International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.7380
***Article Received**: 29 March 2023 **Revised**: 25 May 2023 **Accepted**: 18 June 2023*

Analysis of Behavioral Characteristics of Multiple Blackhole Attacks with TCP and UDP Connections in Mobile ADHOC Networks based on Machine Learning Algorithms

Mrs. Kagita Nandini[1], Sasidhar Thammana[2], Vivek Pitta[3], Segu Prathyusha[4], Varikuti Jaswanth Ram[5],
[1]Assistant Professor, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College,
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, India
kagita.nandini@gmail.com
[2]Student, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, India
sasidhar5901@gmail.com
[3]Student, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, India
vivekpitta1010@gmail.com
[4]Student, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, India
seguprathyu123@gmail.com
[5]Student, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, India
jashujaswanth267@gmail.com

Abstract—In Mobile Adhoc Networks (MANET's), a suit of nodes which are under mobility work together to transmit data packets in a multiple-hop manner without relying on any fixed or centralized infrastructure. A significant obstacle in managing these networks is identifying malicious nodes, or "black holes". To detect black holes, we proposed a method involves broadcasting a Cseq to the neighboring nodes and awaiting the node's response is utilized. This Network is simulated with 25 number of nodes connected with TCP connection and observed the different behavioural characteristics of nodes. Then the connections are changed to UDP and observed the characteristics. Then characteristics are analyzed with different machine learning algorithms. The network is simulated in NS2 environment.

Keywords-MANET's, Blackhole, TCP, UDP, Simulation, Analysis.

I. INTRODUCTION

A. *Mobile Adhoc Networks(MANET'S)*

MANET's contains a decentralized network which is formed by wireless links connecting the devices which is having mobility without the need for central administration. Each device in a MANETs functions as both an agent and a router (AGT and RTR) for forwarding packets for other devices. MANETs are highly dynamic, with a rapidly changing topology caused by the agility of nodes and the unpredictable characteristic of wireless communication. This creates confronts for designing these networks, as traditional network protocols and structures may not be appropriate.

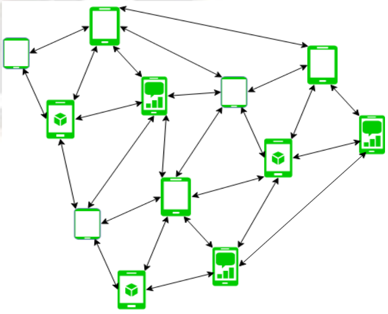


Figure 1 - Mobile Adhoc Networks Representation

Some of the key features of Mobile Adhoc Networks are

a. *Dynamic Behaviour:* Each node has independent behaviour since it may function as a agent and a router.

b. *Low Security:* Security risks are more likely to target wireless networks. Due to the dispersed nature of the function for secure mechanisms, routing, and hosting setup, there is no centralised firewall.

c. *Dynamic Topology*: Network topologies with dynamic topologies, which are often multihop, may change quickly and arbitrarily over time and can provide unidirectional or bidirectional linkages.

d. *Energy Constraints:* Operating under an energy constraint means that some of the nodes relies on batteries or other continuous energy sources. Less memory, power, and lightweight qualities define mobile nodes.

e. *Limited Resources:* The dynamic topology and limited supplies of the devices, such as battery life, power of processing, and bandwidth, are also important considerations. MANETs have numerous practical applications, including emergency response and military operations where traditional communication infrastructure is unavailable, as well as civilian uses like sensor networks, personal area networks, and vehicular networks.

B. Routing Process in MANET'S

In MANET's, routing is a crucial task that enables nodes to discover the most efficient path to forward data packets to their intended destination. Unlike conventional wired networks, MANETs are decentralized networks with no fixed infrastructure or centralized control. Consequently, routing in MANETs is more intricate in traditional networks. Several protocols have been developed to enable efficient communication in MANET's by giving the best route. The AODV protocol is utilized in MANET's to establish and have the routes in an environment dynamically where nodes can move and the topology of the network can frequently change. Unlike traditional routing protocols, AODV sets up the routes only when it is required instead of constantly maintaining a routing table for each individual point within the interconnected system.

To broadcast out a packet to the DN(Destination Node) without a route, a node broadcasts out a RREQ message to its neighbour containing information about SN(Source Node) and DN and a unique identifier for the request. Neighbouring nodes can check their table which contains routing information and forward the RREQ until it reaches either the DN or IN. The DN responds with a RREP message containing information about the most suited shortest path to the DN, including the no.of hops required.

C. Blackhole Attack

The blackhole attack is a security threat that can affect Mobile Adhoc Networks (MANETs). In these type of attacks, malicious node lures packets from the source by deceptively pretending to have the quick route to the target node, but then discards every packet that it receives. The blackhole node can entice traffic by telling itself as it is having the abridged path to DN, and the source nodes may choose it as their next hop. Once the packets are sent through the blackhole node, they are dropped, and the packets are lost.

The blackhole will be identified when SN sends RREQ packets to furnish packets to DN. The blackhole node, with a higher Sno and fewer count of hops, sends RREP to the SN immediately, and the SN starts broadcasting data packets to the attacker. However, the malicious one secretly discards or intercepts the incoming packets without informing the SN that the packets have not reached their intended DN.

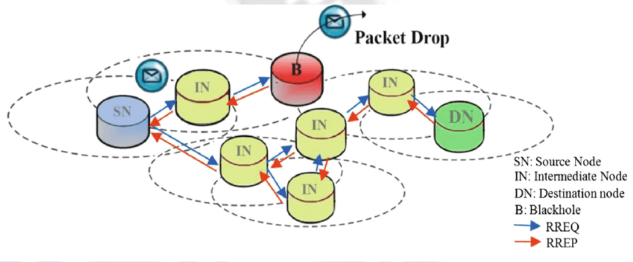


Figure 2 - Blackhole Attack in MANETS

In the Figure-2, we can see that SN means Source Node of the Network, DN means Destination Node of the Network, IN means Intermediate Node of the Network, where SN sends data packets to DN through IN. Here comes the malicious Node B which is Blackhole pretends to have the shortest path from SN to DN and drops all incoming packets.

II. LITERATURE SURVEY

S. Dhende et. al [1] recommended a solution in which the neighbour node which receives the message of opinion which is sent by the SN sends an acknowledgement with a YES packet (YP) through a different path to the SN and a 'NO' packet via the replied node. The SN then waits for time 't' so that it may collect all of the neighbour's node's YES and NO packets. If the SN doesn't get a packet, it notifies its neighbours' IDs in the opinion table (OT) that it doesn't have the optimal path.

N. Sharma et. al [2] has an idea that the Route finding phase and the Monitoring phase are used to find the blackhole assault. This approach employs trap RREQ to catch the blackhole and also uses trusted mode to monitor blackhole behavior of any node. In the Route Discovery phase, Malicious node always replies RREQ supplied by SN and doesn't verify the table which contains routing information. Here, the sender first sends a trap RREQ that has a destination address that isn't actually present in the network. As a result, when an attacker node receives the RREQ, it doesn't check its own table for routing information before sending a reply to the SN. Finally, after receiving the false RREQ response, the SN records the origin of the fake RREP and adds it to the list which contains malicious nodes information.

N. Arya et. al [3] uses a scheme in detecting the attacking behaviour of nodes. Each node in this approach has a Trust table. The status of trust of any node with the neighbors is stored in a trust table. There are 2 columns in the Trust table. The neighboring node's name or identification, followed by its relationship status—which might be Most Reliable, Reliable, or Unreliable—with the neighboring node. Each time a node gets a packet, this table is consulted. When a node first enters a network, they are first regarded as unreliable.

R. Sharma, U. Singh et. al [4] and states that, For the purpose of identification and forestalling of blackhole attacks in MANET's, a cluster direction in the existing AODV protocol has been proposed. To determine the unique difference in the quantity of packets which are received and delivered by a particular node, every node of the unit will acknowledge the cluster head once. All nodes hides the infectious nodes from the particular network if the issue is detected.

V. Trivedi and V. Preethi [5] have made an analysis that Black hole attacks have a significant influence on how well wireless ad hoc network's function. Throughput, Packet delivery ratio (PDR), delay, normalized control overhead, jitter, packet drop, reachability, hop count, and neighbor's node density are the variables that influence network performance. Several performance indicators may suffer as a result assault result of blackhole.

By computing the value of at each node using only the control packets, *J. Thakker*, [6] suggested a method to avoid coordinated attacks, also known as cooperative black hole attacks, which reduce routing overhead.

V. G. Mohite and L. Ragha [7] uses some Blackhole Detection component to gather and examine network node IDs. The system quickly eliminates the particular Node ID(NID) with the check aid of HopAuthentication & DetectingFlow if the type of NID corresponds with the one mentioned in the block table. As a result, it would speed up system performance and decrease the amount of time needed for blackhole detection about this NID.

An approach proposed by *S. Shrestha et. al* [8] involves detecting malicious attacker nodes by considering sequence numbers (Sno) obtained from intermediate next hop nodes. If a SN receives an RREP with a suspicious Sno, it will rebroadcast the RREQ message with a changed Sno equal to the Sno of RREP packet which is sensed from the attacker node. If the responding node is indeed a blackhole node and drops packets from the source, it will broadcast an RREP packet with a higher Sno. Getting RREP packets with even larger sequences after retransmission confirms the identity of the suspect node, and the routes which follows that node will be discarded.

Vimal Kumar et. al [9] suggested a method for detecting blackhole attacks that makes use of the table of route reply which comes this way (CRRT). This table records the DN Sno and the ID of the answered node, is kept by the SN. This data is employed in the process of detection of blackhole attacks in MANETs. The PDR and throughput of the network are improved by the simulation result of the suggested approach.

To get rid of these attacks in MANET's, *Mohammad Al-Shurman, et. al* [11] Park put up two options. The first approach uses shared hop to send packets, and the second one uses the sequence numbers of the most recent packets sent and received to identify problematic nodes. The simulation's findings demonstrated that suggested solutions reduce routing overhead.

Bansal,M et.al [15]evaluated the efficiency of an ad hoc network under a blackhole attack by the simulation process with NS2. They compared the performance measure of the AODV protocol in the presence and absence of a blackhole in a MANET. According to the result, they found that in the absence of a blackhole, there was a less amount of packet loss and an increase in the throughput, as well as a decrease in E-to-E delay.

Suryawamshi et.al [16]assessed and appraised the outcomes of these attacks. They simulated the attack in NS2 tool and observed that a blackhole had negative impact on the total working efficiency of the network, including its connecting nature and data loss. To mitigate the effects of the attack, the authors developed an IDSAODV protocol, which improved the PDR and reduced the loss of packets.

Ming-Yang et.al [17] have introduced an IDS Working, which asses the suspicion level of a node by analyzing the anomalous difference between RREQ's & RREP's transmitted by the node. All nodes within the network implement this IDS, and IN's are not permitted to reply to the RREQs sent by them. If an IN does not broadcast RREQs for a particular route, but sends an RREPLY for that route and is not the DN, its abnormal value is raised in the abnormal node table of neighboring nodes. When the suspicion level of node crosses a set threshold, the node sends "Stop" message to all other IN's in the network, thus collectively sequestering the suspected node.

***Article Received**: 29 March 2023 **Revised**: 25 May 2023 **Accepted**: 18 June 2023*

Gonzalez et. al [18] presented an algorithm which utilizes flow conservation to identify nodes that exhibit persistent mis-behaviour. By setting an appropriate threshold for misbehaviour, the algorithm is capable of distinguishing between good nodes and malicious nodes. However, the algorithm necessitates a definite amount of time to get the necessary data to observe and incriminate malicious behavioural nodes, which results in the average network throughput being less than that of a network with no misbehaving nodes. Consequently, during the initial phase, misbehaving nodes may discard packets which are inspected prior to being blamed and disconnected from the network.

V. Trivedi et. al [5]have presented a strategy for identifying coordinated attacks in addition to a technique for protecting packet delivery history data in [5]. so that the other IN's can recognise internal assaults like grey hole attacks. information at each interaction. by examining these documents. For preventing a cooperative blackhole attack, academics have also suggested using cooperative security agents. By analysing the patterns of data packets, all mechanisms can prevent or detect black hole attacks. We provide a method for the prevention of cooperative blackhole attack that solely makes use of management packets

Adilakshmi Yannam et al. have created a method to recognise cooperative blackhole assaults in MANET's [21]. The study only took into account blackhole nodes that do not cooperate with one another when it comes to multiple blackhole assaults on MANETs. The lack of performance evaluation and scenario simulation is the paper's main drawback.

III. PROPOSED METHODOLOGY

MANETs are highly vulnerable to blackhole attacks which can result in a significant deterioration of the network's performance, integrity, and even render it inoperable. When a substantial no. of nodes taken in the network are compromised and begin to act as blackholes, the entire network may become inaccessible, leaving legitimate nodes unable to communicate with each other. In addition to that, the performance, throughput, and life of the Network will also be decreased by the impact of Blackholes in Network.

In order to enhance the efficiency and working efficiency and throughput of a MANET, it is crucial to identify and detect blackholes within the network. Therefore, a proposed technique utilizing machine learning algorithms has been suggested for blackhole detection in Mobile Adhoc Networks, which can improve both the network's performance and security.

The properties of the Blackholes with different connections are analysed using different Machine Learning Algorithms. The analysis is made based on the TCP and UDP connections for the Blackholes. The analysis is made by considering the following features.

i. *Packets sent:* The no. of packets sent by the SN to DN.
ii. *Packets received:* The no. of packets received by the DN from SN.
iii. *Packets forwarded:* The no. of packets forwarded to the intended DN by the IN.
iv. *Packets dropped:* The no. of packets which are dropped without being forwarded to the DN.
v. *Route Requests:* The no. of route requests sent by the node.
vi. *Route Reply's:* The no. of route reply's given by the node.

The Analysis is made for two connections as described below.

A. *Transmission Control Protocol(TCP) Connection*

The algorithms which are compared with the TCP connected dataset are:

a. *Decision Tree*

This Algorithm is referred as a tree structured classifierr used to solve classification tasks. It operates by repeatedly dividing the dataset into subsets based on a feature's values, until a subset is reached that only contains instances of one class. The resulting model will be present in a tree structure, with internal nodes represents decisions based on a feature, branches representing the outcome of the decision, and leaf nodes represents a class label. To make a prediction, the algorithm follows the decision rules at each internal node until it reaches a leaf node.

Decision trees are easy to interpret and explain since the decision rules are clear and can be visualized as a tree. They are also adaptable and can handle both categorical and continuous input features, and are useful for both binary and multiclass classification problems. However, decision trees may overfit, particularly when the tree is too deep or the dataset is noisy. Techniques like pruning the tree, setting a small no. of samples to split the internal node, or using ensemble methods like random forests can be utilized to avoid overfitting.

- IG = Entropy(S)- [(Average) *Entropy (Every Feature)
- Entropy(s) = -p(true)log2 P(true)- P(false) log2 P(false)
- Gini Index= 1- $\sum_j P_j^2$

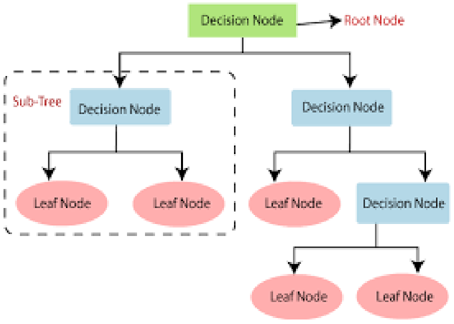


Figure 3 - Decision Tree

b. Support Vector Classifier (SVM)

The Support Vector Classifier (SVC) which is used for classification tasks, which works by finding a hyperplane of particular data points in an n-dimensional space which separates the classes maximally. It is a un-probabilistic binary linear classifier that aims to maximize the margin, which can be taken as the distance b/w the hyperplane and the nearest data points from every class. The support vectors, which are the points next-door to the hyperplane, determine the hyperplane's location and orientation. SVC is useful when dealing with a great number of features compared to the no. of instances or non-linearly separable data. In such cases, a kernel function is used to map input to an n-dimensional space, where a linear hyperplane can be found to separate the classes.

SVC can handle complex decision boundaries and is robust to outliers, making it suitable for both binary and multiclass classification problems. However, it can be computationally expensive to train and tune, particularly with large datasets. To prevent overfitting, regularization parameters can be included in the objective function to minimize the margin and penalize the misclassification of data points. These parameters can be optimized through cross-validation to improve the classifier's performance. The selection of the parameters of kernel function and the function itself can also have a significant impact on the classifier's accuracy.

SVC aims to find a hyperplane which separates data points of the different classes maximally. The hyperplane equation is given below.

$$g(x) = w^T x + b$$

$Value\ of\ g(x)\ depends\ upon\ ||w||$:

1) $Keep\ ||w|| = 1\ and\ maximize\ g(x)\ or,$

2) $g(x) \geq 1\ and\ minimize\ ||w||$

- $W^T * x + b = 0$

The optimization problem can be given with the formula as given below:

- minimize: (1/2) * $|w|^2$

Here, |w| is the Euclidean term of the weight vector "w", and the constraint ensures that each data point is on the correct side of the hyperplane.

- f(x) = sign(w^T * x + b)

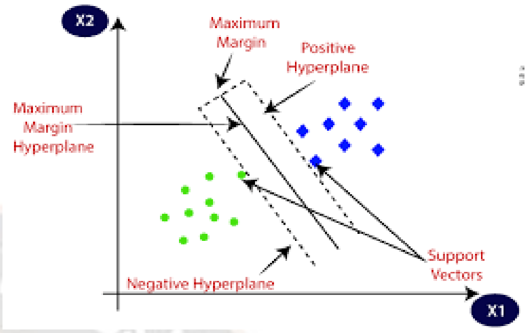


Figure 4 - SVM

c. K-Means Classifier

K-means Classification by clustering is used to group data points into k no. of clusters. The algorithm will start by initializing k no. of cluster centroids, which serve as the center of every cluster. The data points are then assigned to the nearest centroid based on their Euclidean distance, creating an initial partition of the data. The centroids are updated by calculating the mean of data points in every cluster, and the assignment of data points to clusters is updated based on the new centroids. This process is repeated until the assignments no longer change or maximum iterations are reached.

The quality of the clustering is measured using a cost function that calculates the sum of the squared distances between every point and its assigned a centroid. The main aim is to minimize the function of , which results in compact and well-separated clusters. This is computationally proficient algorithm that is commonly used in various applications such as image segmentation, market segmentation, and customer profiling. However, one limitation is that the number of clusters k must be predetermined, which can be difficult in practice. Moreover, k-means clustering may not work well with data that has irregular shapes or varying densities since it assumes that the clusters are spherical and of equal size.

- Assignment: argmin_j ||xi - cj||^2
- Update: cj = (1 / Nj) * sum(xi), where xi will be assigning to the jth cluster and Nj is the no. of data points assigned to that clust

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.7380
***Article Received**: 29 March 2023 **Revised**: 25 May 2023 **Accepted**: 18 June 2023*

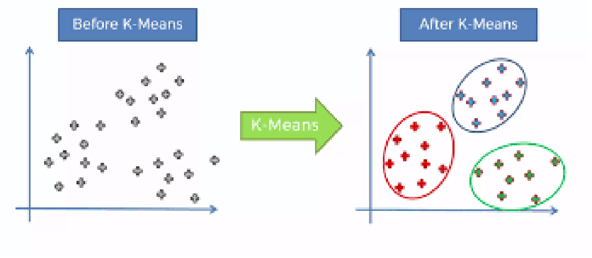


Figure 5- K-Means

B. *User Datagram Protocol(UDP) Connection*

The algorithms which are compared with the UDP connected dataset are:

a. *Logistic Regression*

Logistic regression is a method which is a statistical way used to predict the probability of a binary outcome by using one or more input variables. Unlike linear regression, which is used to predict a continuous outcome, this algorithm models the relation between a dependent variable (the binary outcome) and independent variables (input variables) by applying the logistic function to the linear combination of the input variables.

The output of the logistic function is the probability of the binary outcome for a given set of input variables. To find the coefficients of the input variables that best fit the data, logistic regression uses a maximum likelihood estimation method. Logistic regression is widely used in many applications, such as predicting the probability of customer behavior, disease diagnosis, and loan approval based on credit score. It is a powerful and popular statistical technique in both machine learning and statistical analysis.

- *Sigmoid Function:*

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x}+1}$$

$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_r x_r$
Here, β_0, β_1, β_2, …, β_r are the coefficients of the predictor variables x_1, x_2, …, x_r.

- *Logistic Regression Equation:*

$P(y=1|x) = \sigma(z)$
$P(y=0|x) = 1 - \sigma(z)$

- *Cost Function:*

$J(\theta) = -(m)^{-1} * \Sigma(y\log(h(x)) + (1-y)\log(1-h(x)))$

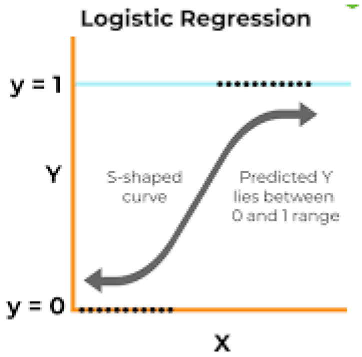


Figure 6 - Logistic Regression

b. *Decision Tree Classifier*

c. *Random Forest Classifier*

Random Forest is used for different tasks such as regression, classification, and more. It creates multiple decision trees on multiple sets of the training data, where every tree will be trained on a random subset of the input features. This process is known as "bagging". To make predictions, every tree is used to predict the outcome, and the complete prediction is determined by the aggregation of predictions of all the trees in the forest, either by taking the mode for classification or the mean for regression. Random forest is a versatile algorithm that has applications in many fields, including finance, marketing, and healthcare. It is particularly useful for dealing with complex relationships between many input variables.

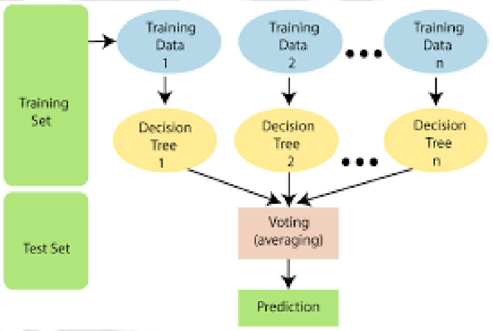
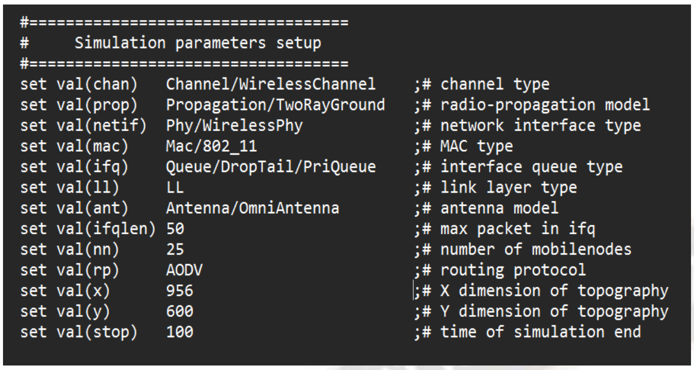


Figure 7 - Random Forest

These are the algorithms which are used for the classification of Blackhole node and good node. Accuracy scores are calculated and compared for the analysis of Blackhole attack over TCP and UDP connections.

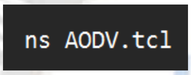
IV. RESULTS

We will simulate the blackhole attack in Mobile ADHOC Networks using NS2 Simulator. NS2 Simulator implements NAM(Network Animator) to show the animation of Blackhole Attack. In order to simulate the blackhole attack in NS2, we need one file with “.tcl” extension. We name that file as “AODV.tcl”. The information like configuration of Network, Number of Nodes, Location of Nodes, Connections between the nodes, Routing Protocol used, Simulation time etc., will be defined in “AODV.tcl” file.



```
#==================================
#     Simulation parameters setup
#==================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     25                         ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      956                        ;# X dimension of topography
set val(y)      600                        ;# Y dimension of topography
set val(stop)   100                        ;# time of simulation end
```

In this way, the information about our Network will be provided in “AODV.tcl” file .The file “AODV.tcl” will be executed in Ubuntu environment with the following syntax**.**

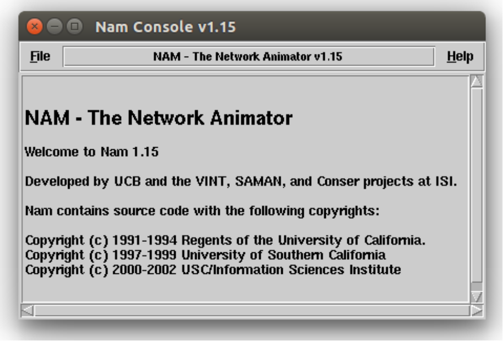


```
ns AODV.tcl
```

Network Animator (NAM):

The visualization tool NAM (Network Animator) is used in ns-2 (Network Simulator 2) to represent the behavior of network simulations in a graphical format. With NAM, the network topology and traffic flow between nodes can be viewed, which aids researchers in analyzing the network behavior and detecting performance bottlenecks.

The primary function of NAM is to provide a user-friendly way of displaying the output produced by the network simulation. NAM shows real-time visual feedback on the network performance, including packet transmission rates, packet drops, and network congestion. Additionally, NAM can help identify routing problems and other network issues by visualizing the routing paths taken by packets. For network simulation researchers, NAM is a valuable tool, as it allows for easy analysis and visualization of complex network simulations. Furthermore, NAM is highly customizable, allowing researchers to tailor the visualization to their specific research questions and needs.



NAM - The Network Animator

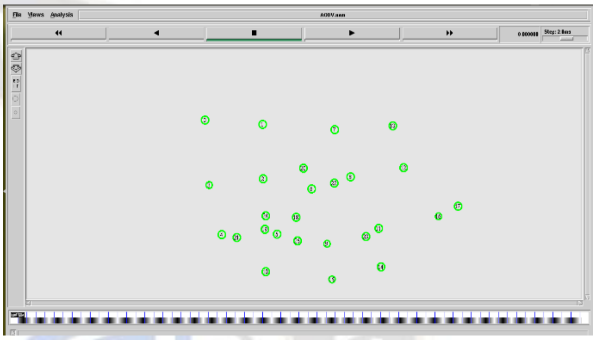
Welcome to Nam 1.15

Developed by UCB and the VINT, SAMAN, and Conser projects at ISI.

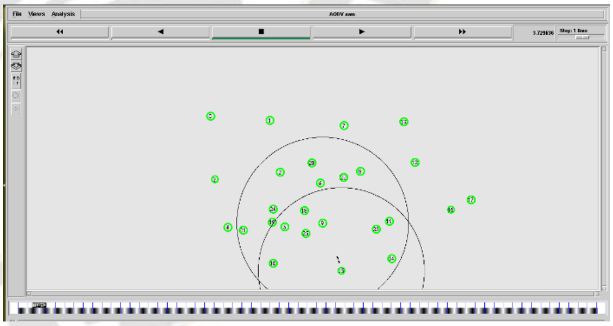
Nam contains source code with the following copyrights:

Copyright (c) 1991-1994 Regents of the University of California.
Copyright (c) 1997-1999 University of Southern California
Copyright (c) 2000-2002 USC/Information Sciences Institute

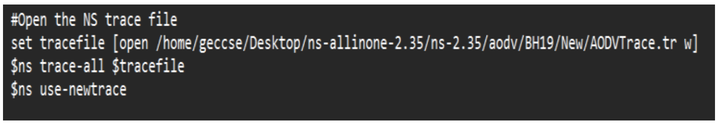
Figure 8 - NAM



Firstly, the Network Animator will be launched and the nodes will be visible and communication will occur between the nodes based on the connection established. The packets will be transmitted between the connected nodes and the remaining nodes will participate in communication by forwarding the incoming packets. The following figure shows the transmission of the packets between the nodes specified.

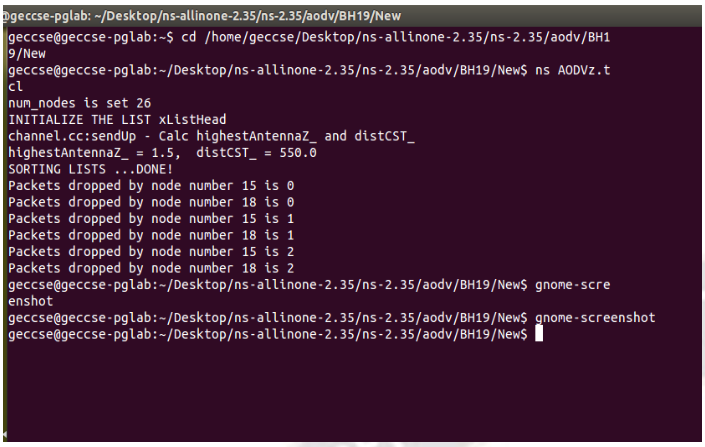


When the “AODV.tcl” file is executed, the corresponding trace file will be generated with “.tr” extension. The name of the trace file will also be mentioned in “AODV.tcl” file itself.



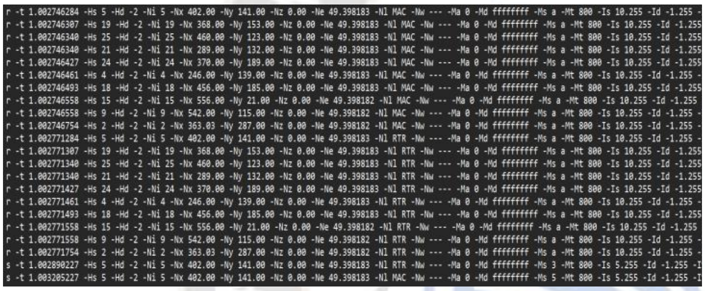
```
#Open the NS trace file
set tracefile [open /home/geccse/Desktop/ns-allinone-2.35/ns-2.35/aodv/BH19/New/AODVTrace.tr w]
$ns trace-all $tracefile
$ns use-newtrace
```

Here the trace file “AODVTrace.tr” will be generated in the mentioned folder after executing the “AODV.tcl” file. After running “AODV.tcl” file Network Animator (NAM) will be opened which consists of the mobile nodes based on the specification our Network Configuration. The number of packets dropped continuously will be generated in the terminal as shown below and then NAM will be opened.



```
@geccse-pglab: ~/Desktop/ns-allinone-2.35/ns-2.35/aodv/BH19/New
geccse@geccse-pglab:~$ cd /home/geccse/Desktop/ns-allinone-2.35/ns-2.35/aodv/BH1
9/New
geccse@geccse-pglab:~/Desktop/ns-allinone-2.35/ns-2.35/aodv/BH19/New$ ns AODVz.t
cl
num_nodes is set 26
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
Packets dropped by node number 15 is 0
Packets dropped by node number 18 is 0
Packets dropped by node number 15 is 1
Packets dropped by node number 18 is 1
Packets dropped by node number 15 is 2
Packets dropped by node number 18 is 2
geccse@geccse-pglab:~/Desktop/ns-allinone-2.35/ns-2.35/aodv/BH19/New$ gnome-scre
enshot
geccse@geccse-pglab:~/Desktop/ns-allinone-2.35/ns-2.35/aodv/BH19/New$ gnome-screenshot
geccse@geccse-pglab:~/Desktop/ns-allinone-2.35/ns-2.35/aodv/BH19/New$ █
```

The corresponding trace file which is generated will be present in the below format.



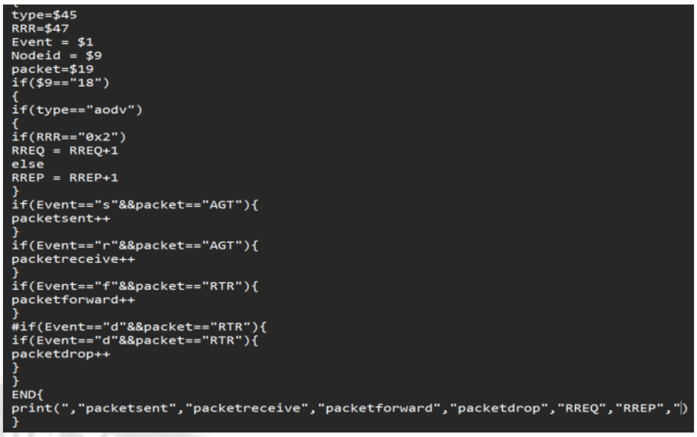
```
r -t 1.002746284 -Hs 5 -Hd -2 -Ni 5 -Nx 402.00 -Ny 141.00 -Nz 0.00 -Ne 49.398183 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255 -
r -t 1.002746307 -Hs 19 -Hd -2 -Ni 19 -Nx 368.00 -Ny 153.00 -Nz 0.00 -Ne 49.398183 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002746340 -Hs 25 -Hd -2 -Ni 25 -Nx 460.00 -Ny 123.00 -Nz 0.00 -Ne 49.398183 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002746340 -Hs 21 -Hd -2 -Ni 21 -Nx 289.00 -Ny 132.00 -Nz 0.00 -Ne 49.398183 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002746427 -Hs 24 -Hd -2 -Ni 24 -Nx 370.00 -Ny 189.00 -Nz 0.00 -Ne 49.398183 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002746461 -Hs 4 -Hd -2 -Ni 4 -Nx 246.00 -Ny 139.00 -Nz 0.00 -Ne 49.398183 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255 -
r -t 1.002746493 -Hs 18 -Hd -2 -Ni 18 -Nx 456.00 -Ny 185.00 -Nz 0.00 -Ne 49.398183 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002746558 -Hs 15 -Hd -2 -Ni 15 -Nx 556.00 -Ny 21.00 -Nz 0.00 -Ne 49.398182 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002746558 -Hs 9 -Hd -2 -Ni 9 -Nx 542.00 -Ny 115.00 -Nz 0.00 -Ne 49.398182 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255 -
r -t 1.002746754 -Hs 2 -Hd -2 -Ni 2 -Nx 363.03 -Ny 287.00 -Nz 0.00 -Ne 49.398182 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255 -
r -t 1.002771284 -Hs 5 -Hd -2 -Ni 5 -Nx 402.00 -Ny 141.00 -Nz 0.00 -Ne 49.398183 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255 -
r -t 1.002771307 -Hs 19 -Hd -2 -Ni 19 -Nx 368.00 -Ny 153.00 -Nz 0.00 -Ne 49.398183 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002771340 -Hs 25 -Hd -2 -Ni 25 -Nx 460.00 -Ny 123.00 -Nz 0.00 -Ne 49.398183 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002771340 -Hs 21 -Hd -2 -Ni 21 -Nx 289.00 -Ny 132.00 -Nz 0.00 -Ne 49.398183 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002771427 -Hs 24 -Hd -2 -Ni 24 -Nx 370.00 -Ny 189.00 -Nz 0.00 -Ne 49.398183 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002771461 -Hs 4 -Hd -2 -Ni 4 -Nx 246.00 -Ny 139.00 -Nz 0.00 -Ne 49.398183 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255 -
r -t 1.002771493 -Hs 18 -Hd -2 -Ni 18 -Nx 456.00 -Ny 185.00 -Nz 0.00 -Ne 49.398183 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002771558 -Hs 15 -Hd -2 -Ni 15 -Nx 556.00 -Ny 21.00 -Nz 0.00 -Ne 49.398182 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255
r -t 1.002771558 -Hs 9 -Hd -2 -Ni 9 -Nx 542.00 -Ny 115.00 -Nz 0.00 -Ne 49.398182 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255 -
r -t 1.002771754 -Hs 2 -Hd -2 -Ni 2 -Nx 363.03 -Ny 287.00 -Nz 0.00 -Ne 49.398182 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms a -Mt 800 -Is 10.255 -Id -1.255 -
s -t 1.002890227 -Hs 5 -Hd -2 -Ni 5 -Nx 402.00 -Ny 141.00 -Nz 0.00 -Ne 49.398183 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 3 -Mt 800 -Is 5.255 -Id -1.255 -I
s -t 1.003205227 -Hs 5 -Hd -2 -Ni 5 -Nx 402.00 -Ny 141.00 -Nz 0.00 -Ne 49.398183 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 5 -Mt 800 -Is 5.255 -Id -1.255 -I
```

As the trace file format is not understandable, we will apply some "awk scripts" to get the information which is present in the trace file.

Following are the fields which we have collected using "awk scripts"

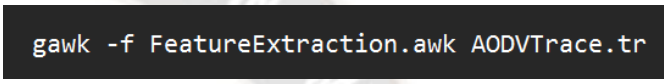
- Packets sent.
- Packets received.
- Packets forwarded.
- Packets dropped.
- Number of Route Requests.
- Number of Route Reply's.

The awk script which is used to get the information of above fields is "FeatureExtraction.awk" and the information will be retrieved like below.



```
type=$45
RRR=$47
Event = $1
Nodeid = $9
packet=$19
if($9=="18")
{
if(type=="aodv")
{
if(RRR=="0x2")
RREQ = RREQ+1
else
RREP = RREP+1
}
if(Event=="s"&&packet=="AGT"){
packetsent++
}
if(Event=="r"&&packet=="AGT"){
packetreceive++
}
if(Event=="f"&&packet=="RTR"){
packetforward++
}
#if(Event=="d"&&packet=="RTR"){
if(Event=="d"&&packet=="RTR"){
packetdrop++
}
}
END{
print(","packetsent","packetreceive","packetforward","packetdrop","RREQ","RREP",")
}
```

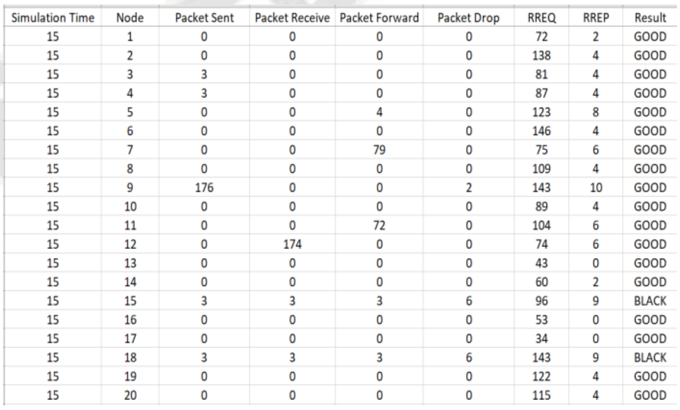
This "awk Script" will be applied to trace file with the following syntax.



```
gawk -f FeatureExtraction.awk AODVTrace.tr
```

By applying "FeatureExtraction.awk" for all the nodes i.e 25 at different simulation times from 5 to 200 with the break of 5 i.e, 5, 10, 15, 20, 25, …, 200, i.e a total of around 40 different times which comes to a total of around (40*25)1000 entries in the dataset, the information like packets sent, packets received, packets forwarded, packets dropped, number of route requests, number of route reply's are collected into a dataset.

Different connections have been established between the nodes which are malicious. Firstly, the nodes are connected over TCP (Transmission Control Protocol) connection and "FeatureExtraction.awk" is applied on all the nodes at specified simulation times and dataset is collected. And then, the nodes are connected over UDP (User Datagram Protocol) connection and "FeatureExtraction.awk" is applied on all the nodes at specified simulation times and dataset is collected. Datasets are shown below at different connections.

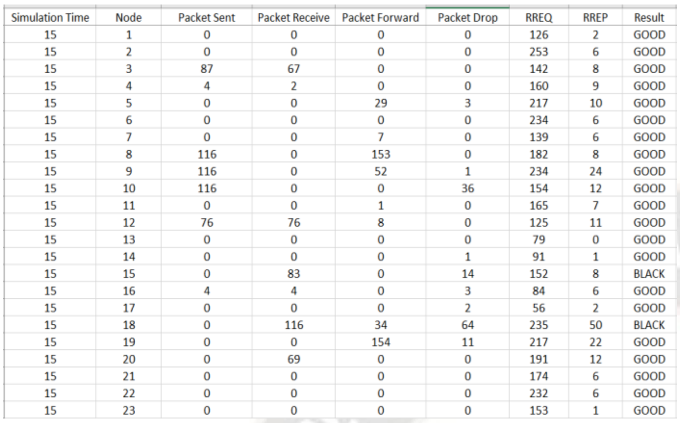


Simulation Time	Node	Packet Sent	Packet Receive	Packet Forward	Packet Drop	RREQ	RREP	Result
15	1	0	0	0	0	72	2	GOOD
15	2	0	0	0	0	138	4	GOOD
15	3	3	0	0	0	81	4	GOOD
15	4	3	0	0	0	87	4	GOOD
15	5	0	0	4	0	123	8	GOOD
15	6	0	0	0	0	146	4	GOOD
15	7	0	0	79	0	75	6	GOOD
15	8	0	0	0	0	109	4	GOOD
15	9	176	0	0	2	143	10	GOOD
15	10	0	0	0	0	89	4	GOOD
15	11	0	0	72	0	104	6	GOOD
15	12	0	174	0	0	74	6	GOOD
15	13	0	0	0	0	43	0	GOOD
15	14	0	0	0	0	60	2	GOOD
15	15	3	3	3	6	96	9	BLACK
15	16	0	0	0	0	53	0	GOOD
15	17	0	0	0	0	34	0	GOOD
15	18	3	3	3	6	143	9	BLACK
15	19	0	0	0	0	122	4	GOOD
15	20	0	0	0	0	115	4	GOOD

The above dataset shows the information about packets sent, packets received, packets forwarded, packets dropped, route requests, route reply's at different nodes at simulation time 15 when the malicious nodes are connected over TCP connection.

***Article Received**: 29 March 2023 **Revised**: 25 May 2023 **Accepted**: 18 June 2023*

In the above condition, Node-4 and Node-14 are communicating over TCP connection. Node-9 and Node-18 are communicating over TCP connection. Node-3 and Node-12 are communicating over UDP connection. i.e, the malicious nodes Node-15 and Node-18 are under TCP connection.

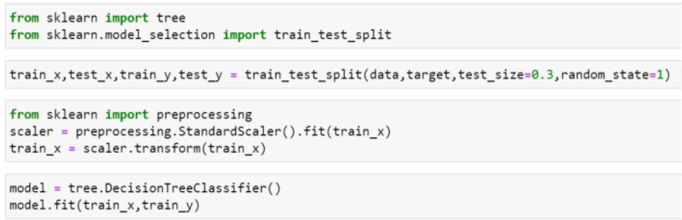


Simulation Time	Node	Packet Sent	Packet Receive	Packet Forward	Packet Drop	RREQ	RREP	Result
15	1	0	0	0	0	126	2	GOOD
15	2	0	0	0	0	253	6	GOOD
15	3	87	67	0	0	142	8	GOOD
15	4	4	2	0	0	160	9	GOOD
15	5	0	0	29	3	217	10	GOOD
15	6	0	0	0	0	234	6	GOOD
15	7	0	0	7	0	139	6	GOOD
15	8	116	0	153	0	182	8	GOOD
15	9	116	0	52	1	234	24	GOOD
15	10	116	0	0	36	154	12	GOOD
15	11	0	0	1	0	165	7	GOOD
15	12	76	76	8	0	125	11	GOOD
15	13	0	0	0	0	79	0	GOOD
15	14	0	0	0	1	91	1	GOOD
15	15	0	83	0	14	152	8	BLACK
15	16	4	4	0	3	84	6	GOOD
15	17	0	0	0	2	56	2	GOOD
15	18	0	116	34	64	235	50	BLACK
15	19	0	0	154	11	217	22	GOOD
15	20	0	69	0	0	191	12	GOOD
15	21	0	0	0	0	174	6	GOOD
15	22	0	0	0	0	232	6	GOOD
15	23	0	0	0	0	153	1	GOOD

The above dataset shows the information about packets sent, packets received, packets forwarded, packets dropped, route requests, route reply's at different nodes at simulation time 15 when the malicious nodes are connected over UDP connection. In the above condition, Node-4 and Node-16 are communicating over TCP connection. Node-3 and Node-12 are communicating over TCP connection. Node-8 and Node-18 are communicating over UDP connection. Node-9 and Node-15 are communicating over UDP connection.Node-10 and Node-20 are communicating over UDP connection. Node-3 and Node-12 are communicating over UDP connection. i.e, the malicious nodes Node-15 and Node-18 are under UDP connection.

These two datasets are analyzed using Machine Learning Algorithms. For all the algorithms, 70% of the dataset is used are training set and 30% of the dataset is used as testing set. Accuracy scores of different Machine Learning algorithms for TCP connected dataset are shown below.

- *Decision Tree:*

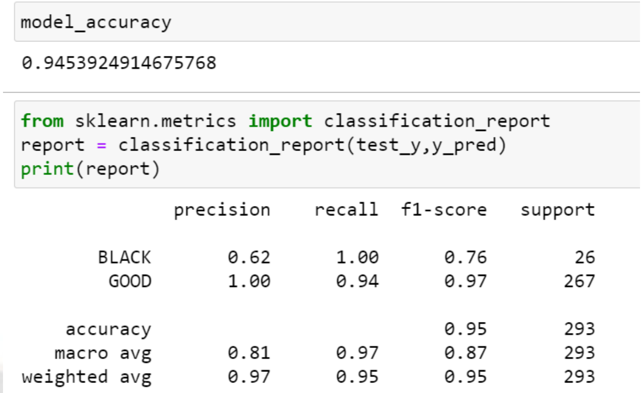


```
from sklearn import tree
from sklearn.model_selection import train_test_split
```

```
train_x,test_x,train_y,test_y = train_test_split(data,target,test_size=0.3,random_state=1)
```

```
from sklearn import preprocessing
scaler = preprocessing.StandardScaler().fit(train_x)
train_x = scaler.transform(train_x)
```

```
model = tree.DecisionTreeClassifier()
model.fit(train_x,train_y)
```



```
model_accuracy
```

```
0.9453924914675768
```

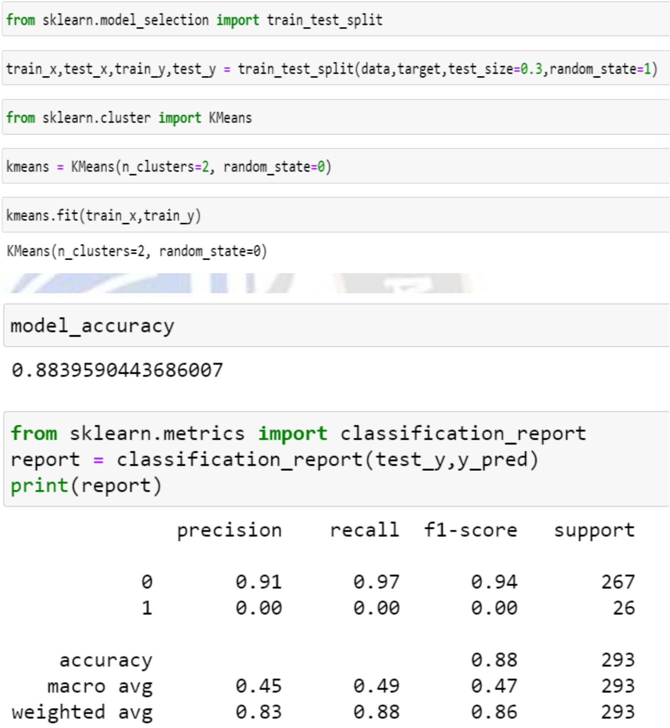
```
from sklearn.metrics import classification_report
report = classification_report(test_y,y_pred)
print(report)
```

```
              precision    recall  f1-score   support

       BLACK       0.62      1.00      0.76        26
        GOOD       1.00      0.94      0.97       267

    accuracy                           0.95       293
   macro avg       0.81      0.97      0.87       293
weighted avg       0.97      0.95      0.95       293
```

- *K-Means:*



```
from sklearn.model_selection import train_test_split
```

```
train_x,test_x,train_y,test_y = train_test_split(data,target,test_size=0.3,random_state=1)
```

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans(n_clusters=2, random_state=0)
```

```
kmeans.fit(train_x,train_y)
```

```
KMeans(n_clusters=2, random_state=0)
```

```
model_accuracy
```

```
0.8839590443686007
```

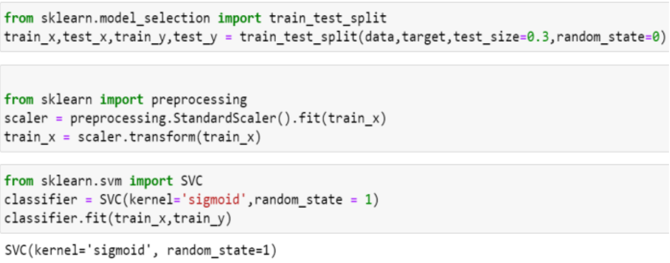
```
from sklearn.metrics import classification_report
report = classification_report(test_y,y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.91      0.97      0.94       267
           1       0.00      0.00      0.00        26

    accuracy                           0.88       293
   macro avg       0.45      0.49      0.47       293
weighted avg       0.83      0.88      0.86       293
```

- *SVM:*



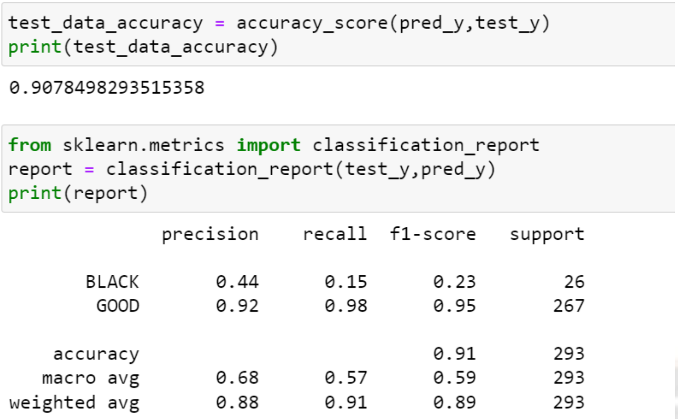
```
from sklearn.model_selection import train_test_split
train_x,test_x,train_y,test_y = train_test_split(data,target,test_size=0.3,random_state=0)
```

```
from sklearn import preprocessing
scaler = preprocessing.StandardScaler().fit(train_x)
train_x = scaler.transform(train_x)
```

```
from sklearn.svm import SVC
classifier = SVC(kernel='sigmoid',random_state = 1)
classifier.fit(train_x,train_y)
```

```
SVC(kernel='sigmoid', random_state=1)
```

Article Received: 29 March 2023 Revised: 25 May 2023 Accepted: 18 June 2023



```
test_data_accuracy = accuracy_score(pred_y,test_y)
print(test_data_accuracy)
```

```
0.9078498293515358
```

```
from sklearn.metrics import classification_report
report = classification_report(test_y,pred_y)
print(report)
```

```
              precision    recall  f1-score   support

       BLACK       0.44      0.15      0.23        26
        GOOD       0.92      0.98      0.95       267

    accuracy                           0.91       293
   macro avg       0.68      0.57      0.59       293
weighted avg       0.88      0.91      0.89       293
```

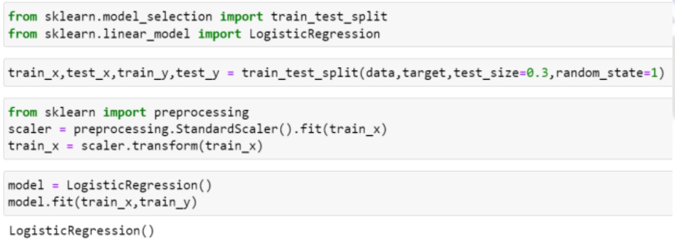
The order of algorithms which shows better accuracy for the TCP connected dataset are shown below.

TABLE I. ACCURACY FOR TCP DATASET

	Accuracy	Precision	Recall	f1-score
Decision Tree	94.53%	97%	95%	95%
SVM	90.78%	88%	91%	89%
K-Means	88.39%	83%	88%	86%

Accuracy scores of different Machine Learning algorithms for UDP connected dataset are shown below.

- *Logistic Regression:*



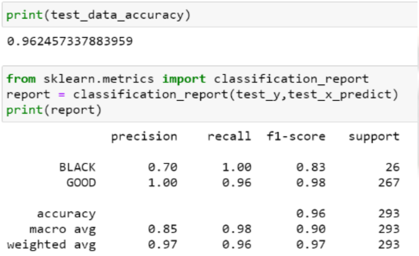
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

```
train_x,test_x,train_y,test_y = train_test_split(data,target,test_size=0.3,random_state=1)
```

```
from sklearn import preprocessing
scaler = preprocessing.StandardScaler().fit(train_x)
train_x = scaler.transform(train_x)
```

```
model = LogisticRegression()
model.fit(train_x,train_y)
```

```
LogisticRegression()
```



```
print(test_data_accuracy)
```

```
0.962457337883959
```

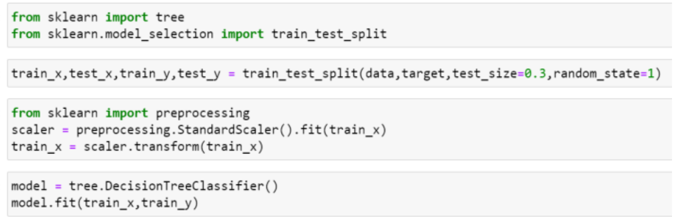
```
from sklearn.metrics import classification_report
report = classification_report(test_y,test_x_predict)
print(report)
```

```
              precision    recall  f1-score   support

       BLACK       0.70      1.00      0.83        26
        GOOD       1.00      0.96      0.98       267

    accuracy                           0.96       293
   macro avg       0.85      0.98      0.90       293
weighted avg       0.97      0.96      0.97       293
```

- *Decision Tree:*

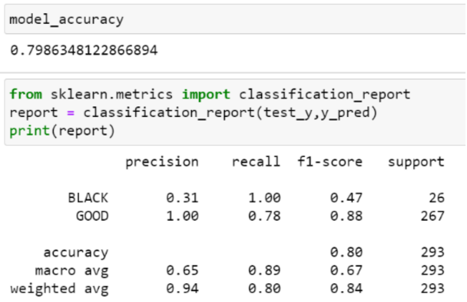


```
from sklearn import tree
from sklearn.model_selection import train_test_split
```

```
train_x,test_x,train_y,test_y = train_test_split(data,target,test_size=0.3,random_state=1)
```

```
from sklearn import preprocessing
scaler = preprocessing.StandardScaler().fit(train_x)
train_x = scaler.transform(train_x)
```

```
model = tree.DecisionTreeClassifier()
model.fit(train_x,train_y)
```



```
model_accuracy
```

```
0.7986348122866894
```

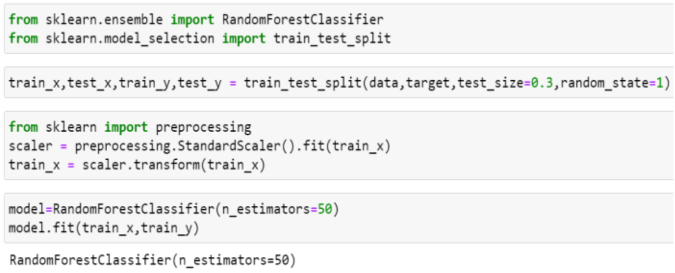
```
from sklearn.metrics import classification_report
report = classification_report(test_y,y_pred)
print(report)
```

```
              precision    recall  f1-score   support

       BLACK       0.31      1.00      0.47        26
        GOOD       1.00      0.78      0.88       267

    accuracy                           0.80       293
   macro avg       0.65      0.89      0.67       293
weighted avg       0.94      0.80      0.84       293
```

- *Random Forest Classifier:*



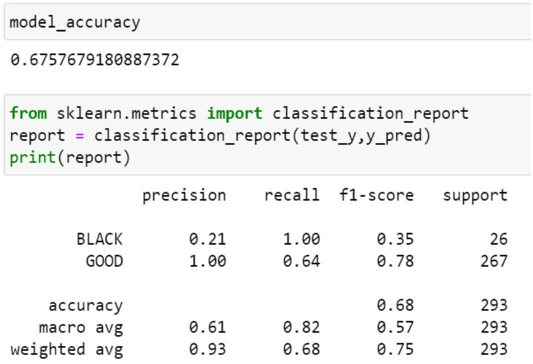
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
```

```
train_x,test_x,train_y,test_y = train_test_split(data,target,test_size=0.3,random_state=1)
```

```
from sklearn import preprocessing
scaler = preprocessing.StandardScaler().fit(train_x)
train_x = scaler.transform(train_x)
```

```
model=RandomForestClassifier(n_estimators=50)
model.fit(train_x,train_y)
```

```
RandomForestClassifier(n_estimators=50)
```



```
model_accuracy
```

```
0.6757679180887372
```

```
from sklearn.metrics import classification_report
report = classification_report(test_y,y_pred)
print(report)
```

```
              precision    recall  f1-score   support

       BLACK       0.21      1.00      0.35        26
        GOOD       1.00      0.64      0.78       267

    accuracy                           0.68       293
   macro avg       0.61      0.82      0.57       293
weighted avg       0.93      0.68      0.75       293
```

The order of algorithms which shows better accuracy for the UDP connected dataset are shown below.

TABLE II. ACCURACY FOR UDP DATASET

	Accuracy	Precision	Recall	f1-score
Logistic Regression	96.24%	97%	95%	95%
Decision Tree	79%	94%	80%	84%
Random Forest	67%	93%	68%	85%

V. CONCLUSION

This work presented the detection mechanism to detect blackhole attack in MANET's. The Network is created and simulated with 25 nodes and these nodes are connected with TCP and UDP connections. With these connections the characteristics like Packets sent, Packets received, Packets forwarded, Packets dropped, Number of route requests, Number of Route reply's are observed and collected in the form of dataset. Then, Machine Learning Algorithms are implemented to analyze their characteristics of these nodes. When two blackholes are connected over TCP connection, Decision Tree Classifier shows better accuracy of 94.53% when compared to Support Vector Classifier SVM (90.78%) and K-Means Classifier (88.39%) in the detection of blackholes in Mobile Adhoc Networks.

On other side, When two blackholes are connected over UDP connection, Logistic Regression shows better accuracy of 90.784% when compared to Decision Tree Classifier (79.86%) and Random Forest Classifier (67.57%) in the detection of blackholes in Mobile Adhoc Networks.

In Future Scope, extra features like throughput, delay, sequence number, CTS and RTS values can be considered to enhance the performance of Machine Learning Algorithms.

REFERENCES

[1] S. Musale, S. Dhende "SAODV: Black hole and gray hole attack detection protocol in MANETs," *2017 International Conference on Wireless Communications, Signal Processing and Networking*, Chennai, India, 2017, pp. 2391-2394, doi: 10.1109/WiSPNET.2017.8300188.

[2] N. Sharma , "Detection as well as removal of black hole and gray hole attack in MANET," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, India, 2016, pp. 3736-3739 , doi: 10.1109/ICEEOT.2016.7755409

[3] N. Arya, "Detecting and avoiding of worm hole attack and collaborative blackhole attack on MANET using trusted AODV routing algorithm," *2015 International Conference on Computer, Communication and Control (IC4)*, Indore, India, 2015, pp. 1-5, doi: 10.1109/IC4.2015.7375649.

[4] V. K. Saurabh, "Cluster-based technique for detection and prevention of black-hole attack in MANETs," *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2017, pp. 489-494, doi: 10.1109/ICECA.2017.8212712.

[5] V. Trivedi "Depictive Analysis of MANETs under Black Hole Attack," *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, Mysore, India, 2017, pp. 1116-1120, doi: 10.1109/CTCEEC.2017.8455010.

[6] J. Thakker, J. Desai and L. Ragha, "Avoidance of co-operative black hole attack in AODV in MANET," *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, 2016, pp. 1049-1053, doi: 10.1109/WiSPNET.2016.7566297.

[7] V. G. Mohite and L. Ragha, "Security agents for detecting and avoiding cooperative blackhole attacks in MANET," *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, Davangere, India, 2015, pp. 306-311, doi: 10.1109/ICATCCT.2015.7456900.

[8] S. Shrestha, "Securing Blackhole Attacks in MANETs using Modified Sequence Number in AODV Routing Protocol," *2020 8th International Electrical Engineering Congress (iEECON)*, Chiang Mai, Thailand, 2020, pp. 1-4, doi: 10.1109/iEECON48109.2020.229555.

[9] Vimal Kumar "An adaptive approach for detection of blackhole attack in mobile ad hoc network," Procedia Computer Science, vol. 48, 2015, pp. 472–479.

[10] Shalini, A. K. ., Saxena, S. ., & Kumar, B. S. . (2023). Designing A Model for Fake News Detection in Social Media Using Machine Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering, 11(2s), 218 –. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2620

[11] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park, "Black hole attack in mobile ad hoc networks," in ACMSE'04, April 2–3, 2004, pp. 96–97

[12] Walsh, Gianluca & group, The & Harrow, Jennifer & Psomopoulos, Fotis & Tosatto, Silvio. (2020). DOME: Recommendations for supervised machine learning validation in biology.

[13] Ghosh, Chandril. (2022). Machine Learning. 10.1007/978-3-031-14634-3_4.

[14] D. V S and V. S, "Behaviour Analysis and Detection of Blackhole Attacker Node under Reactive Routing Protocol in MANETs," *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, Bangalore, India, 2018, pp. 1-5, doi: 10.1109/ICNEWS.2018.8903972.

[15] Bala,A.,Bansal,M.,Singh,J.(2009)"Performance Analysis of MANET under BlackholeAttack" In Proceedings of IEEE International Conference on Networks and Communications, NETCOM '09.,pp.141 – 145

[16] Suryawanshi, Ranjeet and Tamhankar,Sunil (2012) "Performance Analysis And Minimization Of Black Hole Attack In MANET"International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 2, Issue4,pp.1430- 1437

[17] Ming-Yang Su, Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems,Computer Communications, Volume 34, Issue 1, 2011,Pages 107-117,ISSN 0140-3664.

[18] Oscar F. Gonzalez, Godwin Ansa, Michael Howarth and George Pavlou, Detection and Accusation of Packet Forwarding Misbehavior in Mobile Ad-Hoc Networks, Journal of Internet Engineering, vol. 2(1), pp. 181–192, (2008).

[19] G., Maria. (2022). An Approach to Machine Learning. 10.47716/MTS.B.978-93-92090-08-0.

[20] S. Chaturya, Y. Adilakshmi, MBDS: Message Authentication Code (MAC) Based Black Hole Detection System in Manets, International Journal of Innovative Technology and Research, ISSN: 2320-5547, pp. 7980-7987, Feb-Mar 2018, indexed by Google Scholar.

[21] Adilakshmi Yannam, G.V.S.N.R.V.Prasad , "Trust Aware Intrusion Detection System to Defend Attacks in Manets." International Journal of Innovative Technology and Exploring Engineering, 8(7), 1298-1306. ISSN: 2278-3075. BEIESP. (2019) Retrieval Number: F3815048619/19..

[22] Yannam, Adilakshmi & Prasad, Dr. (2019). "Cooperative Intrusion Detection System to Enhance the Security in MANET." Journal of Advanced Research in Dynamical and Control Systems. 11. 100-109.

[23] Muhammad Khan, Machine Learning for Predictive Maintenance in Manufacturing: A Case Study , Machine Learning Applications Conference Proceedings, Vol 1 2021.

[24] Joseph Miller, Peter Thomas, Maria Hernandez, Juan González, Carlos Rodríguez. Machine Learning for Decision Support in Uncertain Environments. Kuwait Journal of Machine Learning, 2(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/205

[25] Ms. Madhuri Zambre. (2012). Performance Analysis of Positive Lift LUO Converter . International Journal of New Practices in Management and Engineering, 1(01), 09 - 14. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/3

[26] Hussain, I., Vinay, P., Deepthi, P. V. L., Mohan, R., & Adilakshmi, M. Y. (2020). "Black hole Attack Detection Using Machine Learning Algorithms in MANET – Performance Comparison." International Research Journal of Engineering and Technology, e-ISSN: 2395-0056, June 2020 , 7(6), pp. 6047-6051.