ADAPTIVE ORDERED WEIGHTED AVERAGING FOR ANOMALY DETECTION IN CLUSTER-BASED MOBILE AD HOC NETWORKS

M. RAHMANIMANESH AND S. JALILI

Abstract. In this paper, an anomaly detection method in cluster-based mobile ad hoc networks with ad hoc on demand distance vector (AODV) routing protocol is proposed. In the method, the required features for describing the normal behavior of AODV are defined via step by step analysis of AODV and independent of any attack. In order to learn the normal behavior of AODV, a fuzzy averaging method is used for combining one-class support vector machine (OCSVM), mixture of Gaussians (MoG), and self-organizing maps (SOM) one-class classifiers and the combined model is utilized to partially detect the attacks in cluster members. The votes of cluster members are periodically transmitted to the cluster head and final decision on attack detection is carried out in the cluster head. In the proposed method, an adaptive ordered weighted averaging (OWA) operator is used for aggregating the votes of cluster members in the cluster head. Since the network topology, traffic, and environmental conditions of a MANET as well as the number of nodes in each cluster dynamically change, the mere use of a fixed quantifier-based weight generation approach for OWA operator is not efficient. We propose a condition-based weight generation method for OWA operator in which the number of cluster members that participate in decision making may be varying in time and OWA weights are calculated periodically and dynamically based on the conditions of the network. Simulation results demonstrate the effectiveness of the proposed method in detecting rushing, RouteError fabrication, and wormhole attacks.

1. Introduction

Mobile ad hoc network (MANET) is an infrastructureless network contains a number of wireless nodes which form the network dynamically without any central administration. Ad hoc on demand distance vector (AODV) [56, 57] is a widely used routing protocol for MANETs that fully trusts all participants and has no security consideration, and as such is vulnerable to several attacks and misbehavior.

In this paper, an anomaly-based attack detection method in MANETs with AODV routing protocol is proposed. In the method, the required features for describing the behavior of AODV protocol are defined via step by step analysis of AODV and independent of any attack, where the total of 118 features are attained through this meticulous analysis. We also utilize and train three one-class classifiers

Received: November 2011; Revised: April 2012; Accepted: August 2012

Keywords and phrases: Ordered weighted averaging weight generation, Mobile ad hoc network, Anomaly detection.

namely, one-class support vector machine (OCSVM) [60], mixture of Gaussians (MoG) [8], and self-organizing maps (SOM) [34] and combine them together for learning the normal profile. In the proposed method, a fuzzy averaging method is used for combining one-class classifiers in which the classifiers and subsequently the network nodes are allowed to consider uncertainty in their decision making, which improves performance and robustness of the proposed method.

We assume that the mobile ad hoc network is cluster-based. Each node collects its own data in the network and there is no need for monitoring the behavior of a node by its neighbors. The votes of cluster members are periodically transmitted to the cluster head and final decision on attack detection is carried out in the cluster head. In doing so, an adaptive ordered weighted averaging (OWA) operator is used for aggregating the votes of cluster members in the cluster head. The most important issue in defining the OWA operator is how to obtain the associated weighting vector. Since the network topology, traffic, and environmental conditions of a MANET as well as the number of nodes in a cluster dynamically change, the mere use of a fixed pre-defined quantifier-based weight generation approach for OWA operator is not efficient. We propose a dynamic condition-based weight generation method in which the number of cluster members that participate in decision making may be varying in time and OWA weights are calculated periodically and dynamically based on the environmental conditions of the network. In the proposed method, the cluster members have fuzzy votes, meaning that they are allowed to retain their uncertainty in decision making and to defer the decision making until the votes to be aggregated in the cluster head.

The remainder of this paper is organized as follows. In Section 2, we provide preliminaries including a brief overview of AODV protocol and its vulnerabilities, one-class classification problems, group decision making, and clustering of MANETs. In Section 3, we survey the extensions of OWA operators, OWA weight generation approaches, applications of OWA operators, and related works on classifier-based anomaly detection in MANETs. In Section 4, we present our anomaly detection method and our method for protocol-based definition of features. Attack models and simulation results are shown in Section 5 and finally the paper is closed with our conclusions.

2. **Preliminaries**

In this section, we provide some preliminaries including an overview of AODV routing protocol, the attacks that we launched in the network to evaluate our method, the concept of one-class classification, group decision making, and clustering of mobile ad hoc networks.

2.1. **Overview of AODV Protocol.**

AODV [56, 57] is a reactive routing protocol for MANETs where each node maintains a routing table. As such, data packets do not have routes from their sources to their destinations. AODV uses two mechanisms, namely, route discovery and route maintenance.

The process of route discovery begins when the source node wants to send data to the destination and there is no valid route for that destination in its routing table. In this case, the source node broadcasts a RouteRequest packet in the network.

After receiving the RouteRequest packet, a node either sends a RouteReply packet to the source or rebroadcasts the received RouteRequest packet after increasing its hop count. Finally, the RouteRequest packet reaches a node (probably the destination itself) that knows a route to the destination. In this case, it will unicast the RouteReply packet to the node that it has received the RouteRequest packet from it. After receiving a RouteReply packet, each intermediate node will maintain in its routing table the address of the node that it has received the RouteReply packet from, and subsequently, it will forward the RouteReply packet so that it can reach the source node. As soon as the source node receives the first RouteReply packet, it can start transmitting data packets and subsequently, it can update its routing table when a RouteReply packet with a better route is received.

The process of route maintenance begins when a broken link is detected or the next node is inaccessible. In this case the node that detects this will send a RouteError packet to all its active neighbors for that destination, and they will also send the packet to their active neighbors for that destination and this action will continue.

2.2. **Attack Implementation.** In this paper, the following attacks are launched in the network to evaluate the proposed method:

1- Falsifying route errors (with the name of RERR Fab.): In this attack, a malicious node fakes some RouteError packets, which can lead to the destruction of the main route, leading to the denial of service.

2- Rushing [29]: In this attack, a malicious node tries to absorb network traffic by getting the RouteRequest packet to the destination faster.

3- Wormhole [30]: In this attack, same as the rushing attack, two malicious nodes try to absorb network traffic by sending the control packets through a hidden private link (called tunnel). The wormhole can drop data packets or forward some of them selectively.

2.3. **One-Class Classification.** In our anomaly detection problem at the time of describing the normal behavior of AODV protocol, there is no attack data and as a result, the training process should be done with the dataset containing the feature vectors belong to just one class (i.e., the target class). When there are such problems, in order to train, we have to use one-class classifiers [62] that can train specifications of one existing class. Tax [62] grouped one-class classifiers into three types, namely, the density methods (e.g. Gaussian model, mixture of Gaussian models, and Parzen density estimator), the boundary methods (e.g., k-centers, nearest neighbor, and one-class support vector machine), and the reconstruction methods (e.g., k-means, learning vector quantization, principal component analysis, and self-organizing maps).

In all one-class classification methods two distinct elements should be identified. The first one is a measure for the distance $d(x)$ or resemblance $p(x)$ of a feature vector x to one existing class (represented by the training dataset). The second one is a threshold θ on this distance or resemblance. For prediction, a new feature vector x is labeled as normal when the distance to the target class is smaller than the threshold ($d(x) < \theta$) or when the resemblance is larger than the threshold ($p(x) > \theta$) [62].

In this paper, to learn the normal behavior of AODV protocol, we use three diverse one-class classifiers, each of them is one of the above mentioned types:

1- One-class support vector machine (OCSVM) with RBF kernel [60]. The OCSVM maps the input data into a high dimensional feature space using kernel function and then finds the maximal margin hyper plane in the mapped space which best separates the training data from the origin.

2- Mixture of Gaussian models (MoG) with PPCA covariance type [8]. A mixture of Gaussians is a linear combination of normal distributions. When the number of Gaussians is known, the mean and covariance of the individual Gaussians can be estimated by an expectation maximization routine [62].

3- Self-organizing maps (SOM) [34]. The self-organizing maps as a type of artificial neural network produces a similarity graph of input data and converts the nonlinear statistical relationships between high dimensional data into simple geometric relationships of their image points on a two dimensional grid of nodes (called a map) [34].

2.4. **Group Decision Making.** A group (or multi-criteria) decision making process is defined as a decision situation in which there are two or more experts, each of them characterized by its own perception, attempt to reach a collective decision. The main problem in fuzzy logic-based group decision making is how to aggregate the experts' opinions to obtain a group decision in such a way that some criteria are satisfied [13, 15, 20, 12, 72].

In this paper, we deal with group decision making at two levels. The first one is in the combination of one-class classifiers in which the fuzzy averaging method is used and the second one is in the aggregation of the votes of cluster members in the cluster head in which the ordered weighted averaging is used.

A fuzzy subset A of a set X is characterized by assigning to each object x of X the degree of membership of x in A:

$$A = \{(x, \mu_A(x)) | x \in X\}, \tag{1}$$

where $\mu_A(x)$, $0 \le \mu_A(x) \le 1$, is the degree of membership of x in A. The closer the membership degree $\mu_A(x)$ is to 1, the more x belongs to A.

The step in which the objects in X are determined the degree to which they belong to each of the appropriate fuzzy sets is called fuzzification. Fuzzification is performed by using membership functions. A membership function, in general, is a curve that defines how each object in the input space is mapped to a membership value between 0 and 1. There are many membership functions such as triangular, trapezoidal, Gaussian, sigmoid, bell and so on [13]. Once the inputs are fuzzified, the fuzzy inference rules are applied for group decision making.

In this paper, the fuzzy averaging method is used for combining one-class classifiers (i.e., the experts). Let $x \in R^d$ be an object in the input space. Suppose there exists a finite set of alternatives $A = \{a_1, a_2, \ldots, a_K\}$ as well as a finite set of experts $E = \{e_1, e_2, \ldots, e_L\}$, and each expert $e_i \in E$ provides the degree of membership of x to each of alternatives a_j in A by membership function $\mu_{e_i,a_j}(x)$. Fuzzy averaging for aggregating the opinions of the experts is performed as [28]:

$$V_{a_j} = \frac{1}{L} \sum_{i=1}^{L} \mu_{e_i,a_j}(x), \quad j = 1, \ldots, K. \tag{2}$$

We also use OWA operator for aggregating the votes of cluster members in the cluster head. Yager [72] introduced OWA operator for multi-criteria decision making. Let $X = (x_1, x_2, \ldots, x_n)$ be an n-dimensional vector. An OWA operator of dimension n is a mapping $F : R^n \to R$, that has an associated n-dimensional weighting vector $W = (w_1, w_2, \ldots, w_n)$, such that $\forall i$, $w_i \in [0, 1]$, $\sum_{i=1}^{n} w_i = 1$, and

$$F(X) = \sum_{j=1}^{n} w_j y_j, \tag{3}$$

where, y_j is the j^{th} largest element of the bag $< x_1, x_2, \ldots, x_n >$.

OWA operator provides a parameterized family of aggregation operators between the minimum and the maximum. By selecting different weighting vectors, different aggregation operators can be implemented. If $w_1 = 1$ and $w_j = 0$ for $j \neq 1$, then the aggregated value obtained is $max\{x_i\}$. If $w_n = 1$ and $w_j = 0$ for $j \neq n$, then the aggregated value obtained is $min\{x_i\}$. If $\forall i, w_i = 1/n$, then the aggregated value is $avg\{x_i\}$.

2.5. **Clustering of Mobile Ad Hoc Networks.** In a clustering scheme, the mobile nodes in MANET are divided into different virtual groups in which the nodes that are geographically adjacent are allocated into the same cluster according to some rules. Yu and Chong [81] mentioned two benefits for MANET with cluster-based structure. First, the cluster structure facilitates the spatial reuse of resources to increase the system capacity. It can save much resources used for retransmission resulting from reduced transmission collisions. The second benefit is in routing, because the set of cluster heads can normally form a virtual backbone for inter-cluster routing, the generation of routing information can be restricted in this set of nodes. Chen et al. [14] utilized clustering of network to simplify the task of ad hoc network management in which the clustering causes it to be more fault tolerant and message efficient. Additionally, our experiments demonstrate that decision aggregation in cluster heads improves performance of the anomaly detection method and consequently the security of the networks.

Several methods have been proposed for clustering of MANETs [81, 14, 4, 64]. According to their objectives, these methods can be divided into several categories such as energy efficient clustering, load balancing clustering, low maintenance clustering, mobility aware clustering, and dominating set-based clustering [81]. In this paper we use a graph-based clustering schemes proposed in [14], where it ensures a low message overhead and form clusters in such a way that it does not induce frequent cluster changes. In our algorithm, the node with minimum ID among neighboring nodes which have not joined any other cluster forms a cluster and becomes the cluster head. Note that because the node with minimum ID considers only its one hop neighbors in forming the cluster, the nodes in a cluster are one hop away from the cluster head and at most two hops away from any other cluster mate when the cluster is formed. Furthermore, each node is member of one and only one cluster.

Chen et al. [14] argues on different ways of assigning IDs to the nodes. For instance, suppose that a node's life time is represented by a k-bit number. Then, an ID of the form EH where E is the current age of the node and H is its hardware address can be considered in the lowest ID clustering (LIC) algorithm.

2.6. **Comparing the Performance of Learning Approaches.** The area under receiver operating characteristics (ROC) curve (AUC) has often been used as the main evaluation criterion for comparing the performance of classification or data mining algorithms. Bradley [9] examine six different classification schemes on six real world medical data sets and found that AUC exhibits several desirable properties compared with accuracy. Ling et al. [38] proved theoretically and verified empirically that AUC is, in general, a better measure than accuracy.

In this paper, to compare the performance of different evaluated algorithms and alternatives in detection of attacks, we utilize an adopted version of AUC called WAUC. WAUC measures the area between ROC curve and random guess line as well as it considers the importance of less false positive rates as compared with the higher one.

Assume $ROC_{A,G}$ indicates detection rate versus false positive rate curve for attack A by classification algorithm G. Weighted area under the ROC curve ($WAUC(A,G)$) is defined as:

$$WAUC(A,G) = \int_{x=0}^{mfp} (mfp - x) * (ROC_{A,G}(x) - x)dx, \tag{4}$$

where mfp is the maximum false positive rate.

For overall comparison of classification algorithms, we sum the WAUC of different attacks by those algorithms. Suppose K attacks $(A_1, A_2, \dots, A_K)$ are launched in the network. Power of classification algorithm G ($PoCA(G)$) is defined as:

$$PoCA(G) = \frac{\sum_{i=1}^{K} WAUC(A_i, G)}{K}. \tag{5}$$

3. Literature Review

In this section, the related works are examined in four groups. In Section 3.1, we survey the extensions of the OWA operators. In Section 3.2, the approaches for generating the weights of OWA operators are presented. In Section 3.3, some applications of OWA operators are introduced, and finally in Section 3.4, the classifier-based anomaly detection methods in AODV-based MANETs will be examined.

3.1. **Extensions of the OWA Operators.** A large variety of OWA operators have been proposed in the literature. The Linguistic OWA (LOWA) operator [28] is an extension of the OWA operator that uses linguistic assessments instead of numerical values for providing individuals' opinions. The induced OWA (IOWA) operator [80] uses order inducing variables in the re-ordering step of the arguments. The heavy OWA (HOWA) operator [75] deals with the situations where the available information is independent and this aspect needs to be considered in the aggregation. The generalized OWA (GOWA) operator [76] add to the OWA operator an additional parameter controlling the power to which the argument values are raised. In the continuous OWA (COWA) operator [77], the given argument is a continuous valued interval rather than an exact argument. Some other extensions of the OWA operator are induced uncertain linguistic OWA operator [71], Type-1 OWA operator [85], induced generalized OWA operator [49], quasi-induced OWA operator [49], continuous generalized OWA operator [84], induced heavy OWA operator [48], uncertain heavy OWA operator [48], uncertain induced heavy OWA operator [48], fuzzy induced generalized OWA operator [50], fuzzy induced quasi-arithmetic OWA operator [50], and also generalized hybrid averaging [45], and induced generalized hybrid averaging [45].

3.2. **Generating the Weights of OWA Operators.** It is clear that the type of aggregation performed by an OWA operator depends upon the form of the associated weighting vector. A number of approaches have been proposed in the literature to obtain the OWA weights.

A commonly used approach is to obtain the weights based on some pre-defined quantifier function [72, 73, 74]. Yager [72] proposed a quantifier guided method for calculating the weights of OWA operator as:

$$w_j = Q(\frac{j}{n}) - Q(\frac{j-1}{n}), j = 1, \ldots, n, \tag{6}$$

where $Q : [0,1] \to [0,1]$ is a regular monotonically non-decreasing quantifier function such that $Q(0) = 0, Q(1) = 1$ and if $x > y$, then $Q(x) \geq Q(y)$.

Another commonly used approach is to generate the desired weights under a given orness level, which is usually formulated as a constrained optimization problem:

optimize $V_{OWA}(W)$

subject to

$$orness(W) = \frac{1}{n-1}\sum_{i=1}^{n} w_i(n-i) = \alpha, \quad 0 \leq \alpha \leq 1, \tag{7}$$
$$\sum_{i=1}^{n} w_i = 1, \quad w_i \in [0,1], \quad i = 1, \ldots, n.$$

The objective to be optimized can be entropy [54, 24, 79, 69], Renyi entropy [43], variance [25], and maximum disparity [66, 3, 22]. O'Hagan [54] suggested a maximum entropy approach for determining OWA operator weights which formulates the weight determination problem as a constrained nonlinear optimization model with a pre-defined degree of orness as its constraint and the entropy as its objective function (i.e., $V_{OWA}(W) = disp(W) = -\sum_{i=1}^{n} w_i \ln w_i$). The resulting OWA operators are called maximum entropy OWA (ME-OWA) operators. Fuller and Majlender [25] suggested a minimum variance approach, which minimizes the variance of OWA operator weights under a given degree of orness (i.e., $V_{OWA}(W) = \sum_{i=1}^{n} w_i^2$). Majlender [43] proposed a parametric class of OWA operators with maximal Renyi entropy or entropy of degree l for any level of orness (i.e., $V_{OWA}(W) = H_l(W) = \frac{1}{1-l}\log_2 \sum_{i=1}^{n} w_i^l$). Wang and Parkan [66] proposed a minimax disparity approach for obtaining OWA operator weights. The proposed approach generates the weights by minimizing the maximum difference between any two adjacent weights (i.e., $V_{OWA}(W) = Max|w_i - w_{i+1}|, \quad i = 1, \ldots, n-1$). Liu [39, 40] proved that both of the minimum variance problem and the minimax disparity problem have the same equidifferent form, which is composed of a weighting vector of nonnegative arithmetic progression and zeros. These equidifferent form OWA weighting vectors can be obtained in an analytical way instead of solving the quadratic or linear programming problem.

Filev and Yager [23, 80] proposed a method for learning the weighting vector from observational data. The method is the same as the types of learning algorithms used in neural networks and is also based upon the gradient descent method. Sadiq and Tesfamariam [59] suggested an approach for generating OWA weights using probability density functions (PDFs) in which the mean and standard deviation of the PDFs are determined using the number of criteria in the aggregation process. Emrouznejad [21] and Ahn [1] considered the preferences of alternatives across all the criteria to determine the OWA weights. Some other weight generation approaches are Liu [39, 40], Yager [78], Llamazares [41], Wang and Parkan [67], Wang et al. [65], Szidarovszky and Zarghami [61], Renaud et al. [58], and Ahn [2].

The approach presented in this paper is different from those of the previous ones. In our proposal, we do not fix the OWA weighting vector either based on a pre-defined quantifier function or based on any optimization or learning approach on the weights, but we generate the OWA operator weights according to dynamic environmental conditions of the MANET in each time slot.

3.3. **Applications of the OWA Operators.** The OWA operator has been used in a wide range of applications such as decision making, neural networks, data mining, network security, and image processing. Li et al. [37] utilized OWA operator for trust management in large-scale peer-to-peer computing in which multiple factors are incorporated to reflect the complexity of trust, and the weights of multiple factors are dynamically assigned by weighted moving average and ordered weighted averaging (WMA-OWA) combination algorithms. Lo and Chen [42] used a type of OWA operator in risk assessment and prevention in information systems to ensure information security. In their method, fuzzy linguistic quantifiers-guided maximum entropy OWA (FLQ-MEOWA) operator is used to aggregate impact values assessed by experts. De and Diaz [17] proposed a model for result merging based on analytical network process (ANP) and OWA operator in which a meta search engine passes a query to multiple search engines and combines and re-ranks results returned by them in one merged list.

Merigo and Casanovas [46, 47] used OWA operator in business decision making regarding the selection of investments. Aydin et al. [6] utilized OWA operator for creating a decision support system for site selection of wind turbines using geographic information system (GIS) tools. Zhou et al. [86] suggested Alpha-level approach to implementing the type-1 OWA operator and applied it to breast cancer treatments. Nin et al. [52] utilized OWA operator for calculation of the reputation of the nodes in collaborative private networks. McElroy and Gader [44] presented a method for encoding and decoding lattice-based associative memories using two families of OWA operators. Ghaderi et al. [26] used OWA operator for combining two feature selection methods to improve the precision of text classification algorithms. Cho [16] formalized modular neural networks as information sources and used OWA operator for combining neural outputs. Wu and Liu [68], Jazebi et al. [33], and Wu et al. [70] used OWA operator for fusion of classifiers to be used in fish disease diagnosis system, protein fold recognition, and E-learning respectively.

3.4. **Anomaly Detection in MANET.** Huang et al. [31] proposed a method for anomaly detection in MANETs with AODV and DSR routing protocols. For this purpose, they uses cross-feature analysis to determine correlations among different features, i.e., the classification problem is $\{F_1, F_2, \cdots, F_{i-1}, F_{i+1}, \cdots, F_L\} \rightarrow F_i, \forall \quad 1 \leq i \leq L$, where $\{F_1, F_2, \cdots, F_L\}$ is the feature set. Consequently, anomaly detection problem is transformed into a set of classification sub-problems, where each sub-problem chooses a different feature as class label. A total of 141 features in 2 categories were defined in [31], divided into traffic related features (132 features), and topology and route related features (9 features). For evaluating the proposed method, 3 classifiers, i.e., C4.5, RIPPER, and NBC were used, where it was shown that C4.5 yields better results.

Nakayama et al. [51, 36] proposed anomaly detection that utilizes principal component analysis (PCA). PCA is used for statistical analysis of the feature vector in each time interval by utilizing the projection distance obtained from the first principal element in PCA, which shows the deviation of the extracted feature vector from the saved normal profile. The proposed method is adaptive, meaning that the training dataset will be gradually updated based on the conditions of the network. The proposed method defines 14 features in 3 categories, namely, path finding features, path abnormality features, and AODV characteristic feature.

Cabrera et al. [10, 11], proposed an anomaly-based intrusion detection system (IDS) for MANETs with AODV and OLSR routing protocols, in which a 3-level cluster-based MANET with normal nodes, cluster heads, and a manager node is assumed. Furthermore a local IDS is attached to each node that calculates the local anomaly index via machine learning. Anomaly index indicates the difference between current operation of the nodes and a baseline of normal operation. The calculated anomaly index is periodically transmitted to its cluster head, which calculates the cluster-level anomaly index to be transmitted to the manager node for calculating the network-level anomaly index. In their scheme, a clustering algorithm is used for updating the clusters and cluster heads using a graph-based clustering scheme [14]. The proposed method uses 28 features for describing the normal behavior of AODV routing protocol, which are a subset of features in [31], and include AODV topology and route features, as well as AODV traffic features. The anomaly index is calculated by using cross-feature analysis performed by C4.5 classifier. Using a higher-level anomaly index improves the performance as compared to using lower-level indices.

Zhang et al. [82, 83] used 2 features for describing the normal behavior of AODV and utilized SVM and RIPPER classifiers independently for learning the normal profile. Deng et al. [19] proposed SVM-based anomaly detection method with 4 features. Deng et al. [18], utilized SVM and KNN classifiers independently for learning the normal profile and used 24 features for describing the normal behavior of AODV. Avram et al. [5] proposed a method for anomaly detection in MANETs by utilizing SOM classifier with 6 features.

Note that existing methods use crisp classification in which the feature vectors are classified based on their deviation from the saved normal profile. In crisp classification, a threshold is used as a discriminator between the normal and the abnormal feature vectors and the normal (abnormal) feature vectors that are close to the defined threshold, are the same with those that are far from it. In contrast, the fuzzy classification enables us to distinguish the feature vectors that make different deviations from the saved normal profile and to consider these differences in overall attack detection. Furthermore, existing schemes use only one classifier (or if multiple classifiers are used, they are not combined). This means that a classifier's blind spot is its inherent weakness that cannot be cured. In contrast, we combine three diverse classifiers to cover blind spots of a classifier by the other classifiers, and obtain better results. Our approach is also based on a thorough analysis of AODV, which is not the case in the existing schemes.

4. **Anomaly Detection**

Figure 1 shows our proposed anomaly detection method in each node of the network. In the proposed method, the required features for describing the normal behavior of AODV protocol are defined by a protocol-based approach. For modeling the normal profile, we use a fused combination of three different one-class classifiers, namely, OCSVM, MoG, and SOM that are trained on normal profile of the network. Training the method, is off-line and the trained model is stored in each node for anomaly detection. Subsequently, for actual network traffic, the deviation from the saved normal profile is measured to identify the attack according to the measured deviation.

We assume that the mobile ad hoc network is cluster-based. The clustering and cluster membership must be updated in each time slot according to the changes in the topology of the network. As it was said, the LIC method is used for clustering the network [14]. Furthermore, each node collects its own data in the network and there is no need for monitoring the behavior of a node by its neighbors. At the end of each time slot, each node gives the extracted feature vector related to current network operation to three utilized one-class classifiers, MoG, OCSVM, and SOM. Subsequently, the fuzzy averaging is performed on the outputs of the classifiers to shape the fuzzy decision of the node.

The vote of each cluster member is then transmitted to the corresponding cluster head for making the final decision. Figure 2 shows the proposed method in each cluster head after receiving the votes of cluster members. As it can be seen in Figure 2, the cluster head aggregates the received votes by using the OWA operator and maps them to *Normal* or *Attack*, and then announces the final decision to cluster members.

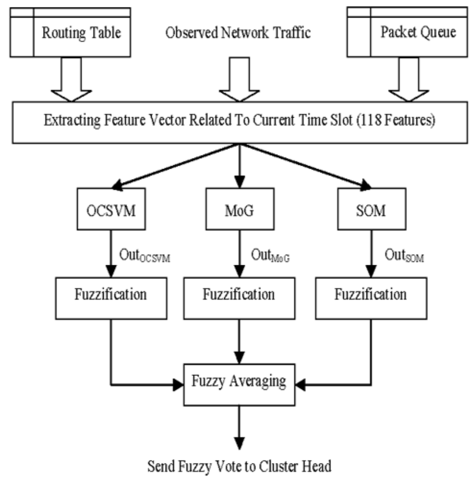

FIGURE 1. Our Proposed Anomaly Detection Method in Each Node

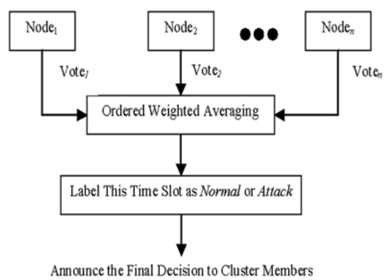

FIGURE 2. Vote Aggregation in Each Cluster Head

Our approach can be summarized as:

1- We allow the nodes to consider uncertainty in their decision making and to defer the decision making until the votes to be aggregated in the cluster head, which improves the performance of the proposed method via global fuzzy decision making instead of the local crisp one. We also use adaptive ordered weighted averaging for aggregating the votes of cluster members in the cluster head in which the amount of information that can be assured by the cluster head is determined according to the dynamic environmental conditions of the MANET.

2- We use the fuzzy averaging method for combining three diverse one-class classifiers, which improves the performance of the proposed method and causes the proposed method to be robust and less dependent on a single classifier. In this manner, weaknesses of a classifier are covered by other classifiers. Furthermore, the fuzzy classification enables us to distinguish the feature vectors that make different deviations from the saved normal profile and to consider these differences in overall attack detection.

In the reminder of this section, we describe the proposed method in details.

4.1. **Protocol-Based Feature Definition.** We defined the required features for describing the normal behavior of AODV protocol via step by step analysis of AODV. The analyses were carried out on the fields of RouteRequest, RouteReply, RouteError, Data packets, the node's ingoing and outgoing traffic, routing table and its changes, each node's input queue, ..., including the following phases:

1- Recognizing all possible functions of AODV, for example by extended finite state automata (EFSA) [56, 57, 32, 7].

2- Constructing a functionality tree for each function in EFSA. The functionality tree can be derived from existing documentations or RFCs. The constructed trees should seek all functionality paths of the protocol.

3- Logging the information in each functionality path.

4- Performing statistics and time-based analysis on the collected information in each time slot.

As an example, the function *Receive RouteReply* of AODV protocol is analyzed, as shown in Figure 3.

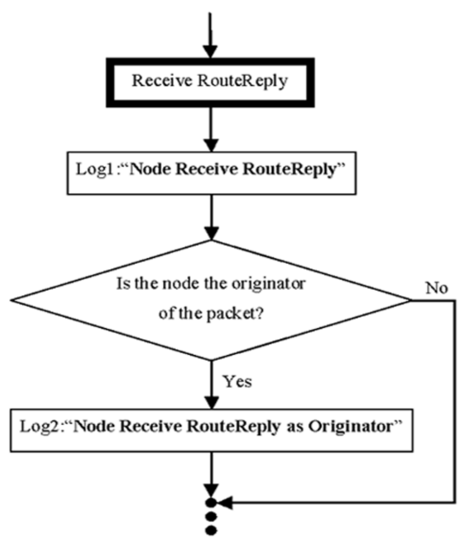

FIGURE 3. Protocol-Based Feature Definition

In Receive RouteReply function, when a node receives a RouteReply packet, first it logs **"Node Receive RouteReply"**. Subsequently, the node checks whether it is the originator of the received packet. If so, it logs **"Node Receive RouteReply as Originator"**, and this action will continually repeat until it reaches the leaf of the functionality tree. At the end of each time slot, the collected information is analyzed statistically. So, the number of Log1 is considered as a feature, i.e., the number of received RouteReply packets, and the number of Log2 is considered as another feature, i.e., the number of received RouteReply which the node is originator of route discovery mechanism.

By this method, a comprehensive description of AODV's normal behavior is obtained. A total of 118 features, divided into 5 categories, were obtained by this method as listed in Table 1. The complete set of defined features is shown in Appendix 1.

	Category	Number of Features
A	Traffic Related Features	58
B	Routing Table Related Features	37
C	Extracted Fields of Observed Packets	16
D	Packet Queue Features	3
E	Route Discovery Related Features	4

Table 1. Classification of Defined Features into 5 Categories

4.2. **Combining One-Class Classifiers.** Researchers are continually seeking to improve the performance of classifiers in classification problems, where combining classifiers is an approach to achieve this goal. Combining classifiers can improve the accuracy and robustness of the classification at the expense of increased complexity [35]. To clarify the need for combining classifiers, suppose one selects a classifier for anomaly detection according to its high performance in detecting existing attacks. If in the future, a new attack would be launched in the network which is in the blind spot of the utilized classifier, then the attack will not be detected. Whereas, if more than one classifier are utilized, the weakness of a given classifier would be covered by the other classifiers, especially when the classifiers are diverse and make different output patterns for the same input feature vectors [35, 55, 27].

Several methods have been proposed in the literature for combining classifiers such as averaging, fuzzy averaging, voting, fuzzy voting, and decision templates [62, 35, 63]. In this paper, the fuzzy averaging method is used for combining one-class classifiers. The idea of fuzzy-output classification is to consider uncertainty in the decision making of classifiers. It allows the state of the extracted feature vector r (which is representative of the current state of the network) to be a fuzzy variable, meaning that to be neither fully abnormal, nor fully normal, but partly abnormal and partly normal to a given degree. In other words, we can say "r is normal with a degree of 0.7" instead of "r is normal" in the crisp version. Furthermore, r is allowed to be neutral in addition to normal and abnormal.

For fuzzy averaging, the outputs of classifiers are first fuzzified using membership functions (see Figure 1). The degree of membership of a feature vector r to three fuzzy sets $Abnormal$, $Neutral$, and $Normal$ are calculated as:

$$\mu_{C,Abnormal}(r) = \begin{cases} 1 & \alpha \leq P_C(r) < \theta - \delta \\ \frac{\theta - P_C(r)}{\delta} & \theta - \delta \leq P_C(r) < \theta, \\ 0 & \theta \leq P_C(r) \leq \beta \end{cases}$$

$$\mu_{C,Neutral}(r) = \begin{cases} 0 & \alpha \le P_C(r) < \theta - \delta \\ \frac{P_C(r)-\theta}{\delta} + 1 & \theta - \delta \le P_C(r) < \theta, \\ \frac{\theta - P_C(r)}{\delta} + 1 & \theta \le P_C(r) < \theta + \delta \\ 0 & \theta + \delta \le P_C(r) \le \beta \end{cases} \tag{8}$$

$$\mu_{C,Normal}(r) = \begin{cases} 0 & \alpha \le P_C(r) < \theta \\ \frac{P_C(r)-\theta}{\delta} & \theta \le P_C(r) < \theta + \delta, \\ 1 & \theta + \delta \le P_C(r) \le \beta \end{cases}$$

where δ is the length of neutrality that specified by the user, θ is the discriminator of normal class from abnormal class in the crisp version, C is the utilized one-class classifier, r is the extracted feature vector, and $P_C(r)$ is the output of classifier C for input r (i.e., as a measure of resemblance). The proposed membership function is graphically presented in Figure 4, in which, we consider the state of r, as a fuzzy variable that is *Abnormal*, *Neutral* or*Normal* to a given degree according to the output of utilized classifier for input r. The parameters α, β, θ, and δ may be different for each utilized one-class classifier.

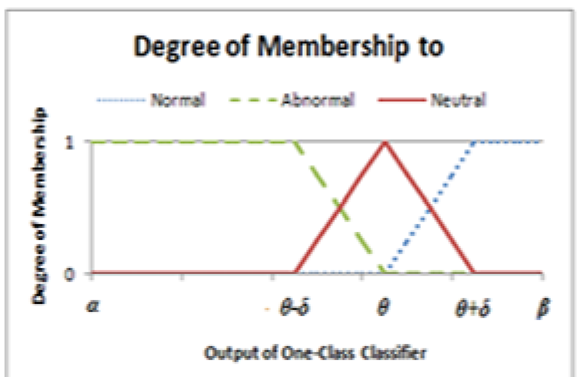

FIGURE 4. Membership Functions for Fuzzifying the Outputs of One-Class Classifiers

Once the outputs of classifiers are fuzzified, the fuzzy inference rule is applied to form the vote of node j as:

$$V_{j,class}(T) = \frac{1}{L}\sum_{i=1}^{L} \mu_{C_i,class}(r), \tag{9}$$

where T denotes the time slot, L is the number of utilized classifiers, C_i is i^{th} classifier, and $class \in \{Abnormal, Neutral, Normal\}$. The vote of a node is a triplet:

$$V_j(T) = (V_{j,Abnormal}(T), V_{j,Neutral}(T), V_{j,Normal}(T)). \tag{10}$$

The generated vote is then transmitted to the cluster head for making the final decision.

4.3. **Aggregating the Votes of Cluster Members in the Cluster Heads.** The combined vote of classifiers in each node is transmitted to the corresponding cluster head for making the final decision. The cluster head then aggregates the received votes according to:

$$FV(T) = arg\ max_{class} \sum_{j=1}^{n} w_j(T) * B_{j,class}(T), \tag{11}$$

where n is the number of nodes in the cluster, $class \in \{Abnormal, Neutral, Normal\}$, $B_{j,class}(T)$ is the j^{th} largest element of the bag $< V_{1,class}(T), V_{2,class}(T), \ldots, V_{n,class}(T) >$ and $w_j(T)$ is the associated OWA weight. The cluster head then announces the final decision to the cluster members, and all cluster members label time slot T according to the received vote.

4.4. **Condition-Based Weight Generation.** An important issue in the definition of OWA operator is how to obtain the associated weighting vector. One approach for weight generation of OWA is to give some semantics to the weights, meaning that to calculate OWA weights based on some pre-defined policies or desired criteria. Since the network topology, traffic, and environmental conditions of a MANET as well as the number of nodes in a cluster dynamically change, the mere use of a fixed pre-defined quantifier-based weight generation approach for OWA operator is not efficient. Therefore, we use an adaptive condition-based weight generation approach in which the OWA weights are calculated periodically and dynamically based on the environmental conditions of the network. The idea is to calculate OWA weighting vector $W(T) = (w_1(T), w_2(T), \ldots, w_n(T))$ such that: 1) it associates higher weights to the nodes that have more certainty in their decisions (i.e., $orness(W) > 0.5$), and 2) the more the rate of changes of the network, the votes of the fewer nodes are reliable (i.e., $orness(W)$ is greater). Inspired form [51], the cluster heads use the following two metrics for measuring the rate of changes of the network in time slot T:

1- The changes in the neighbor set of the cluster head between time slot T and time slot $T-1$ (denoted as $NC(T)$). This metric represents the rate of changes in the topology of the network. Assume that for a given cluster head in time slot T, its neighbor set is $NS(T)$, then:

$$NC(T) = \frac{|NS(T) - NS(T-1)| + |NS(T-1) - NS(T)|}{2 * n}, \tag{12}$$

where n is the number of nodes in the cluster.

2- The total changes of routing table of the cluster head in time slot T (denoted as $RTC(T)$). Let $RTA(T)$ be the number of routes added to routing table, $RTR(T)$ be the number of routes removed from routing table, $RTU(T)$ be the number of routing table updates, and $RTS(T)$ be the size of routing table, then:

$$RTC(T) = \frac{RTA(T) + RTR(T) + RTU(T)}{RTS(T)}. \tag{13}$$

According to the above mentioned policies and based on these two defined metrics, $w_j(T)$ is calculated as:

$$w_j(T) = e^{-RTC(T)*NC(T)*j}, j = 1, \ldots, n. \tag{14}$$

The weights $w_j(T)$ are constrained by:

$$\sum_{j=1}^{n} w_j(T) = 1. \tag{15}$$

Figure 5 graphically shows the OWA weights obtained in different network conditions compared to the ones obtained via fixed quantifier-based method.

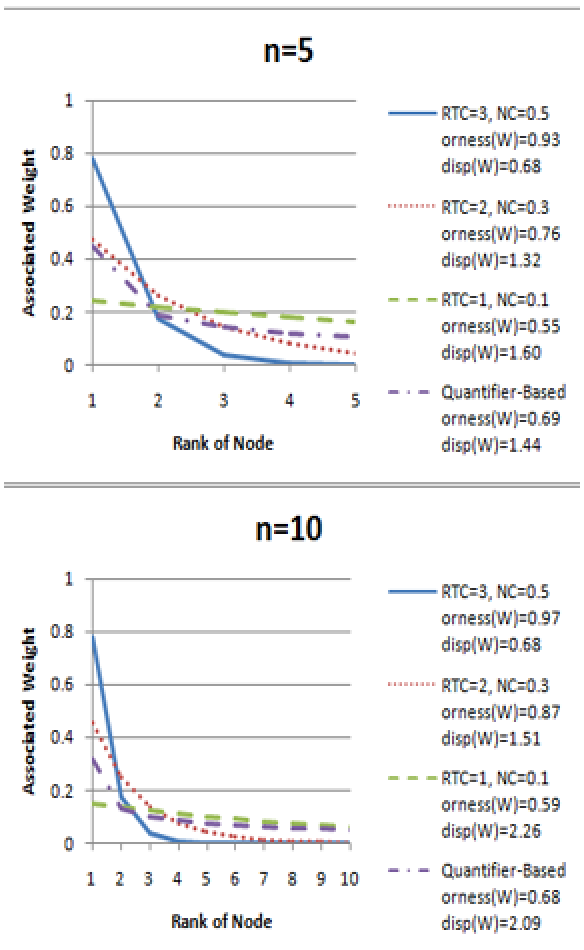

Figure 5. Condition-Based Weight Generation Compared to the Quantifier-Based Weight Generation, with the Quantifier $Q(x) = \sqrt{x}$

5. Simulation Results

The simulation was carried out by using Network Simulator NS-2 version 2.34 [53]. Our experiments are based on 50 wireless mobile nodes distributed in a 1000*1000 meters area, which follow the random way-point mobility model with the maximum speed of 5 m/s and a pause time of 10 seconds. Network traffic type is constant bit rate (CBR), data packet size is 512 bytes, and the maximum number of connections is 40 packets per second. Simulation time is 3000 seconds and each time slot is 30 seconds. Regular nodes normally perform routing as well as anomaly detection. In contrast, each malicious node launches attacks. The simulations were done on 100 different traffic patterns and movement scenarios and the average results are shown.

We launched three different attacks, namely, RouteError fabrication (RERR Fab.), rushing, and wormhole in the network to evaluate our method, where the number of attackers for RERR Fab. attack is 1, and for wormhole and rushing attacks is 2. Figure 6 compares two criteria for each of four environments (i.e, one normal and three attack environments). As it is shown in Figure 6, the launched attacks decrease packet delivery ratio and increase end to end delay except for wormhole that decreases end to end delay.

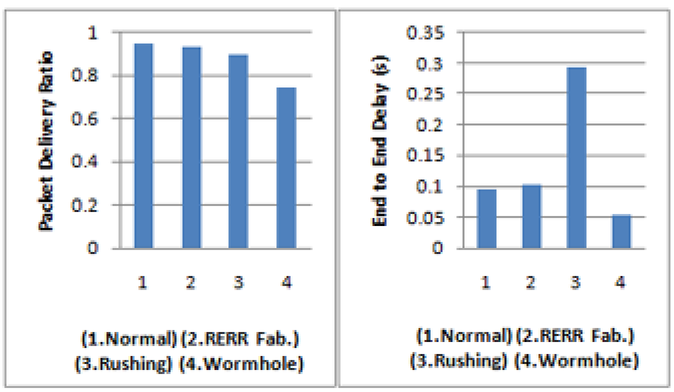

Figure 6. Impacts of the Attacks on the Network

Considering the information of Figure 6, different launched attacks in the network have different impacts and consequently, the deviation from the normal profile in these attacks will be different. The ROC curves for different attacks by the model obtained from 3 utilized classifiers as well as the combination of them are shown in Figure 7, where it shows that the launched attacks have different detection rates. Note that the RERR Fab. attack in general has the most detection rates compared to the wormhole attack that has the least detection rate. Furthermore as it can be seen, combining classifiers significantly improves the detection rates of attacks.

5.1. Evaluations and Discussion.

In this section, we aim to evaluate some alternatives of the proposed method and to present their impacts on the performance of the method. The alternatives are:

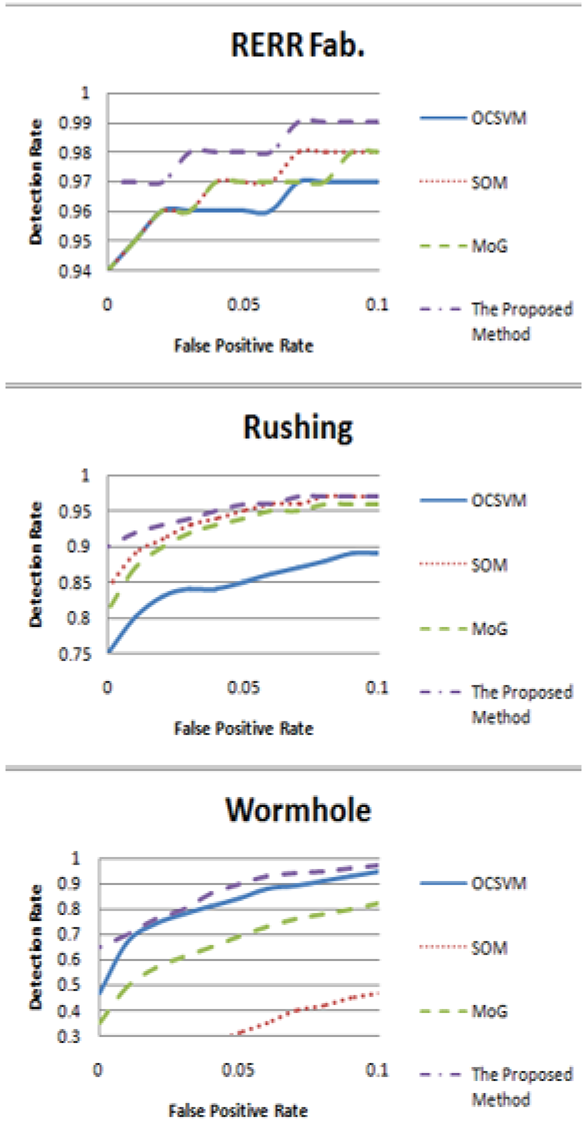

FIGURE 7. ROC Curves for Different Attacks by Different Learning Methods

Case 1: The network is flat instead of cluster-based.

Case 2: The network is cluster-based, but a crisp classification and attack detection is performed instead of the fuzzy one.

Case 3: The votes of cluster members is aggregated using averaging.

Case 4: The votes of cluster members is aggregated using fixed quantifier-based ordered weighted averaging with quantifier $Q(x) = \sqrt{x}$.

The ROC curves for these four alternative cases are shown in Figures 8 and 9. Comparing the ROC curve of Case1 and the proposed method shows that clustering the network improves performance of the anomaly detection method. The improvement is attained via global fuzzy decision making instead of the local crisp one. Comparing the ROC curve of Case2 and the proposed method shows that uncertainty in decision making of the nodes has significant impact on the performance of the anomaly detection method. Comparing the ROC curve of Case3 and the proposed method shows that we must have more assurance to the nodes that have more certainty in their decisions and associate higher weights to them. Comparing the ROC curve of Case4 and the proposed method shows that in the definition of the weights of OWA operator, we must consider dynamic environmental conditions of the networks. As it can be seen, the performance of the method decreases significantly in each four cases.

5.2. **Comparison of Different Alternatives.** The comparison of different alternatives based on their PoCA is shown in Table 2. Note significant differences among PoCA of the proposed method and all other alternatives.

The Method	PoCA
OCSVM	0.443
SOM	0.368
MoG	0.429
case1- Flat Network	0.352
case2- Crisp Voting	0.452
case3- Averaging	0.463
case4- Quantifier-Based OWA	0.469
The Proposed Method	0.480

Table 2. Comparison of PoCA for Different Methods

6. Conclusions

In this paper, an anomaly detection method in cluster-based mobile ad hoc networks with AODV routing protocol was proposed. In the method, the required features for describing the normal behavior of AODV were defined via step by step analysis of the protocol and not based on attack analysis. We used the fuzzy averaging method for combining three one-class classifiers, namely, SOM, MoG, and OCSVM to learn the normal behavior of AODV protocol in each node. Our experiments clearly show that uncertainty in decision making of the nodes, resulting from fuzzy-output classification, improves performance of the anomaly detection method. In the proposed method, an adaptive OWA operator was used for aggregating the votes of cluster members in the cluster head in which the associated OWA weights are calculated periodically and dynamically based on the environmental conditions of the networks. The condition-based weight generation approach outperforms

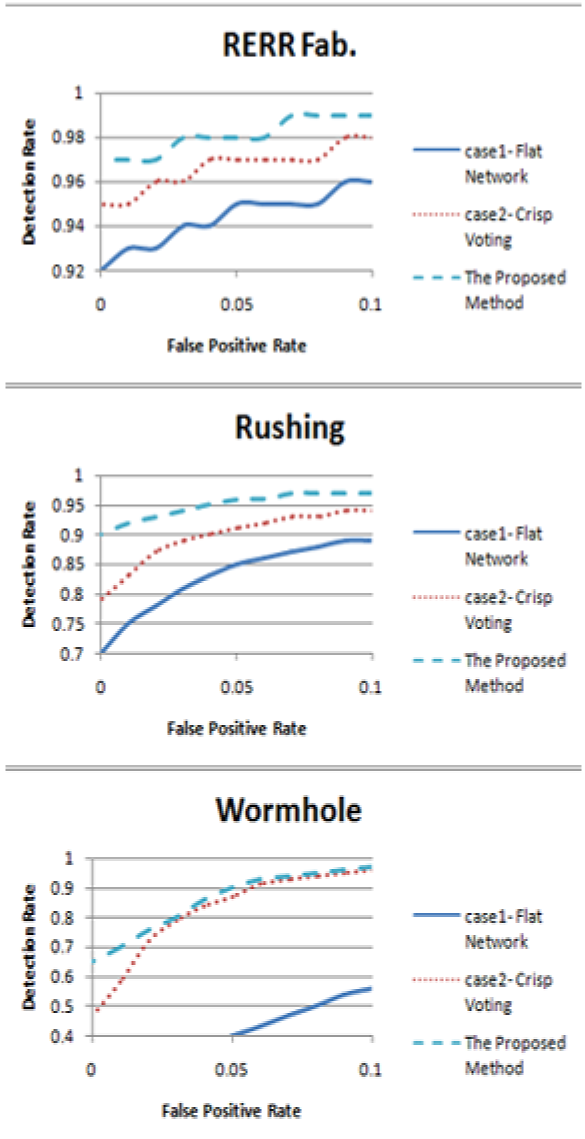

FIGURE 8. ROC Curves for Different Attacks in Different Alternative Cases

quantifier-based weight generation approach in detecting attacks.

Acknowledgements. This work was supported in part by the Iran Telecommunication Research Center (ITRC).

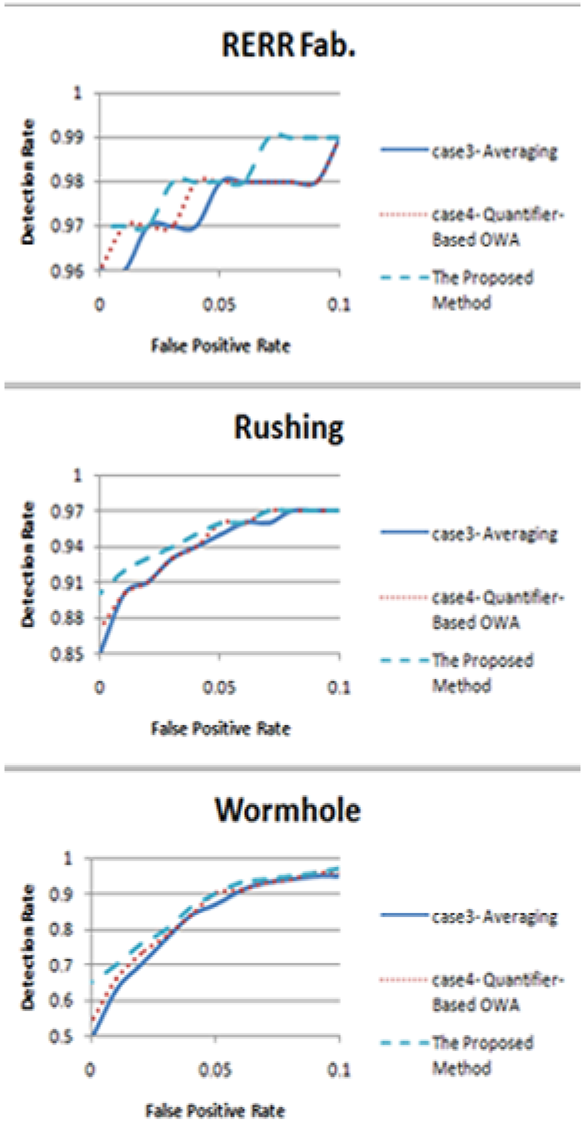

Figure 9. ROC Curves for Different Attacks in Different Alternative Cases

7. Appendix 1. Defined Features

A) Traffic Related Features (58 Features)

1-4) Number of [sent/ received] [packets/ control packets]

5-16) Number of [sent/ received/ forwarded] [RouteRequest/ RouteReply/ RouteError/ Data] packets
17-19) Number of replicated [RouteRequest/ RouteReply/ Data] packets received by node
20-21) Number of sent RouteReplys of which the node is [destination of route discovery/ an intermediate node that has a fresh route to destination]
22-27) Number of [RouteRequest/ different RouteRequest] packets received by node of which the node is [originator/ destination/ neither originator nor destination] of route discovery mechanism
28-29) Number of received RouteReplys of which the node is [originator/ destination] of route discovery mechanism
30-31) Number of different RouteReplys received with [new/ the same] originator-destination pair
32-33) Number of received Data packets of which the node is [creator/ destination]
34) Number of Data packets received by node of which the sender is creator
35-38) Number of dropped [RouteRequest/ RouteReply/ RouteError/ Data] packets
39) Number of dropped RouteRequests for TTL
40-41) Number of dropped [RouteReply/ Data] packets for NRTE
42) Number of dropped Data packets for CALLBACK
43) Max number of sent RouteRequests to the same destination
44-47) Max number of sent [RouteReply/ Data] packets to the same [receiver/ destination]
48-53) Max number of received [RouteRequest/ RouteReply/ Data] packets from the same [source/ sender]
54) Max number of received RouteErrors from the same Sender
55-58) [Max/ Average] difference between reception time of first and second copy of the same [RouteRequest/ RouteReply]

B) Routing Table Related Features (37 features)

1) Size of routing table
2) Number of valid routes div by number of invalid routes
3) Number of in repair routes
4-5) Route [add/ remove] count
6-8) [Max/ Min/ Average] number of valid routing table entries
9) Total routing table changes
10-11) Routing table changes [from its own packets/ from overheard packets]
12-14) Routes newly added by information from [destination of Data packet/ originator of RouteRequest/ destination of RouteReply]
15-17) Number of routing table updates by information from [RouteRequest originator/ RouteReply destination (the node is originator of route discovery)/ RouteReply destination (the node is in route from originator to destination)]
18-21) Number of successful routing table lookup for [valid/ invalid] routes, Number of unsuccessful routing table lookup, Rate of routing table successful lookup
22-24) [Max/ Min/ Average] length of valid routes in routing table

25-27) [Max/ Min/ Average] sequence number difference of valid routes in routing table
28) Number of times that hop count fields of routing table entries fixed (for invalid routes)
29-31) Number of times that hop count fields of routing table entries updated (for valid routes) - Total changes of these fields- Rate of changes
32-37) Number of times that sequence number fields of routing table entries [fixed (for invalid routes)/ updated (for valid routes)] - Total changes of these fields- Rate of changes

C) Extracted Fields of Observed Packets (16 Features)

1-8) [Max/ Min/ Average] difference between [hop count/ sequence number] field of received RouteRequest and [hop count/ sequence number] field of related entry in routing table for RouteRequest originator- Number of these packets
9-16) [Max/Min/Average] difference between [hop count/sequence number] field of received RouteReply and [hop count/sequence number] field of related entry in routing table for RouteReply destination - Number of these packets

D) Packet Queue Features (3 Features)

1-2) [Max/ Average] queue length
3) Number of added packets to queue

E) Route Discovery Related Features (4 Features)

1) Number of completed route discovery mechanisms
2-4) [Max/ Min/ Average] route discovery time

References

[1] B. S. Ahn, *Preference relation approach for obtaining OWA operators weights*, International Journal of Approximate Reasoning, **47** (2008), 166-178.

[2] B. S. Ahn, *Parameterized OWA operator weights: an extreme point approach*, International Journal of Approximate Reasoning, **51** (2010), 820-831.

[3] G. R. Amin and A. Emrouznejad, *An extended minimax disparity to determine the OWA operator weights*, Computers & Industrial Engineering, **50** (2006), 312-316.

[4] V. S. Anitha and M. P. Sebastian, *(k, r)-dominating set-based, weighted and adaptive clustering algorithms for mobile ad hoc networks*, IET Communications, **5(13)** (2011), 1836-1853.

[5] T. Avram, S. Oh and S. Hariri, *Analyzing attacks in wireless ad hoc network with self-organizing maps*, in Proceedings of Fifth Annual Conference on Communication Networks and Services Research (CNSR 07), (2007), 166-175.

[6] N. Y. Aydin, E. Kentel and S. Duzgun, *GIS-based environmental assessment of wind energy systems for spatial planning: a case study from Western Turkey*, Renewable and Sustainable Energy Reviews, **14** (2010), 364-373.

[7] K. Bhargavan, C. A. Gunter, M. Kim, I. Lee, D. Obradovic, O. Sokolsky and M. Viswanathan, *Verisim: formal analysis of network simulations*, IEEE Transactions on Software Engineering, **28(2)** (2002), 129-145.

[8] C. Bishop, *Neural networks for pattern recognition*, Oxford University Press, 1995.

[9] A. P. Bradley, *The use of the area under the ROC curve in the evaluation of machine learning algorithms*, Pattern Recognition, **30(7)** (1997), 1145-1159.

[10] J. B. D. Cabrera, C. Gutierrez and R. K. Mehra, *Infrastructures and algorithms for distributed anomaly-based intrusion detection in mobile ad hoc networks*, In Proceedings of IEEE Military Communications Conference (MILCOM 2005), (2005), 1831-1837.

[11] J. B. D. Cabrera, C. Gutierrez and R. K. Mehra, *Ensemble methods for anomaly detection and distributed intrusion detection in mobile ad hoc networks*, Information Fusion, **17(8)** (2008), 96-119.

[12] N. Cagman and S. Enginoglu, *Fuzzy soft matrix theory and its application in decision making*, Iranian Journal of Fuzzy Systems, **9(1)** (2012), 109-119.

[13] Z. Chen, *Consensus in group decision making under linguistic assessments*, Ph.D. Dissertation, Kansas State University, 2005.

[14] W. Chen, N. Jain and S. Singh, *ANMP: ad hoc network management protocol*, IEEE Journal on Selected Areas in Communications, **17(8)** (1999), 1506-1531.

[15] H. Chen and L. Zhou, *An approach to group decision making with interval fuzzy preference relations based on induced generalized continuous ordered weighted averaging operator*, Expert Systems with Applications, **38** (2011), 13432-13440.

[16] S. Cho, *Fuzzy aggregation of modular neural networks with ordered weighted averaging operators*, International Journal of Approximate Reasoning, **13** (1995), 359-375.

[17] A. De and E. D. Diaz, *A fuzzy ordered weighted average (OWA) approach to result merging for meta search using the analytical network process*, In Proceedings of Second International Conference on Emerging Applications of Information Technology (EAIT), (2011), 17-20.

[18] H. Deng, R. Xu, J. Li, F. Zhang, R. Levy and W. Lee, *Agent-based cooperative anomaly detection for wireless ad hoc networks*, In Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS 06), 2006.

[19] H. Deng, Q. A. Zeng and D. P. Agrawal, *SVM-based intrusion detection system for wireless ad hoc networks*, In Proceedings of IEEE Vehicular Technology Conference, (2003), 2147-2151.

[20] M. Dursun, E. E. Karsak and M. A. Karadayi, *A fuzzy multi-criteria group decision making framework for evaluating health-care waste disposal alternatives*, Expert Systems with Applications, **38** (2011), 11453-11462.

[21] A. Emrouznejad, *MP-OWA: the most preferred OWA operator*, Knowledge-Based Systems, **21** (2008), 847-851.

[22] A. Emrouznejad and G. R. Amin, *Improving minimax disparity model to determine the OWA operator weights*, Information Sciences, **180** (2010), 1477-1485.

[23] D. Filev and R. R. Yager, *On the issue of obtaining OWA operator weights*, Fuzzy Sets and Systems, **94** (1998), 157-169.

[24] R. Fuller and P. Majlender, *An analytic approach for obtaining maximal entropy OWA operator weights*, Fuzzy Sets and Systems, **124** (2001), 53-57.

[25] R. Fuller and P. Majlender, *On obtaining minimal variability OWA operator weights*, Fuzzy Sets and Systems, **136** (2003), 203-215.

[26] M. A. Ghaderi, N. Yazdani, B. Moshiri and M. Tayefeh Mahmoudi, *A new approach for text feature selection based on OWA operator*, In Proceedings of 5th International Symposium on Telecommunications (IST), (2010), 579-583.

[27] G. Giacinto, R. Perdisci, M. Del Rio and F. Roli, *Intrusion detection in computer networks by a modular ensemble of one-class classifiers*, Information Fusion, Special Issue on Applications of Ensemble Methods, **9(1)** (2008), 69-82.

[28] F. Herrera, E. Herrera-Viedma and J. L. Verdegay, *Direct approach processes in group decision making using linguistic OWA operators*, Fuzzy Sets and Systems, **79** (1996), 175-190.

[29] Y. C. Hu, A. Perrig and D. B. Johnson, *Rushing attacks and defense in wireless ad hoc network routing protocols*, in Proceedings of ACM Workshop on Wireless Security (WiSe 2003), (2003), 30-40.

[30] Y. C. Hu, A. Perrig and D. B. Johnson, *Wormhole attacks in wireless networks*, IEEE Journal on Selected Areas in Communications, **24(2)** (2006), 370-380.

[31] Y. Huang, W. Fan, W. Lee and P. Yu, *Cross-feature analysis for detecting ad hoc routing anomalies*, in Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS 03), (2003), 478-487.

[32] Y. A. Huang and W. Lee, *Attack analysis and detection for ad hoc routing protocols*, in Proceedings of Recent Advances in Intrusion Detection, (2004), 125-145.

[33] S. Jazebi, A. Tohidi and M. Rahgozar, *Application of classifier fusion for protein fold recognition*, In Proceedings of Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 09), (2009), 171-175.
[34] T. Kohonen, *self-organizing maps*, Springer-Verlag, 3rd edition, 2001.
[35] L. I. Kuncheva, *Combining pattern classifiers, methods and algorithms*, John Wiley and Sons, 2004.
[36] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour and Y. Nemoto, *A self-adaptive intrusion detection method for AODV-based mobile ad hoc networks*, In Proceedings of IEEE International Conference on Mobile Ad hoc and Sensor Systems (IEEE MASS 05), (2005), 773-780.
[37] X. Li, F. Zhou and X. Yang, *A multi-dimensional trust evaluation model for large-scale P2P computing*, Journal of Parallel and Distributed Computing, **71** (2011), 837-847.
[38] C. X. Ling, J. Huang and H. Zhang, *AUC: a statistically consistent and more discriminating measure than accuracy*, In Proceedings of 18th International Conference on Artificial Intelligence, (2003), 329-341.
[39] X. Liu, *The solution equivalence of minimax disparity and minimum variance problems for OWA operator*, International Journal of Approximate Reasoning, **45** (2007), 68-81.
[40] X. Liu, *A general model of parameterized OWA aggregation with given orness level*, International Journal of Approximate Reasoning, **48** (2008), 598-627.
[41] B. Llamazares, *Choosing OWA operator weights in the field of social choice*, Information Sciences, **177** (2007), 4745-4756.
[42] C. Lo and W. Chen, *A hybrid information security risk assessment procedure considering interdependences between controls*, Expert Systems with Applications, **39** (2012), 247-257.
[43] P. Majlender, *OWA operators with maximal Renyi entropy*, Fuzzy Sets and Systems, **155** (2005), 340-360.
[44] J. McElroy and P. Gader, *Generalized encoding and decoding operators for lattice-based associative memories*, IEEE Transactions on Neural Networks, **20(10)** (2009), 1674-1678.
[45] J. M. Merigo and M. Casanovas, *Fuzzy generalized hybrid aggregation operators and its application in decision making*, International Journal of Fuzzy Systems, **12(1)** (2010), 15-24.
[46] J. M. Merigo and M. Casanovas, *Decision-making with distance measures and induced aggregation operators*, Computers & Industrial Engineering, **60** (2011), 66-76.
[47] J. M. Merigo and M. Casanovas, *Induced aggregation operators in the Euclidean distance and its application in financial decision making*, Expert Systems with Applications, **38** (2011), 7603-7608.
[48] J. M. Merigo and M. Casanovas, *Induced and uncertain heavy OWA operators*, Computers & Industrial Engineering, **60** (2011), 106-116.
[49] J. M. Merigo and A. M. Gil-Lafuente, *The induced generalized OWA operator*, Information Sciences, **179(6)** (2009),729-741.
[50] J. M. Merigo and A. M. Gil-Lafuente, *Fuzzy induced generalized aggregation operators and its application in multi-person decision making*, Expert Systems with Applications, **38** (2011), 9761-9772.
[51] H. Nakayama, S. Kurosawa, A. Jamalipour, Y. Nemoto and N. Kato, *A dynamic anomaly detection scheme for AODV-based mobile ad hoc networks*, IEEE Transactions on Vehicular Technology, **58(5)** (2009), 2471-2481.
[52] J. Nin, B. Carminati, E. Ferrari and V. Torra, *Computing reputation for collaborative private networks*, in Proceedings of 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC 09), (2009), 246-253.
[53] *NS-2 (network simulator version 2)*, URL: http://www.isi.edu/nsnam/ns/ns-documentation, 2010.
[54] M. O'Hagan, *Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic*, In Proceedings of 22nd Annual IEEE Asilomar Conference on Signals, Systems, Computers, (1988), 681-689.

[55] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto and W. Lee, *McPAD : a multiple classifier system for accurate payload-based anomaly detection*, Computer Networks, Special Issue on Traffic Classification and Its Applications to Modern Networks, **5(6)** (2009), 864-881.
[56] C. Perkins and E. Royer, *Ad hoc on demand distance vector routing*, In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 99), (1999), 90-100.
[57] C. Perkins, E. Royer and S. Das, *Ad hoc on demand distance vector routing*, IETF RFC 3561, 2003.
[58] J. Renaud, E. Levrat and C. Fonteix, *Weights determination of OWA operators by parametric identification*, Mathematics and Computers in Simulation, **77** (2008), 499-511.
[59] R. Sadiq and S. Tesfamariam, *Probability density functions-based weights for ordered weighted averaging (OWA) operators: an example of water quality indices*, European Journal of Operational Research, **182** (2007), 1350-1368.
[60] B. Scholkopf, J. C. Platt, J. S. Taylor, A. J. Smola and R. C. Williamson, *Estimating the support of a high dimensional distribution*, Neural Computation, **13** (2001), 1443-1471.
[61] F. Szidarovszky and M. Zarghami, *Combining fuzzy quantifiers and neat operators for soft computing*, Iranian Journal of Fuzzy Systems, **6(1)** (2009), 15-25.
[62] D. M. J. Tax, *One-class classification*, Ph.D. Dissertation, Delft University of Technology, 2001.
[63] D. M. J. Tax, M. V. Breukelen, R. P. W. D. Duin and J. Kittler, *Combining multiple classifiers by averaging or by multiplying?*, Pattern Recognition, **33(9)** (2000), 1475-1485.
[64] J. A. Torkestani and M. R. Meybodi, *Clustering the wireless ad hoc networks: a distributed learning automata approach*, Journal of Parallel and Distributed Computing, **70(4)** (2010), 394-405.
[65] Y. M. Wang, Y. Luo and X. Liu, *Two new models for determining OWA operator weights*, Computers & Industrial Engineering, **52** (2007), 203-209.
[66] Y. M. Wang and C. Parkan, *A minimax disparity approach for obtaining OWA operator weights*, Information Sciences, **175** (2005), 20-29.
[67] Y. M. Wang and C. Parkan, *A preemptive goal programming method for aggregating OWA operator weights in group decision making*, Information Sciences, **177** (2007), 1867-1877.
[68] Y. Wu and J. Liu, *A new method for fish disease diagnosis system based on rough set and classifier fusion*, In Proceedings of International Conference on Artificial Intelligence and Computational Intelligence (AICI 09), (2009), 24-27.
[69] J. Wu, B. L. Sun, C. Y. Liang and S. L. Yang, *A linear programming model for determining ordered weighted averaging operator weights with maximal Yager's entropy*, Computers & Industrial Engineering, **57** (2009), 742-747.
[70] Y. Wu, X. Tan and S. Gu, *A learning evaluation system based on classifier fusion for E-learning*, In Proceedings of IEEE International Symposium on IT in Medicine & Education (ITIME 09), (2009), 749-752.
[71] Z. Xu, *Induced uncertain linguistic OWA operators applied to group decision making*, Information Fusion, **7** (2006), 231-238.
[72] R. R. Yager, *On ordered weighted averaging aggregation operators in multi-criteria decision making*, IEEE Transactions on Systems, Man and Cybernetics, **18** (1988), 183-190.
[73] R. R. Yager, *Families of OWA operators*, Fuzzy Sets and Systems, **59** (1993), 125-148.
[74] R. R. Yager, *Quantifier guided aggregation using OWA operators*, International Journal of Intelligent Systems, **11** (1996), 49-73.
[75] R. R. Yager, *Heavy OWA operators*, Fuzzy Optimization and Decision Making, **1(4)** (2002), 379-397.
[76] R. R. Yager, *Generalized OWA aggregation operators*, Fuzzy Optimization and Decision Making, **3(1)** (2004), 93-107.
[77] R. R. Yager, *OWA aggregation over a continuous interval argument with applications to decision making*, IEEE Transactions on Systems, Man, and Cybernetics, **Part B 34** (2004), 1952-1963.
[78] R. R. Yager, *Centered OWA operators*, Soft Computing, **11** (2007), 631-639.

[79] R. R. Yager, *On the dispersion measure of OWA operators*, Information Sciences, **179** (2009), 3908-3919.
[80] R. R. Yager and D. P. Filev, *Induced ordered weighted averaging operators*, IEEE Transaction on Systems, Man and Cybernetics, **29** (1999), 141-150.
[81] J. Y. Yu and P. H. J. Chong, *A survey of clustering schemes for mobile ad hoc networks*, IEEE Communications Surveys & Tutorials, **7(1)** (2005), 32-48.
[82] Y. Zhang and W. Lee, *Intrusion detection in wireless ad hoc networks*, In Proceedings of 6th Annual International Conference on Mobile Computing and Networking, (2000), 275-283.
[83] Y. Zhang, W. Lee and Y. Huang, *Intrusion detection techniques for mobile wireless networks*, ACM Wireless Networks, **9(5)** (2003), 545-556.
[84] L. G. Zhou and H. Y. Chen, *Continuous generalized OWA operator and its application to decision making*, Fuzzy Sets and Systems, **168** (2011), 18-34.
[85] S. M. Zhou, F. Chiclana, R. I. John and J. M. Garibaldi, *Type-1 OWA operators for aggregating uncertain information with uncertain weights induced by type-2 linguistic quantifiers*, Fuzzy Sets and Systems, **159** (2008), 3281-3296.
[86] S. M. Zhou, F. Chiclana, R. I. John and J. M. Garibaldi, *Alpha-level aggregation: a practical approach to type-1 OWA operation for aggregating uncertain information with applications to breast cancer treatments*, IEEE Transactions on Knowledge and Data Engineering, **23(10)** (2011), 1455-1468.

Mohammad Rahmanimanesh, Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Islamic Republic of Iran
E-mail address: `rahmanimanesh@modares.ac.ir`

Saeed Jalili*, Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Islamic Republic of Iran
E-mail address: `sjalili@modares.ac.ir`

*Corresponding author