Analysis of node attack in MANET and Intrusion detection using AI

Aakruti Ambasana, Mudit Parmar, and Md Rubayatur Bhuyian

Abstract—Mobile Ad-hoc Network (MANET) is a decentralized wireless network that is self-configuring, self-organizing, and does not rely on preexisting infrastructure. Due to its open and dynamic structure, MANET is more susceptible to intrusion, affecting the overall network functionality, reliability, and performance. Active attacks like black-hole attacks, hello flooding attacks, and denial of service (DoS) attacks are common types of attacks in MANET. Nodes in a MANET work simultaneously as a host and a router to do transmission and reception. As a result, it is easier to attack them, and any node can be under attack at any point in time. Dynamically changing topology makes it is even more challenging to detect the attacker node. It is imperative to identify the malicious nodes within the shortest possible time to immediately isolate the compromised nodes and save the rest of the network. In this paper, different node attacks in a MANET are modeled and simulated in OMNET++ for different mobility models and node speed. Obtained results are analyzed in terms of data packet delivery ratio (PDR), end-to-end delay, network throughput, and energy consumption. This helps us understand the impact of different node attacks in MANET and can be used to enhance the performance of the existing protocols. Moreover, this paper also presents an offline intrusion detection system (IDS) using an artificial neural network (ANN) that identifies the attacker node in the MANET. This is achieved by classifying normal and threat patterns, and the approach is validated using data obtained from OMNET++ simulation. The results show that the IDS is 97.80% accurate and can detect attacker nodes successfully.

Index Terms—Node Attack, Arificial Neural Network, AODV, Mobility Model, MANET.

1 Introduction

Mobile Ad-Hoc Networks (MANET) offer access and exchange of information between users regardless of their geographic location or proximity to any central infrastructure. MANET finds numerous applications in various fields such as sensor networks, environmental monitoring, military, private sectors, etc. [1]. In a typical network, a central node coordinates activities like routing, packet forwarding, and network management, whereas in MANET, these tasks are performed by every node in the network [2]. The network topology changes dynamically in MANET due to nodes' unpredictable movement in all directions at varying speeds. Connectivity between the nodes is maintained through wireless links, and the nodes work together to carry data from one another. Nodes in MANET work both as a host and a router simultaneously to do transmission, and reception [3]. The decentralized character makes MANET more flexible and robust.

In MANET, nodes are typically energy-constrained, and bandwidth is limited and low compared to wired networks. As a result, routing protocols are designed in such a way as to optimize the resources, although the primary goal of such routing protocols is to adapt to changing topology [4]. Routing protocols for MANET can be divided into two categories: Proactive and Reactive. In proactive routing protocol, routing information is maintained in a table and up-to-date routing information is obtained by periodically sending control messages. It uses link-state routing algorithms, which frequently flood the link information about its neighbors [5]. Reactive or on-demand routing protocols create routes as per source host requirement, and these routes are maintained as long as they are needed [6]. Distance-vector routing algorithms is used for such protocols [7]. Ad hoc On-Demand Distance Vector (AODV) Routing protocol, Dynamic Source Routing (DSR) protocol, and Distance Sequenced Distance Vector (DSDV) are some popular routing protocols used in MANET.

MANET is more vulnerable to different types of attacks and threats by design. Wireless links make MANET more susceptible to attacks [7] compared to wired networks, as in wired networks, an attacker requires physical access to the network. In MANET, any node may be attacked at any point in time. Attacks in MANET are divided into two categories: Active and Passive attacks. In Active attacks, the attacker nodes work to affect the MANET operation. Dropping the forwarded data, altering the connection links, draining the node's battery by requesting them to forward a huge amount of data are common types of active attacks. In passive attacks, the attacker node does not affect the communication operation; instead, it eavesdrops on the communication between two nodes [8]. Having no central node to monitor the network makes it easy for the attacker to deploy an attack in MANET.

Mobility configuration plays a vital role in MANET. It is hard to predict the behavior of an attacker node in a dynamically changing environment. Due to this, a malicious node might go undetected sometimes. So it is very important to understand the impact of an attacker node on the overall network in MANET. To do so, different attacks in MANET, such as black hole attack, hello flooding attack, and UDP flood attack were modeled, and simulations were performed using OMNET++. The obtained results were analyzed in terms of data packet delivery ratio (PDR), end-to-end delay, network throughput, and energy consumption. The behavior of an attacker node under different mobility models and speed was observed and the impact on the overall network was accessed. It was observed that in most cases attacker node is one of the most energy-consuming nodes. Again, only energy consumption may not be a good indicator to

identify a malicious node. Additional information about the node is required to detect an attacker node accurately. This paper also presents an offline intrusion detection system (IDS) using an artificial neural network (ANN) that identifies the attacker node in the MANET. This is achieved by classifying normal and threat patterns, and the approach is validated using data obtained from OMNET++ simulation. The results show that the IDS is 97.80% accurate and can detect attacker nodes successfully.

In summary, the paper makes the following contributions:

- Analyzes the behaviour of an attacker node and the impact of an attack on the overall network.
- Proposes an offline intrusion detection system (IDS) using an artificial neural network (ANN).

The rest of the paper is organized as follows. Section 2 gives an overview of different attacks and mobility models and discusses the AODV routing Protocol. Section 3 describes the general configuration and network parameters used for the simulations. Section 4 presents the implementation of different attacks and their outputs, respectively. Section 5 discusses the impact of different attacks on the overall network 6 presents the intrusion detection technique using AI. Section 7 presents a summary and discusses possible extensions of the work.

2 Background

2.1 AODV Routing protocol

AODV is a reactive routing protocol that builds routes between nodes when needed. AODV uses a distance-vector routing algorithm that keeps only information about next hops to adjacent neighbors and costs for paths to all known destinations [9]. This information about the neighbors and the cost is stored in the routing tables. All the nodes have a sequence number, which helps it avoid the looping problem, making it a loop-free routing protocol. The AODV proves to be robust in the case of mobile ad-hoc networks where the network has self-configuring and self-organizing mobile nodes. In such mobile ad-hoc networks, it is not possible to have a fixed infrastructure as the nodes are moving and the topology varies with time. When we are using AODV, and a mobile node moves out of the range of the other mobile node, the old connections break, and a new path is built. The other benefits of using AODV as the routing protocol are that it is scalable and requires very little bandwidth as there is no global periodic routing advertisement.

The route discovery process is initiated when a new route is to be built. Route discovery is done in two steps. The first step is Reverse Path Setup; in this step the source initiates the route finding process by sending Route Request (RREQ) message. This message is broadcasted and is forwarded by all the nodes that have not seen this message before. RREQ travels to various destination nodes and automatically sets up the reverse paths to these nodes [10]. The second step in Forward Path Setup, when the RREQ reaches a node that is the destination node or a node that contains the route to the destination, then that node sends out a Route Reply (RREP) message to identify that a route has been found. The route from source to destination is saved in routing table. Route Reply Acknowledgement (RREP-ACK) and Route Error (RERR) control messages are used for path maintenance. The sender sends out RREP-ACK to ensure that the link from sender to destination is valid. RERR is sent by source, destination, or path node and is sent when the link breaks and a new route needs to be discovered.

The management of the routing table is an important part of AODV. The entries in the routing table contain the information about the destination node, next-hop, number of hops, sequence number for the destination, and expiration time for the route request. The destination sequence number acts as a timestamp, and its value is changed in the routing table if the node receives a new route with a greater sequence number. This ensures that the table has the latest route. The route request expiration time helps to get rid of the entries that do not lie on the source to destination path.

2.2 Mobility models

Mobility models have an important role in mobile ad-hoc networks. They are responsible for deciding the movement of mobiles nodes in the context of location, direction, and velocity with time. Mobility models help to simulate the real-world behavior of mobility nodes. The mobility models can be grouped based on some characteristics that they display. The mobility models can be grouped as stationary, deterministic, trace-based, stochastic, or combining [11]. The stationary mobility models have only one fixed position and no movement. Deterministic models use non-random mathematical models for their motion, whereas stochastic models use random mathematical models for movement. Trace-based mobility models replay recorded motions, and combining mobility models are just a combination of one or more simpler models. In our simulations, we have used stochastic mobility models. The types of stochastic mobility models used are:-

2.2.1 Mass mobility Model

This is a model for a mobile host with a mass. The time period for which a node will move in a specific direction depends on the random number that is generated by the parameter change interval, which takes the average of the time provided to it and the standard deviation. The node moves in the direction of a normally distributed random number with an average of previous direction and standard deviation. The mass mobility model is implemented by setting the .mobility.typename as “MassMobility”. Along with this, we have also defined the change Interval and angle Delta parameters.

2.2.2 Random Waypoint mobility model

In the random waypoint mobility model, the nodes first begin with pausing for a few seconds and then randomly select a destination node and move towards it with a randomly selected speed between 0 and the max speed. After reaching that destination, it pauses for a while again, repeats randomly selecting a destination node, and starts moving towards it with a random speed [12]. It is implemented by setting the mobility.tynename= “RandomWayPointMobility”. For both mobility models, the speed of mobility can be configured at the beginning of the simulation and range from 0 to a maximum of 20m/s.

2.3 Node Attack

Mobile ad hoc networks are vulnerable to security attacks because of their properties, like lack of a central entity, and also the nodes are mobile and dynamic. The resources in wireless sensor networks are limited; hence, sophisticated security techniques and protocols cannot be applied to them. In contrast to this, the attacker node in the mobile ad-hoc networks can have more equipped resources. The denial of service attacks can go through all the layers of the protocol stack. The probability of attack in mobile ad-hoc networks is higher than in wired networks because gaining excess to network physically is tough. There are various node attacks in the mobile ad-hoc networks like selective forwarding attack, black-hole attack, hello flooding attack, sink-hole attack, UDP flooding attack, Sybil attack, etc. In our paper, we have implemented the following three attacks.

2.3.1 Hello Flooding Attack

The Hello flooding attacks can be caused by a node that broadcasts a Hello packet with very high transmitting power so that a large number of nodes even far away in the network choose it as a node in the route [13]. This gives the attacker node probability of being selected as a route. After the malicious node receives packets from these nodes, it drops all the packets.

2.3.2 Black Hole Attack

The Black-Hole attack is also a popular type of denial of service node attack. In the Black-hole attack, the attacker node drops all the packets that it receives instead of forwarding them. This attack is different from the hello flooding attack. In this attack, the transmitting power for the attacker node is the same as that of the other nodes in the network. There is no increased probability of the attacker node getting selected.

2.3.3 UDP Flood Attack

The UDP flooding attack is a kind of attack in which the network is bombarded with a huge number of UDP Packets, so that the nodes are choked out of resources. This is one of the most common type of denial of service attack. In this, a UDP source sends many packets to target nodes reducing their resources.

3 Design Parameters

This section outlines the general configurations and the network parameters that we have used to simulate different attacks.

Table 1 shows the general configuration for the overall simulation layout. The area we have selected is of length 500m and width 500m. The initial position of the sender and the receiver are fixed, and it's set to opposite corners. This is done so that messages from the sending node are routed through other nodes in the network to reach the receiver node. The other 23 nodes start from random locations within the area and stay within the defined area throughout the simulation regardless of mobility scheme and speed. Out of the 23 nodes, any node could be an attacker node and is set manually before the simulation is performed.

TABLE 1
General Configuration.

General Configuration	
Simulation time	30s
Number of nodes	25
Area	500mx500m
Intial position of HOSTA	(20m,20m)
Intial position of HOSTB	(480m,480m)
Position of intermediate nodes	Random
Mobility Model	MassMobility, RandomWayPoint
Speed	1-20ms

Table 2 shows the specific network configuration opted for simulating the node attack. AODV has been used as the routing protocol to perform all the simulations. The sender node sends a UDP packet of 1000 Bytes to the receiver node roughly around 15ms. To mimic an actual scenario, under normal operating conditions, other nodes in the network communicate with each other at a lesser frequency and the packet size is set to 500 Bytes. Depending on the attack types, these parameters were changed accordingly.

TABLE 2
Routing protocol parameters.

Network parameters	
Communication type	Wireless
MAC Protoco	IEEE 802.11g
Type of traffic	UDP
Routing protocol	AODV
Type of IP	IPv4
Source port	5000
Destination port	5000
Packet length	1000 Bytes
Send interval	exp(15ms)
Host bitrate	1Mbps

Table 3 shows the power consumption parameters of transmitter and receiver at different states. Under normal operating conditions, a radio transmitter's transmitting power is set to 1.4mW. This parameter was varied depending on the attack type, but the other parameters were kept the same throughout all the simulations.

TABLE 3
Power consumption parameters.

Power consumption parameters.	
Sleep Power Consumption	1mW
Switching Power Consumption	1mW
Receiver Idle Power Consumption	2mW
Receiver Busy Power Consumption	5mW
Receiver Receiving Power Consumption	10mW
Transmitter Idle Power Consumption	2mW
Transmitter Transmitting Power Consumption	100mW
Radio transmitter power	1.4mW
Battery type	Inet Simple Battery

Figure 1 shows the overall layout of the simulation. The mobility model used for the simulation was mass mobility,

and the speed at which the nodes were traveling was 4 m/s. The line attached each node depicts the path it followed during the simulation. The packet sent count represents the number of UDP packets sent by HOSTA, and the packets received count represents the number of packets received by HOSTB. Packet delivery ratio (PDR), throughput, and end-to-end delay, all these performance matrices were calculated based on the count stated above.

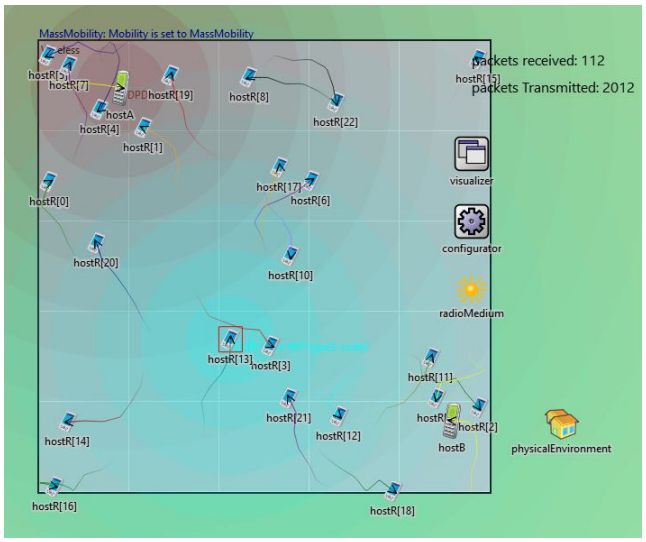

Fig. 1. Simulation Layout.

4 Implementation and Analysis of Attacks

4.1 Hello Flooding Attack

In Omnet++, the hello flooding attack is simulated by making the transmission power of the malicious node high. This is done by setting the .wlan*.radio.transmitter.power high as compared to the other nodes [14]. And we set *.malicioushost.forwarding= false so that the attacker node doesn't forward any packets. The total number of nodes considered for this simulation are 25, out of these 2 nodes are set as malicious nodes which have a transmitting power of 10mW and have forwarding parameter set to false. The transmitting power of the other nodes is 1.4mW.

It can be seen in fig. 2 that the source node has consumed the most energy since the transmitter transmitting power consumption is 100mW and all the packets are transmitted by the source . For the time interval between 3.5 to 12.8 seconds, the packets are being dropped by the malicious nodes so no power is being consumed for forwarding by the relay nodes.

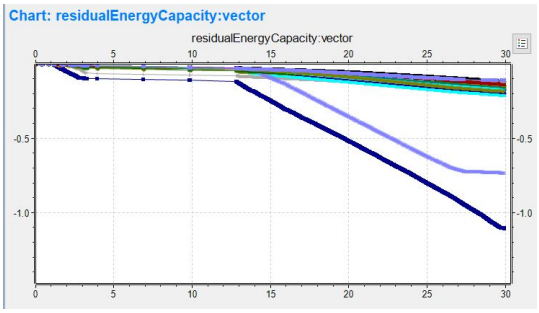

Fig. 2. Energy consumption vs Time.

The end-to-end delay histogram for sink node for network under hello flooding attack is shown in fig. 3. The end-to-end delay for this simulation mostly ranges from 0.05 seconds to 0.9 seconds with a few extreme cases having delay value more than 2 seconds. The most common value for delay is between the range 0.2 to 0.25 seconds.

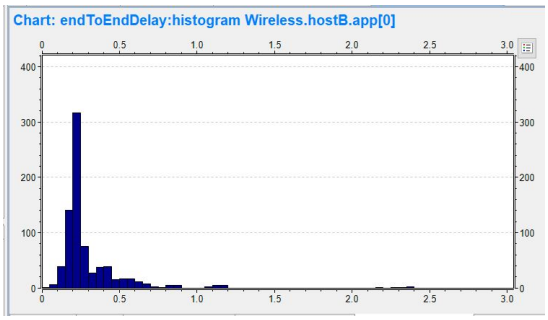

Fig. 3. Histogram of end-to-end delay.

Figure 4 depicts that the throughput vector graph has 0 values for some corresponding values of time in seconds. For this specific simulation we can see that the throughput value for Hello flooding attack is zero for 0 to 2 seconds, 3.5 to 12.8 seconds and 27.4 to 28.6 seconds time intervals. This is because during these time intervals the route followed from the source to destination has the malicious node which drops the packets. Hence the packets are not delivered and the throughput is zero.

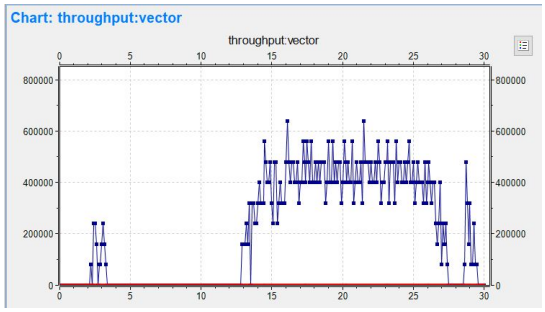

Fig. 4. Throughput under Hello Flooding Attack.

4.2 Black Hole Attack

It is implemented by changing code in .ini file such as $*.maliciousHost.forwarding = false$ in Omnet++. This means whenever attacker node receives any packet it will drop it.

Lets consider, out of 25 nodes in network, 3 nodes are attackers. The sender tends to send many packets through this three nodes. As attacker node receives the packet, it will drop it. As a result, we can say that throughput will be decreased, packet delivery ratio will be dropped.

The fig. 5 indicates the Residual Energy Capacity of 25 nodes. The Host A which is sender node have highest residual energy of -1.8 because it have transmitted all the packets. The energy consumed for transmitting packet is 100mW and energy consumed for receiving the packet is 10mW. Whenever attacker node receives a packet, it drop it. So, the energy consumption will be low because we are not

transmitting any packet. Where as, all the intermediate node including attackers and receiver node have residual capacity between -0.1 to -0.3 because only node A have transmitted all the packets.

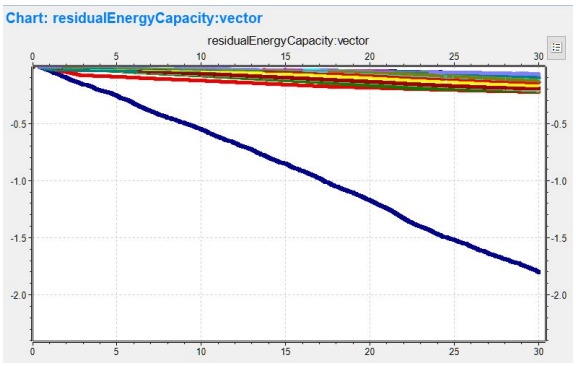

Fig. 5. Energy consumption vs Time

The fig. 6 shows end-to-end delay histogram. In this, most of the packets have end-to-end delay between 1 to 2.5 time unit. Under normal scenario in AODV, the shortest path is always chosen for transmitting the packets from sender to receiver. Therefore, end-to-end delay is less when there is no black hole node in network. The end-to-end delay is used for comparing the time required to transmit the packet under normal scenario and when attack is happening. Based on that difference we can find out that whether the network is under attack or not. Therefore, the average end-to-end delay when network is under attack is higher than the normal scenario [15], [16].

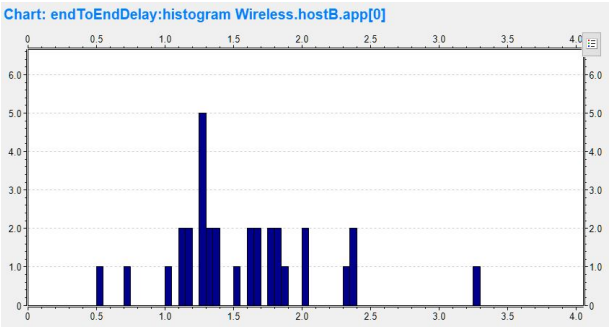

Fig. 6. Histogram of end-to-end delay.

The fig. 7 indicated the throughput vs time. The simulation time is 30 sec.

$$Throughput = \frac{Packets\ Received * Packet\ size}{Time}$$

The throughput is high, if the number of packets received is more in that time. Therefore, in normal scenario throughput will be higher because packets are transmitted from sender to receiver. Apparently, when attack is happening on network then throughput will be low because attacker will drop the packets. Therefore, from fig. 7 we can clearly see that at what time, attacker was active and dropping the packets in network. We can see that after 5 to 7 seconds no packets are transmitted because throughput is 0. If throughput is low but greater than 0 then some packets are transmitted but if throughput is 0 then no packets are transmitted in that time period [15], [16].

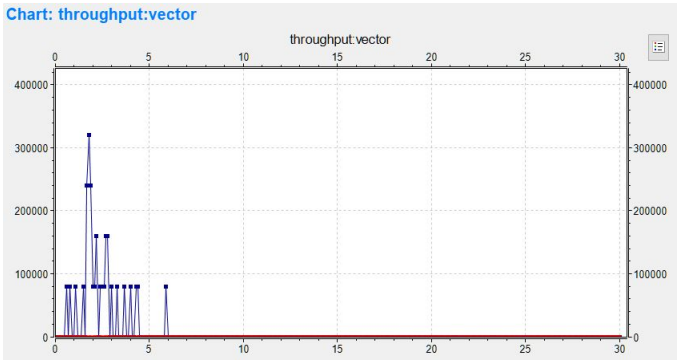

Fig. 7. Throughput under Black Hole Attack

4.2.1 UDP Flood Attack

To simulate this in Omnet++, we randomly selected a few nodes out of all the relaying node as malicious nodes. The malicious nodes send UDP packets at a higher frequency to other relaying nodes causing network congestion. The traffic in the network increases abruptly, and the relaying nodes fail to handle the regular traffic of the network. This causes an overall decrease in the number of packets received by the UDP sink node. For the results shown below we have set two attacker nodes, one source node, one sink node and the rest are relay nodes.

From fig. 8 we can see that that the source node (red) and the attacker nodes (blue and cyan) have the highest power consumption. From the power consumption data, it is not evident which one is the attacker node and which one is the sender node. As a result, further investigation into other performance matrices is required to identify the attacker node. Due to the random movement of the nodes, sometimes these nodes can also act as relay nodes, which might significantly impact their energy consumption pattern.

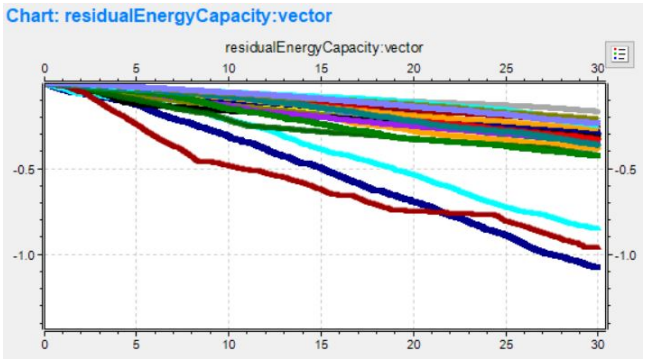

Fig. 8. Energy consumption vs Time.

Figure 9 shows the histogram of the end-to-end delay for sink node under UDP flood attack. In the case of a UDP flood attack, the end-to-end delay for the sink node mostly ranges from 1.0 seconds to 2.0 seconds, with a few instances having a delay value of more than 3.0 seconds.

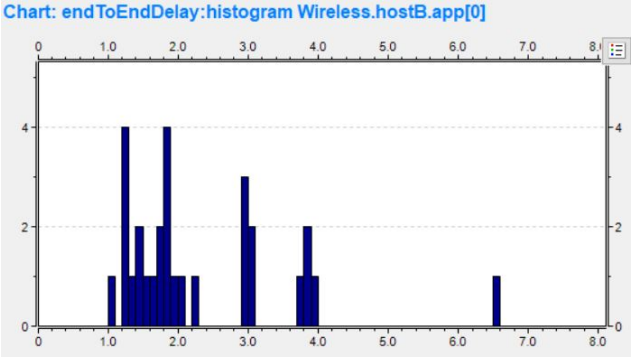

Fig. 9. Histogram of end-to-end delay.

The throughput of the sink node under the UDP flood attack is shown in fig. 10. In most cases, we can see that the packets were dropped due to network congestion, and the throughput is zero. The spike in the graphs represents packets received by the sink node, and the value indicates the corresponding throughput. UDP flood attack significantly reduces the normal traffic of the network.

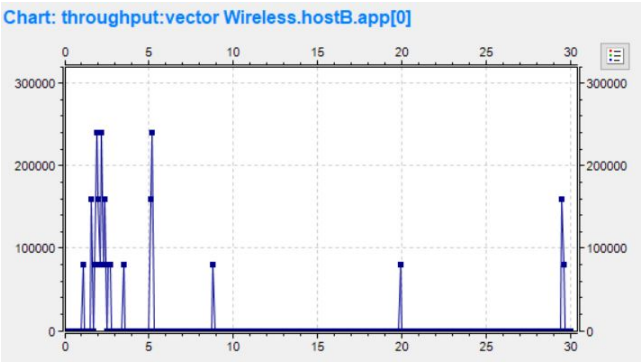

Fig. 10. Throughput under UDP Flood Attack.

5 Comparison of Results

In this section, we compare the different types of attacks and their impact on throughput, end-to-end delay, and packet delivery ratio of the network with change in mobility speed, the number of attackers, and the total number of nodes in the network. For the first three graphs in this section, the mobility model used is mass mobility, and the number of attackers is set to two. In the following three graphs, we compare the above-stated parameters based on the number of attacker nodes. We have considered the mobility model as mass mobility, and the mobility speed is fixed to 12m/s. In the last three graphs, we check the same performance matrices by varying the total number of nodes in the network.

It can be seen in fig. 11 that the simulations without any node attack have higher throughputs for the different speeds, as compared to a network in which malicious nodes are present. Since when no attacker is present, all the nodes forward the packets, and the transmission of packets takes place smoothly. In a mobile ad-hoc network, since the position of the nodes is random, the throughput highly depends on the number of relay nodes required and if any node among them is a malicious node. When the network attack is of type UDP flooding, the throughput is generally low, as the malicious nodes are utilizing all the resources of the nodes, and the speed with which the nodes move has minimal impact [14] [15] [16].

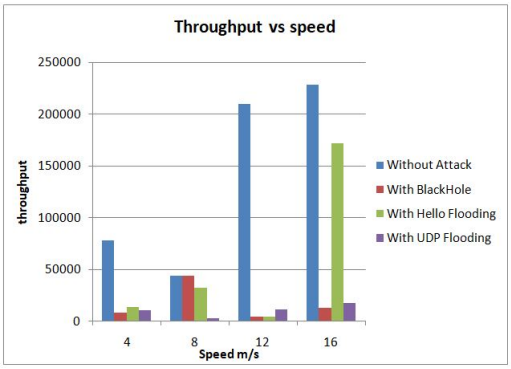

Fig. 11. Throughput vs Speed.

Figure 12 shows the impact of mobility speed on end-to-end delay. The end-to-end delay is maximum for all the different mobility speeds for the networks with UDP flooding attack. The value for the end-to-end delay is for mobility speed 4 for UDP flooding attack is around 4.96s, which is the highest compared to others. In a mobile ad-hoc network, the nodes are mobile; therefore, the end-to-end delay depends on the position of the source and sink node and has different values in different cases [14] [15] [16].

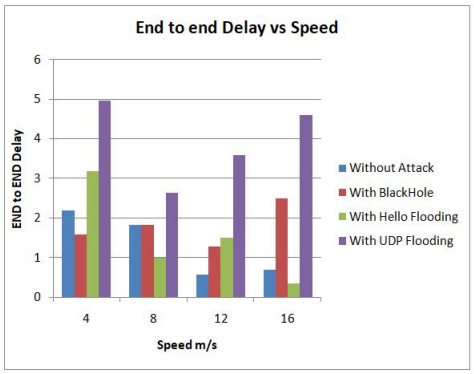

Fig. 12. End-to-end delay vs Speed.

As the bar chart in fig. 13 depicts, the network without any attack has a higher packet delivery ratio than the networks with malicious nodes. For mobility speed 16m/s, the simulation environment with two attacker nodes gives a packet delivery ratio of 0.4461, which is higher than all the other scenarios depicted. The network with hello flooding attack with speed 16m/s shows a higher packet delivery ratio than the hello flooding attack simulated with speeds 4 ,8 and 12m/s. The possible reason could be due to the random positioning of the attacker node. Attacker nodes moved away from the sender node; as a result, packets were not relayed through them [14] [15] [16].

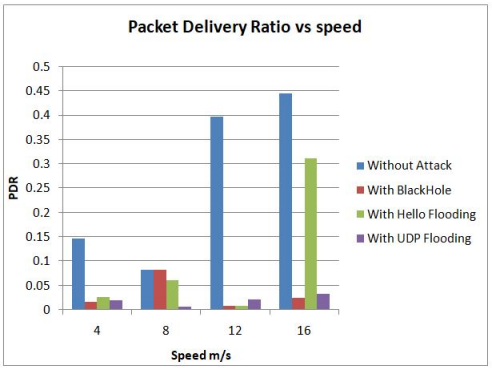

Fig. 13. Packet delivery ratio vs Speed.

In fig. 14, the throughput is maximum for all the attack types when the number of attacker nodes is 1. As the number of attacker nodes increases, the throughput of the network decreases. When the number of malicious nodes causing Hello flooding is set to 4 in the simulating environment of 25 nodes, we get a throughput of 0. For four nodes causing UDP flooding, we only get a throughput of 266.66bps [14] [15] [16].

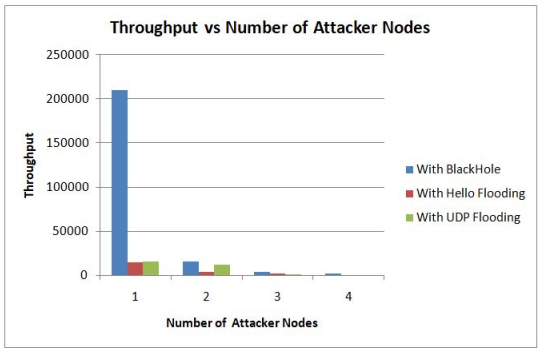

Fig. 14. Throughput vs Number of attacker Node.

The fig. 15 depicts that the end-to-end delay is the least for the black-hole attack and is generally high for UDP flooding attacks. As the number of attacker nodes causing black-hole increases, the end to delay tends to increase, but in case the malicious node is not a part of the route, the end-to-end delay is less [14] [15] [16].

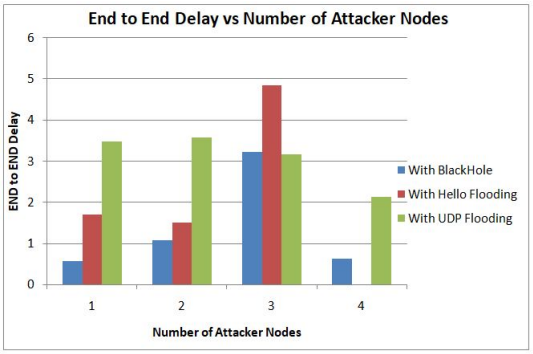

Fig. 15. End-to-end delay vs Number of attacker nodes.

It can be seen in fig. 16 that as the number of attacker nodes increases the packet delivery ratio decreases. When four are set as malicious nodes in a simulated network of 25 nodes, the packet delivery ratio falls below 0.005. When only one attacker is present in the network causing a black-hole attack, the packet delivery ratio is close to 0.39; a reason for this can be that the malicious node is not a part of the route for a majority of simulation time [14] [15] [16].

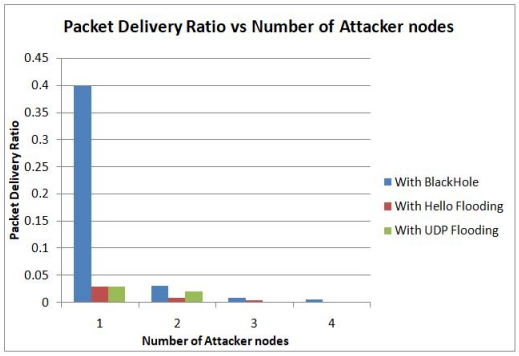

Fig. 16. Packet delivery ratio vs Number of attacker nodes.

Figure 17 shows the comparison between the network behaviour without any attacks, and network behaviour with various attacks. In fig. 17, throughput is plotted against the number of nodes. We can see that when the blackhole attack and hello flooding attack is performed on network, throughput is decreased because attacker node drops the packets it receives. Apparently, throughput of UDP Flooding Attack will be high because new packets are generated by attacker nodes and flooded the network with the UDP packets. So, throughput with UDP Packet flooding attack is higher compared to without attack scenario. From this graph we can observe that by increasing number of nodes, throughput will not be affected. The result of throughput will be affected by mobility model and mobility speed and position of node in network. In MANET all the host in network are mobile including attacking nodes and location of all the nodes is also random. We cannot predict the approximate throughput by number of nodes in network because it does not follow any trends. But we can determine whether the attack is happening in network or not. If throughput is less than normal scenario then blackhole attack or hello flooding attack is performed on network. Apparently, if throughput is greater than normal scenario then UDP Flooding attack is performed on network [14] [15] [16].

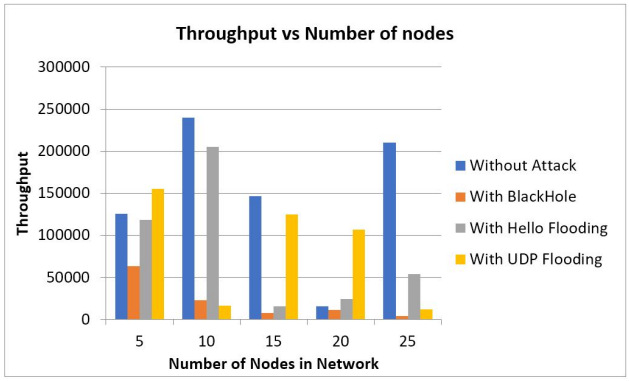

Fig. 17. Throughput vs Number of nodes.

Figure 18 shows that if number of nodes in network increases, then what will be the end to end delay, under various attacks or without any attack on network. End to end delay is high in network with attacks than network without attack because AODV chooses shortest path under normal scenario so end to end delay is less under normal scenario. The end to end delay with respect to number of nodes is not changed. The end to end delay is affected by position of all the nodes in network and mobility of nodes in network. Sometimes attacker node is near to source node or receiver node but some times it will be further because of randomness of mobility [14] [15] [16].

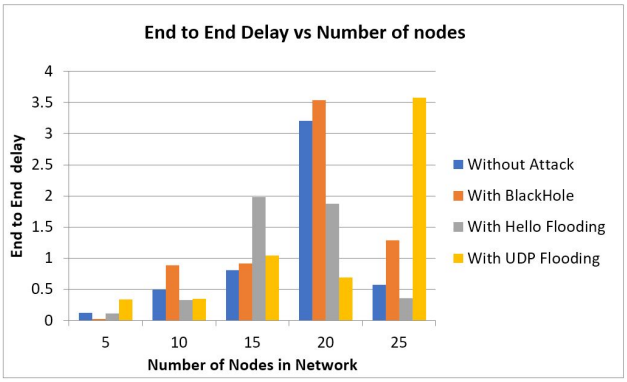

Fig. 18. End to End delay vs Number of nodes.

Figure 19 shows the packet delivery ratio with respect to increase in number of nodes in network. The packet delivery ratio under normal condition without attack in network is greater than network when blackhole attack or Hello Flooding attack is performed. Because whichever packets the attacker node receives are dropped by it. Apparently, UDP Flooding attack is generating UDP packets and flooding the network with it, so the packet delivery ratio of UDP Flooding attack is greater than normal scenario of network. The Packet delivery ratio is also highly affected by mobility and random position of nodes rather than increase in number of nodes in network [14] [15] [16].

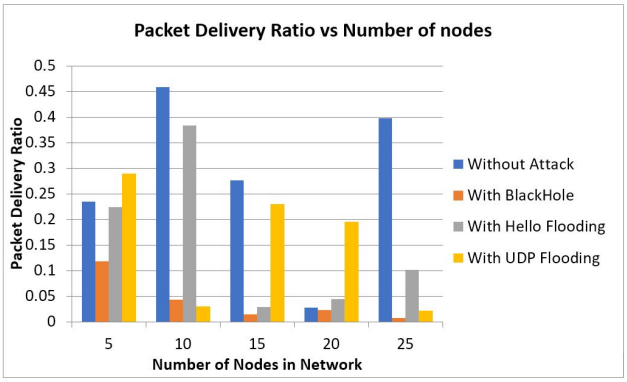

Fig. 19. Packet delivery ratio vs Number of nodes.

6 Intrusion Detection using AI

Artificial Neural Network (ANN) is trained to learn the behavior of the attacker in different kinds of attacks. In ANN, we have three layers: input, hidden, and output layers. Many sigmoid units or logistic units are connected in a neural network to produce an output, so it is also called Multi-Layer Perceptron. The neural network determines the probability that a node is an attacker or normal for the given input. The probability of output is predicted by logistic regression. The sigmoid unit based on the activation function applied will determine the class of the target output. There are two classes in target outcome, which are Attacker node or Normal node.

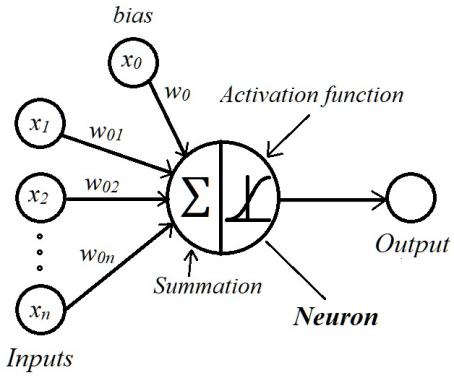

Fig. 20. One Sigmoid Unit.

One Sigmoid unit contains one neuron as shown in fig. 20. Each neuron has summation unit and activation function. The multiple inputs are connected with one neuron. Each connection has its corresponding weight.

$$net_j = \sum_{i=0}^{m} x_i w_{ij}$$

where i is the number of neurons in the current layer, j is the number of nodes in the next layer, and net_j represents the summation unit. This summation will be passed to an activation layer which can be sigmoid, ReLU, softmax, etc. The weights for each connections will be updated based on the back propagation method [17].

We are training our dataset using the Supervised Learning method . It is a kind of offline training where the output class label is associated with each input sample [17]. In our case, if the node is an attacker or not is specified as the target outcome.

6.1 Why use ANN to detect malicious nodes?

The different parameters decide whether a node is an intruder or not in different type of attack. Like in a Blackhole attack, the intruder drops all the packets it receives. So, parameter like the number of packets forwarded by all nodes is used to detect the attacker. Whereas in Hello Flooding, transmitting Power is increased, so that every time the path is chosen where attacker node is intermediate node. After that, attacker node drops all the packets it receives. So, parameters like packet forwarded by each node, energy consumption by each node, the transmitter power are used to detect the attacker. Therefore, different parameters like packet sent, packet received and packet forwarded, residual energy capacity, forwarding difference are used to determine the outcome. The forwarding difference is the number

of packets the relay node received minus the number of packets it transmitted. Based on this, we will get to know which node is dropping the packets in between. The end-to-end delay, throughput, and packet delivery ratio will be used to analyze the network activities.

So, to learn the behavior or pattern of the attacker, Artificial Neural Network is used. It learns the pattern from these features and determines the outcome of whether the node is normal node or attacker node.

6.2 Network design

There are 11 features of each node in network from which we can analyze the behaviour of network. The network design is shown in fig. 21. We have 11 neurons in the input layer, based on which we determine whether the node is an attacker node or normal node in a network. These features are the type of Mobility models, Mobility speed, number of packets received, number of packets sent, Residual Energy Capacity, number of forwarded packets sent (mac), number of forwarded packets received (mac), forwarded packets difference, number of packets dropped, Transmission power, and type of attack.

There are four hidden layers, where first hidden layer has 32 neurons and the second layer has 64 neurons. Then we have a Dropout layer with value 0.2 for Generalization. The next layer has 32 neurons, and after that, the next layer has eight neurons.

The **activation function** used in hidden layers is ReLU because it is suitable for overcoming problems like gradient vanishing or exploding values. The output layer contains one neuron, which tells whether, according to the input provided, the node is an attacker node or not. The activation function used in the output layer is sigmoid because our problem needs binary classification.

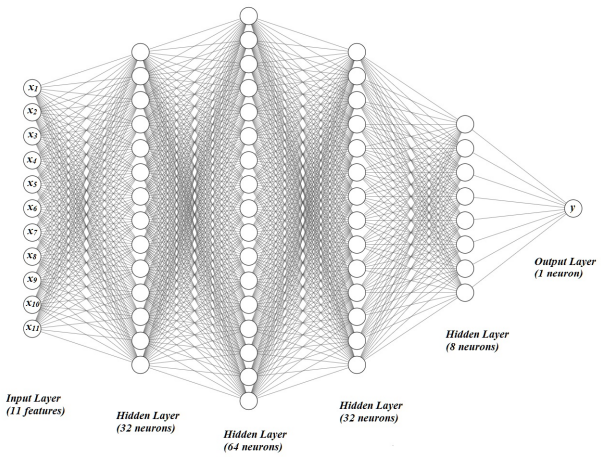

Fig. 21. Structure of Artificial Neural Network

The value of hyper parameters for our ANN are :

- **Loss Function: Binary Cross-Entropy**
  The measurement of uncertainty in the classification results associated with a given distribution is called entropy. The value of entropy is 0 if all the nodes are classified correctly. While training, all the data points are considered to calculate the binary cross-entropy loss, effectively fitting the distribution.
- **Optimizer: Adam**
  In this neural network, we are using Adam optimizer. Adam means Adaptive Moment Estimation. It provides an algorithm for gradient descent optimization. It is efficient with a large dataset and needs less memory.
- **Metrics: MSE and Accuracy**
  In this neural network, we have calculated Mean Square Error (MSE) because we are implying logistic regression on output layer. Accuracy tells how many nodes neural network detected correctly.

6.2.1 *Problems with neural network*

- **Overfitting**
  We implemented the Early Stopping Method for overcoming the overfitting problem. In Early Stopping, we constantly check the validation set loss value. After every epoch, if the loss is decreased, we train for more number of epochs, but once the validation set loss value increases, we should stop the training at that point; otherwise, the model will overfit [18].
- **Dropout layer for Generalization**
  In the dropout layer, we specified value 0.2. It means that 20% of neurons values will be zero. It helps the model not to memorize the pattern or the attacker node but to learn the pattern in the model.

6.2.2 *Results of Neural Network*

To understand the feature behaviour of each node, we need to analyze the correlation between them. The correlation between the input features is presented by heatmap in fig. 22.

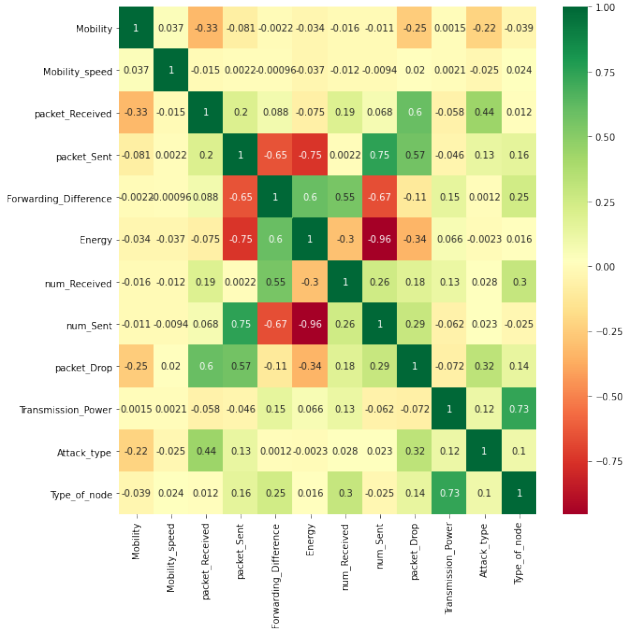

Fig. 22. Feature Correlation

From fig. 22, we can say that residual energy has a strong negative correlation with the number of packets forwarded. It means if a greater number of packets are sent, more energy

will be drained for that node. The packet dropped is correlated to the packet sent and packets received. Forwarding difference is negatively correlated to the number of packets sent or the number of forwarded packets transmitted and positively correlated with the number of forwarding packets received. The transmission power is highly correlated with the type of node, which can be an attacker node or a normal node.

The number of datapoints collected from Omnet++ simulations for each attack is specified in table 4.

TABLE 4
Number of datapoints collected from Omnet++

Normal Scenario	Blackhole Attack	Hello Flooding Attack	UDP Flooding Attack
375	700	1000	200

The training set has total 2047 data points, and the test dataset has 228 data points. The validation set is 15% from the training dataset. The accuracy, binary cross entropy loss value, and mean square error for training and validation set is shown in table 5:

TABLE 5
Accuracy and loss values for Training and Validation set

	Training Set	Validation Set
Accuracy	98%	99%
Binary Cross Entropy Loss	0.0540	0.0351
Mean Square Error	0.0148	0.0095

The test dataset accuracy is 97%. The fig. 23 shows that, we are training our dataset for 72 epochs because after that the validation loss value starts increasing and Early Stopping parameter stops the training [18].

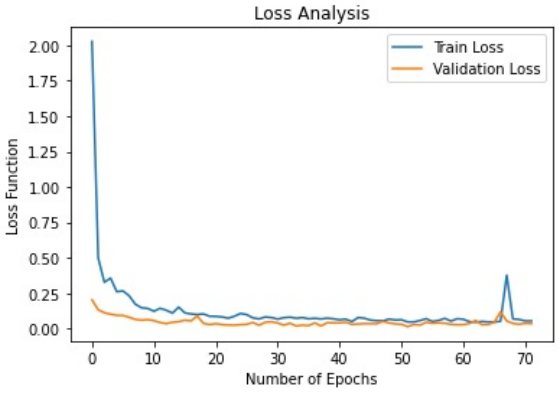

Fig. 23. Loss Analysis

The Confusion Matrix determines how many datapoints are classified correctly and incorrectly. The confusion matrix have mainly 4 values: TP for True Positive, TN for True Negative, FN for False Negative, and FP for False Positive. These values are determined by comparing actual values with predicted values. The Confusion Metrix generated for Testing data is shown in table 6.

TABLE 6
Confusion Matrix

		Predicted Values	
		Normal	Attacker
Actual values	Normal	TP = 210	FN = 4
	Attacker	FP = 1	TN = 13

7 Discussion and Conclusion

In this paper, different attacks in MANET are simulated, and the impact of an attack is analyzed in terms of PDR, end-to-end delay, energy consumption, and throughput under different mobility models and speed cases. In most cases, it is observed that the attacker nodes are the ones with high power consumption and low PDR. A combination of different features is required to accurately identify a malicious node in MANET, as the random movement of nodes poses additional uncertainty. The study performed here can be used to enhance the performance of the existing protocols. In addition to that, this paper also presents an offline intrusion detection system that uses an artificial neural network (ANN) to identify the attacker node in the MANET. The classification is done based on normal and threat patterns, and the technique achieved a 97.80% accuracy. The IDS using ANN can detect attacker nodes successfully, and the result is validated using data obtained from the OMNET++ simulation. In the future, more attacks can be introduced to test the reliability of our method. Furthermore, recurrent neural network and convolutional neural network approaches can be investigated for IDS.

REFERENCES

[1] L. Raja and S. S. Baboo, "An overview of manet: Applications attacks and challenges," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 1, pp. 408–417, 2014.

[2] P. Vinayakray-Jani, "Security within ad hoc networks," in *Position Paper, PAMPAS Workshop*. Citeseer, 2002.

[3] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad hoc networks*, vol. 1, no. 1, pp. 13–64, 2003.

[4] S. Mirza and S. Z. Bakshi, "Introduction to manet," *International research journal of engineering and technology*, vol. 5, no. 1, pp. 17–20, 2018.

[5] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE personal communications*, vol. 6, no. 2, pp. 46–55, 1999.

[6] G. Jayakumar and G. Gopinath, "Ad hoc mobile wireless networks routing protocols–a review," *Journal of Computer science*, vol. 3, no. 8, pp. 574–582, 2007.

[7] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 275–283.

[8] M. M. Alani, "Manet security: A survey," in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*. IEEE, 2014, pp. 559–564.

[9] P. K. Maurya, G. Sharma, V. Sahu, A. Roberts, M. Srivastava, M. Scholar *et al.*, "An overview of aodv routing protocol," *International Journal of Modern Engineering Research (IJMER)*, vol. 2, no. 3, pp. 728–732, 2012.

[10] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, 1999, pp. 90–100.

[11] Inet.omnetpp.org. (2021) Node mobility — inet 4.3.0 documentation. [Online]. Available: https://inet.omnetpp.org/docs/users-guide/ch-mobility.html.

[12] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[13] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[14] S. Alanazi, K. Saleem, J. Al-Muhtadi, and A. Derhab, "Analysis of denial of service impact on data routing in mobile ehealth wireless mesh network," *Mobile Information Systems*, vol. 2016, 2016.

[15] H. Shanthi and E. Anita, "Performance analysis of black hole attacks in geographical routing manet," 2014.

[16] A. Yasin and M. Abu-Zant, "Detecting and isolating black-hole attacks in manet using timer based baited technique," *Wirel. Commun. Mob. Comput.*, vol. 2018, pp. 9 812 135:1–9 812 135:10, 2018.

[17] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 1–6.

[18] M. Moradi and M. ZULKERNINE, "A neural network based system for intrusion detection and classification of attacks," 02 2014.