ISSN:2231-0711
www.ijcset.net

Secure Leader Election using Machine Learning

Doris Rachel K[1], Dr MHM Krishna Prasad[2]

[1]*Dept of CSE, UCEV – JNTUK*
Vizianagaram, Andhra Pradesh, India

[2]*Head, Dept of IT, UCEV – JNTUK*
Vizianagaram, Andhra Pradesh, India
{dorisrachel.k, krishnaprasad.mhm}@gmail.com

Abstract: **This paper discusses research in developing general and systematic methods for intrusion detection and based on the classification elect the leaders in network for those are classified as high reliability. The key ideas are to use data mining techniques to discover consistent and useful patterns of system features that describe program and user behavior, and use the set of relevant system features to compute classifiers that can recognize anomalies and known intrusions. The performance of the classification algorithms is evaluated under different traffic conditions and mobility patterns for the Black Hole, Forging, Packet Dropping, and Flooding attacks. The obtained experimental results indicate that the Support Vector Machines exhibit high accuracy for almost all simulated attacks and that Packet Dropping is the hardest attack to detect.**

Keywords *:* **MANET, Classification algorithm, detection techniques.**

1. INTRODUCTION

The adoption of Mobile Ad hoc networks (MANETs) has increased in recent years mainly due to their important advantages and their broad applicability. MANETs can be defined as dynamic peer-to-peer networks that consist of a collection of mobile nodes. The nodes employ multi-hop information transfer without requiring an existing infrastructure. Although MANETs are characterized by great flexibility and are employed in a broad range of applications, they also present much inherent vulnerability that increases their security risks. Due to their dynamic and cooperative nature, MANETs demand efficient and effective security mechanisms in order to be safeguarded. Intrusion prevention can be used as a first line of defense in order to reduce possible intrusions but undoubtedly, it cannot eliminate them. Intrusion detection using classification algorithms can help us to effectively discriminate “normal” from “abnormal” behavior and thus, detect possible intrusions. Therefore, intrusion detection, serving as a second line of defense, is an indispensable part of reliable communication in MANETs.

Although some use of classification algorithms was present in all of the previous works, almost none contained comparisons between methods, apart from. Thus, there is a lack of evidence to support the use of one algorithm compared to others, when it comes to intrusion detection in MANETs. Furthermore, there is virtually no data on the performance of such algorithm under different traffic conditions (i.e. mobility, number of malicious nodes), and how such meta-algorithmic parameters such as the sampling interval should be selected. The selection of the sampling interval is particularly important, as there could be a trade-off between good classification performance and quick response.

The security of a computer system is compromised when an intrusion takes place. An intrusion as ``any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource". Intrusion prevention techniques, such as user authentication (e.g. using passwords or biometrics), avoiding programming errors, and information protection (e.g., encryption) have been used to protect computer systems as a first line of defense. Intrusion prevention alone is not sufficient because as systems become ever more complex, there are always exploitable weakness in the systems due to design and programming errors, or various ``socially engineered" penetration techniques. For example, after it was first reported many years ago, exploitable ``buffer overflow" still exists in some recent system software due to programming errors. The policies that balance convenience versus strict control of a system and information access also make it impossible for an operational system to be completely secure.

Intrusion detection is therefore needed as another wall to protect computer systems. The elements central to intrusion detection are: *resources* to be protected in a target system, i.e., user accounts, file systems, system kernels, etc; *models* that characterize the ``normal" or ``legitimate" behavior of these resources; *techniques* that compare the actual system activities with the established models, and identify those that are ``abnormal" or ``intrusive".

2. PROPOSED INVASION DETECTION MODEL

The IDS architecture we adopt is composed of multiple local IDS agents, which are responsible for detecting possible intrusions locally. The collection of all the independent IDS agents form the IDS system for the

MANET. Each local IDS agent is composed of the following components:

Packet Sniffer: is responsible for selecting local audit data and activity logs.

Intrusion Detection Engine: is responsible for detecting local intrusions using local audit data. The local intrusion detection is performed using a classification algorithm. Firstly, it performs the appropriate transformations on the selected labeled audit data. Then, it computes the classifier using training data and finally applies the classifier to test local audit data in order to classify it as "normal" or "abnormal".

Response Engine: If an intrusion is detected by the Detection Engine then the Response Engine is activated. The Response Engine is responsible for sending a local and a global alarm in order to notify the nodes of the mobile ad hoc network about the incident of intrusion.

3. ALGORITHMIC COMPARISONS AND QUALITY METRICS

When comparisons are made between algorithms, it is important to use the same measure of quality. For a given classification algorithm f : X -> Y , where X is the observation space and Y is the set of classes, a common measure of quality is the classification error C measured over an independent test set D,

$$\hat{E}(C|D) = \frac{1}{|D|} \sum_{d \in D} C(f(x_d), y_d),$$

where xd is the observation of example d and yd is its class and C(y0;y) = 0 when
y = y0 and 1 otherwise. However, it is important to note that in most of the literature, the Detection Rate (DR) and the False Alarm (FA) rate are used instead:

$$DR = \frac{TP}{TP+FN}, \qquad FA = \frac{FP}{TN+FP}$$

where TP, TN, FP, FN, denote the number of true (TP & TN) and false (FP & FN) positives and negatives respectively. The goal of an effective intrusion detection approach is to reduce to the largest extent possible the False Alarm rate (FA) and at the same time to increase the Detection Rate (DR).

4. CLASSIFICATION MODELS

A specific instance of an MLP can be viewed simply as a function g: X -> Y, where g can be further defined as a composition of other functions zi : X -> Z . In most cases of interest, this decomposition can be written as g(x) = Kw0z(x) with x2X, w being a parameter vector, while K is a particular kernel and the function z(x) = [z1(x); z2(x); :::] is referred to as the hidden layer. For each of those, we have zi(x) = Ki(v0 ix) where each vi is a parameter vector, V = [v1;v2; :::] is the parameter matrix of the hidden layer and finally Ki is an arbitrary kernel.

$$P(Y = y|X = x, M = m), \qquad y = g(x).$$

The case where there is no hidden layer is equivalent to zi = xi, which corresponds to the Linear model, the second model into consideration. The GMM, the third model under consideration, will be used to model the conditional observation density for each class, i.e. P(X = xjY = y;M = m). This can be achieved simply by using a separate set of mixtures Uy for modeling the observation density of each class y. Then, for a given class *y* the density at each point *x* is calculated by marginalizing over the mixture components *u 2 Uy*, for the class, dropping the dependency on m for simplicity:

$$P(X = x|Y = y) = \sum P(X = x|U = u)P(U = u|Y = y).$$

$$Z = \sum_{y \in Y} P(X = x|Y = y)P(Y = y)$$

It does not depend on *y* and where we have again dropped the dependency on m The fourth model under consideration is the Naıve Bayes model which can be derived from the Gaussian Mixture Model (GMM) when there is only one Gaussian mixture.

5. SIMULATION ENVIRONMENT

In order to evaluate this approach simulated a mobile ad hoc network (MANET). Assumptions included that the network has no preexisting infrastructure and that the employed ad hoc routing protocol is the Ad hoc On Demand Distance Vector (AODV). Simulations are conducted within the JNS-1.7 library. Simulation models a network of 50 hosts placed randomly within an 850 x 850 m2 area. Each node has a radio propagation range of 250 meters and the channel capacity was 2 Mbps. The nodes in the simulation move according to the 'random way point' model. At the start of the simulation, each node waits for a pause time, then randomly selects and moves towards a destination with a speed uniformly lying between zero and the maximum speed. On reaching this destination it pauses again and repeats the above procedure till the end of the simulation. The minimum and maximum speed is set to 0 and 20 m/s, respectively, and pause times at 0, 200, 400, 700 sec. The simulation time of the experiments was 700 sec, thus a pause time of 0 sec corresponds to the continuous motion of the node and a pause time of 700 sec corresponds to the time that the node is stationary.

Each node is simulated to generate Constant Bit Rate (CBR) network traffic. The size of the packets sent by each node varies from 128 to 1024 bytes. The sampling interval dictates both the interval for which the statistical features are calculated, and the period between each

classification decision. Simulation is having four different types of attacks:

Flooding attack: Simulated a flooding attack for multiple paths in the network layer, where each malicious node sends forged RREQ packets randomly to all nodes of the network every 100 msec.

Forging attack: Simulated Forging attack for RERR packets, where each malicious node modifies and broadcasts (to a selected victim) a RERR packet every 100 msec leading to repeated link failures.

Packet dropping attack: Simulated selective packet dropping attack, where each malicious node drops all RERR packets leading legitimate nodes to forward packets in broken links.

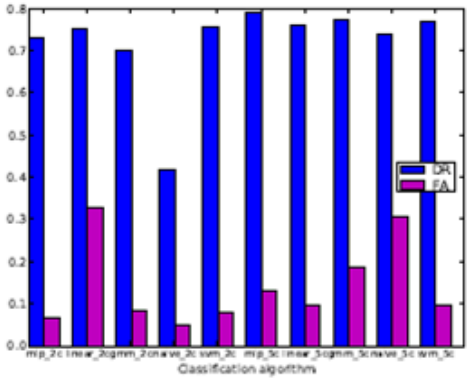
Black Hole attack: In a black hole attack a malicious node advertises spurious routing information, thus receiving packets without forwarding them but dropping them. In the black hole attack we have simulated the scenario where each time a malicious-black hole node receives a RREQ packet it sends a RREP packet to the destination without checking if the node has a path towards the selected destination. Thus, the black hole node is always the first node that responds to a RREQ packet and it drops the received RREQ packets. Furthermore, the malicious-black hole node drops all RREP and data packets it receives if the packets are destined for other nodes.

A very important decision to be made is the selection of feature vectors that will be used in the classification. The selected features should be able to represent the network activity and increase the contrast between “normal” and “abnormal” network activity. Selected the following features from the network layer:

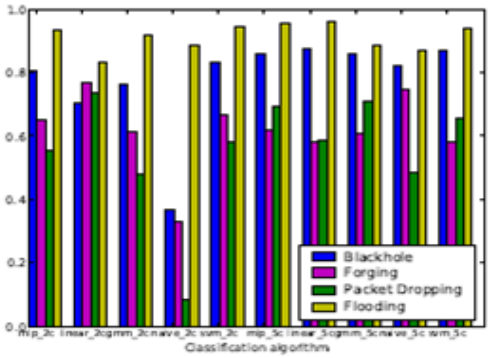
- RREQ Sent: indicates the number of RREQ packets that each node sends.
- RREQ Received: indicates the number of RREQ packets that each node receives.
- RREP Sent: indicates the number of RREP packets that each node receives.
- RError Sent: indicates the number of RError packets that each node receives.
- RError Received: indicates the number of RError packets that each node sends.
- Number of Neighbors: indicates the number of one-hop neighbors that each node has.
- PCR (Percentage of the Change in Route entries): indicates the percentage of the changed routed entries in the routing table of each node.
- PCH (Percentage of the Change in number of Hop): indicates the percentage of the changes of the sum of hops of all routing entries for each node
- =

Based on the above measures every node will be classified as normal or abnormal. Considering the percentage of normal behavior by giving some threshold, elect the node as leader along with energy levels also.

6. SIMULATION GRAPHS



(a) Average Detection Rate and False Alarm



(b) Detection Rate of each type of attacks

Figure 1: Comparison of all Classification algorithms

7. CONCLUSION

This paper presents a performance comparison of five efficient and commonly used classification algorithms. The proposed model used features from the network layer and evaluated the performance of these algorithms for the detection of four serious attacks in MANET’s viz., the Black hole, Forging, Packet Dropping and Flooding attack based on the performance of the testing datasets.

Furthermore, from the experimental observations, it is concluded that the most efficient classifier for detecting all four types of attacks simultaneously is the SVM classifier for multiclass classification although the MLP classifier presents a satisfying Detection Rate (DR) and also a quite high False Alarm (FA) rate. The easiest attack to be detected is the Flooding attack, while the most difficult attack to detect is the Packet Dropping attack. The proposed work also investigated the impact of how the number of malicious nodes in the network and the mobility of the network affect the performance of the classification algorithms in detecting intrusions. Furthermore, the classification algorithms present effective detection of attacks in MANET’s with medium mobility.

REFERENCES

1)J.Mc Dermottand D.Goldschlag,"Towards a model of storage jamming", Proceedings of the IEEE Computer Security Foundations Workshop, Kenmare, Ireland, June 1996, pp. 176-185

[2]Pramote Luenam, PengLiu ,"ODAM: An On-the-fly Damage Assessment and Repair System for Commercial Database Applications", Dept. of Info. Systems, UMBC Baltimore, MD 21250.

[3]W.Lee, SJStolfo, KWMok, "Datamining approaches for Intrusion detection", Proceedings of the 7th USENIX SecuritySymposium, 1998.

[4] P. Liu and S. Jajodia,"Multi-phase damage confinement in Database systems for intrusion tolerance", Proceedings of the 14th IEEE Computer Security Foundations Workshop, June 2001, pp.191 – 205.

[5] Ashoka Savasere, Edward Omiecinski, Shamkant B. Navathe,"An Efficient Algorithm for Mining Association Rules in Large Databases", Proceedings of the 21st International Conference onVery Large Data Bases, San Francisco, CA, USA, pp. 432 – 444,1995.

[6]Yi Huand Brajendra Prasad, "A Data Mining approach for Database Intrusion Detection", ACM Symposium on Applied Computing, 2004, x(y): 711 –716.