Proposing A Distributed Model For Intrusion Detection In Mobile Ad-Hoc Network Using Neural Fuzzy Interface

Meisam Yadollahzadeh Tabari[1], Hamid Hassanpour[2], Ali Movaghar[3]
(1) Department of Computer Engineering, Nour Branch, Islamic Azad University, Nour, Iran
(2) Department of Computer Eng. & IT, Shahrood University of Technology,Shahrood,Iran
(3) Department of Computer Engineering, Sharif University of Technology,Tehran, Iran
iust_tabari@yahoo.com, h_hassanpour@yahoo.com, movaghar@sharif.edu

Received: 2011/02/15 ;Accepted: 2011/04/13 Pages: 85-96

Abstract

Security term in mobile ad hoc networks has several aspects because of the special specification of these networks. In this paper a distributed architecture was proposed in which each node performed intrusion detection based on its own and its neighbors' data. Fuzzy-neural interface was used that is the composition of learning ability of neural network and fuzzy Ratiocination of fuzzy system as our architecture engine. Our scrutiny showed that fuzzy-neural interface had a good ability for separating normal data from abnormal. The ability of proposed architecture was evaluated in accuracy and overhead case.

Keywords: *Mobile ad-hoc network, security, attack, intrusion detection, neural-fuzzy interface*

1. Introduction

Mobile ad-hoc networks consist of a set of nodes that can be related with each other without the need for any predefined infrastructure. The characteristics of these networks, such as speed in construction and its infrastructure-less architecture, cause to perform an important role in different fields especially in military and emergency applications. Security concern in these networks is currently the most important research case. Nodes mobility, wireless communication, dynamic changes of the network topology, lack of any central point for gathering behavioral and functional data, lack of any distinct line of defense and limited resource of these networks prepare a good environment for attackers to accomplish their malicious purpose. These attacks are developed with increasing these networks usage [1,2,3], so emphasizing only on prevention mechanism is not enough and we need a secondary defensive line for securing such networks. Intrusion detection system (IDS), as a solution for detecting misbehavior functions, has been widely used. These systems accomplish detection process by gathering required information from their monitoring environment which is called audit data and addressing it to a detection engine. Totally there are two techniques for intrusion detection in these networks: *misuse detection*, and *Anomaly detection*.

The first technique accomplishes intrusion detection by comparing current network pattern with the well known attack patterns and the latter by comparing it with the

pattern that has been produced from normal network behavior. If there is a deviance between them, we can assume that the network is under attack. With the consideration that the first approach should maintain the attacks pattern database which is a costly term in these networks and also that it doesn't have the ability of detecting zero day attack which the later has, anomaly detection is the best approach for these networks.

Regardless of used technique there are some architecture which intrusion detection were done based on them. As a whole there are three kinds of architectures:

Stand-alone Architecture: This architecture is based on a self-contrained approach for detecting malicious actions at each network node.

Cooperative Architecture: In the cooperative IDS architectures an intrusion detection engine is installed in every node monitoring local audit data and providing intrusion detection. To resolve inconclusive intrusion detections and detect more accurately advanced types of attacks, detection engines may cooperate with engines of neighbour's nodes through the exchange of audit data or detection outcomes.

Hieratical Architecture: In the hierarchical IDS architectures the network nodes are divided into cluster-heads and cluster members. The latter typically run a lightweight local intrusion detection engine, while the former run a comprehensive engine that processes raw or pre-processed audit data from all the cluster members [4].

Good IDS in these networks should accomplish intrusion detection by concerning its characteristics such as low battery. Hence, the spotted datasets should be in lowest weight as possible and also IDS should have a fine accuracy for detecting intrusions.

The main concern of this paper is proposing a distributed architecture that use neural-fuzzy interface as its detection engine. The rest of the article consists of the following sections: In section 2 a short glance at related works that have been done in this field was proposed. In section 3 the routing protocol (AODV) as our work's conditions was explained. The presented attacks in these networks was analysed in section 4. Also, suggested architecture was presented in section 5. Implementation and evaluating of proposed architecture was presented in section 6. The conclusion of our work is in section 7.

2. Related Works

Generally, existing approaches for anomaly detection in mobile ad-hoc networks may use a classifier, finite state machine, or game theory. In this research, a classifier was used for intrusion detection. Hence, classifier-based intrusion detection approaches in the literature were reviewed in this section.

This method is based on the fact that the normal behavior of the system is apart from the abnormal one. This classifying system can predict the following event of the system with considering its current treatment and if the happened event does not have any correspondence with the predicted event, it will be detected as an anomaly. Along with setting up the classifier, some of the features in the system that has a high data value should be used. Classifier divides the dataset of the nodes' data to distinct and absolute groups by testing the value of features. In this part, some algorithms like SVM, RIPER or $C_{4.5}$ as well as decision tree can be used. Zhang and Lee [5], proposed the first high level system for intrusion detection in ad-hoc networks. This paper introduced a

distributed and communicative system for intrusion detection based on anomaly detection techniques. Their proposed system is used as a guide for many systems which has been designed later. In this system, they gathered and used information from network and application layers.

Huang and Lee [6] improved their previous work by proposing a cluster based intrusion detection system. Their proposed architecture decreases the use of energy for intrusion detection in mobile ad-hoc networks. A series of features that are extracted from node's routing table and detect normal case from the abnormal are used in their research using $C_{4.5}$ algorithm. This system has the ability to detect the source of the attack if the intruder node is in one-hope distance from detector's node. In [7], the authors evaluated the use of a hierarchically distributed system and a completely distributed system. They used data from network layer and a Support Vector Machine (SVM) for intrusion detection. In this paper, hierarchically distributed system was proved to have higher power than a completely distributed system. Lui et al [8] summed up information from MAC Layer, and then gained the normal profile of the system with performing cross-feature algorithm on this data by proposing a completely distributed system. This profile was used for intrusion detection in the next stage. [9] Cooperative detection method was proposed by analyzing feature for detecting sink-hole attack in the context of Dynamic Source Routing (DSR). The evaluation of their work was done in terms of detection rate and detection time.

3. AODV Routing Protocol

AODV is one of the most popular reactional routing protocols designed for mobile ad-hoc networks. In this protocol, routes are established on demand and its mechanism provides a rapid dynamic network connection featuring low processing loads and low memory consumption with very good packet arrival rate. AODV uses two functions for routing in the network, Route discovery which is used to find a path from source node to destination and Route maintenance which is used to handle route changes in the network. First function uses Route Request (RREQ) and Route Reply (RREP) messages while the second uses Route Error (RERR) and Hello .AODV does not utilize source routes and instead refers to a routing table for each node in the network and updates its content while receiving a routing message. Each routing table has one entry for any accessible path in the network. Each entry includes < Destination IP address, Destination sequence number, Hop count, Next hop, Precursor list, Expiration time>. The overall algorithm of this protocol is that a source node will fill networks with packets of RREQ. When the first copy of the RREQ reaches its destination, the destination sends back a reverse route using RREP. This will establish the shortest path between the requester and destination. In details, when a source node needs to send a packet to a destination, it first checks its routing table and if the path toward the destination is out of date, or there is no path toward the destination, it would broadcast RREQ to all nodes in the network. For guarantying loop-less routing, each intermediate node receiving an RREQ would first judge whether it is the source node or if such an RREQ is repeated or not. If yes, this RREQ would be dropped, if not, the RREQ would be processed and broadcasted again. This will add one unit to Hop count. In processing the RREQ, an intermediate node first checks if a corresponding reverse route exists in its routing table, if not, the node would create an entry for a reverse route. If there is

already a reverse route, the intermediate node checks the content of this entry. If the destination sequence number in this entry is smaller than the source sequence number in the RREQ (a larger number means newer information), or if the two sequence numbers are the same, but the hop count recorded by the routing table is larger (smaller hop count means shorter path), the data in the entry would be replaced by the data in the RREQ. Then, if this intermediate node has a route toward the destination, and the route is not expired, the intermediate node would send RREP back to the source by the reverse route. However, if the intermediate node does not have a (forward) route to the destination, it will broadcast the RREQ to continue searching a route to the destination node. For route maintenance function in AODV, each mobile node would send Hello messages frequently, thus, each node knows its neighbors with one-hop distance. If one node has not received any Hello message from a neighboring node within a certain time, the node would send an RERR message to the nodes that are recorded in the corresponding precursor list in the routing table. The nodes receiving an RERR would remove the compromised route from their routing tables [10,11,12,13].

AODV does not use any special control and security mechanism; hence there is no solution for controlling suspicious behaviour such as, eavesdropping, dropping or changing the content of the packets. Some protocols such as SAR [14], SAODV [15] will secure AODV against some type of attacks, but this will be done with high operating cost and low efficiency.

4. Attacks in Mobile Ad-Hoc Networks

There are two different kinds of attacks in ad-hoc wireless networks based on their techniques:

Route Logic Compromise*:* In this technique, the intruder node injects incorrect routing messages to the entire network that causes disorder in network routing and corrupts the process of sending packets from source to destinations. Black-Hole and Flooding attacks use this technique as an example. In Black-Hole attacks the intruder node sends a propitious RREP to any received RREQ disregarding its routing table, therefore this node's RREPs are sent a lot sooner than the other node's, and also received faster. Other nodes identify the intruder node as a proper route and start sending their packets through this node. Next, all the received packets and the RREQs will be destroyed by the intruder node [16]. Also in flooding attack, the intruder node sends a flood of route requests in the network using all the bandwidth and finally preventing the network from working properly and disturbing the process of packet sending for the other nodes in the network. [17]

Packet Distortion: Usually this technique does not cause any disorder in nodes routing process, but the intruder node intercepts the network traffic and eventually tries to destroy the packets sent by the other nodes in the network. An example of this is in packet dropping attacks, the intruder node tries to destroy or change the packets check sum and also destroys all the PREQ, RREP and REEE packets according to their frequency or their type. There are three different types of packet dropping. In random dropping, the intruder node destroys packets in random frequency and in constant

dropping, all of the sent packets are destroyed. Finally, in periodic dropping the intruder node destroys the packets in a period to avoid being identified by the IDS. [9]

In addition, there are some attacks that have overlap with these kinds. In sink-hole attack, the intruder node tempts to gather the network traffic for example. Later, it alters or drops the traffic silently, causing disorder in the network. A sink-hole node makes use of a sequence number to create bogus RREQs. It observes the target node's sequence number carefully from the RREQs of target node and generates a bogus RREQ whose source is the target node with a higher sequence number than the sequence number of the target node [18].

5. Proposed Architecture

In a wireless environment, each node should have its own IDS agent to perform intrusion detection, since it cannot rely on other nodes' IDS that may leave the network at any time. Ad-hoc networks also do not have traffic concentration points allowing intrusion detection at a centralized location. In other words, it emphasizes the need for each node to have its own intrusion detection module [19]. On the other hand, the dynamic and cooperative nature of ad-hoc networks suggests that the designed IDS should be in a dynamic and cooperative fashion. There can be numerous states in which, nodes cannot investigate that the network is under intrusion or not by the use of their own IDS. In these cases cooperation between nodes should occur to accomplish intrusion detection process. So in our architecture which is a distributed and cooperative architecture the IDS Agent is active in all nodes and with observation of its audit data and all its neighbors audit data accomplish intrusion detection process. A conceptual model for this architecture was shown in figure 1. As shown in figure 1 our proposed architecture composed of three main modules that were described as below:

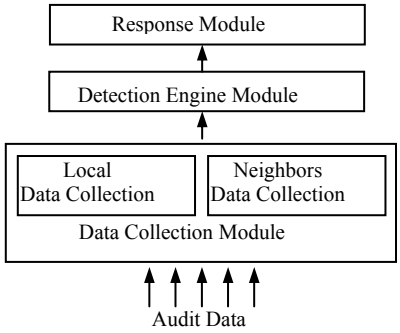

Figure 1. a conceptual model for proposed architecture

- ***Data Collection Module:*** The main function of this module is gathering nodes' movements and behavioral information and then sending them to detection engine. This can be obtained via information from various network layers. Required information from current node and the neighbor's nodes which are in its radio range with one-hop distance should be gathered. Therefore this module was composed of Local data collection for gathering current node information and Neighbors data collection for gathering neighbor data. For this purpose, first, each node should know its neighbors which are in its radio range. Therefore, for a sample $node_A$, it generates and sends a HELLO MESSAGE to all the other nodes. Each node which sends ACK

to this message in a fixed predetermined time will be node_A's neighbor. For each node a neighbor list was assumed which will be updated after a period of time. For a sample node_A, assuming *n* features for gathering required information with *m* neighbors, its feature set can be formulated as follow:

$$f_A = f_1, f_2, \ldots\ldots f_n, f_{1_1}, f_{1_2}, \ldots f_{1_n}, \ldots\ldots\ldots f_{m_n}$$

It should be mentioned that each f_i is the specified feature for node_A and each f_{I_J} is J[th] feature for one of the neighbors of node_A as I.

- ***Detection Engine Module:*** This module has the responsibility of detecting intrusion. anomaly detection technique for each IDS detection engine which uses neural-fuzzy interface was used in this research. For this purpose first the ANFIS (Adaptive Neural-Fuzzy Interface System) should be trained [20] with a labeled learning data, and then should be applied on an unlabeled data gathered from current network activities to predict its states as normal or abnormal. Denoting each state as *i*, our detection function was $\varphi(i): \text{R}^\text{n} \rightarrow \text{C}_\text{i}$ which $\text{R}^\text{n} = \{\lambda_0, \lambda_1, \lambda_2, \ldots\ldots\ldots, \lambda_\text{n},\}$ was the feature set and $\text{C}_{\text{i=\{normal,abnormal\}}}$ was the output class. As said before the feature set value for each node will be gathered from its own and its neighbors audit data, so each nodes should has its neighbors profile too. As shown in figure 2 the detection engine module in node 6 should have node1,2,3,4,5,7 profiles and this nodes have node6 profile too.

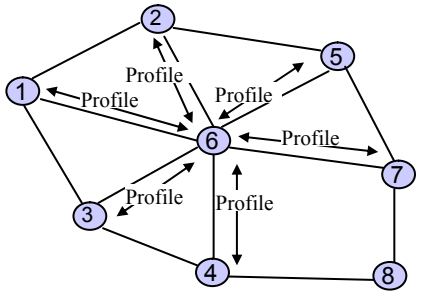

Figure 2. example of profile distribution in proposed architecture

- **Response Module**: This module had the responsibility of repercussion to detected attack in the network. Once a node detects an attack on the network, propagates re-authentication message through network or universal announce was done in this module.

5.1 Neural-Fuzzy Interfaces

Fuzzy logic is an appropriate tool for non-deterministic modeling. This method is used in mobile ad-hoc networks for securing and routing purposes [21]. Using fuzzy logic for intrusion detection in networks takes place for two reasons. First, there are no distinct boundaries between normal and abnormal states. Second, the security term is a fuzzy logic in nature [22]. Each member of fuzzy set was shown by a membership function which has a value between 0 and 1.Zero shows that the value is not a member of the set, and 1 shows the full dependency to the set. Fuzzy systems use fuzzy rules for classifying. A fuzzy rule has a template like if , antecedent then consequent. Fuzzy rules

can address a set of available data to its favorite class. Variety in if-then-else rules and its membership functions is highly related to the knowledge that from the system. On the other hand, there is no systematic way for converting human knowledge to fuzzy system knowledge. This problem would be solved by combining neural network learning ability with logic operation of fuzzy systems and creating a neural-fuzzy system.

ANFIS was used for this purpose. Figure 3 showed a precise structure of our used ANFIS architecture. As shown in this figure, an ANFIS structure was composed of five layers. Layers 1, 2, and 3 make the antecedent part of fuzzy system rules; and layer 4 makes the consequent part of it. Below each of the layers functions was described:

First layer: Nodes in this layer have the responsibility of receipting input layer as shown in equation 1:

$$f_i^1(x) = x \tag{1}$$

Second layer: The nodes in this layer calculate the membership function. The membership function that was used for each ANFIS node was a triangle function which was shown in Figure 5 and was defined as follow in equation 2:

$$f_i^2(x) = \mu_{Aj}(x) \tag{2}$$

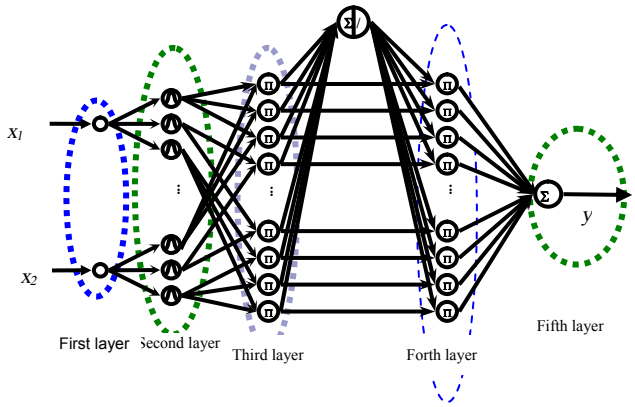

Figure 3. a precise structure of our used ANFIS architecture

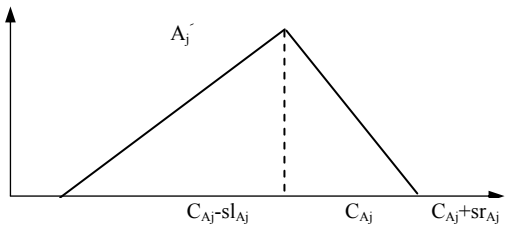

Figure 4. used triangle function

$$\mu_{Aj}(x) = \begin{cases} 1 - \dfrac{x_i - c_{Aj}}{sr_{Aj}}, & x_i \in [c_{Aj}, c_{Aj} + sr_{Aj}] \\ 1 - \dfrac{x_i - c_{Aj}}{sl_{Aj}}, & x_i \in [c_{Aj} - sl_{Aj,} c_{Aj}] \\ 0 & 0.W \end{cases} \quad (3)$$

Third layer: This layer is called rules layer. The nodes in this layer were assigned for creating antecedent part of the rules. Output function of each node was computed as below in equation 4:

$$f_i^3(x_1, x_2, \dots x_p) = \prod_{i=1}^{p} (x_i) \quad (4)$$

Fourth layer: In this layer, which is named Consequent layer, the output of each node was computed as below in equation 5:

$$f_i^4 = \frac{A_i^4}{\sum_{i=1}^{l} A_i^4} \quad (5)$$

Fifth layer: This layer has the responsibility of defuzzification of the output. As shown in Figure 5 and grasped from these equations, defuzzification is performed by calculating the center of gravity method as in equations 6, 7 and 8:

$$f_i^5(x_1, x_2, \dots x_t) = \frac{\sum_{i=1}^{t} centroid(B_i, x_i)\ Area(B_i, x_i)}{\sum_{i=1}^{t} Area(B_i, x_i)} \quad (6)$$

Which

$$Area(B_i, x_i) = \sum_{i=1}^{n} \min\ (\mu_{B_i}(y_i), x_i) \quad (7)$$

$$Area(B_i, x_i) = \frac{\sum_{i=1}^{n} y_i \min\ (\mu_{B_i}(y_i), xy_i)}{\sum_{i=1}^{n} \min\ (\mu_{B_i}(y_i), xy_i)} \quad (8)$$

This interface first was used in learning phase to produce network normal profile and then in testing phase for estimating current state as normal or abnormal.

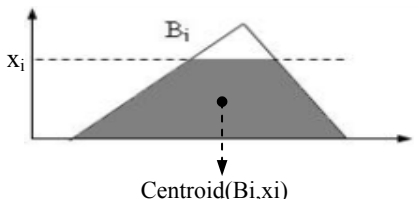

Figure 5. Defuzzification done by the center of gravity method

6. Implementation and Evaluation

Our implementation model had 30 nodes that move in an area with 850 m^2 ×1000 dimension under AODV routing protocol and random way point model with a communication link as 2 Mbps capacity. At the start of simulation each node by selecting a random destination, moved toward it by a speed that range from 0 to a maximum value. Then by stopping there at a specific pause time, selected another

destination and moved toward it. After implementing this scenario, the data from selected features of the node should be gathered. Selected features should have high information gain and have the ability to discriminate between normal and abnormal class. Four different subsets for node's features in ad-hoc networks were identified:

1- Features related to network routing packet and data packet propagation.

2- Feature related to rate of change in node's route table i.e. percentage of change in node distance (PDIST); percentage of change in route entries (PCR); percentage of change in traffic (PSTC); percentage of change in number of hops (PCH);percentage of change in bad routes (PCB); percentage of change in updated routes (PCU); percentage of change in stale routes(PCS) (routing table is a good reference for networks activities and its rate of change shows the network moving topology and node's relationship).

3- Physical and dynamic node's feature.

4- Feature related to other nodes in the network.

Plenty of instant from each subset were selected. Next, good instants from each subset should be selected with high information gain and low entropy. For this purpose Weka feature selector was selected. Finally, below features were selected for our implementation.

Velocity: indicates nodes speed.

Network Allocation Vector (NAV): this feature shows the network medium occupancy measure for transforming a message.

Transmission Traffic Rate: This message represents outgoing routing packet rate per node.

Reception Traffic Rate: This feature shows incoming routing packet rate in each node.

Retransmission Rates of DATA Packets: This feature reveals the number of data packet retransmitted in a node.

Active Neighbor Node Count: denotes the number of active neighbors that communicate in routing process.

PCR (Percentage Of Change in Route entry): This feature exhibits percentage of change in the route entry of each node that is calculated by equation 9:

$$PCR = \frac{|S_2 - S_1| + |S_1 - S_2|}{S_1} \quad (9)$$

$|S_2 - S_1|$ shows newly added route entry to a specific route table in $(t_2\text{-}t_1)$ time period and $|S_1 - S_2|$ shows the deleted route entry in this period.

PCH (Percentage Of Change in Hop count): This feature points to the percentage of change in all route entry the node route entry that calculated by equation 10:

$$PCH = \frac{|H_2 - H_1|}{|H_1|} \quad (10)$$

$|H_2 - H_1|$ shows changed hop-count in $(t_2\text{-}t_1)$ period.

14 pairs of CBR were used as all nodes in the network involved for communication between each other. For modeling short term profile of network until long term profile each node was logged for 2,5,30,50 and 180 seconds interval after 100 seconds from start of simulation (for preventing casual and incongruous data to our dataset that might cause disturbance for producing normal network profile). A sample vector of learning dataset is like < $feature_1$,…… $feature_n$, *Sampling interval* >. Then the acquired data to our Adaptive Neural-Fuzzy System Interface System (ANFIS) should be rendered. For this reason this algorithm was implemented in MATLAB environment. In this implementation, the membership function was set (MFS) to 5 and early stopping technique was used for solving over-fitting problem. The attacker node type was used as [23]. The insider adversary in this attack is allowed to do anything that a legitimate network node can do. It takes part in the ongoing transmission, drops the legitimate packets that it receives, modifies the legitimate packets before it forwards them to the next hop or tries to reveal the message sent from the source to the destination. This scenario was logged every 10 seconds. The evaluation was done based on detection accuracy and detection overhead. In detection accuracy percentage of Detection Rate (DR) and False Positive Ratio (FPR) versus number of Attacker node was used for our evaluation metrics. Figures 6 and 7 showed the results versus nodes speed. With the increasing of the nodes speed nodes detection rate was decreased and their false positive rate was increased. But in the worst case it was not lower than 76% for detection rate and upper than 2.5% for false alarm. In overhead case the architecture overhead and traditional stand-alone architecture were compared by calculating sum of transmitted packet byte which was shown in figure 8. As seen, this value is not greater than 1.7%

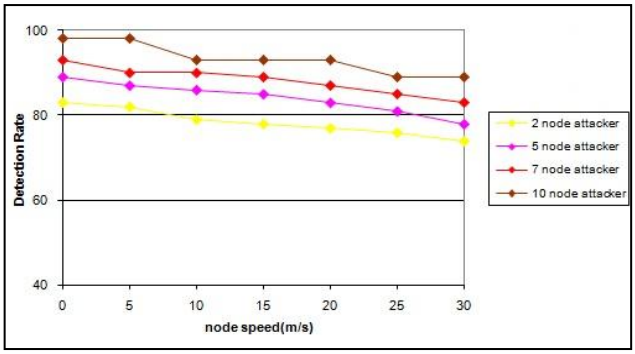

Figure 6. Detection Rate versus node speed for different numbers of intruder nodes.

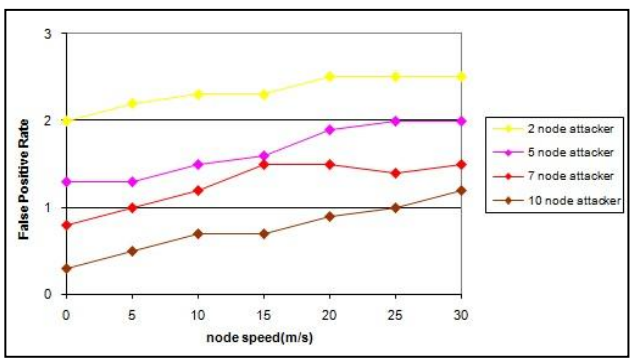

Figure 7. False Positive Rate versus node speed for different numbers of intruder nodes

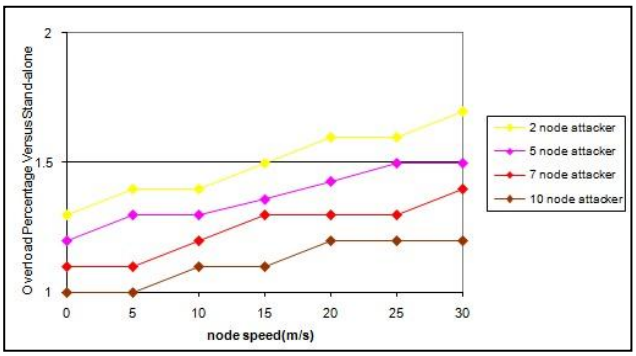

Figure 8. Overload Percentage by traditional stand-alone architecture versus node speed for different numbers of intruder nodes.

7. Conclusions

In this research, a distributed architecture based on anomaly detection technique using neural-fuzzy interface as its detection engine was used. The architecture composed of three modules as data collection module, detection module and response module. The first module was responsible for collecting required feature's data from its audit data and its neighbors which is in its one-hop communication. The detection module was responsible for detection intrusion based on its own and its neighbor's profile. Anomaly detection was used in this module with ANFIS as its detection engine. In response module what was done, was propagating network alarm. Then, a comparison was done on the results based on two aspects. First was detection accuracy which Detection Rate and False Positive Rate were used on it. The next was detection overhead. The proposed architecture overhead was compared with traditional stand-alone which no communication occurred between nodes. Totally, results showed that our architecture had a good accuracy with low overhead.

References

1. Stefan K. Stafrace, Nick Antonopoulos,"Military tactics in agent-based sinkhole attack detection for wireless ad hoc networks" Computer Communications 33 (2010) 619–638
2. Makki S & Pissinou N & Huang H. (2004). "The Security issues in the ad-hoc on demand distance vector routing protocol (AODV)", In Proceedings of the 2004 International Conference on Security and Management (SAM'04), p 427-432.
3. Komninos N & Vergados D & Douligeris C. (2007). "Detecting Unauthorized and Compromised Nodes in Mobile Ad-Hoc Networks", Journal in Ad Hoc Networks, Elsevier Press, 5(3), p 289-298.
4. C. Xenakis, C. Panos, and I. Stavrakakis, "A comparative evaluation of intrusion detection architectures for mobile ad hoc networks" journal of computers & securi ty 30 (2011) 63 e80
5. Zhang Y & Lee W & Huang Y. (2003). "Intrusion Detection Techniques for Mobile Wireless Networks", ACM WINET 2003.
6. Yi-an Huang & Wenke Lee. (2003). "A cooperative intrusion detection system for ad hoc networks", Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, October 31, Fairfax, Virginia.
7. Deng H & Zeng Q & Agrawal D.P. (2003). "SVM-based Intrusion Detection System for Wireless Ad Hoc Networks", In: Proceedings of the IEEE Vehicular Technology Conference (VTC03), p 2147-2151.

8. Liu Y & Li Y & Man H. (2005). "MAC Layer Anomaly Detection in Ad Hoc Networks", In: Proceedings of 6th IEEE Information Assurance Workshop, West Point, New York.
9. Djenouri D & Mahmoudi O & Bouamama M & Llewellyn-Jones D & Merabti M. (2007). "On SecuringMANET Routing Protocol Against Control Packet Dropping", In: Proceedings of IEEE International conference on Pervasive Services (ICPS' 07), Istanbul, Turkey, p 100-108.
10. Hu Y & Perrig A & Ariadne Johnson. (2002). "A secure on-demand routing protocol for ad hoc networks", In Proceedings of the Eighth Annual International Conference on Mobile Computing andNetworking (MobiCom 2002), September.
11. Papadimitratos P & Hass Z. (2002). "Secure routing for mobile ad hoc networks", in SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS), San Antonio.
12. Ming-Yang Su. (2010). "A wormhole-avoidance routing protocol by anomaly detection in mobile ad hoc networks", computer&security29 journal(2010), p 208-224.
13. Charles E. Perkins & Elizabeth M. Belding-Royer & Samir Das. (2003). "Ad Hoc On Demand Distance Vector (AODV) Routing", IETF RFC 3561.
14. S. Yi & P. Naldurg & R. Kravets. (2001). "Security Aware Ad Hoc Routing for Wireless Networks", In Proceedings of the 2nd ACMSymposium on Mobile Ad Hoc Networking and omputing,Poster Session, , Long Beach, California, p 299-302.
15. M. G. Zapata & N. Asokan. (2002). "Securing Ad-Hoc Routing Protocols", In Proceedings of the 2002 ACM Workshop on Wireless Security, p 1–10, Atlanta, GA.
16. Shurman M.AI & Yoo S.M. & Park S. (2004). "Black Hole Attack in Wireless Ad Hoc Networks", In:Proceedings of ACM 42nd Southeast Conference (ACMSE 04), p 96-97.
17. Yi P & Hou Y.F & Zhong Y & Zhang S & Dai Z. (2005) "Flooding Attack and Defence in Ad hoc Networks", In: Systems Engineering and Electronics, 17(2), p. 410-416.
18. Gisung Kim & YounggooHan & SehunKim. (2010). "A cooperative-sinkhole detection method for rmobile ad hoc networks", in Int. J. Electron. Commun. (AEÜ), 64, p 390–397.
19. N.Komninos & C.Douligeris. (2009). "LIDF: Layered intrusion detection framework for ad-hoc networks", Ad Hoc Networks 7 journal. p 171–181.
20. Jang R. & S J. (1993). "ANFIS: Adaptive-Network-based Fuzzy Inference Systems", IEEE Transactions on Systems, Man, and Cybernetics, 23(3), p 665-685.
21. Jing N, Jiangchua W, Ji L, Xin H, Zheng Z,"An adaptive fuzzy logic based secure routing porotocol in mobile ad hoc network", in *J. Nie et al. / Fuzzy Sets and Systems 157 (2006) 1704 – 1712*
22. Bridges, Susan M & Rayford M. Vaughn. (2000). "Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection", Proceedings of the Twenty-third National Information Systems Security Conference, Baltimore, MD.
23. E.Ayday, F.Fekri, " A protocol for data availability in Mobile Ad-Hoc Networks in the presence of insider attacks", journal of Ad Hoc Networks 8 (2010) 181–192