

Agent-based Distributed Intrusion Detection Methodology for MANETs*

Hongmei Deng, Roger Xu, Frank Zhang, Chiman Kwan, and Leonard Haynes
Intelligent Automation Inc., Rockville, MD 20855

Abstract—*Intrusion detection, as a complementary mechanism to intrusion prevention, is necessary to secure wireless Mobile Ad hoc Networks (MANETs). In this paper we propose a practical agent-based distributed intrusion detection methodology for MANETs. A two-step intrusion detection procedure has been developed to effectively detect anomalies and identify attack types using distributed intrusion detection agents. The approach is efficient in dealing with large amount of system audit information with the growing network size. In addition, the distributed agent based implementation provides inherent flexibility and scalability. The performance of the approach has been evaluated via extensive simulations.*

Keywords: MANET, Intrusion/Anomaly detection.

1. Introduction

In mobile ad hoc networks (MANETs), mobile nodes are not bounded to any centralized control like base stations or access points. This feature offers great flexibility for establishing communications, while at the same time makes MANETs very vulnerable to various attacks [1]-[3].

A number of research activities have been devoted to develop security mechanisms for MANETs. They can be generally categorized into four groups: secure routing, trust and key management, service availability protection, and intrusion detection. The intrusion detection is necessary since developing a system that is absolutely secure is generally impossible. There are always some weak links, by which the attackers can get access to the network. Moreover, encryption and authentication mechanisms can efficiently prevent intrusions, but they can not totally eliminate them, especially when the intrusions initiate from inside. Thus, intrusion detection is suggested as a complimentary mechanism when all other approaches fail and the attackers successfully access the network.

Although the intrusion detection is not a new concept and it has been an active research area for over two decades, traditional intrusion detection systems can not be directly applied to MANETs due to its unique

characteristics. In this paper, we present an agent-based distributed intrusion detection methodology for MANETs. The proposed approach addresses the underlying characteristics of MANETs and provides specific support to enhance their securities. First, the intrusion detection is implemented in a distributed manner. In traditional networks, network traffic monitoring and intrusion detection are usually performed at gateways or routers. In MANETs, however, it is impossible to find such a traffic concentration point. Moreover, transmitting all the data collected locally to a central server would generate too much overhead, and also cause a significant processing delay, even if the central server can be virtually defined. Therefore, in our approach, the intrusion detection is distributed to each individual mobile node.

Second, a two-step procedure to effectively detect intrusions and identify attack types has been developed. In the first step, an unsupervised anomaly detection model, learned only from normal network behaviors, is applied to detect anomalies. In the second step, the intrusions are classified based on their behavior patterns. Conventional intrusion detection approaches need clearly-labeled dataset for training. Using unsupervised anomaly detection in the first step eliminates the time-consuming labeling process, and makes the approach efficient in dealing with a large amount of network data in real MANET environments. Moreover, the approach is able to detect new attacks. It is a beneficial feature as the MANET is a new wireless communication paradigm, it is still under development and a lot of attack types have not yet been identified.

Third, the concept of software agent technology has been applied in the system implementation. The intrusion detection agents are distributed throughout the network, thus, there is no single point of failure. In addition, this distribution of resources provides inherent flexibility and scalability, as agents may be easily added, removed, or even actively deployed.

The organization of the paper is as follows. In Section 2, the proposed distributed intrusion detection methodology including overall intrusion detection procedure and intrusion detection algorithms are described. Performance evaluation is given in Section 3. In Section 4, we present some related work. Concluding remarks are given in Section 5.

*. This research was supported by Army research Office under contract W911NF-05-C-0030.

2. Proposed distributed intrusion detectoin methodology

The proposed intrusion detection methodology is a distributed intelligent agent-based system. It performs intrusion detection in a fully distributed manner. Each node works as an independent intrusion detection agent, responsible for detecting intrusion locally based on the data collected by itself. By distributing the intrusion detection agents throughout the network, it takes full advantages of the distributed agent architecture, and is scalable, adaptive, and reconfigurable. In the scheme, when a malicious node is found, an alarm message will be broadcasted to the network. Each node also makes a final decision based on the detection reports from other nodes. To avoid drastic flooding over the network caused by broadcasting local detection results, the alarm messages are restricted to several hops since the neighboring nodes usually play more important roles than far away nodes.

2.1 Two-step intrusion detection

Within the distributed intrusion detection methodology, a two-step procedure (Figure 1) to detect intrusions and identify the attack types is employed. In the first step, an anomaly detection model is built to detect anomalies using a collected feature set. If the detection result of the first step is true, the feature set will be labeled as an UNKNOWN type and go through the second step - intrusion identification. Otherwise, the process stops. To identify each known attack type and recognize the unknown attacks, several identification models are applied, and each of them corresponds to an individual attack type. After the second step, all the identified anomalies are relabeled with the corresponding attack type, and all the unidentified anomalies remain to be of an UNKNOWN type for further investigation.

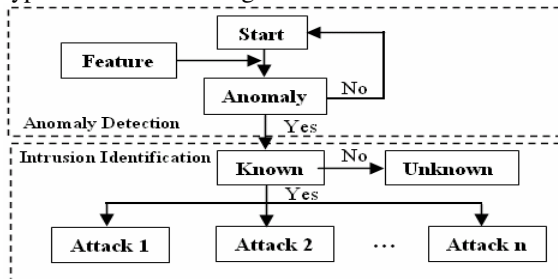


Figure 1. Overall procedure of intrusion detection

The main purpose of the two-step intrusion detection procedure is to detect various attacks (known/unknown) and separate them from normal network behaviors, such that the security response can

be conducted whenever an attack occurs. We take a data-centric point of view and consider intrusion detection as a classification problem. The task of an intrusion detection is to determine whether an audit record is normal or not, which is analagous to predicting the class label of the record.

The anomaly detection is transferred to an unsupervised classification or outlier detection problem, in which the attacks are detected by monitoring system behavior for evidence of deviation from the normal behavior. Since the anomaly detection model is learned only from normal network patterns, it can effectively capture the anomalous behavior even from attacks that have not been seen. The intrusion identification process is similar to the traditional supervised classification problem, which can be formulated as follows. In the training phase, we have profiles of all available attack types. Then a learning algorithm is applied to the training data to learn a classifier that identifies the characteristics of each attack types. Given a test dataset, the classifier can then recognize the known attack types by predicting the class label of each record.

The proposed two-step intrusion detection methodology has several advantages over the traditional signature-based intrusion detection approaches. First, the approach builds an anomaly detection model in the first phase, so that it has the capability to handle huge amount of system audit information with the growing network size. Unsupervised anomaly detection model usually has less complexity than signature-based approaches. In addition, the anomaly detection model has the potential to detect new unknown attacks. The fact is that the number of newly created attack types mounted on mobile computing environments is constantly increasing as more and more network appliances become mobile and wireless. Comparing the anomaly detection process with the intrusion identification, anomaly detection is more interesting since it avoids the time-consuming labeling process by only requiring the training data under normal network conditions. Moreover, anomaly detection is insensitive to the buried noise (intrusions) within training datasets.

2.2 Intrusion Detection Algorithms

In our scheme, the Support Vector Machine (SVM) is proposed for building both anomaly detection and intrusion identification models. We have implemented one-class SVM (1-SVM) [4] based anomaly detection and intrusion identification models. For performance comparison, we have also implemented K-Nearest Neighbors (KNN) for anomaly detection, and multi-class SVM (2-SVM), Probabilistic neural network (PNN), and KNN for intrusion identification.

We choose SVM classifier in both the anomaly detection and intrusion identification processes for two reasons. Firstly, SVM has been proven to be able to provide a good noise-tolerance performance on pattern recognition problems. Secondly, SVM performs classification in a virtually mapped high dimensional space instead of the original feature space, thus it can usually achieve higher classification accuracy than other approaches. The 1-SVM extended the idea of 2-SVM by Schölkopf for estimating the support of a high-dimensional distribution. Given a training dataset without any class information, the 1-SVM constructs a decision function that takes the value +1 in a small region capturing most of the data points, and -1 elsewhere. The strategy in this technique is to map the input vectors into a high dimension feature space corresponding to a kernel, and construct a linear decision function in this space to separate the dataset from the origin with maximum margin. In SVM, there are two parameters, γ and μ . Both of them control the generalization ability of the decision function.

KNN is a well-known classification technique, in which a training dataset is used to classify each test patterns based on the majority of the labels within its k-nearest neighbors. When using KNN for anomaly detection, a slightly modified algorithm is applied. Using only the normal data to train the KNN algorithm, the testing sample will be categorized according to the distance to its k-nearest neighbors.

The PNN is a layered neural network. The first layer computes distances from the input vector to the training input vectors, the second layer sums these contributions for each class of inputs to produce a net output as a vector of probabilities. Finally, a compete transfer function picks the maximum of these probabilities, and produces a 1 for that class and a 0 for the other classes. The PNN algorithm can generalize well, but with high computational complexity.

3. Performance Evaluation

The intrusion detection methodology is implemented using both the ns-2 [6] network simulator and IAI's proprietary network simulator (NetSim) [11]. The NetSim is a distributed MANET simulator based on the software agent infrastructure.

3.1 Simulation Environment

The network topology covers an area of 1000m by 1000m, and the number of mobile nodes simulated is 50. The maximum number of connections is set to be 20 and the simulation time is 10000 seconds. In the simulation, each node moves from a random starting point to a random destination with a randomly chosen speed (the speed is uniformly distributed between 0 to

20m/s). Once the destination is reached, another random destination is targeted after a pause time. We choose AODV as the underlying routing protocol. The sampling period is 5s and 10s.

3.2 Attacks

A number of attacks has been identified in [1][2]. Here we have simulated three common attacks.

- Blackhole attack [2][9]: a malicious node listens to a route request packet, and responds with a claim of having an extremely short route to the destination node, even if it does not have any such route.
- Routing request flooding attack (RREQ flooding): the malicious node deliberately floods the whole network with meaningless route discovery messages in order to exhaust the network bandwidth and effectively paralyze the network. In our implementation, the attack node can flood the network with random ROUTE REQUEST messages by specifying the inter-packet intervals.
- Routing request disrupt attack (RREQ disrupt): slightly different from RREQ flooding, the malicious node deliberately disrupts the route between randomly selected SOURCE node and DESTINATION node by sending ROUTE REQUEST packets. The purpose is to disrupt the route between the SOURCE and DESTINATION node.

3.3 Feature Selection

Since routing attacks attempt to make the routing protocol malfunction by sending bogus messages or intentionally dropping packets, the system behavior, represented by routing packets propagation, routing table changes and the data packet exchange/transmission might be different from those under normal network conditions. In addition, the network topology has some effect on the routing behavior. Thus, we mainly consider the features in the following aspects: 1) routing packet propagation, 2) route table changes, 3) data packet transmission, 4) velocity. Table 1 show the features used in the intrusion detection process. In total we have 30 features.

Table 1. Features for intrusion detection

Features	description	n
velocity	Speed of mobile node	1
Routing packet propagation: RREQ, RREP, RRER, HELLO, (send/receive/forward/drop)	Four flow directions for each type.	16
Data packet transmission: Transport layer (send/receive)	Two flow directions at the transport layer,	6

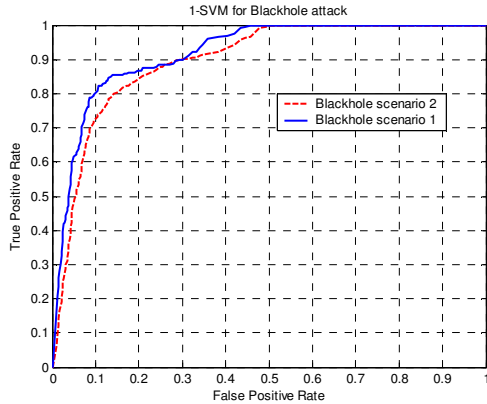
Routing layer (send/receive/forward/drop)	but four directions at the network layer.	
Route table changes: Add/remove/find/notice/repair, Total changes, Average route length	Newly added, removed stale routes, found in cache, eavesdropped, broken routes.	7

3.4 Anomaly Detection

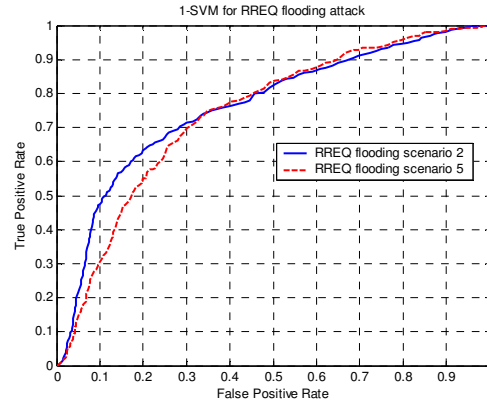
The Receiver Operating Characteristic (ROC) curve is used as a performance evaluation metric. We first simulate normal network activities (without any intrusive activities involved) in different scenarios: 1) the initial network topology is randomly generated, 2) each node follows a random mobility model, and 3) the network traffic patterns are generated in a random way. We extract the pre-defined features from the network log files with a sampling period of 5 seconds, and thus generate a normal dataset. The normal dataset is further divided into two parts with equal size, one for training, and the other part for testing.

We also implement the three attacks described in the previous section. The same procedures are applied to extract features from the simulated attacks and generate an attack dataset. The difference is that the whole attack dataset is used as testing samples to evaluate the performance of the anomaly detection scheme. The performance of the 1-SVM based anomaly detection model is evaluated under the following scenarios.

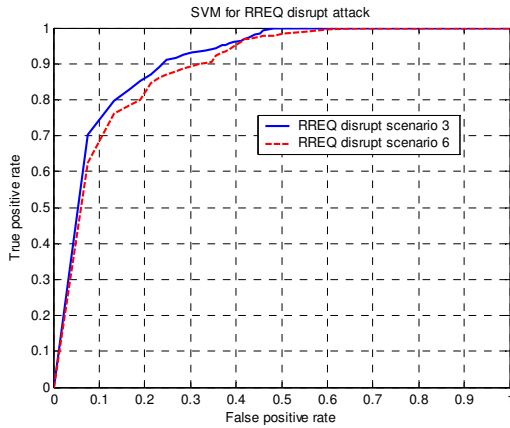
- (1) The blackhole attack is conducted 2 times in the whole simulation, and each time it lasts for 2000s.
- (2) Same as scenario 1, but the RREQ flooding attack is conducted.
- (3) Same as scenario 1, but the RREQ disrupt attack is conducted.
- (4) The black hole attack is conducted 8 times in the whole simulation, and each time it lasts for 200s.
- (5) Same as scenario 4, but the RREQ flooding attack is conducted.
- (6) Same as scenario 4, but the RREQ disrupt attack is conducted.
- (7) All the three attacks are conducted in one simulation, each attack lasts for 200s period.



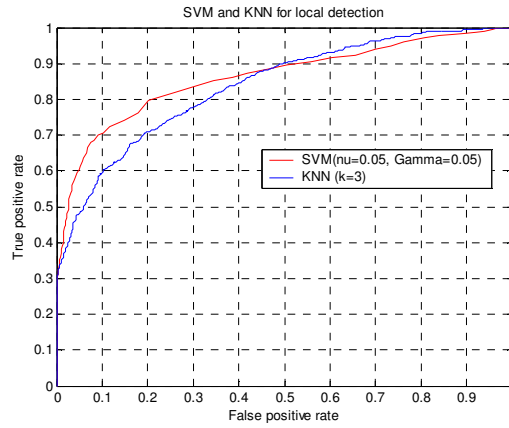
(a) Blackhole attack



(d) RREQ flooding attack



(c) RREQ disrupt attack

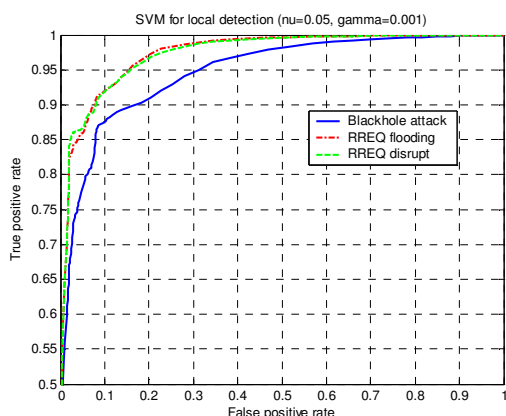


(d) Mixed attacks

Figure 2. Anomaly detection results using 1-SVM with sampling period of 5s

The first three scenarios correspond to the single attack case, but with a longer running time. Scenarios 4) to 6) change the attack running time to 200s to challenge the intrusion detection algorithm since the short time attack is more difficult to detect. In the last scenario, we mix all these attacks together. The experimental results are shown in Figure 2.

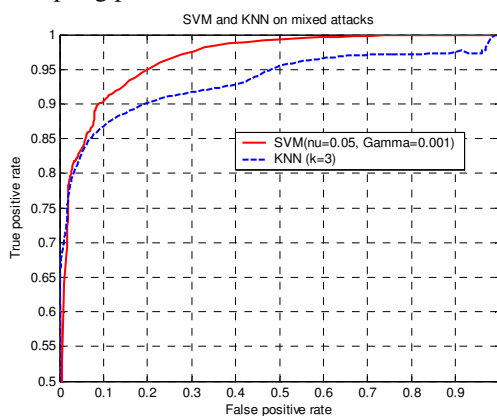
We have seen that the 1-SVM based anomaly detection can capture the attack behaviors even they only last for a short time. It can detect single type of attacks and also mixed attacks. It is not surprising since the anomaly detection model is an unsupervised approach. The decision boundary is built only based on the normal training data, and all the attacks are new to



(a) Individual attacks

it without much difference. The two parameters of 1-SVM used here are $\mu=0.05$, $\gamma=0.05$, respectively.

We use the same data set in scenario 7 to evaluate the performance of KNN. Figure 2(d) shows the anomaly detection result. KNN can also detect the intrusions, but with lower accuracy. 1-SVM achieves higher detection performance than KNN. In Figure 3 we also shows the performance of detection when sampling period of 10s is used. The same observation can be obtained that both the 1-SVM and KNN based anomaly detection approaches can detect mixed attacks, but 1-SVM achieves a higher detection performance. In addition, the detection performance using sampling period of 10s is better than that using sampling period of 5s.



(a) Mixed attacks

Figure 3. Anomaly detection results with sampling period of 10s

3.5 Intrusion Identification

After the anomalies are detected, the next step is to classify them. We have developed two methods to identify the attack types, multi-class classification and 1-SVM based intrusion recognition.

First we consider identifying different attack types as a multi-class classification problem, and each class corresponds to an individual attack type. We still add normal data type into the model because we would like to find the hidden normal patterns which were misclassified by the anomaly detection models. Thus, a four-class classification model is built. We have investigated two types of classification algorithms, 2-SVM and PNN. The misclassification matrix is used as the performance measure.

Table 2 and Table 3 show the intrusion detection results of 2-SVM based and PNN-based approaches. We have observed that both the SVM and PNN have achieved over 90% detection performance for the black hole attack and RREQ disrupt. For the RREQ flooding, the SVM-based approach achieves a detection rate of 83.75% and the PNN obtains 76.38%. The SVM-based approach has a lower false alarm rate and gets 98.95% accuracy for identifying normal data types, while PNN

captures 89.84% of the normal data samples correctly. We have also observed that a lot of RREQ flooding samples have been misclassified as normal data types. The reason is that the RREQ flooding behavior sometimes looks like normal network activities if the intruder selects a low flooding rate.

Table 2. Detection results of multi-class SVM

	Normal	Black hole	RREQ flooding	RREQ disrupt
Normal	98.95%	0.00%	0.85%	0.20%
Black hole	7.00%	92.75%	0.00%	0.25%
RREQ flooding	13.88%	2.38%	83.75%	0.00%
RREQ disrupt	7.12%	0.00%	0.37%	92.50%

Table 3. Detection results of PNN

	Normal	Black hole	RREQ flooding	RREQ disrupt
Normal	89.84%	0.45%	8.51%	1.20%
Black hole	2.50%	97.12%	0.13%	0.25%
RREQ flooding	21.88%	1.25%	76.38%	0.50%
RREQ disrupt	4.37%	0.00%	3.25%	92.37%

The advantage of multi-class classification approach is that it can quickly identify the known attack types in only one run. However, the model is trained on fixed number of known attack types. When new attacks occurs, our unsupervised intrusion detection model successfully captures them, but the multi-class detection model still misses them and misclassifies them as some known attack types. To solve this problem, we have also developed a 1-SVM based intrusion identification model.

We built one model for each attack type. The model is built in an unsupervised way (using 1-SVM) such that only the specified attack data is used during the training process. In each model, only two classes are considered, the specific data type to be detected and all the other types. For example, to identify whether the black hole attack is occurred or not, we need first pre-build a <blackhole-against-rest> model using unsupervised learning. The model is then applied to find the instances of black hole attacks.

Table 4. 1-SVM based Intrusion Recognition

	TPR	TNR
Model for Blackhole	96.86% ±0.67%	97.86% ±0.35%
Model for RREQ-flooding	89.00% ±1.17%	71.82% ±3.11%
Model for RREQ-disrupt	89.79% ±1.49%	81.05% ±1.11%

Table 5. KNN based Intrusion Recognition

	TPR	TNR
Model for Blackhole	91.00% ±1.33%	98.30% ±0.30%
Model for RREQ-flooding	89.21% ±1.50%	53.89% ±0.97%
Model for RREQ-disrupt	83.25% ±1.57%	63.99% ±3.57%

Table 4 and Table 5 show the performance of intrusion recognition using 1-SVM and KNN. The true positive rate (TPR) and true negative rate (TNR) are used as the performance measure. It should be noted that the TPR and TNR lost its original meaning here. TPR is the ratio of correctly recognized attacks under consideration, and TNR is the ratio of correctly recognized as other types. We run the experiments 10 times, and report the average performance and standard variance. From Table 4 and Table 5, we have seen that SVM and KNN can effectively recognize Blackhole attack, but have a lower performance in identifying the other two types. SVM still outperforms KNN in intrusion recognition.

4. Discussions

During our experimental studies, we have also realized several issues related to the proposed approach and there is further room to enhance its performance.

First, the proposed anomaly detection scheme intends to enhance the security level of MANETs, but it may also become the target of attackers. Similar to other common intrusion detection systems, the approach addresses the network security issues through a combination of auditing and analysis, monitoring and alarm, and vulnerability assessment. The audit logs are used by the intrusion detection agent for detection of potential intrusions. In reality the audit logs and the intrusion detection mechanism itself may become the target of malicious users. More generally, the problem is central to the creation of survivable network systems since any mechanism designed to assist a network in detecting and recovering is a tempting target for intruders. Further technologies in tamper resistant software/hardware may provide some type of solutions.

Second, the parameter determination in the proposed intrusion detection model has a large effect on system performance. We use 1-SVM for both anomaly detection and intrusion identification. The 1-SVM has a parameter μ and γ to control the tradeoff between the fraction of data points in the region and the generalization ability of the decision function. In traditional two-class or multi-class SVM algorithm, the problem is relaxed since we have access to more than one class of training data. Using the hold-out approach, a good pair of μ and γ can be obtained. However, in 1-SVM, only one class of training data is available. It is more difficult to find a good decision boundary. One option is to use the evolving method proposed by [12] to select the best combination of γ and μ .

Besides, the approach needs to be further enhanced as the current attackers become more and more sophisticated. Some attacks may present a slow multi-stage behaviors, and some attacks may include cooperative activities from multiple attackers. Further, truly sophisticated attacks may occur very rarely with a short execution time.

5. Related work

The research in securing wireless MANETs has attracted increasing attentions recently, but it is still in its early stage. A few papers have suggested using intrusion detection to enhance the security of MANETs. The distributed intrusion detection and response architecture proposed by Zhang and Lee [3] provides an excellent guide for the later works on designing an intrusion detection system in MANETs.

Sergio Marti et al. [7] introduced *Watchdog* and *Pathraiter* techniques that improve throughput in MANETs by identifying the misbehaving nodes that

agree to forward the packets but never do so. The *Watchdog* can be considered as a simple version of intrusion detection agent to identify misbehaving nodes, and the *Pathrater* works as the response agent to help routing protocols avoid these nodes.

There are several researches [13][14] focusing on detecting malicious packet dropping. Rao and Kesidis [13] presented a statistical approach using estimated congestion at intermediate nodes to make decision about malicious packet dropping. In [14], Buttyan and Hubaux proposed a tamper-resistant module at each node that counts packet forwarding events. The counter is used to monitor cooperative behaviors and penalize non-cooperation nodes.

Y. Huang et al [9] presented an anomaly detection approach using cross-feature analysis. They observed that a strong inter-feature correlation exists in normal network traffic and demonstrated that the approach of cross-feature analysis is very efficient to capture this between-feature relationship, thus a good anomaly detection performance has been obtained. One weakness of the approach is its high computational complexity. To build an intrusion detection model, a large number of sub-models are generated and the number equal to the number of features within the feature set. In [10], Y. Huang and W. Lee extended the previous idea and further built a rule-based intrusion identification mechanism. They addressed the resource constraints of mobile nodes and investigated the possibility of using cooperative detection to reduce the energy consumption.

6. Conclusions

In this paper, we proposed an agent-based distributed intrusion detection methodology. Within the proposed methodology, a two-step intrusion detection model has been developed to provide the capability of dealing with large amounts of network data and detecting new attack types. This paper mainly focused on anomaly detection, and transferred anomaly detection to an unsupervised classification problem. We have developed I-SVM based anomaly detection approaches. It should be noted that the proposed intrusion detection system is not specific to any routing protocol and any attack types. It can also be applied to other types of attacks in other layers as long as appropriate features are selected.

References

- [1] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," IEEE Network Magazine, Vol. 13,
- [2] H. Deng, W. Li, and Dharma P. Agrawal, "Routing Security in Ad Hoc Networks," IEEE Communications Magazine, Special Topics on

- Security in Telecommunication Networks, Vol. 40, No. 10, October 2002.
- [3] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks," Proceedings of the 6th International Conference on Mobile Computing and Networking, MobiCom 2000, pp. 275-283, August 2000.
- [4] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, "Estimating the support of a high-dimensional distribution," Neural Computation, v 13, no 7, pp. 1443-1472, 2001.
- [5] E. Eskin, A. Arnold, M. Prerau, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data," Data Mining for Security Applications, 2002.
- [6] K. Fall and Varadhanm, The ns Manual (formerly ns Notes and Documentation), 2000.
- [7] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proceedings of the 6th International Conference on Mobile Computing and Networking (MOBICOM'00), pp.255-265, August 2000.
- [8] Kachirski, and R. Guha, "Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks," Proceedings of the 36th Hawaii International Conference on System Sciences, 2003.
- [9] Y. Huang, W. Fan, W. Lee and P. S. Yu, "Cross-Feature Analysis for Detecting Ad-hoc Routing Anomalies," Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS'03), May 2003.
- [10] Y. Huang, and W. Lee, "A cooperative intrusion detection system for ad hoc networks," In Proc. of the 1st ACM workshop on Security of ad hoc and sensor networks, pp. 135-147, 2003.
- [11] Cybele, Intelligent Automation Inc., <http://www.i-a-i.com/view.asp?tid=36>.
- [12] Q. A. Tran, Q. L. Zhang, and X. Li, "Evolving Training Model Method for One-Class SVM," Proceedings of the 2003 IEEE International Conference on Systems, Man & Cybernetics (SMC 2003), pp 2388~2393, Washington D.C., 2003.
- [13] R. Rao and G. Kesidis, "Detection of malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited," Brazilian Journal of Telecommunications, 2003.
- [14] L. Buttyán and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," Technical Report No. DSC/2001/046, Swiss Federal Institute of Technology, Lausanne, August 2001.