Intrusion Detection for Mobile Ad-Hoc Networks based on a Non-Negative Matrix Factorization Method

Carlos Mex-Perera[1] José Zamora-Elizondo[1] Raul Monroy[2]

[1] Center for Electronics and Telecomunications, ITESM, Campus Monterrey
Av. Eugenio garza Sada 2501 Sur. Col. Tecnologico
Monterrey, N.L. , CP 64849 Mexico
[2] Computer Science Department, ITESM, Campus Estado de Mexico
Carretera al lago de Guadalupe, Km. 3.5, Estado de Mexico, CP 52926, Mexico
{carlosmex, A00791990, raulm}@itesm.mx

Abstract. In this paper we focus on intrusion detection in Mobile Ad Hoc Networks (MANETs), we propose a novel method for intrusion classification based on a Non-Negative Matrix Factorization (NMF) model. Feature vectors derived from statistics collected in the routing tables of the mobile nodes are used to form an input matrix for the NMF algorithm, which creates a behavior profile by building a matrix W of basis. Such matrix is later used to test unseen vectors. The distance between the test vector and its reconstruction is compared to a threshold level to obtain a decision about the existence of normal behavior. The results of the simulations show that the method might be suitable for its deployment in MANETs.

1 Introduction

A Mobile Ad hoc Network (MANET) is a low-cost, rapid-deployment, self-configuring network of mobile nodes (and associated hosts) connected by wireless links. Nodes cooperate one another forwarding packets so that each node may communicate beyond its wireless transmission range. They are free to move forming an arbitrary topology, which changes rapidly and unpredictably. MANETs have captured increasing interest as they are suitable for emergency situations.

Security is required in many MANET applications, including military operations and disaster relief. However, MANETs are vulnerable to a number of attacks. At a communication level, an intruder can easily inject bogus packets or eavesdrop on communication. At a network level, an intruder can easily attempt a malicious router misdirection. MANET's vulnerabilities cannot always be dealt with using techniques that were designed in the context of wired networks [1]. This is particularly the case for ad hoc routing: the routing problem is magnified as soon as we no longer assume a trusted environment [2]. This is because it is not easy to distinguish an ordinary change in the network topology from a change caused by a collection of compromised nodes.

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 131-140

Security mechanisms, such as authentication and encryption can be used as the first line of defense against attacks in MANET. However, they still cannot provide protection for attacks generated by a malicious inside node. Intrusion detection mechanisms are necessary to detect this type of attacks. To mitigate the problem, Intrusion Detection Systems (IDS), as a complementary mechanism, is designed to protect the availability, confidentiality and integrity of critical networked information systems. The goal of a IDS is to distinguish those nodes that perform an attack, such nodes are known as intruders.

In recent years the problem of IDS for MANEts has been devoted an special interest, there are a number of papers that have proposed different mechanisms for IDS for MANET, such as [3–5].

In this paper, we focus on the problem of intrusion detection for MANETs in the network layer, we propose a IDS based on a Non-Negative Matrix Factorization [6] model. Although NMF has been used previously in [7], for profiling program and user behaviors for host-based IDS, it has not been applied yet in MANETs. This work presents results that show that NMF could be applied in MANETs.

2 Intrusion Detection

Intrusion detection is concerned with the timely discovery of any activity that jeopardizes the integrity, availability or the confidentiality of an IT system. A Misuse Intrusion Detection System (MIDS) annotates as an attack any known pattern of abuse. MIDSs are very effective in detecting known attacks but are usually bad at detecting novel attacks. An Anomaly IDS (AIDS) annotates as an attack any activity that deviates from a profile of ordinary computer usage. Unlike MIDSs, AIDSs are capable of detecting novel attacks. However, they frequently tag ordinary computer usage as malicious, yielding a high false positive detection rate.

Depending on the activity it observes, an IDS can be placed at either of three points: a host, a network or an application. A host IDS usually audits the functionality of the underlying operating system, but can also be set to watch critical resources. An application IDS scrutinizes the behavior of an application. It commonly is designed to raise an alarm any one time the application executes a system call that does not belong to a pre-defined set of system calls, built by some means, an object-code analysis. A network IDS analyzes network traffic in order to detect mischievous activities within a computer network. A denial of service attack resulting from flooding a network with packets can be pinpointed only at this level.

An IDS should perform a number of tasks. In particular, it should [8]:

- identify the appearance of patterns of a known attack or of deviations from normal computer usage;
- identify flaws or vulnerabilities in the system configuration;
- audit the integrity of critical system or data files; and
- highlight user violations of a security policy.

Additionally, an ad hoc network IDS should [1]

- not add any extra weakness to the computer system under surveillance;
- consume little system resources; and
- run continuously in a transparent manner.

2.1 Building an Anomaly detection Model

This work is focused on obtaining profiles from statistics computed from the routing tables of the nodes of the MANET. Attacks can be presented in a great number of ways, for instance an intruder node can fake routing information and all nodes that receive such information might be building their routing tables with erroneous entries. Another way to create an attack is by dropping packets, thus the rest of the nodes would not be updating their routing information as expected in absence of intruders. In both type of attacks, connectivity among the nodes would be affected according to the severity of the attack. In our study, we implemented random packet dropping attack which is a pattern distortion technique.

In order to build an anomaly detection model, statistics of the routing tables when intruders are not present in the MANET are considered. Such statistics are then properly formatted to generate data vectors to train a classifier.

Once the learning phase has been performed, the classifier is ready to be used to test unseen vectors. The result of the classification is a decision about if the observed vector corresponds or not to a normal behavior. If an abnormal behavior is obtained, then it is considered that an intruder is affecting the routing protocol.

3 The Proposed Method

The proposal is based on non-negative matrix factorization, which is a method aimed to represent data using non-negativity constraints. The idea is the representation of a given object using the addition of its parts, which are considered as positive contributions to the whole object. NMF has been applied to many fields, including face recognition and text classification tasks [6]. Considering that intrusion activities in MANETs might affect audit data as positive contributions, we propose the use of NMF for intrusion detection.

3.1 Non-negative Matrix Factorization

Given a database represented by a n x m matrix V, where each column is an n-dimensional vector with positive data belonging to the original database (m vectors), we can obtain an approximation of the whole database (V) by

$$V_{i\mu} \approx (WH)_{i\mu} = \sum_{a=1}^{r} W_{ia} H_{a\mu} \quad (1)$$

Where the dimensions of the matrix W and H are n x r and r x m, respectively. Usually, r is chosen so that $(n+m)r < nm$. This results in a compressed version of the original data matrix. Each column of matrix W contains a basis vector while each column of H contains encoding coefficients needed to approximate the corresponding column in V. The following iterative learning rules are used to find the linear decomposition [6]:

$$H_{a\mu} \longleftarrow H_{a\mu} \sum_i (V_{i\mu}/(VH)_{i\mu}) W_{ia} \tag{2}$$

$$W_{ia} \longleftarrow W_{ia} \sum_\mu (V_{i\mu}/(WH)_{i\mu}) H_{a\mu} \tag{3}$$

$$W_{ia} \longleftarrow W_{ia}/\sum_j W_{ja} \tag{4}$$

The above unsupervised multiplicative learning rules are used iteratively to update W and H. The initial values of W and H are fixed randomly. With these iterative updates, the quality of the approximation of Equation 1 improves monotonically with a guaranteed convergence to a locally optimal matrix factorization [6].

3.2 IDS Based on Non-negative Matrix Factorization

IDS Based on NMF includes three steps: features selection, classifier training, classifier test and decision, as it is shown in Figure 1.

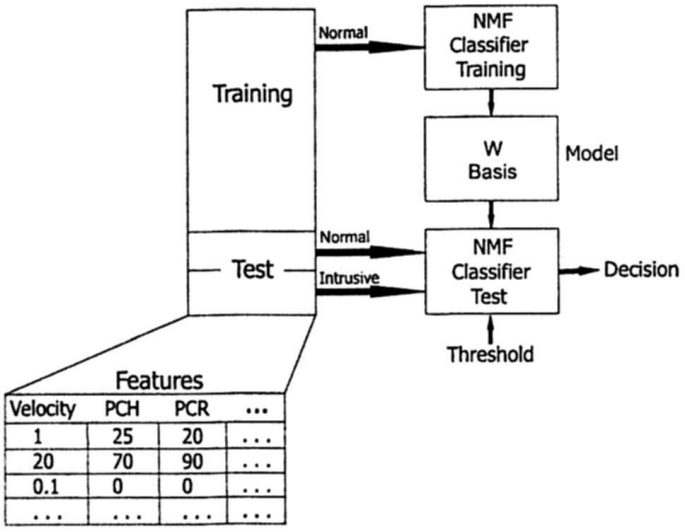

Fig. 1. System Architecture.

3.3 Features selection

Because we focus on routing attacks, we need to specify the trace data to be used that will present evidence of normalcy or anomaly. We define a trace data as the set of features aggregated into a single data set, which describes all changes in routing tables for a single node. Routing tables have information to know the next hop to reach a destination node and the number of hops for that route. Due to the movement of the nodes, the routing tables are updated regularly. These changes in the routing tables can be computed as features for a classifier. Following the work described in intrusion detection of MANETs [3], and based on our experiment results, we use features associated with routing caches and topological movement of mobile nodes in order to characterize their normal changes. Figure 1 shows some fictional features for a node. All features are detailed in Table 1 and the meaning of each feature is further explained in the Notes column.

Features	Explanation	Notes
PCR	% of changed routes	Deleted and increased routing entries
PCH	% of changes in the sum of hops	Average length of routes
PCB	% of change of bad routes	Broken routes
PCS	% of change of stale routes	Stale routes being removed
PCU	% of change of updated routes	Routes updated via overhearing.

Table 1. Local features ad-hoc routing protocol

We use percentages as measurements because of the dynamic nature of mobile networks (i.e., the number of nodes/routes is not fixed).

3.4 Classification Training

The data set for normal behavior represented by V (which is the matrix transpose of training data, see Figure 2) is approximately factorized into two matrices $W_{training}$ and H by the iterative updating rules given by Equations 2 - 4. Each column of matrix $W_{training}$ contains a basis vector, while each column of H represents the coefficients needed to approximate the corresponding column in V. Figure 2 shows that given a set of multivariate n-dimensional data vectors, the vectors are placed in the columns of an $n \times m$ matrix V where m is the number of examples in the data set. For instance, a column of matrix V contains a vector data with the values of PCR, PCH, PCU, PCB, PCS and velocity at a given time t, others columns of V have the same features but taken at different times. This matrix V is then approximately factorized in $W_{training}$ and H.

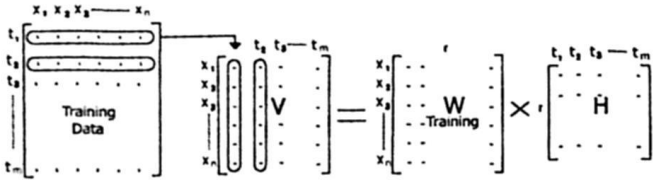

Fig. 2. NMF Classifier training.

3.5 Classification Test and decision

Equation 1 can be rewritten column by column as $v \approx Wh$ where v and h are the corresponding columns of V and H, respectively. Each vector v is approximated by a linear combination of the columns of W, weighted by the components of h. Based on this, it is found that given a column v of matrix V_{test} and using the basis $W_{training}$ learned from normal training data set, we can find a representative vector of encoding coefficients h by the update rule in Equation 2, see Figure 3. We then reconstruct v as $v' = W_{training} \times h$.

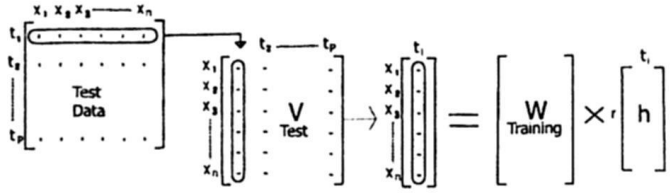

Fig. 3. NMF Classifier test.

The Euclidean distance given by Equation 5 between v and v' is used to calculate the similarity between these two vectors.

$$\Delta = \| v - v' \|^2 = norm(v - v') \tag{5}$$

If the test data vector is normal, then it is expected that the test data vector v will be very similar to its reconstructed version v', and the resulting distance between them will be very small. Therefore the testing vector is classified as normal if

$$\Delta < \varepsilon \tag{6}$$

where ε is defined as a threshold. Otherwise it is classified as intrusive, on this property our intrusion classification model is based.

4 Experimental Studies

In order to study how NMF classifier can be used to construct anomaly detection models for MANET routing, we have conducted the following simulation experiments.

4.1 Simulation Model

We chose one specific ad-hoc wireless protocol as the subject of our study, Ad-hoc On-Demand Distance Vector (AODV) [9]. In the simulation, 50 mobile nodes move in a 1000 X 1000 meter square region. We apply the random way-point model to emulate node mobility patterns. The maximum pause time between movements is 300 seconds, the minimal movement speed is 1 m/s, and the maximal movement speed is 20 m/s. 16 source-destination pairs are selected randomly to generate Constant Bit Rate (CBR) traffic as the background traffic. The transmission range is set to 250 meters. We simulate a routing disruption attack, where the attacker node drops packets belonging to the routing protocol, the attacker was randonmly chosen among 50 nodes. We run the simulation 3000 seconds in order to get normal data traces from all nodes. For each data trace, we collect (PCR, PCH, PCU, PCB, PCS and velocity) feature values every 3 seconds after a warm-up time period of 300 seconds. The data at the last 100 seconds are discarded. Values are treated in a similar way presented in previous works, in our case all features except the velocity are discretized into five uniformly distributed levels, ranging from 0 to 100% while velocity feature is discretized into 10 levels. We split the data collected during the 3000 seconds simulation into 2 parts, the first one corresponding to the first 2400 seconds will be used as training data and the rest 600 seconds will be used as normal testing data. Data collected from the routing tables of all nodes forms the training and testing vectors, each training data trace has 800 items and each normal testing data trace has 200 items. In this way, we get 50 x 800 = 40000 items for training data, and 50 x 200 = 10000 items for normal testing data. To get data of intrusive behaviors, we let the simulation run 600 seconds. For each run, we let the attack script start at time 100 seconds, and ends at time 300 seconds. We get 50 x 200 = 10000 items for intrusive testing data.

4.2 Simulations results

For evaluation purposes of the detection performance of the proposed method, we obtain Receiver Operating Characteristic (ROC) curves. An ROC curve is a parametric curve that is generated by varying the threshold of the intrusive measure, which is a tunable parameter. The ROC curve can be used to determine the performance of the system for different operating points. The ROC curve is the plot of False Alarm Rate, calculated as the percentage of decisions in which normal data are flagged as intrusive versus Missing Alarm Rate, calculated as the percentage of intrusive behavior falsely classified as normal. Training data and test data are used to tune the parameters of the classifier. The dimension r

is typically much lower then either dimension of matrix V. We have trained the model using different basis r from 2 to 5 in order to determine a suitable value of r. See Figure 4, among the ROC curves depicted the best results are obtained for $r = 3$. It is interesting to observe that NMF classifier has different operating points and we can be positioned in any point of the curve depending on the necessities. For comparison purposes we run the simulation with the same data using RIPPER [10], which is a well known rule based classifier. For RIPPER it is obtained a false alarm rate (1.67%) and the missing alarm rate (71.26%), which for many cases is not acceptable. Besides, it is not possible to adjust the sensibility of the detection for RIPPER. However, with the NMF method it is possible to adjust the operating point following a trade-off between the false alarm and missing alarm rates.

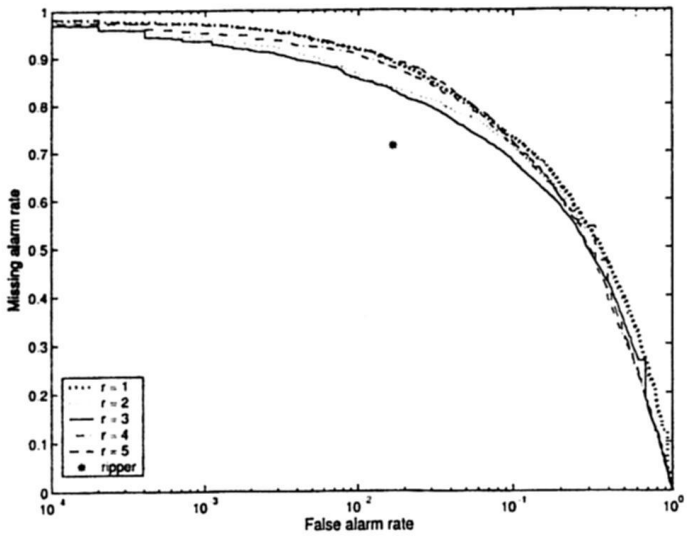

Fig. 4. ROC performance curves comparing NMF using from r = 2 to r = 5 basis and RIPPER

In our experiments, we performed for training 50 iterations (update rules) of the NMF algorithm while for testing phase we only performed 10 iterations. The reason for this reduction in the number of iterations in the test phase relies on the fact that the NMF algorithm converges quickly for normal test data while it does not converge if the test data is abnormal. Table 2 shows the training times of the classifier in seconds for different sizes of the training data. Despite the code for NMF was not optimized, it is observed that our model is faster compared to the classifier using RIPPER. In more detail, our model can learn from different size of normal training data in a short time even if the the amount of audit data is quite large. We measured the time to process test data of the proposed

method for several amounts of data, the results are listed in Table 3. From the results it can be concluded that performance of the NMF based method is very attractive for real time applications where actions for response must be taken as soon as possible.

Training Data	NMF	RIPPER
40000	7.0300	364.4
20000	3.1040	309.88
10000	2.1330	233.25
5000	1.4930	102.04
1000	0.9820	43.17
500	0.9310	30.34

Table 2. Learning time(seconds) NMF vs. RIPPER

NMF would be suitable for scenarios where computational resources are limited since the processing is very fast. Another advantage of using NMF is that if we want to update our model, we only need to update the matrix $W_{training}$, which results in a reduced computational cost.

Test Data	Processing time
10000	5.7480
5000	2.4130
1000	0.4400
500	0.2400

Table 3. Testing time(seconds) using NMF for different size of testing data

5 Conclusion

The experiment results demonstrate that Non-Negative Matrix Factorization might work on MANETs. Normal behavior profiles of a routing protocol can be established and used to detect anomalies. However, more research has to be done to obtain operating points with both low false and missing alarm rates. Thus, less error rates should be reached to meet the typical goals for a real IDS. It is suggested that the preprocessing stage should be improved for better performance of the detection methods. On the other hand, the model can easily achieve real-time intrusion classification based on dimensionality reduction and on a simple classifier. In the proposed method, NMF reduce the high dimensional

data vectors and classifies in a low dimensional space with high efficiency and low usage of system resources. The NMF algorithm does not seem sensible to parameter selection, furthermore a small amount of data is sufficient to train the classifier without reducing the performance. Updating the profiles can be easily implemented in the NMF model, a number of iterations of the algorithm can be used for updating purposes.

The low computational expense of the processing allows a real-time performance of intrusion classification. Future work might involve research considering more attack scenarios in MANETs, not only at the routing layer, but also at other layers. More security-related features can be drawn after the analysis of the threats. This could facilitate the construction of better detection models.

Acknowledgments. The authors would like to acknowledge the Cátedra de Biométricas y Proto colos Seguros, ITESM, Campus Monterrey, Cátedra de Seguridad, ITESM, Campus Estado de Mexico and Regional Fund for Digital Innovation in Latin America and the Caribbean (FRIDA), who supported this work.

References

1. P. Albers, O. Camp, J. M. Parcher, B. Jouga, L. Me, R. Puttini, "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches", The 1st International workshop on Wireless Information Systems" (WIS 2002), in the 4rth International Conference on Enterprise Information Systems, 2002.
2. W. Wang, Y. Lu, B. K. Bhargava, "On Vulnerability and Protection of Ad Hoc On-demand Distance Vector Protocol", International Conference on Telecommunications (ICT'2003), France, March 2003 pp. 375 - 382 Vol.1.
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. Hongmei Deng, Qing-An Zeng, Agrawal D.P., "SVM-based Intrusion Detection System for Wireless Ad Hoc Networks", Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th, Volume 3, 6-9 Oct. 2003 pp. 2147 - 2151 Vol.3
5. Buschkes R., Kesdogan D., Reichl P., "How to increase security in mobile networks by anomaly detection", Computer Security Applications Conference, 1998, Proceedings., 14th Annual 7-11 Dec. 1998 pp. 3 - 12
6. D. D Lee and H Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization", Nature, vol. 401, pp. 778-791, 1999.
7. Wei Wang, Xiaohong Guan, Xiangliang Zhang, "Profiling program and user behaviors for anomaly intrusion detection based on non-negative matrix factorization", Decision and Control, 2004. CDC. 43rd IEEE Conference on Volume 1, 14-17 Dec. 2004 pp. 99 - 104 Vol.1
8. H. Debar, M. Dacier, A. Wespi, "Towards a taxonomy of intrusion-detection systems", Computer Networks 31, 1999, pp. 805-822.
9. S. Marti, T.J. Giuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", Proceedings of MobiCom 2000, pp. 255-265.
10. W.W. Cohen, "Fast effective rule induction", in: Proceedings of the 12th International Conference on Machine Learning (Morgan Kaufmann, San Mateo, CA, 1995) pp. 115-123.