IST-Africa 2017 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2017
ISBN: 978-1-905824-57-1

Design and Implementation of an Intrusion Detection System using MLP-NN for MANET

Innocent MAPANGA[1], Vinod KUMAR[2] ,Wellington MAKONDO[3],
Tripathi KUSHBOO[4] Prudence KADEBU[5] , Wadzanayi CHANDA[6]
[1,3,5,6]Harare Institute of Technology, P.O Box BE277 Belvedere, Harare, +263, Zimbabwe
Tel: +263782471127, Fax: +263 4 741 406, Email: imapanga@hit.ac.zw
[2]*Delhi Technological University, Shahbad Daulatpur, Bawana Road, Delhi-110042, India*
Tel: +91-11-27871018, Fax: +91-11-27871023, Email: vinodkumar@dce.edu
[4]*Amity University Gurgaon, Punchgaon, Gurgaon (Manesar) – 122 413, Haryana*
Tel: +91-7523827231, Email: ktripathi@ggn.amity.edu

Abstract: Communication in Mobile Ad hoc (MANETS) networks for end-to-end delivery of packets is achieved cooperatively. This model assumes that an intermediary node will always forward traffic originating from other nodes willingly, other than traffic emanating from the node itself. Conversely, in hostile environments where we find most applications of our ad hoc networks, an always cooperative and submissive behavior on behalf of the other nodes of the network cannot be presumed as the ultimate action undertaken by all the nodes. Our focus in this paper is on detecting the presence of malicious nodes that selectively or randomly drop packets intended for other destination nodes, we further classify each packet drop attack, according to its attack type by observing and analysing how each packet drop attack affect the network characteristics. Using a simulated MANET environment and MLP-NN modelling we can illustrate an Intrusion detection System that can successfully detect malicious packet dropping attacks with great accuracy.

Keywords: Intrusion Detection Systems, MANET, Multi-layer Perceptron, Artificial Intelligence, Network Security, Machine Learning.

1. Introduction

Mobile Ad hoc Network (MANET) is a collection of nodes which are mobile and self-organize themselves into a network, with no fixed topology. As such nodes can freely roam around, join or leave the network randomly [1]. MANETs can be established devoid of any infrastructure hence are becoming very useful, especially in environments that are geographically constrained for instance in next generation of battlefield applications envisioned by the military as well as in applications like disaster recovery and message exchanges in rescue missions [2]. Also, each node can function as a router, utilizing its multi hop routing facility [18]. This eliminates the need for a dedicated router or access point for communication between nodes. However, the MANET is vulnerable especially due to its continuously changing topology, open medium, lack of central monitoring point as well as no clear defensive mechanism. [5] For instance any un-trusted node is capable of joining the network, subsequently posing threats to it. This can be done either by dropping the packets or by providing wrong information to the network among other things.

Ensuring and enforcing security in MANET is of prime importance since data in transit may be confidential and delivery of packets must be ensured by the network. Attacks targeted at MANETs can emanate either from within or outside the network and most times attacks come from trusted nodes within the network. [7]The Wireless medium is susceptible to attacks, due to ease of access into the network [24]. The costs of damages in the event of an attack as a result of malicious activities in the network can have unbearable consequences hence a need for systems that can monitor data flow within the network in order to circumvent possible malicious activities. Such a system for monitoring the network is called an intrusion detection system (IDS).[8] An IDS collects and evaluates information from different areas within a node or network to discover suspicious patterns that may signify an attack or attempt to compromise a system. [21] IDS design are based on two approaches namely, anomaly detection and misuse based IDS. Misuse-based IDS looks for behavior corresponding to predefined intrusion or vulnerability signatures. [10] Anomaly detection based IDS searches for abnormal network traffic, which can either be a violation of acceptable thresholds for an occurring event or a violation of a user's normal behaviour in the network. [11]

2. Research Problem and Objectives

In this paper the problem addressed is shown diagrammatically in Figure 1 below. A set of n nodes formulating a MANET having m nodes acting maliciously by dropping packets either continuously or selectively is illustrated. A fraction m of the nodes deployed in the network is assumed to be misbehaving. Packets within the traffic are dropped as they are moved from a source to a destination. Given a path P of length k, we make an assumption that a set of m malicious nodes, where $|m| \leq k$, are present on P. These nodes can be located anywhere along P.

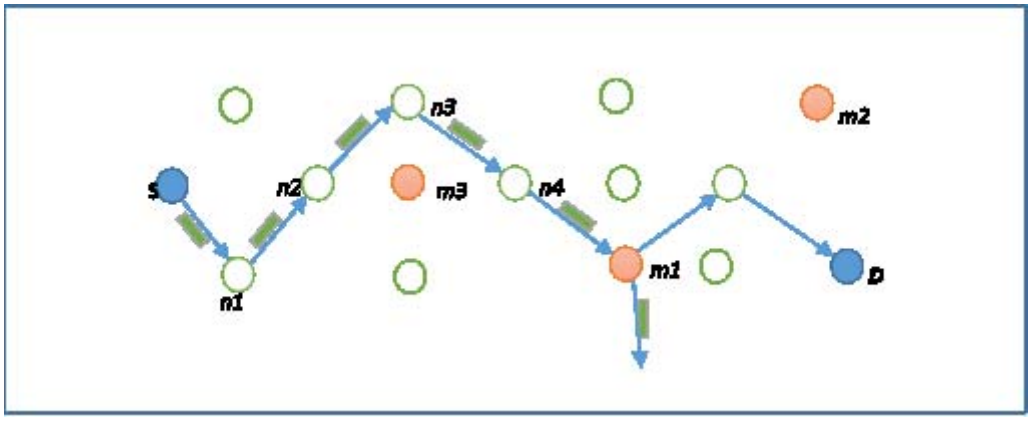

Figure 1: Problem Scenario

In Figure 1 S the source node is sending traffic to the destination node D along P. Node m_1 is dropping all the packets it is receiving. Our goal is to identify m_1, provide evidence of its misbehaviour, and identify the type of attack manifesting on m_1, so that we can classify it accordingly for further action to mitigate the attack.

In our model, we consider several types of attacks that lead to an ultimate packet dropping action, for example: Wormhole attack, Black hole attack and several others. Each attack can be observed to affect network characteristics differently and analysis of the network characteristics can give us the state of the network, describing whether it is under an attack or in normal operation. [14]

In the first phase of our work, we simulate different attacks using ns-2.35 simulator under the AODV routing protocol and in this phase we collect network characteristic data that is crucial for the next phase. The second and final phase we use the collected data of our desired parameters for training a neural network used for attack classification and detection in the Matlab platform

3. Proposed Methodology

This research motivate the use of a technique that hinges on Multilayer Perceptron Neural Network for detection and classification of packet drop attacks. The detection will be performed on the collected network characteristic data. In this technique we collect and analyse locally available data. We mainly focus on nodes that participate in the route discovery process successfully. The data collection, analysis, detection and classification components form the core of the detection technique. Local data collected such as route request and route error messages, are used to extract important parameters that affect network characteristic data which will be passed as input for training the detection engine. This will in turn be used to detect and classify misbehaviour in the network. This information is gathered by the data collection component during the duration of the simulation period. Collected data is passed on to the analysis module that extracts useful information or parameters from control messages being exchanged in the network, for use as input in the second phase involving training the detection engine. The detection engine will check for any deviation from normal behavior and classify the attacks according to their types as well.

Our proposed solution has a set of two nodes namely (i) regular nodes which do not pose any threat to the MANET. Regular nodes are responsible for exchanging routing information and sending or forwarding data packets to a destination on behalf of other nodes and perform actions regarded as normal in the network [14]. The second type is a (ii) malicious node with a built-in attack mechanism aimed at causing undesirable effects in the network, this node drops all data packets but responds to all routing information

The methodology and the algorithm used in our work is shown diagrammatically below. This technique relies on readily available information at different network levels, to detect the presence of malicious nodes.

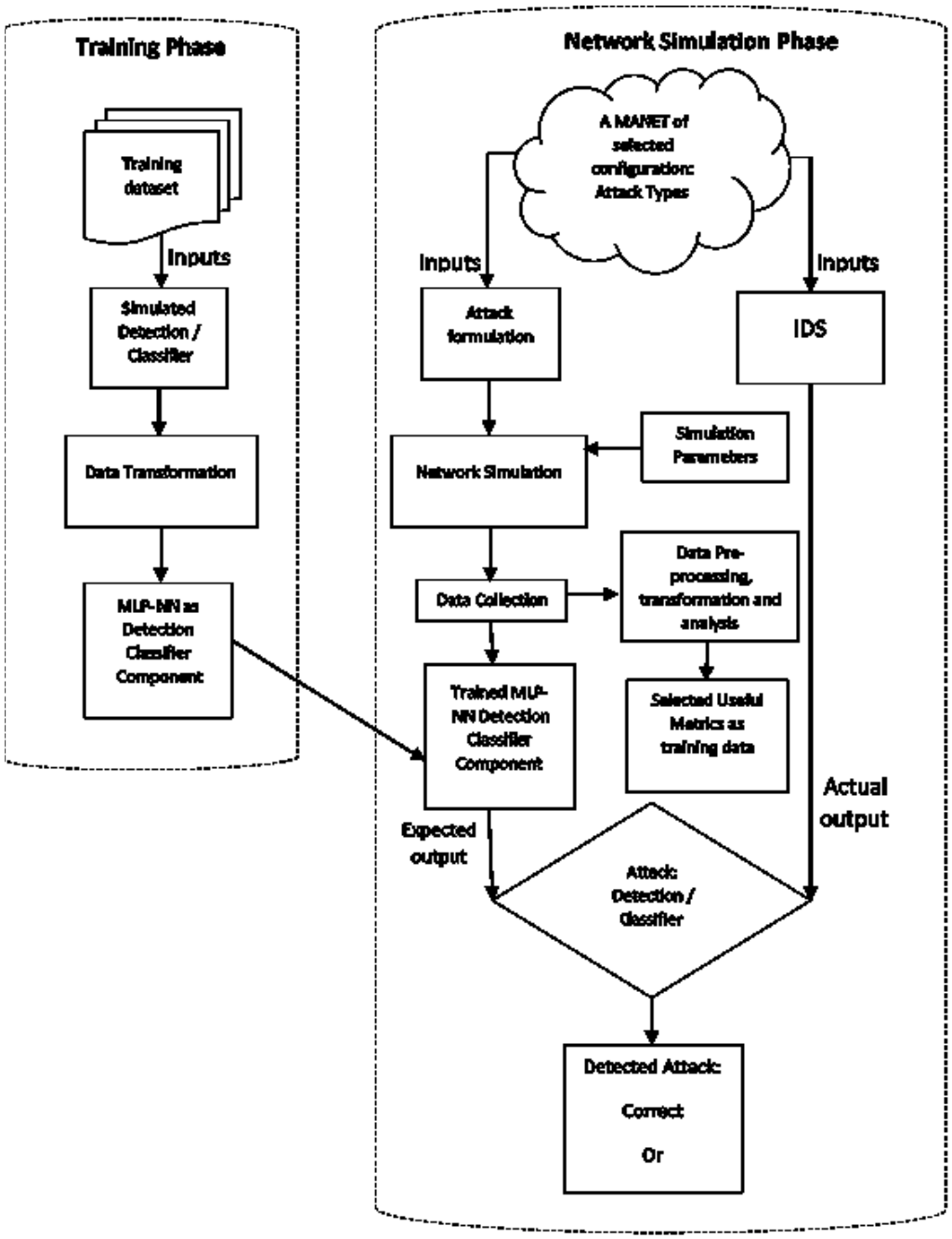

Figure 2: Proposed IDS Architecture

4. Simulation and Results

Part of our work is simulated via Network Simulator (NS) 2.35. NS-2 has as its prime aim to support research in networking at various institutions undertaking networking research [17]. New protocols can be developed and traffic patterns can be studied in NS-2.

A. Network Simulation Scenario

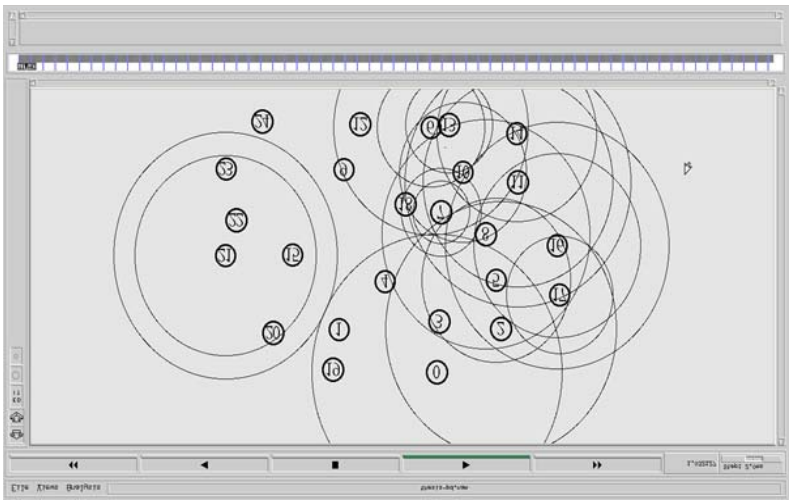

Figure 3: Nodes in transmission action with their hearing ranges

B. *Trace file*

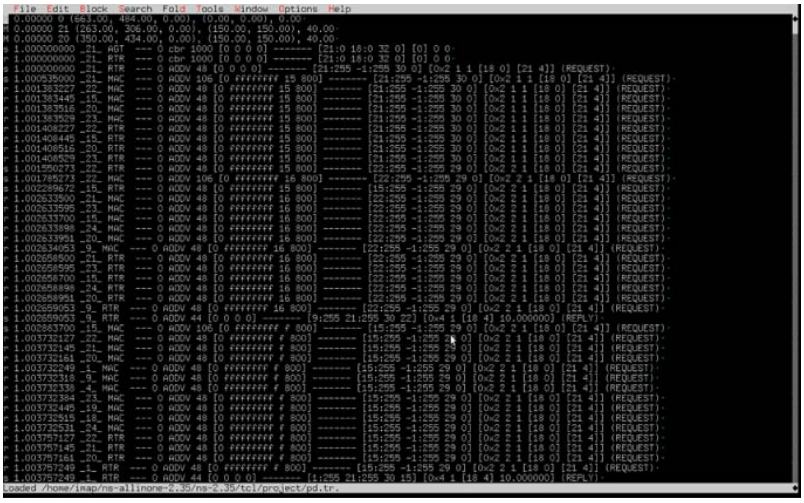

Figure 4: Trace file after the initial simulation in the network

The effectiveness of the MLP-NN model depends on the training done as well as the data used. The collection of data for training is a critical problem. This can be obtained several ways as including by using real traffic and by using simulated traffic. In our case we used simulated traffic to arrive at our data sets, which we then divided into three subsets. The first subset is the training set, which is used for training and updating the ANN parameters. The second subset is the validation set.

Sample Collected data for a Normal network scenario from our simulation (*each row shows the value of one of the selected parameters*) can be shown below

In this phase we make use of the parameters obtained in phase one above as our intrusion detection evaluation data set with an extract shown in the table 1 above. The sample version of the dataset included 6000 records. A subset of the data that contained the desired attack types and a reasonable number of normal events were selected manually. The final dataset used in this study included 2500 records. MLP-NN is used to train the detection and classification engine.

RREQ	224	196	206	219	190	202	170	188	240	196

Sent										
RREQ Replies	36	51	49	61	34	32	36	48	60	62
Route Error Messages	0	2	2	1	0	0	1	2	2	1
Packet Data Sent	1241	1241	1241	1241	1241	1241	1241	1241	1241	1241
Packet Data Received	1240	1192	992	1020	900	962	980	1021	1011	960
Packet Data Dropped	0	56	249	221	139	341	279	261	220	230

Table 1: Sample Data

C. Training and Validation Method

This section details an implementation of a Multilayer layer Perceptron-NN done in MATLAB using the Neural Network Toolbox [16], for the purpose of detection and classification. We constructed the MLP-NN with the desired neurons per each layer, with a desired activation function. The training data (feature vectors) and the corresponding target or desired outputs are fed to the neural network to begin our training. The implemented neural networks had 6 input neurons (equal to the dimension of the feature vector) and output neurons equal to the number of classes desired.

The number of inputs fed to the IDS would be exact as those defined as the MLP parameters. In the simulation of the IDS the input data types are taken into consideration notably the length as well as the data range for numerical inputs. A real test case training dataset obtained from data dumps of Network Simulation from NS2 was used to train the MLP-NN. Two scenarios modelled as normal and adversarial are passed to MLP as 2 dimensional arrays in CSV file format. In order to train our MLP-NN, data was mapped into the input space of the MLP-NN subsequently allowing learning to be conducted from numerical values, which is usually the case.

D. *Feedforward MLP-NN*

Figure 5 depicts an MLP-NN of a feedforward type that comprises of layers of nodes resembling a directed graph, having interconnection between the layers. It has a single input layer, a single hidden layer, and a single output layer. An activation function is linked to every node in the hidden and output layers. Signals coming from the preceding layers are constantly fed to the next layer in the forward direction for further computations.

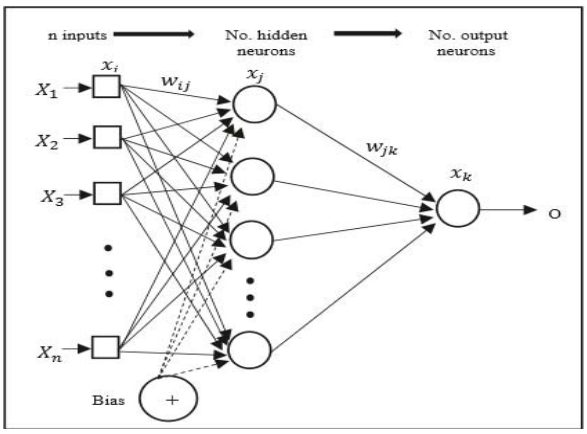

Figure 5: Multi-layer Perceptron Neural Network

An activity in a neuron is denoted by *Xi*. A connection that exist between the neurons are denoted by w, the weight factor or strength. Therefore for a node i linked to node j, then the linking weight is represented as *wij*. All inputs are multiplied by the weighting factor to give them strength. All arriving signals to the neuron are summed up as given in equation (1). The inner sum corresponds to:

$$x_{ij} = \sum_{i=1}^{n} x_i w_{ij} + w_0 \qquad (1)$$

where w_0 is the bias for the weight

The summed input result is passed as an argument to the activation function for further computation, which will give result or output value of a node. The result of the activation function denoted as f is given as *xj*,

$$x_j = f(\bar{x_j}) \qquad (2)$$

and to sum it all comprehensively,

$$x_j = f(\sum_{i=1}^{n} x_i w_{ij} + w_0) \qquad (3)$$

E. Activation functions and Reduction

The activation function acts as the decision engine of the neuron acting upon the summed input it receives. The activation function determine when it is appropriate for a neuron to be active subject to the stated threshold when reached or not. In our work, the training was subjected to a number of activation functions that include the linear, sigmoid and hyperbolic tangent activation functions. An activation function that best addressed the problem was chosen after conducting experiments to evaluate the activation functions with respect to the problem at hand.

The entire network results are obtained after the output neurons are activated. The obtained results are then availed either to humans for their use or to another process requiring them. If the desired output is not obtained, a deviation might have occurred during training signalling an error. The error can be corrected by adjusting the weight denoted as wij that are between the neuron links. The measure of error has to be established first.

Given neurons xa, xb and xc and desired outputs as di= (dki, dki+1, dki+3) then the measure of error can be found by the equation below:

$$E_i = (d_{ki} - x_{a1}) + (d_{ki+1} - x_{a2}) + (d_{ki+3} - x_{a3}) \quad (4)$$

So if we had *n* outputs, the measure of error will be generalized as

$$E_i = \sum_{i=1}^{n} (d_{ki} - x_{ai})\, 2 \quad (5)$$

In order to get the overall difference for the data set presented to the MLP-NN, we have to sum the output over all input vectors where the number of inputs is denoted by *s*.

$$E_i = \sum_{k=1}^{s} \sum_{k=0}^{n} (d_{ki} - x_{ai})2 \quad (6)$$

$$E_s = \sum_{s} (d_{ki} - x_{ai})\, 2 \quad (7)$$

The resultant output is a function of the contribution of many determinants including inputs, weights and the error too. This error needs to be eradicated if the results are to match what is desired in the training before we can trust our MLP-NN to work a real life environment. Since the error E is as a result of all the weights, it becomes prudent to determine the effects of each given weight w's change on the overall calculated error. For a given n number of weights, the error will be denoted by:

$$E_i = f(w) = f(w_1, w_2, \ldots\ldots, w_n) \quad (8)$$

If the given function f is continuous and differentiable, that will be the same as computing the derivative of E with respect to wi of $\partial/\partial wi$ in the point w. An error term $\bar{\in}$ has to be computed therefore for each node as given by the below equation:

$$\bar{\in}_i = \frac{\partial + E_p}{\partial \bar{x}_i} \quad (9)$$

A weight update function will have to be computed and established to enable updating of weights linking two given nodes, this can be denoted as follows:

$$\Delta w_{ki} = -\eta\, \bar{\in}_i\, x_i \quad (10)$$

Where the learning rate is denoted by η. The learning rate affects the convergence speed as well as the stability of the weights during training. Training continues until the error has been minimized .a weight matrix that gave the minimum is created and saved for use in the testing phase and beyond.

F. Testing Phase

In this phase the MLP-NN is subjected to test data and its results are observed and checked against the expected or desirable results. Human intelligence is then sort to compare the results with any deviations being taken note of.

The inaccuracy on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training similar to the training set error. However, when the ANN begins to over-fit the data, the error on the validation set will typically

begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights that produced the minimum error on the validation set are retrieved [12].

There are at least four different chosen categories of packet dropping attacks in our MANET environment including selfish, sleep deprivation attack, Blackhole attack among a many other attacks. Table 2 shows detailed information about the number of records from normal and three attack types included in training, validation, and testing sets.

Record types	**Training set**	**Validation set**	**Test set**
Desired / Normal	2500	500	1000
Selfish packet droppers	2500	500	1000
Blackhole	2500	500	1000
Sleep deprivation	2500	500	1000

Table 2: Distribution of data vectors

This research was aimed to detect the malicious packet droppers and solve the multi class problem brought about by several attacks. Our scenario has a categorization which detail a set of different attacks, nevertheless open for adding more classes. Our output layer gives a variety of outputs which we shall term outputs states representing various classes, a good example will be to suppose that state [0] represents normal desired scenario, while [1] represents a malicious attack known as sleep deprivation and another state [2] representing packet dropper.

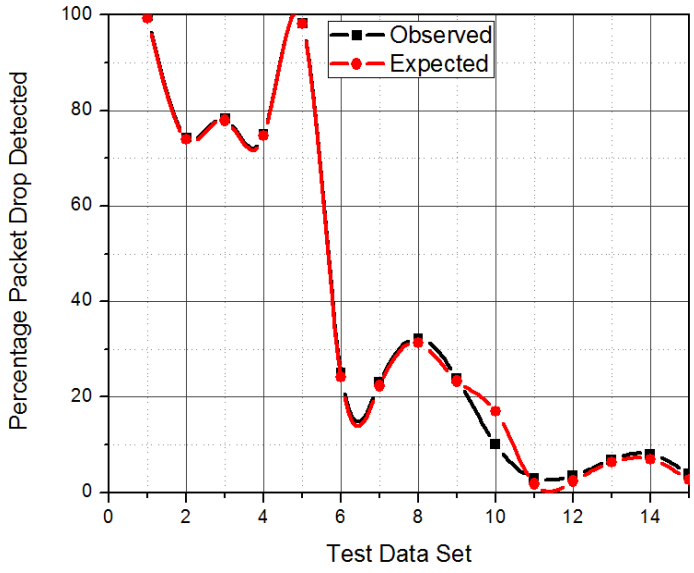

Figure 6: Packet drop detection rate in percentage terms

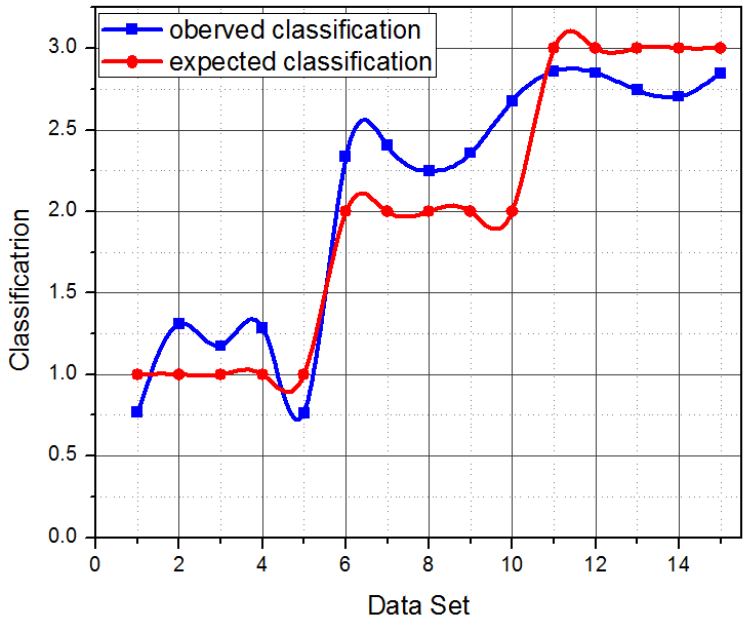

Figure 7: Attack classification

5. Conclusion and Future Work

This section summarizes our research work and reflects on our proposed future work.

A. *Conclusion*

Some features inherent in Mobile ad hoc networks (MANET) makes them susceptible to several attacks such as free access for everyone to the wireless medium, an ever-changing topology, distributed collaboration, and limitations in their abilities like memory and power. MANET's security can be aided by Intrusion Detection Systems (IDS) which can act as a second line of defense that is crucial to the overall security of the Network. Nevertheless, designing an IDS in MANET is a daunting task. Most wired IDS in the market today lack an approach that is distributive. Core to our research problem is the design of a scheme that is a scalable approach of intrusion detection for a MANET. This research utilise MANET as the core assessment platform.

To evaluate the detection and classification effectiveness of our technique we carried out a number of simulations. The detection / classification technique was evaluated on the basis of verifying false positives, accuracy and false negatives.

B. Detection Effectiveness and Accuracy

i. Percentage of Dropped Packets

We compute the ratio of dropped packets calculated over all source/destination pairs every 100 epochs given as

$$PD = \frac{\#\,packets\ dropped}{\#\,packets\ sent\ by\ sources} \times 100\% \qquad (14)$$

Where $\#\,packets\ dropped = \#packets\ sent - \#packets\ recieved$

We measured the percentage of dropped packets when misbehaving nodes were dropping the packets. We also varied the number of nodes misbehaving in the network with capabilities to drop packets haphazardly. We went on to evaluate the performance of our detection engine as measured by the detection effectiveness derived as:

$$DE = \frac{\#\ malicious\ nodes\ detected}{\#\ malicious\ nodes\ present} \times 100\% \qquad (15)$$

C. *Classification*

For the purpose of classification tasks in our system, we used the metrics true positives (TP), true negatives (TN), false positives (FP), and false negative (FN) in helping us to compare how effectively our classifier classified the results under test against our expected output. Our detection /classifier engine will be evaluated according to the estimation it will do, to evaluate whether it is close enough to the expected solution or not. Based on the prediction outcome a judgment can be arrived at whether it is TP, FP, FN or TN. For example if we define our investigation from the above cases for an approximate situation.

D. Future Work

Ongoing researches have not dedicated much time to this area of securing MANETs by a way of intrusion detection systems. The research undertaken in this paper detail our preliminary work on IDS in MANETs. Many notable exciting, thought-provoking impending directions are promising in this research area:

Our focus has been on the network layer were we used AODV routing protocol, we will however seek to extend our technique to other layers such as Medium Access Control (MAC) layers or application layers. Explore how to design more detection strategies especially at the hands of Zero day attacks and other sophisticated attacks. Broaden the scope beyond MANETs, to incorporate other technologies like the internet of things.

References

[1] G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp.529-551, April 1955.

[2] S. Buchegger and J.-Y. L. Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP'02), pages 403{410, 2002.

[2] M. Jakobsson, J.-P. Hubaux, and L. Buttyan. A micropayment scheme encouraging collaboration in multi-hop cellular networks. In Proceedings of Financial Crypto 2003, 2003.

[3] K. Liu, J. Deng, P. Varshney, and K. Balakrishnan. An acknowledgment based approach for the detection of routing misbehaviour in Manets. IEEE Transactions on Mobile Computing, 6(5):536{550, May 2007.

[4] Y. Liu and Y. R. Yang. Reputation propagation and agreement in mobile ad-hoc networks. In Proc. of IEEE Wireless Communication and Networking Conference (WCNC'03), pages 1510{1515, March 2003.

[5] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehaviour in mobile ad hoc networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), pages 255{265, 2000.

[6] S. Buchegger and J.-Y. L. Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP'02), pages 403{410, 2002

[7] M. Jakobsson, J.-P. Hubaux, and L. Buttyan. A micropayment scheme encouraging collaboration in multi-hop cellular networks. In Proceedings of Financial Crypto 2003, 2003.

[8] B. Culpepper, H. Tseng, N. Center, and C. Monett Field. Sinkhole intrusion indicators in DSR MANETs. In Proceedings of Broadband Networks. First International Conference on, pages 681{688, 2004.

[9] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In Network and Distributed System Security Symposium (NDSS), pages 131{141, 2004.

[10] Y. Hu, A. Perrig, and D. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In Proceedings of the 2nd ACM workshop on Wireless security, pages 30{40, 2003.

[11] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. Chang. Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach. In Proceedings of IEEE Wireless Communications and Networking Conference, volume 2, 2005.

[12] K. Liu, J. Deng, P. Varshney, and K. Balakrishnan. An acknowledgment based approach for the detection of routing misbehaviour in Manets. IEEE Transactions on Mobile Computing, 6(5):536{550, May 2007.

[13] Y. Liu and Y. R. Yang. Reputation propagation and agreement in mobile ad-hoc networks. In Proc. of IEEE Wireless Communication and Networking Conference (WCNC'03), pages 1510{1515, March 2003.

[14] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehaviour in mobile ad hoc networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), pages 255{265, 2000.

[15] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In Proceedings of the Sixth IFIP Conference on Security Communications and Multimedia (CMS02), 2002.

[16] E. Ngai, J. Liu, and M. Lyu. On the intruder detection for sinkhole attack in wireless sensor networks. In Proceedings of the IEEE International Conference on Communication (ICC), 2006.

[17] A. Pirzada and C. McDonald. Circumventing sinkholes and wormholes in wireless sensor networks. In Proceedings of the International Conference on Wireless Ad Hoc Networks (IWWAN), 2005.

[18] R. Poovendran and L. Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. Wireless Networks, 13(1):27{59, 2007.

[19] B. Culpepper, H. Tseng, N. Center, and C. Monett Field. Sinkhole intrusion indicators in DSR MANETs. In Proceedings of Broadband Networks. First International Conference on, pages 681{688, 2004.

[20] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto. Detecting blackhole attack on AODV-based mobile ad hoc networks by dynamic learning method. International Journal of Network Security, (3):338{346, 2007.

[21] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In Network and Distributed System Security Symposium (NDSS), pages 131{141, 2004.

[22] Y. Hu, A. Perrig, and D. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In Proceedings of the 2nd ACM workshop on Wireless security, pages 30{40, 2003.

[23] S. Buchegger and J.-Y. L. Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP'02), pages 403{410, 2002.

[24] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM 2003), San Francisco, CA, USA, April 2003.

[25] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In Proceedings of the 2003 ACM Workshop on Wireless Security, pages 30–40, San Diego, CA, USA, 2003. ACM Press.

[26] Wellington Makondo, Raghava Nallanthighal, Innocent Mapanga, Prudence Kadebu. Exploratory test oracle using multi-layer perceptron neural network. In Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India.