On an IDS Model for Mobile Ad Hoc Networks

Fabio Buiati[1], Javier García Villalba[1], Robson de Oliveira[1], Helena Rifà-Pous[2]

[1] Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Sistemas Informáticos y Programación (DSIP)
Facultad de Informática, Universidad Complutense de Madrid (UCM)
C/ Profesor José García Santesmases s/n
Ciudad Universitaria s/n, 28040 Madrid
fabio@fdi.ucm.es, javiergv@sip.ucm.es, robson@fdi.ucm.es
http://www.ucm.es/info/gass
[2] Safelayer Secure Communications S.A.
Ed. World Trade Center S4
Moll de Barcelona s/n
08039 Barcelona (Spain)
hrifa@safelayer.com

Resumen Manet security has a lot of open issues. Due to its characteristics, this kind of network needs preventive and corrective protection. In this paper, we focus on corrective protection proposing an anomaly IDS model for Manet. The design and development of the IDS are considered in our 3 main stages: normal behavior construction, anomaly detection and model update. A parametrical mixture model is used for behavior modeling from reference data. The associated Bayesian classification leads to the detection algorithm. MIB variables are used to provide IDS needed information. Experiments of DoS and scanner attacks validating the model are presented as well.

1. Introduction

Security of Mobile ad hoc networks (Manet) is an active topic in recent research. Most of current work on Manet security focuses on some kind of preventive protection design (e.g. authentication [1]). However, as network entities in a Manet consist of general purpose hardware and software equipments, usually without good physical protection, occurrence of malfunctioning and compromised entities in such networks cannot be neglected. Therefore, security must be designed in a way that the network service remains robust even in presence of misbehaving nodes. In general treat models, the compromising of a network entity leads to revealing all confidential information to the intruder, which allows for most of preventive security mechanisms to fail. Intrusion detection and response systems (IDS) are a common approach in such scenarios where a corrective security mechanism is required to cope with the limitations of preventive only security mechanisms.

In respect to the IDS design, two basic approaches can be considered: misuse and anomaly intrusion detection. In misuse detection, an attack signature must be explicitly provided, leading to a positive identification of an attack occurrence. If the source of the attack (e.g. compromised node) can also be identified as part of the detection process, a simple corrective (response) action consists in excluding the attacker node from the network. This is the case for security systems based on the preventive and corrective protection by combination of strong authentication and misuse IDS [2]. Anomaly detection has a completely different base. The current behavior of the monitored system (e.g. network) is repeatedly compared with some reference behavior, which is previously stated (normal behavior). In this case, as existence of attacks is not explicitly realized, the problem source cannot be precisely identified. Thus, corrective (response) actions must concentrate on mitigation of attack effect.

In this paper, we propose the design of an IDS following the anomaly detection approach. We are especially interested in detecting anomalous network traffic behavior due to packet flooding (e.g. DoS) and scan attacks in mobile ad hoc networks.

Our first contribution is the presentation of an anomaly IDS conception. This design is based on statistical modeling of reference behavior using mixture models [3] in order to cope with an observable traffic composed by mixture of different traffic profiles due to different network applications. The detection algorithm is based on Bayesian classification criteria.

The second contribution is the adaptation on the statistical model in order to model network traffic behavior in Manet. Standard MIB variables are used as observations of the traffic behavior (during reference model establishment and detection). Simulations with ns-2 are conducted in order to validate this approach.

The remaining of this paper is organized as follows: Section 2 gives an overview of related works. Section 3 presents the anomaly IDS design. Section 4 presents the Manet traffic characterization and defines the behavior model construction. Finally, section 5 presents our conclusions and proposed future works.

2. Related Work

The IDS project for Manet is not a complete new issue and this subject has already been treated recently. Y. Zhang and W. Lee [4] introduce the basic requisite for this special kind of IDS. This architectural design was explored in V. Mittal and G. Vigna [5] who present an IDS formed by various sensors to detect attacks against the routing protocol that monitors promiscuously the network links. In a previous work, R. Puttini *et al.* [6] present the design of a fully-distributed IDS architecture.

In [7], G. Vigna *et al.* proposes an IDS for Manet that is essentially projected to reinforce the security of the routing protocol. In [2], Puttini R *et al.* propose a new security model for protection of Manet routing protocol. The salient features in this design are: combination of preventive and corrective protection, self-organized conception of security services and fully localized solutions. In the work at [.8] it is presented a security solution based in a modified version of AODV that uses a mechanism of intrusion detection combined with a token system that is used to grant the node access

to the routing services. However, this solution does not incorporate any preventive solution (authentication).

Y. Huang et al. [9] and C.-Y.Tseng et al. [10] present projects of IDS for Manet based on detection by anomaly strategy. Finally, a strategy of detection and response to intrusion to deal with non-cooperative nodes in ad hoc networks is presented by S. Marti et al. [11].

In this paper we present a completely new anomaly IDS design, based on statistical models for detecting DoS and scan attacks in Manet networks.

3. Anomaly IDS Design

This section presents our anomaly IDS model [3]. The idea is to build a behavior model that takes into account multiple use profiles and allows *a posteriori* Bayesian classification of data as part of the detection algorithm. A reference audit data set representing the normal system behavior is used to create the model with a learning procedure[1].

Before starting to describe the model, we should note that audit data must be mapped into random variables (e.g. into a number-based domain). Hereafter, we admit that audit data can be represented by a set of realizations of a continuous random vector **y**, which probability distribution function (pdf) will be modeled[2].

A. Behavior Model

Parametrical Mixture Model and EM-Algorithm

In our behavior model, the pdf of the (d-dimensional) random vector **y**, whose realizations are mapped from the audit data domain, are represented by a parametrical mixture model [12]. The mixture model fundamental equation, giving the probability of $\mathbf{y}_i$, can be formally expressed as:

$$p(\mathbf{y}_i) = \sum_{k=1}^{K} p(z_k) g_k(\mathbf{y}_i, \boldsymbol{\theta}_k) \cdot \quad (\mathbf{1})$$

Where: yi is the i-th observed data; z is the hidden vector that indicates which source (profile) data comes from (e.g. zk = 1 if data comes from cluster k and z = 0, otherwise); gk are kernel distribution functions with respective parameters θk, each of

1 Obtaining good initial reference information set is not straightforward as assuring a data set to be representative for every expected behavior is usually difficult.

2 Some data types are numerical by nature and are easily mapped. In this paper we admit input (reference and activity) data to be numerical, continuous and unbounded. This is not the case for every data type founded in real systems and special mapping and distributions are need when dealing with non-numerical, non-continuous or bounded data.

them modeling one of the use profiles; K is the model order corresponding to the number of sources being modeled.

The unknown parameters in the model (Equation (1)) are the set of cluster probabilities ($p(z_k)$) and the parameters of kernel distribution functions of each cluster ($\boldsymbol{\theta}_k$), represented by $\boldsymbol{\Psi} = [p(z_1), p(z_2),..., p(z_k), \boldsymbol{\theta}_1, \boldsymbol{\theta}_2,..., \boldsymbol{\theta}_k]$. An iterative algorithm of optimizing the unknown vector $\boldsymbol{\Psi}$ by a maximum likelihood (ML) criterion has been defined and is called the expectation-maximization (EM) algorithm [13]. We let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2,..., \mathbf{y}_n]^T$ be an observed n-dimensional realization vector of $\mathbf{y}$ (which we like to model). $\mathbf{Y}$ is regarded as the reference data containing representative normal behavior information and are used to fit $\boldsymbol{\Psi}$ using the EM algorithm. This algorithm permits both log-likelihood and model parameter estimation to be done in an iterative manner. A detailed discussion of the EM-algorithm is out of the scope of this paper. The reader is asked to refer to [3,4] for a more general description of the EM-algorithm.

In the particular case of Gaussian mixture models (GMM), the Equation (1) should be rewritten replacing the general distrbutions (g_k) by the normal distribution (represented by ϕ) and the distribution parameters $\boldsymbol{\theta}_k$ by the mean vector ($\boldsymbol{\mu}_k$) and covariance matrix ($\mathbf{R}_k$), as stated at Equation (2), where the probabilty $p(z_k)$ are also replaced by the pondering factor w_k, for simplicity of notation.

$$p(\mathbf{y}_i) = \sum_{k=1}^{K} w_k \phi(\mathbf{y}_i, \boldsymbol{\mu}_k, \mathbf{R}_k) \tag{2}$$

For completeness, we provide the EM recursion equations (Equations (3)-(6)) for the Gaussian mixture models:

$$p(k \mid \mathbf{y}_i) = \frac{w_k^i \phi(\mathbf{y}_i, \boldsymbol{\mu}_k^i, \mathbf{R}_k^i)}{\sum_{k'=1}^{K} w_{k'}^i \phi(\mathbf{y}_i, \boldsymbol{\mu}_{k'}^i, \mathbf{R}_{k'}^i)} \tag{3}$$

$$w_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i) / n \tag{4}$$

$$\boldsymbol{\mu}_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i) \mathbf{y}_i \Big/ \sum_{i=1}^{n} p(k \mid \mathbf{y}_i) \tag{5}$$

$$\mathbf{R}_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})^T \Big/ \sum_{i=1}^{n} p(k \mid \mathbf{y}_i) \tag{6}$$

Optimal Entropy-Based Estimation of Model Order

For the propose of the EM-algorithm, the model order K must be provided because it is useful to be able to estimate the most probable number of partitions.

As described in [14], this "ideal partitioning" should be obtained by minimizing Shannon entropy given observed data, which can be evaluated for each observation by Equation (7):

$$H_K = -\sum_{k=1}^{K} p(k \mid \mathbf{y}_i) \log(p(k \mid \mathbf{y}_i)) \tag{7}$$

The expected value of this entropy is evaluated taking the mean of H_K over all observed data[3] (Equation(8)):

$$E^*(H_K) = -\sum_{i=1}^{n} \sum_{k=1}^{K} p(k \mid \mathbf{y}_i) \log(p(k \mid \mathbf{y}_i))/n \tag{8}$$

where: E^* denotes an expectation estimator and H_K is the measure we are interested in.

We proceed fitting K_{max} models with different order (K = 1, 2, ..., K_{max}) and we evaluate the expected entropy (8) for each of them. The model which results in a minimum of this measure will be considered the optimum model. The complete algorithm of the *learning* phase can be summarized as follows:

EM-Algorithm with Model Order Estimation

1. K = 0, H_{opt} = 0, K_{opt} = 1.
2. K = K+1.
3. Fit the K-order model to data using the EM-Algorithm (eqs. 2-6).
4. Calculate expected value of H_K (Equation (8)).
5. If H_K < H_{opt} then Hopt = H_K ; K_{opt} = K; and $\boldsymbol{\Psi} = \boldsymbol{\Psi}_{opt}$.
6. If K < K_{max}, then repeat (2).
7. Update actual model order K with optimal model order: K = K_{opt}.
8. Update actual model parameters $\boldsymbol{\Psi}$ with optimal model parameters $\boldsymbol{\Psi}_{opt}$.

B. Anomaly Detection

During detection, the behavior model has been already fitted and is available for making inferences about a new data presented to the system. Our objective is to define some penalty λ, which varies from 0 (zero) to 1 (one) (e.g. $0 \leq \lambda \leq 1$), indicating the degree of normality concerning this realization from certainly abnormal ($\lambda = 0$) to a certainly normal ($\lambda = 1$) behavior.

3 This should be easily verified by simple inspection of entropy expression. A formal treatment can be found in [5].

We have defined a detection procedure formed by two basic steps: a (Bayesian) classification inference and a cluster pertinence inference.

The classification inference is straightforward for parametrical mixture models and consists of evaluation of the posterior cluster probabilities conditioned to new data **y'**, p(k | **y'**), for k = (1, 2,..., K).

Cluster pertinence inference is a little more complex. The considered approach consists in evaluating the probability of new data being contained in some pertinence interval (Π_k), defined as a function of cluster distribution parameters (μk and $\mathbf{R}$ k, for example) and the observation y', which should be formally expressed as following (Equation (9)):

$$p(\mathbf{y}' \in \Pi_k \mid k) = \int_{\Pi_k} g_k(\mathbf{y}, \boldsymbol{\theta}_k) d\Pi_k \tag{9}$$

Such probability should, indeed, look like some kind of cumulative distribution function (cdf), if we define Π_k as stated in Equation (10), below[4]:

$$\Pi_k = \left\{ \mathbf{y} \in \Re^d \mid \frac{\left\| (\mathbf{y} - \boldsymbol{\mu}_k) \right\|^2}{\left\| \mathbf{R}_k \right\|} \geq \gamma^2 \right\} \tag{10}$$

where: $\| \ \|^2$ and $\| \ \|$ denote some type of norm operators, and γ is a constant that should depend on **y'**.

Finally, detection penalty should be defined as (Equation (11)):

$$\lambda(\mathbf{y}') = \sum_{k=1}^{K} p(k \mid \mathbf{y}') p(\mathbf{y}' \in \Pi_k \mid k) \tag{11}$$

4. Manet Traffic Characterization and Behavior Model Construction

The goal here is to construct a model of behavior to characterize the normal traffic conditions in a Manet. Knowing there isn't a common place about which traffic pattern would be typical in a Manet, the characterization of what would be a normal traffic should be done for each case.

Also, it may be difficult to obtain real samples of Manet traffic in operation which are free of possible intrusion vestiges. An alternative is the execution of simulations. Thus, the pretension here is to validate a behavior intrusion detection process using simulated data.

4 This is a good choice for symmetrical kernel distributions, as the Gaussian distribution used in our experiments. Asymmetrical distributions should have different definitions.

In order to create our normal traffic profile for simulation, we use the following assumptions:

- Control traffic: basically consisted of the traffic generated by the routing protocol (UDP) and ARP (neither UDP nor TCP).
- Applications: four kinds of traffic generated by different applications in all of the network nodes are considered. Their parameters are adjusted to produce an average occupation of the wireless links of around 20% of total capacity.
- The simple remote session (telnet) uses TCP; the generated traffic is bidirectional; the interval between messages is defined by a Poisson process; and multiple sessions are opened between different origins/destinations, being the origin and destination nodes (uniformly distributed), the starting time (Poisson process) and the session burst (normally distributed) randomly defined.
- The blast data transfer (FTP) uses TCP; the "file" size is random (normally distributed); and multiple transfers between different origins/destinations are done, being the origin and destination nodes (uniformly distributed) and the starting time (Poisson process) randomly defined.
- The constant bit rate (CBR) data transfer (videoconference) uses UDP; the CBR rate is fixed at 128 kbps; there are multiple transfers between different origins/destinations, being the origin and destination nodes (uniformly distributed), the starting time (Poisson process) and the session duration (normally distributed) randomly defined.
- The simple application of asynchronous question-answer (ping) uses ICMP; it always send 4 requisitions, separated in time by 1 second; an answer is always sent; and multiple transfers between different origins/destinations are done, being the origin and destination nodes (uniformly distributed) and the starting time (Poisson process) randomly defined.
- Mobility model: the random waypoint algorithm model developed by CMU is adopted[5]. A Manet of 50 nodes in a 250m x 250m area and a transmission range of 50m is used, for simulation purposes, resulting in an average neighborhood of 6.28 nodes.
- The simulation time for model construction is 1000 seconds.

Our objective is to fit a Gaussian mix model to the traffic generated in accordance with the premises defined above, in order to detect traffic anomalies caused by DoS and scan attacks. A crucial issue here is the definition of which variables reflecting the Manet traffic conditions should be modeled (normal behavior characterization) and monitored (detection).

Behavior models are created separately for TCP, UDP, ICMP and IP traffic. As Table 1 shows, for each model a group of pertinent variables is monitored. Table 1 also shows which type of attacks is intended to be detected using a GMM normal behavior model and having as reference data the simulated traffic, generated according to the premises stated above.

5 http://www.monarch.cs.cmu.edu/cmu-ns.html

Table 1. Monitored Variables.Types of attacks that are intended to be detected using GMM normal behavior model

Monitored Variables		
Behavior Model	**Variables to be monitored**	**Possible detected attacks**
TCP	-number/rate of connections or incomings -each connection duration -tcpInErrs[6] -tcpNoPorts[6]	-TFN and TFN2K -stacheldraht -shaft -mstream -TCP scanner
UDP	-udpInDatagrams -udpInErrs[6] -udpNoPorts[6]	-trinoo -TFN and TFN2K -stacheldraht -shaft -UDP scanner
ICMP	-icmpInEchos -icmpOutEchos -icmpInErrs[6]	-smurf -TFN (ping flood) -stacheldraht -shaft
IP	-ipReasmFails[6]	-TFN2K (Traga3)

5. Implementation and Experimental Results

The figure 1 illustrates the simulation data processing to verify the applicability of the behavior intrusion detection techniques to Manet networks. *trafficgen* is a script that is used to generate the *ns-2* input files, allowing for adjustments in the simulation model (e.g. 50 nodes Manet, 250m x 250m area, transmission range of 50m etc.). The *ns-2* package is used for the simulation and generates a trace file containing all the generated packages, forwarded and received in all of the net nodes (*trafego.out*). However, all the MIB variables must be saved and monitored in each node. Therefore, this file is decomposed in several other files, one for each node of the net, by the *ns2tcpdump* program. Inside each file generated by *ns2tcpdump* only the packages

[6] These variables are observed with zero mean and variance in construction data of the reference model, as there is no error conditions in normal traffic generated by simulation. The use of these variables generates singularities in the maximization function of the EM algorithm and, therefore, they are avoided in the results presented in this paper. In real networks, however, these variables present not null values, reflecting the occasional faults of the monitored system/network.

generated, received or forwarded by the same node are actually written. These files are equivalent to a package dump file captured by a net analyzer with capture interface set to non promiscuous mode. These files also transforms the packages traces generated by *ns-2* in packages that look like those captured by a net analyzer: all the fields of the layers 3 and 4 are fulfilled (including IPv4 with 4 bytes) and an absolute timestamp, compatible with the relative time measure used by *ns-2*, is inserted into each package. The results from *ns2tcpdump* are files **.pcap*, which have the format compatible with raw package dump of the *libpcap* library. Once this format is largely supported by several net analyzers, for instance, ethereal, the files *.pcap can be visualized and analyzed by this tools. After that, each one of this files is processed by the *tcpdump2mib* that produces as output (**.mib* files) a list of samples of the MIB variables values sampled in a time interval that can be defined by parameter passing at the command call.

We assume that each Manet node executes one local instance of the IDS, called L-IDS. The L-IDS data collector executes periodic pooling to a local SNMP agent [6]. This is equivalent to processing the IDS algorithms with the values assumed by the MIB variables that are stored inside the *.mib files. It is important to notice that the sampling period passed to the program *tcpdump2mib* (i.e. for the *.mib file generation) does not have to be the same period of pooling used by the L-IDS extractor module. Actually, the pooling period is a lot bigger than the period used by *tcpdump2mib*.

To make the training and the model adjustment, the training events (variable samples) generated in all network nodes are processed in a single L-IDS, providing an GMM adjustment to the reference data (events) that is independent of the Manet node. The result of this stage is distributed to all L-IDSs in the network.

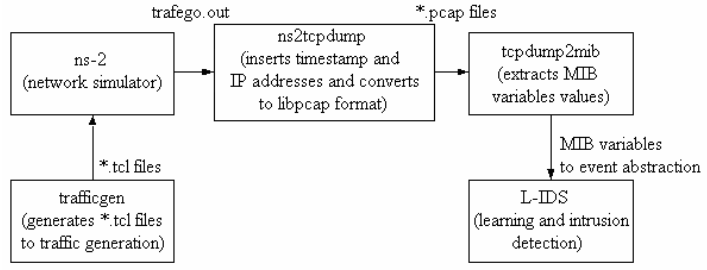

Fig. 1. Simulation process

Two traffic models have been closely analyzed: TCP and UDP. Using these models separately creates an implicit discrimination between all the UDP and TCP generated traffic. Thus, the behavior model using UDP will be useful to model only the videoconference application and the routing protocol. On the other hand, for TCP, the traffic generated by Telnet and FTP applications is modeled.

In case of UDP model, only the udpInDatagrams (UDP datagrams that go into a node) and ipForwDatagrams (IP datagrams forwarded by the node) variables are used. Once this variables are monotonically growing, the observation (updIn; ipForw)

is defined as the learning event generation (realization), which value is obtained subtracting from the present value of "udpInDatagrams; ipForwDatagrams" (current periodic pooling) its predecessor value (previous periodic pooling). The pooling period was adjusted to the same interval of the OLSR TC (equal to three times the HELLO interval, e.g. 6s).

Concerning GMM model for the UDP traffic model, adjusted to the simulation data, the formation of two well defined clusters was observed: the first one, with average of (6,3; 93,9) datagrams and standard deviation of (2,2; 39,7) datagrams. Certainly, this cluster indicates traffic conditions of a node that is not receiving or forwarding any package from the videoconference application. Another cluster, with average of (203; 101) datagrams and with standard deviation of (21,1; 47,1) datagrams resulted from the videoconference traffic (source CBR 128kbps) modeling. Obviously, there is a contribution of the OLSR protocol traffic over the average and the standard deviation of this cluster values. The correlation between the variables are positive, but small (36,7 datagrams).

For the generation of the DoS attack, it is simulated the generation of an UDP CBR (2Mbps) traffic in four randomly chosen origin nodes in direction to an unique destination node. Applying the detention model, anomalous situations are detected in all the nodes that forward the traffic from the origin to the destination. This result is interesting from the point of view of DDoS detention. The detection was only possible thanks to the combined analysis of two variables udpInDatagrams and ipForwDatagrams.

We are also interested in evaluating the response measures that could possibly be activated by the L-IDS in the nodes detecting the attack, in order to mitigate the attack effects. Obviously, the node that receives all generated traffic (from all its neighbors) will quickly become unavailable (the *ns-2* accuses the generation of some forward errors and the disposal of the destined node neighborhood packages). However, although the far nodes are generating/forwarding a non-expressive amount of data, they are not necessarily broken by the attack. As the intrusion detection system identifies anomalies in all nodes in the forward path, these nodes could possible interact to block the forwarding of packages that come from the compromised origin. This forwarding must be blocked based on the enlace address and not based on the IP datagram destination addresses, because these ones are easily faked and, in more advanced DDoS attacks, they are constantly modified (to each package).

In the case TCP model, tcpPassiveOpens (number of open passively connections in the node) and tcpInSegs (number of received segments, including the ones with error and for connection opening) are used as MIB variables. Similarly to the UDP case, a pooling period equal to the OLSR TC interval is defined (e.g. 6s). The observations (tcpPO ; tcpIN) are obtained as the difference between the value of (tcpPassiveOpens; tcpInSegs) in the current and preceding consultation. To avoid singularities (i.e. a formation of a cluster with average zero and small variance for tcpPassiveOpens), the events in which tcpPassiveOpens was equal to zero are discarded as normal in the learning and in the intrusion detection processes. Concerning to the adjustment, in this case, it is observed the formation of two clusters with averages at (1,11; 38,41) and at (1,05; 97,11), shaping respectively the telnet and the ftp.

For the generation of a scanner attack, an origin-destine pair is randomly chosen and this origin sends TCP connection solicitations to the destination, in a rate of 10 solicitations per second. In destination, a drain is made in which, to each 30 connection solicitations, one is accepted (i.e. indicating one "match" with one service that is answering). As long as the values of the MIB variables start to reflect this additional traffic, the attack is detected by the destination node, with a null false negative rate.

Acknowledgements

This work has partially been supported by Programa PROFIT 2005 (Ministerio de Educación y Ciencia, MEC) under Project FIT 360000-2005-65.

6. Conclusions and Future Work

We have presented a new anomaly IDS design for statistic behavior modeling of a network. It uses a parametric Gaussian mixture model for behavior modeling with a Bayesian classification intrusion-detection. This model aims to permit the simultaneous modeling of different types of events (e.g. applications) that have influence on the set of variables available for monitoring.
The preliminary experimental results indicate that this kind of model can be adjusted with a carefully choice of variables to be modeled and monitored. Due to the large cost of monitoring packets in a Manet, we have chosen to use MIB variables. These variables are easily provided by SNMP agents.
However, the proposed intrusion detection model by behavior anomaly is still in its first stages of development and it has just been used with synthetic data that do not represent necessarily the real behavior of a network. Due to this fact, beyond the need of a further validation with real data, the model presents some important limitations that must be investigated and become more flexible. Moreover, the parametric Gaussian mixture model is not suitable for modeling complex data that do not have normal features.
Finally, as future work, we suggest the validation of this model with experiments that use real data. Furthermore, a lot of improvements of model conception pre-conditions can be done, like the use of other types of kernel functions, the use of semi-parametric mixture models [**14**], the adoption of stochastic models (e.g. Markov process) for eliminating the statistic independence pre-condition between the events, among others.

7. References

1. Hao Yang, Haiyun Luo, Fan Y, Songwu Lu, Lixia Zhang, Security in Mobile Ad Hoc Networks: Challenges and Solutions. IEEE Wireless Communications - February 2004 – pp 2-11, 2004.

2. R. Puttini; R. de Sousa.; L. Me – Preventive and Corrective Protection for Mobile Ad Hoc Network Routing Protocols. In Proceedings of 1st International Conference on Wireless On-demand Network Systems in Lecture Notes on Computer Science, Springer, 2004.
3. R. Puttini; Z. Marrakchi and L. Mé - Bayesian Classification Model for Real-Time Intrusion Detection, 22th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MAXENT'2002). August 2002.
4. Y. Zhang and W. Lee – Intrusion detection in wireless ad hoc networks. Proceedings of 6th ACM Annual International Conference on Mobile Computing and Networking (MOBICOM 2000), ACM Press, New York, pp. 275-283, 2000.
5. V. Mittal and G. Vigna. Sensor-based intrusion detection for intra-domain distance-vector routing. In R. Sandhu, editor, Proceedings of the ACM Conference on Computer and Communication Security (CCS'02), Washington, DC, November 2002. ACM Press.
6. R. Puttini; J.M. Percher; L. Me; R. de Sousa - A Fully Distributed IDS for Manet. In Proceedings of 9th IEEE International Symposium on Computers Communications, 2004.
7. G. Vigna, S. Gwalani, K. Srinivasan, E. Royer, R. Kemmerer – A Intrusion detection tool for AODV-based ad hoc wireless network. In Proc. 20th Annual Computer Security Applications Conference (ACSAC2004), 2004.
.8 H. Yang, X. Meng and S. Lu, "Self-Organized Network Layer Security in Mobile Ad Hoc Networks", in the Proceedings of ACM Workshop on Wireless Security – 2002 (WiSe'2002), in conjunction with the ACM MOBICOM2002, September, 2002.
9. Y. Huang, W. Fan, W. Lee, and P. Yu. Cross-feature analysis for detecting ad-hoc routing anomalies. In The 23rd International Conference on Distributed Computing Systems, May 2003.
10. C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. A specification-based intrusion detection system for AODV. In ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), October 2003.
11. S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehaviour in mobile ad hoc networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, Boston, MA, August 2000.
12. G. J. McLachlan, D. Peel, K. E. Basford and P. Adams, "The EMMIX Software for the Fitting of Mixtures of Normal and t –Components", Journal of Statistical Software, v. 04, 1999.
13. Dempster, A. P., Laird, N. M., and Rubin, D. B., Journal of the Royal Statistical Society B 39 ,1-38 (1977).
14. Roberts, S. J., Everson, R., and Rezek, I., Pattern Recognition, 33:5, pp. 833-839 (1999).
15. Johnson, R. A., Wichern, D. A., Wichern, D. W. , "Applied Multivariate Statistical Analysis – 4th Edition", Prentice-Hall, 1998.