62.96	82.96	587.18	6.25	8.23	2.952	1.187	0.680
					30m/s				
0	82.10	57.96	80.75	7.02	3.57	6.60	0.627	1.342	0.703
1	19.63	57.50	79.92	228.57	4.88	8.05	3.059	1.185	0.813
2	8.78	55.69	79.41	898.82	7.09	9.28	4.973	1.327	0.798
					Average				
0	84.35	58.68	83.08	5.89	3.37	5.61	0.569	1.091	0.623
1	24.65	56.37	82.42	180.20	4.54	6.99	3.121	1.056	0.668
2	10.69	59.32	82.06	670.06	6.44	8.28	3.864	1.145	0.692
					Standard deviation values				
0	1.86	3.16	2.47	0.79	0.33	0.69	0.06	0.06	0.07
1	2.18	5.41	2.20	17.87	0.47	0.80	0.25	0.13	0.10
2	0.90	2.35	2.73	146.54	0.48	0.89	0.43	0.19	0.07



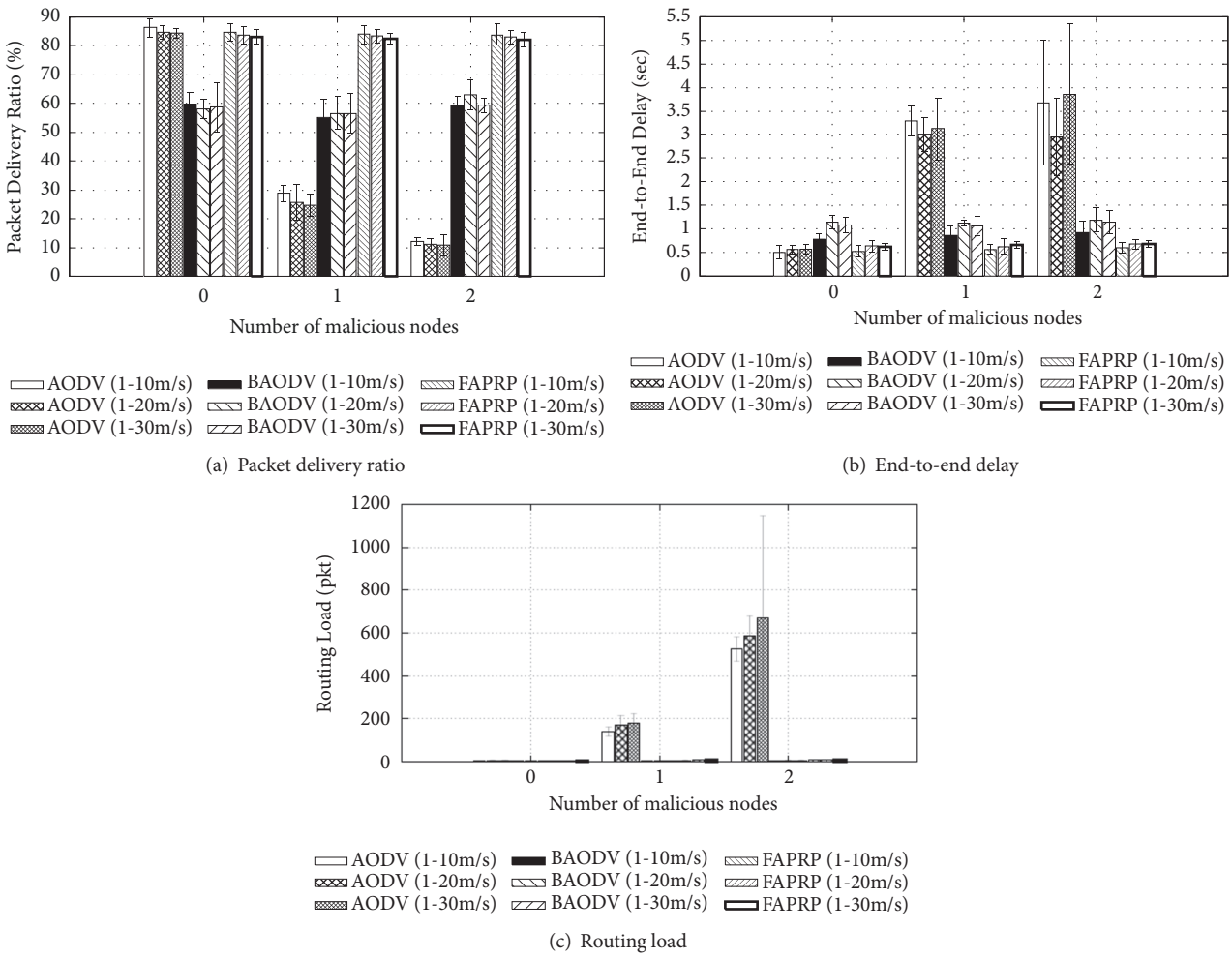
(a) 1-10m/s mobility speed

(b) 1-20m/s mobility speed

(c) 1-30m/s mobility speed

(d) Average of mobility speed

Figure 12: Malicious nodes successful detection ratio.



(a) Packet delivery ratio

(b) End-to-end delay

(c) Routing load

Figure 13: AODV, B-AODV, and FAPRP performances under RREQ flooding attacks.

attacks resulting in lower packet delivery ratios and longer route discovery delays. For our proposed solution, the average end-to-end delay for normal scenarios and mobility speed is about 0.623s. Under flooding attacks, FAPRP average end-to-end delays are about 0.668s and 0.692s when intruder uses one and two malicious nodes, respectively. Clearly, FAPRP achieves shorter end-to-end delay compared to AODV under flooding attack scenarios and B-AODV under both normal and flooding attack scenarios.

(c) Routing Load. The results in Figure 13(c) show that the average routing load for high mobility speed by AODV is about 5.89pkt in the absence of a malicious node. The routing loads are about 180.2pkts and 670.06pkts for one and two one malicious nodes, respectively. The high routing load is caused by the broadcasting of selective fake route request packets by the malicious nodes. For B-AODV in normal scenarios, the routing load is about 3.37pkt. B-AODV average routing load in attacks state is about 4.54pkt when the intruder uses one malicious node and 6.44pkt for two malicious nodes. For our proposed solution, the routing load for normal scenario and high mobility speed is about 5.61pkt. Under flooding attacks, FAPRP average routing load is about 6.99pkts and 8.28pkts when the intruder uses one and two malicious nodes, respectively. B-AODV routing load is, however, better as compared to AODV as it drops many route request packets due to mistake detection. Overall, FAPRP performs as well as AODV in the routing load measure under both normal and flooding attack scenarios due to its high correct detection rate and low mistake rate.

5. Conclusion

In this paper, we introduced the flooding attack detection algorithm based on our proposed route discovery frequency history feature vector and the kNN data mining algorithm to detect and isolate the malicious nodes in the network. We introduced a new FAPRP protocol by integrating FADA into the route request phase of AODV. Using route discovery frequency vector sizes larger than 35, the simulation results show that FADA achieves higher misbehaving detection ratio (above 99.0%) as compared with existing algorithms and lower mistaken rate (below 1.0%). Furthermore, the proposed solution is efficient in that it improves the network

performance in terms of higher packet delivery ratio, smaller end-to-end delay, and reduced routing load compared to AODV and B-AODV protocols.

In the future, we will extend the proposed solution for mitigating the effects of other flooding attacks.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

This research was supported by the DThU (Dong Thap University), Vietnam, under the PhD Thesis (62.48.01.01) supervised by the Hue University of Sciences, Hue University.

Supplementary Materials

We submit the source code for the AODV, B-AODV, and FAPRP protocols and analysis files (.tcl, .awk) for the simulation with this revision. ID, file name, description are as follows: 1, aodv_cc.rar, source code of AODV routing protocol for simulation in NS2.35; 2, fdaodv_cc.rar, source code of FDAODV routing protocol for malicious node simulation in NS2.35; 3, baodv_cc.rar, source code of BAODV routing protocol for simulation in NS2.35; 4, fdbaodv_cc.rar, source code of FDBAODV routing protocol for malicious node simulation in NS2.35; 5, faprp_cc.rar, source code of FAPRP routing protocol for simulation in NS2.35; 6, fdfaprp_cc.rar, source code of FDFAPRP routing protocol for malicious node simulation in NS2.35; 7, scen.rar, 15 network topologies for simulation; 8, TCL.rar (TCL source code is used to write simulation script in ns2), analysis files (.awk) for the simulation; 9, DATA.rar, all simulation data; 10, Figures.rar, all scripts (gnuplot) to create the figures in the paper. *(Supplementary Materials)*

References

[1] J. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester, "An overview of mobile ad hoc networks: Applications and challenges," *Journal of the Communications Network*, vol. 3, no. 3, pp. 60–66, 2004.

[2] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, New Orleans, La, USA, February 1999.

[3] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 234–244, London, UK, 1994.

[4] Z. J. Haas, M. R. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *INTERNET-DRAFT*, pp. 1–11, 2002.

[5] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-Hoc and mesh networks," *Computer Networks*, vol. 56, no. 2, pp. 940–965, 2012.

[6] R. Di Pietro, S. Guarino, N. V. Verde, and J. Domingo-Ferrer, "Security in wireless ad-hoc networks - A survey," *Computer Communications*, vol. 51, pp. 1–20, 2014.

[7] R. Mitchell and I.-R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, pp. 1–23, 2014.

[8] M. Wazid and A. K. Das, "A Secure Group-Based Blackhole Node Detection Scheme for Hierarchical Wireless Sensor Networks," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1165–1191, 2017.

[9] E. C. H. Ngai, J. Liu, and M. R. Lyu, "An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2353–2364, 2007.

[10] S. Gurung and S. Chauhan, "A novel approach for mitigating gray hole attack in MANET," *Wireless Networks*, vol. 24, no. 2, pp. 565–579, 2018.

[11] T. L. Ngoc and T. T. Vo, "Whirlwind: A new method to attack Routing Protocol in Mobile Ad hoc Network," *International Journal of Network Security*, vol. 19, no. 5, pp. 832–838, 2017.

[12] T. T. Vo, N. T. Luong, and D. Hoang, "MLAMAN: a novel multi-level authentication model and protocol for preventing wormhole attack in mobile ad hoc network," *Wireless Networks*, 2018.

[13] Y. Ping, D. Zhoulin, Y. Zhong, and Z. Shiyong, "Resisting flooding attacks in ad hoc networks," in *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, vol. 2, pp. 657–662, Las Vegas, NV, USA, April 2005.

[14] H. Ehsan and F. A. Khan, "Malicious AODV: Implementation and analysis of routing attacks in MANETs," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012*, pp. 1181–1187, UK, June 2012.

[15] D. Gada, R. Gogri, P. Rathod et al., "A distributed security scheme for ad hoc networks," *The Crossroads Journal*, vol. 11, no. 1, pp. 1–14, 2004.

[16] J.-H. Song, F. Hong, and Y. Zhang, "Effective filtering scheme against RREQ flooding attack in mobile ad hoc networks," in *Proceedings of the 7th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2006*, pp. 497–502, Taiwan, December 2006.

[17] P. Yi, Y. Hou, Y. P. Zhong, and Z. L. Dai, "Flooding attack and defence in ad hoc networks," *Journal of Systems Engineering and Electronics*, vol. 17, no. 2, pp. 410–416, 2006.

[18] M. J. Faghihniya, S. M. Hosseini, and M. Tahmasebi, "Security upgrade against RREQ flooding attack by using balance index on vehicular ad hoc network," *Wireless Networks*, vol. 23, no. 6, pp. 1863–1874, 2017.

[19] S. Gurung and S. Chauhan, "A novel approach for mitigating route request flooding attack in MANET," *Wireless Networks*, vol. 24, no. 8, pp. 2899–2914, 2018.

[20] V. Thanh Tu and L. Thai Ngoc, "SMA2AODV: Routing protocol reduces the harm of flooding attacks in mobile ad hoc network," *Journal of Communications*, vol. 12, no. 7, pp. 371–378, 2017.

[21] M. Patel, S. Sharma, and D. Sharan, "Detection and prevention of flooding attack using SVM," in *Proceedings of the 3rd International Conference on Communication Systems and Network Technologies, CSNT 2013*, pp. 533–537, India, April 2013.

[22] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network," *Journal of Electrical and Computer Engineering*, vol. 2014, Article ID 240217, 8 pages, 2014.

[23] DARPA, *The network simulator NS2*, 1995, https://www.isi.edu/nsnam/ns/.

[24] S. K. Sahu, P. Kumar, and A. P. Singh, "Modified K-NN algorithm for classification problems with improved accuracy," *International Journal of Information Technology*, vol. 10, no. 1, pp. 65–70, 2018.

[25] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 2, pp. 1312–1321, San Francisco, Calif, USA, March-April 2003.

[26] S. Gurung and S. Chauhan, "A dynamic threshold based algorithm for improving security and performance of AODV under black-hole attack in MANET," *Wireless Networks*, pp. 1–11, 2017.