Bulletin of the *Transilvania* University of Braşov • Vol 9(58), No. 2 - 2016
Series III: Mathematics, Informatics, Physics, 125-136

A SECURE FRAMEWORK FOR BEEADHOC (BIO/NATURE INSPIRED MANET ROUTING PROTOCOL) USING NEURAL NETWORKS

Hamideh FATEMIDOKHT[1] **and Marjan KUCHAKI RAFSANJANI**[*2]

Abstract

Mobile ad hoc network (MANET) is a group of mobile nodes that are formed dynamically and communicate with each other via wireless links. These networks are characterized by dynamic topology, the absence of central authorities, distributed cooperation and constrained capability. Thus, they are vulnerable to a number of security threats. These vulnerabilities create significant challenges for routing protocols. Bio/Nature-inspired routing algorithms such as BeeAdHoc have been offered for developing routing algorithms for MANETs. But a malicious node can seriously disrupt the routing behavior of this protocol. In this paper, we try to build a security protocol for BeeAdHoc which utilizes Back Propagation Network (BPN). Back-propagation neural network is used for the purpose of anomaly detection. For simulating, we use TrueTime toolbox of MATLAB. The results of our experiments show that our proposed protocol is able to counter the different types of threats.

2010 *Mathematics Subject Classification:* 92B20, 68M10, 90B18.
Key words: Mobile Ad Hoc Networks (MANETs), BeeAdHoc, neural networks, security.

1 Introduction

In mobile ad hoc networks, all nodes are mobile and are connected via wireless links without using a fixed infrastructure and they are established in any environment which is needed [1]. In this network each node has a limited transmission range. Two nodes can communicate directly in the transmission range of each other. Otherwise, the communication between them has to rely on the other nodes. In MANETs because each node is able to move independently in any direction, the network topology can change frequently [2].

[1]Department of Applied Mathematics, Faculty of Mathematics and Computer, *Shahid Bahonar* University of Kerman, Kerman, Iran, e-mail: h.fatemidokht@math.uk.ac.ir
[2]**Corresponding author*, Department of Computer Science, Faculty of Mathematics and Computer, *Shahid Bahonar* University of Kerman, Kerman, Iran, e-mail: kuchaki@uk.ac.ir

Routing for mobile ad hoc networks is a popular research topic. Secure routing is one of the important issues for MANETs. These networks are characterized by dynamic topology, limited bandwidth and limited energy of their nodes. Bio/Nature-inspired routing algorithms such as BeeAdHoc have been presented to develop routing algorithms for MANETs [3]. Unfortunately BeeAdHoc is very vulnerable and a malicious node can disrupt the routing behavior of this protocol.

In this paper, we present a secure protocol for BeeAdHoc that uses Back Propagation Network (BPN). In recent years artificial neural network has been accepted as an efficient tool for modeling complex systems and it has been used for prediction [4]. Back Propagation Neural Network (BPN) model is a supervised learning model whose input vectors and the corresponding target vectors are used in training the network. It is a feed-forward multi-layer network that can effectively classify many types of data. BPN has many possible training algorithms such as gradient descent, momentum and resilient algorithm [5].

The remainder of the paper is organized as follows; In section 2, BeeAdHoc protocol and its security vulnerabilities are investigated. We explain the proposed algorithm in section 3. In section 4, the different types of routing attacks launched by malicious nodes are explained and the routing behavior of BeeAdHoc and BeeAdHoc by BPN is shown and finally the conclusion is in section 5.

2 The analysis of attacks in BeeAdHoc protocol

BeeAdHoc is a reactive source routing algorithm with effective energy for routing in MANETs, which has been inspired from bee behaviors. It uses scout agents to discover new routes and forager agents to transport data from source to destination. When a node is required to send data to a particular destination, the forward scout broadcasts on the network. The intermediate nodes that receive the scout, append their addresses in the source route of the scout until it arrives at the destination. When a forward scout reached the destination, a backward scout is sent back to the source node using link reversal. Once a scout returns to its source node, it advertises the route to other foragers and then foragers transport data to the destination node. On their journey, they collect the information about the network state and evaluate the quality of the traversed path. It was a brief description of the BeeAdHoc protocol; an interested reader can find its complete description in [3].

The security threat analysis of BeeAdHoc in [6] has shown that a malicious node could launch a number of Byzantine attacks which disrupt the normal routing behavior. The attacker will modify the source route in a scout. Also it can forge a scout by spoofing the source address or inserting fake scout ID, or both. Table 1 is the field BeeAdHoc route protocol, which is liable to be attacked.

Table 1: Vulnerable fields in BeeAdHoc packets.

Field	**Modifications**
Type	Modify the packet's type.
Hopcount	When node forward ForwardScout or BackwardScout packet to route's next node, hop count will decide the shortest route path. When attack node sends ForwardScout or BackwardScout packet, it will make Hop count decrease which makes when the destination node (or source node) chooses a single path, it will choose the route with the problem.
Src	Attack node will replace an invalid source IP address.
Dest	Attack node will replace an invalid destination IP address.
Source_route	Attack node can modify the source route in a scout.
Scout ID	Attack node can forge a scout by inserting fake scout ID.

3 The proposed approach

In this section, we explain the detailed treatment of our proposed approach. It contains three phases: picking detection node, BPN local detection and scout authentication.

3.1 Picking detection node

In a MANET, the detection of the information of the neighbor node by every node and the repeated detection of some packets will cause the worst overhead and the waste of resource. In this paper according to the mechanism provided by [7] for intrusion detection, the picking detection node performs the following steps:

1. Calculate the number of connections from self to the neighbor node (Figure 1 (b)).
2. Calculate the radio range self can communicate. It is shown as Figure 1 (c).
3. Compare the total influence on neighbor node, when it is bigger than itself, it will be recommended as detection node. It is shown as Figure 1 (d).

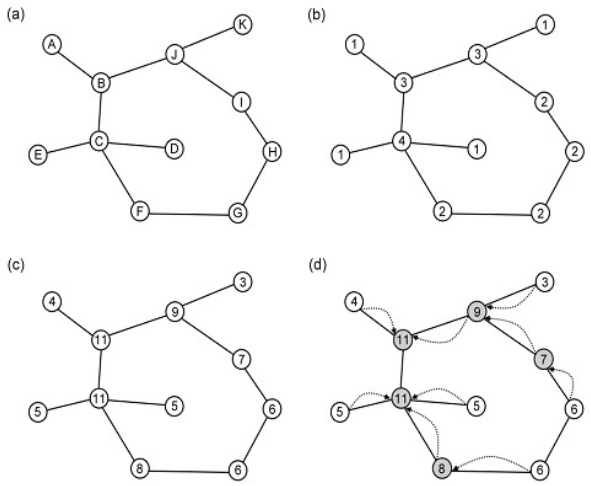


Figure 1: The process of picking detection node [7].

3.2 BPN local detection

This approach for anomaly detection used six items of ForwardScout and BackwardScout packet fields which are liable to be attacked. They are: (1) the types of packet, (2) the hopcount, (3) the source IP address, (4) the destination IP address, (5) the Source_route, (6) scout ID.

Because neural algorithm cannot handle symbol field, it must switch from symbol to numeric. Furthermore, values of features can lie within different dynamic ranges. Thus, features with large values may have a larger influence on the cost function than features with small values [8]. So the features must be normalized to make the number between [0, 1]. For normalizing the features, we use the following equation [7]:

$$e_{normalize} = \frac{e_i - Min(e)}{Max(e) - Min(e)} \tag{1}$$

Where e_i is the one in the present packet field, Min(e) and Max(e) are minimum and maximum of packet field, respectively.

Our proposed approach uses BPN three layer network structures. This is illustrated in figure 2.

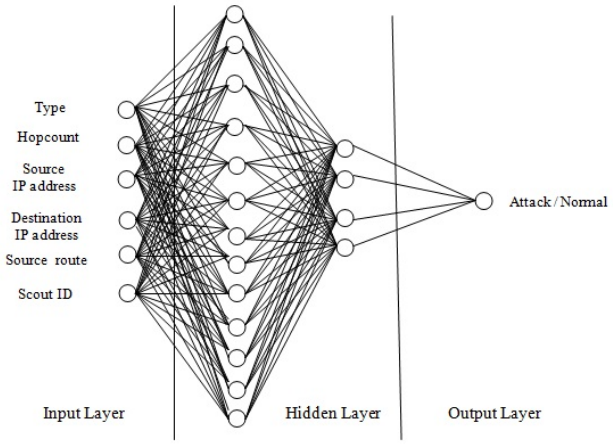


Figure 2: Three layer structure of BPN.

In table 2 all the parameter values of BPN in this approach are demonstrated.

During training, when the IDS is just recently online, weights and biases are initialized by random number. The calculation of the training error of the hidden layer and the output layer for mapping weight continues until the error has been reduced to an acceptable value. Back Propagation Network uses the gradient steepest descend method to attack prediction.

3.3 Scout Authentication

When a source node has data to send to the destination, if there is no route in dance floor, it broadcasts a forward scout to all its neighbors for discovering new routes. This forward scout contains source ID, destination ID, source route and hopcount. When a node receives a forward scout, it can predict an attack by a malicious node with the help of BPN. When a forward scout reached the destination, the destination node unicasts the backward scout back to the source

Table 2: BPN parameters.

Parameters	**Values**
Training function	Trainoss
Input layer	Input neoron: 6
Hidden layer	1. Number of layer: 2 2. First hidden layer neuron: 13 3. Secondly hidden layer neuron: 4 4. Transfer function: Sigmoid
Output layer	1. Output neuron: 1 2. Transfer function: Sigmoid
Learning epoch number	400
Learning efficiency	0.1
Mean squared error	0.01-0.005

node. Once the backward scout is received by the source node, it can verify that the backward scout has not tampered by a malicious node by using BPN. Then it recruits the foragers for transport data to the destination node.

4 Simulation and results

In this section, we investigate the impact of attacks on BeeAdHoc and BeeAdHoc by BPN protocols. The simulation is carried out by using TrueTime toolbox of MATLAB. Table 3 displays simulation parameters. We investigate the details of three attacks launched by malicious nodes. We use node topology shown in Figure 3, which is first used for BeeSec [6].

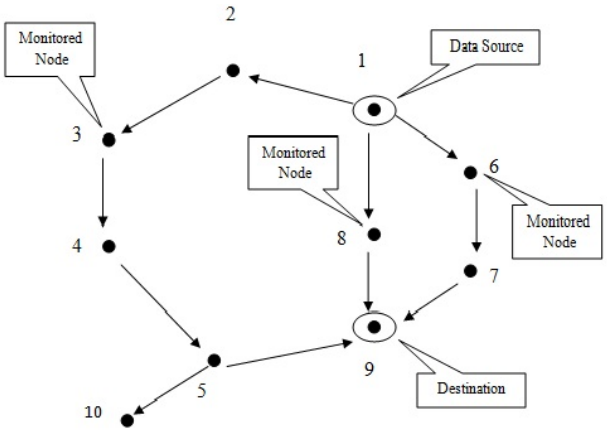


Figure 3: Node Topology selected for attacks.

Attack-1: Forging Forward Scout: This attack is launched 100 seconds after the start of the simulation, when initial route discovery is complete. The attacker node 5 launches forged forward scouts after 100 s of injection of data into network to install a forged route 1-2-3-4-5-9. Figure 4(a) and 4(b) show the impact of this attack on BeeAdHoc and BeeAdHoc by BPN respectively.

Attack-2: Forging Backward Scout: The attack involving spoofed backward scouts is launched by Node 3 at time t = 100 s consequently, Node 3 was successfully able to divert subsequent data packets toward itself on the least suboptimal path 1-2-3-4-5-9. Figure 5(a) and 5(b) show behavior of BeeAdHoc and BeeAdHoc by BPN respectively.

Attack-3: Returning Scout with a Suboptimal Route: For each received scout, malicious node 6 changed the source route to 9-5-4-3-2-1 and instead of broadcasting it further, sent it back as a unicast message. As a result the longer path 1-2-3-4-5-9 got established instead of the desired path 1-6-7-9. The impact of this attack on desired protocols is shown in figure 6.

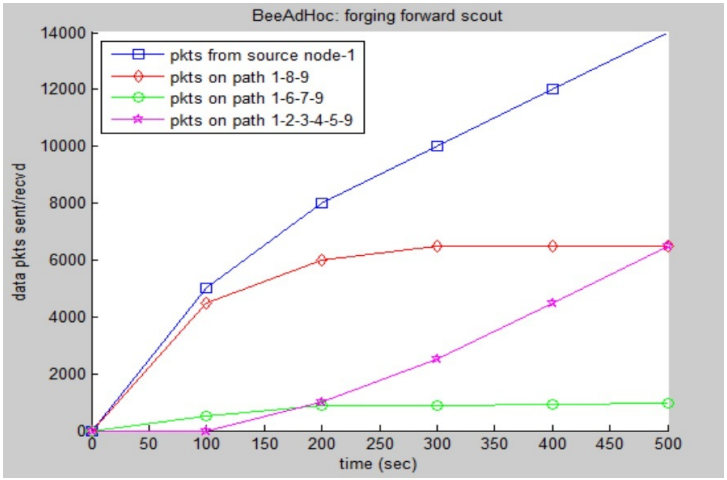
Table 4 is the number of datasets; it includs BeeAdHoc's normal behavior, Forging Forward Scout, Forging Backward Scout and Returning Scout with a Suboptimal Route.

Table 3: Simulation parameters.

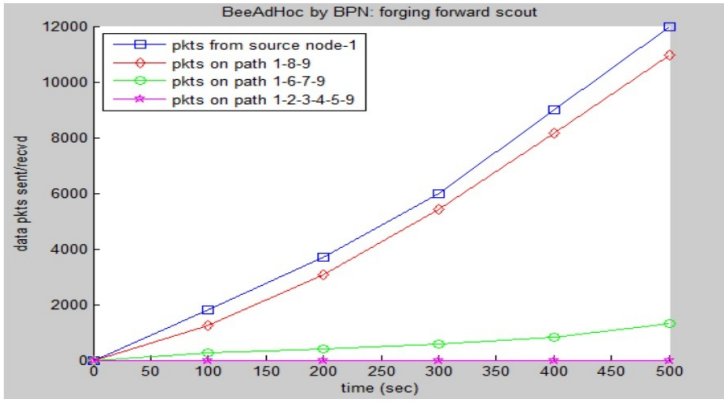
Parameters	**Values**
Routing protocol	BeeAdHoc
Simulation area	$100 * 100m^2$
Number of mobile hosts	10
Number of malicious node	3
Speed of nodes	1 m/sec
Movement model	Random waypoint
Traffic type	CBR
Packet size	512 bytes
Transmission rang	12.5

Table 4: Dataset on the number of samples.

Data type	**Training**	**Test**
Normal sample	100	100
Forging Forward Scout	40	40
Forging Backward Scout	40	40
Returning Scout with a Sub-optimal Route	40	40
Total sample	220	220



(a) BeeAdHoc



(b) BeeAdHoc by BPN

Figure 4: Forager forward scout.

Detection performance of our proposed IDS is measured as follows [7]:

True Positives (TP): It represents that the system detects a normal behavior. In fact, it is a normal behavior.

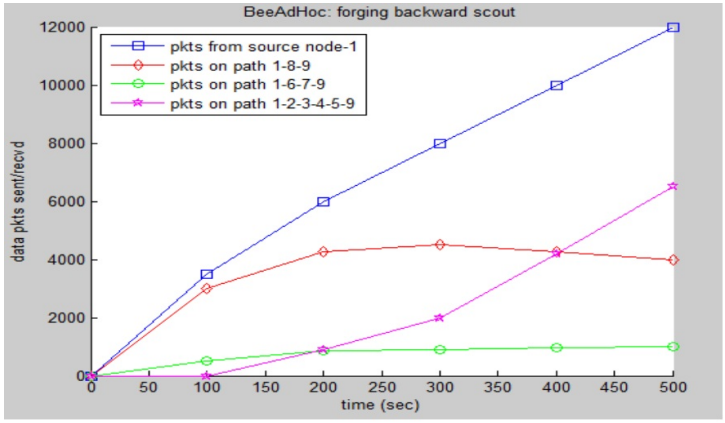
$$True\ Positive = \frac{TP}{TP + FN} \tag{2}$$

True Negatives (TN): It represents that the system detects an attack behavior. In fact, it is an attack behavior.

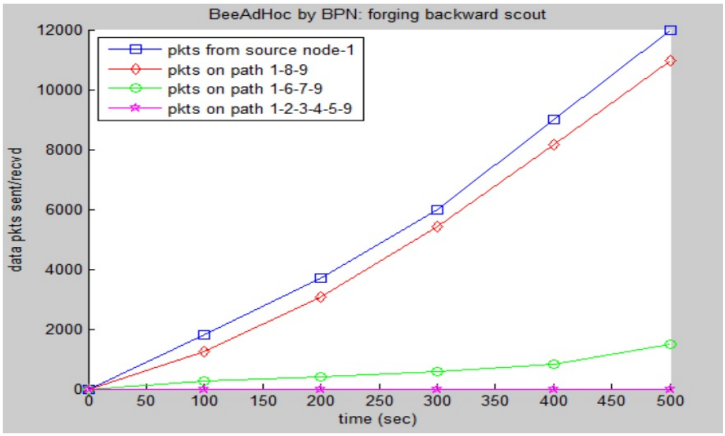
$$True\ Negative = \frac{TN}{TN + FP} \tag{3}$$

False Positive (FP): It represents that the system detects an attack behavior. In fact, it is not an attack.

$$False\ Positive = \frac{FP}{FP + TN} \tag{4}$$



(a) BeeAdHoc



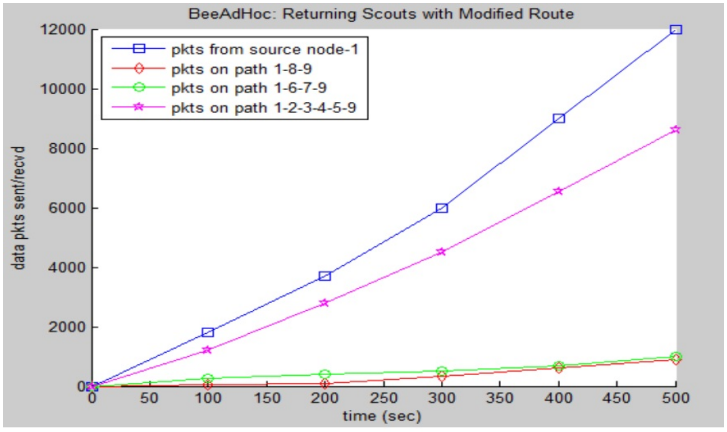
(b) BeeAdHoc by BPN

Figure 5: Forager backward scout.

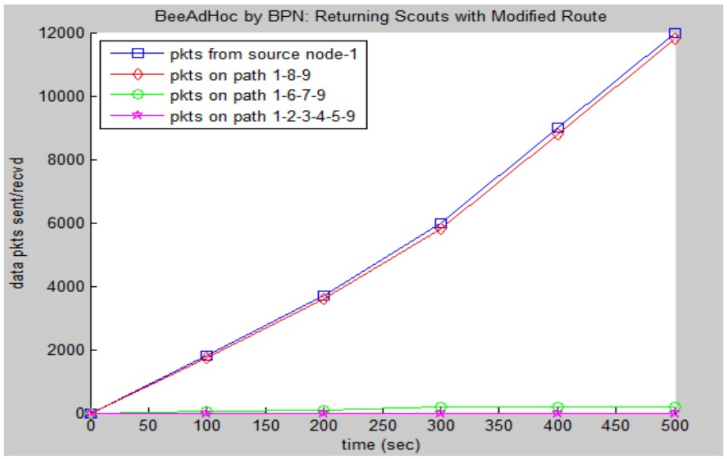
False Negative (FN): It represents that the system detects a normal behavior, but in fact, it is an attack behavior.

$$False\ Negative = \frac{FN}{FN + Tp} \tag{5}$$

In table 5 experimental results for TP, TN, FP and FN averaged over 5 trials are presented.



(a) BeeAdHoc



(b) BeeAdHoc by BPN

Figure 6: Returning Scout with a Suboptimal Route.

5 Conclusion

In this paper, we investigated security vulnerabilities of BeeAdHoc. In this protocol a malicious node is able to launch a number of attacks and disrupt normal behavior of the protocol. We proposed a secure protocol for BeeAdHoc which is designed using Back Propagation Network (BPN). We implemented BeeAdHoc by BPN in MATLAB. Experimental results show that BeeAdHoc by BPN was able to counter the attacks launched by a malicious node. We investigated the performance of our IDS by computing of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Simulation results show high efficiency in detecting internal attack.

Table 5: Experimental results.

TP	TN	FP	FN
0.976	0.98	0.0194	0.0242

References

[1] Basagni, S., Conti, M., Giordano, S., and Stojmenovic, I., *Mobile Ad Hoc Networking*, IEEE Press book, 2004.

[2] Abed, A.K., Oz, G., and Aybay, I., *Influence of mobility models on the performance of data dissemination and routing in wireless mobile ad hoc networks*, J. Computers and Electrical Engineering **40** (2014), 319-329.

[3] Wedde, H.F., Farooq, M., Pannenbaecker, T., and Vogel, B., *BeeAdHoc: An energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior*, Proc. of the ACM Genetic and Evolutionary Computation Conf. 153-160, 2005.

[4] Singh, J.P., Dutta, P., and Pal, A., *Delay Prediction in Mobile Ad Hoc Network using Artificial Neural Network*, Proc. of Technology. 201-206, 2012.

[5] Sangkatsanee, P., Wattanapongsakorn, N., and Charnsripinyo C., *Practical real-time intrusion detection using machine learning approaches*, J. Computer Communications. **34** (2011), 2227–2235.

[6] Mazhar, N., and Farooq, M., *Vulnerability analysis and security framework (BeeSec) for nature inspired MANET routing protocols*, Proc. of the ACM Genetic and Evolutionary Computation Conf. 102-109, 2007.

[7] Shao, M.H., Lin, J.B., and Lee, Y.P., *Cluster-based Cooperative Back Propagation Network Approach for Intrusion Detection in MANET*, Proc. of 10th IEEE International Conference on Computer and Information Technology. 1627-1632, 2010.

[8] Theodoridis, S., and Koutroumbas, K., *Pattern recognition*, Elsevier, 2009.