Network Intrusion Detection Demystified using Classification Trees

[1]Zainab Assaghir, [2]Antoun Yaacoub, [3]Sara Makki
Lebanese University, Faculty of Science, Applied Mathematics,
Beirut, Lebanon
[1]*zainab.assaghir@ul.edu.lb*, [2]*antoun.yaacoub@ul.edu.lb*, [3]*smakke.ch@gmail.com*

***Abstract*—** In a network environment, intrusions pose a serious security problem especially when new intrusion types are discovered which make them difficult to detect. In this work, we use the Classification And Regression Tree (CART) algorithm, a supervised learning technique, on a labeled dataset collected during an attack on the Faculty of Science's web servers, and this in order to classify bad and good connections. As result, the accuracy of the classification reached 99%, maintaining low false-negative and low false-positive rates.

Keywords*-Attack detection; IDS; Decision tree; data mining; classification.*

*****

I. INTRODUCTION

One of the important technology in business sector is Intrusion detection. Along with cybersecurity, it is considered as one of the most active area in research. Nowadays, a single computer/laptop/pad is part of multiple networked and distributed systems and prone for intrusion; intrusions that are increasing in severity and number.

For this, several protective measure have been put in place like firewalls to check the activities of intruders. However, this technique can not guarantee the full protection of the system.

Hence, in order to monitor networks for intrusions and/or attacks more efficiently, a more dynamic mechanism like Intrusion Detection System (IDS) is used and considered as a second line of defense. This system will allow reporting signs intrusions in order to take the correct action [1].

Intrusion Detection System are either host-based or network-based.

In a host-based environment, IDS operates on data collected form a single computer system. However, in a network-based environment, IDS operates on raw data (i.e. packets or data source) collected from the network.

Simply, IDS try to identify attack patters by the way of two techniques: Misuse or Anomaly. In a well known attack, the IDS detection technique is Misuse, however, when changes are detected in the pattern of utilization or in the behaviour of the system, the IDS detection technique is Anomaly.

Most of the IDS currently are expert based system, in the sense that it can detect known attack types as well as generation of false positive alarms. Consequently, machine-learning techniques (i.e. intelligence technique) were embedded to such systems. These techniques extract useful pattern from data as a reference for good/bad traffic behavior from existing data for subsequent classification of network traffic.

Data mining for automated models of intrusion detection was the first area where intelligence technique was used using association rule [2]. Mukkamala *et al.*, Byunghae *et al.* and Ajith *et al.* used neural networks to study the relationship between given input and output in order to generalize them and to extract new relationship between input and output [3, 4, 5]. Besides that, Ajith *et al.* and Susan *et al.* used fuzzy data mining to generalize relationship between input and output vector based on degree of membership [5, 6]. Ajith *et al.*, Quinlan and Pavel *et al.* used decision tree which is a classification method that learns information from a fixed collection of properties or attributes in a top down strategy [5, 7, 8]. Mukkamala *et al.*, Zhang *et al.* and Byung-Joo *et al.* used support vector machine that creates Maximum-margin hyper planes during training with samples from two classes [3, 9, 10]. Zhang *et al.*, Adetunmbi *et al.*, Sanjay *et al.* Used rough set classification to produce a set of compact rules made up of relevant features only suitable for anomalous and misuse detection [9, 11, 12, 13, 14]. Axelsson and Amor *et al.* used bayesian approaches which are powerful tools for decision and reasoning under uncertain conditions employing probabilistic concept representations [15,16].

Before using any of the machine learning algorithms, one major step should be performed: raw network data should be reduced into records containing some within-connection features such as duration,

service, duration, source IP, destination IP, and so on. For this, the success of any learning algorithm depends thoroughly on the identification of the important features. Note that once this selection will reduce drastically the computational cost, over fitting, model size and leads to increase in accuracy.

Sung *et al.* identified important features for intrusion detection using support vector machines and neural networks [17].

In this paper, we apply decision tree (CART algorithm) to classify good and bad connections, using data consist of records of packets, collected during an attack to the faculty of science's web servers.

This paper is organized as follows: in section II we explain briefly the CART algorithm, in section III, we present the results and tree obtained, and finally we conclude.

II. DECISION TREE

Decision tree is a data mining technique that can be used as classification and regression tools [18]. It consists of dividing the space of input variables (regressors, predictors, etc) into smaller regions, separating the data points according to the target values y. Hence, the model consists of recursive partition of the data and a simple model for each final region.

The recursive partition part is represented as a tree that consists of nodes where each terminal node of the tree, called "leaf" represents a final region. We start at the root node of the tree that contains all data points; this node is decomposed in two daughter nodes according to a question asked about the values of regressors. If these daughter nodes are not converted into leaves, same process is applied and they are decomposed each in two nodes, and so on until all nodes are transformed to leaves. Each point moves down in the tree according to its regressors values (questions' answers), until it reaches a leaf. The predicted value is then determined according to the simple model mentioned above. This model for classification is simply the majority class of the train samples that form that leaf.

The difficult tasks of the tree construction is to find the best questions that can lead to an optimal tree (optimal partition), and to find the criteria upon which the node is considered "pure" and converted to a leaf.

Splitting Rules

Starting at the root node, and then repeating the same procedure at each node, the question to be asked is chosen in a way that maximizes the information about y which means minimizing the impurity of the node, when this question leads to two daughter nodes. A score measure is used assess the importance of the variable and their discriminative ability, and thus to be used to split the node. This score is defined as:

$$score(S,t) = I(t) - \sum_{i=1}^{2} \frac{N_i}{N} I(t_i)$$

Where S is the split used to decompose the node t of size N in two daughter nodes t_1 and t_2 of size N_1 and N_2 respectively and $I(.)$ is the impurity measure of the node. The impurity measures commonly used are Entropy and Gini index, which in our case (binary classification) are defined as follows:

$$I_{Entropy}(t) = -\left(\frac{N_+}{N}\log\frac{N_+}{N}\right) - \left(\frac{N_-}{N}\log\frac{N_-}{N}\right)$$

$$I_{Gini}(t) = \left[\frac{N_+}{N}\left(1-\frac{N_+}{N}\right)\right] + \left[\frac{N_-}{N}\left(1-\frac{N_-}{N}\right)\right]$$

Where N_+ and N_- represents the number of each class (positive and negative) in the node t.

Stopping Criterion

The choice to stop the procedure mentioned previously on a specific node and converting it to a terminal node depends on several factors:

- $I(T) = 0$, which means that all the data points in the node are from the same class.
- $score(S,T) < threshold$, which means that the splitting is not beneficial enough.
- If the number of the records in the node are less than a certain threshold.
- When reaching a maximal size of the tree.

Tree pruning

This construction will always lead to a complex tree that may overfit the train set, and perform poorly on the test set. We can solve this issue by pruning the obtained complex tree. We consider a complexity parameter CP that associates a penalty of having a complex tree. Hence, the quantity we aim to minimize to prune the tree (T) is:

$$R_{CP}(T) = \frac{R(T)}{R(T_0)} + CP \times |T|$$

Where $R(T)$ is the misclassification rate of the tree T, and T_0 is the first node of the tree, $|T|$is the number of leaves of the tree. This quantity represents a trade-off between the complexity of the tree and its performance on the train data.

III. RESULTS

Data consist of 483013 records of connections, collected during an attack on the Faculty of Science's web servers. Data are presented as a sequence of TCP packets, labeled as "bad" representing attacks and system intrusions (82%), and "good" representing normal connections (18%). The data is divided into two parts: a train set composed of 70% of the data to construct the tree and the remaining 30% are used to validate the model and its accuracy.

Each record is described using a set of explanatory variables, out of which the most discriminating ones are

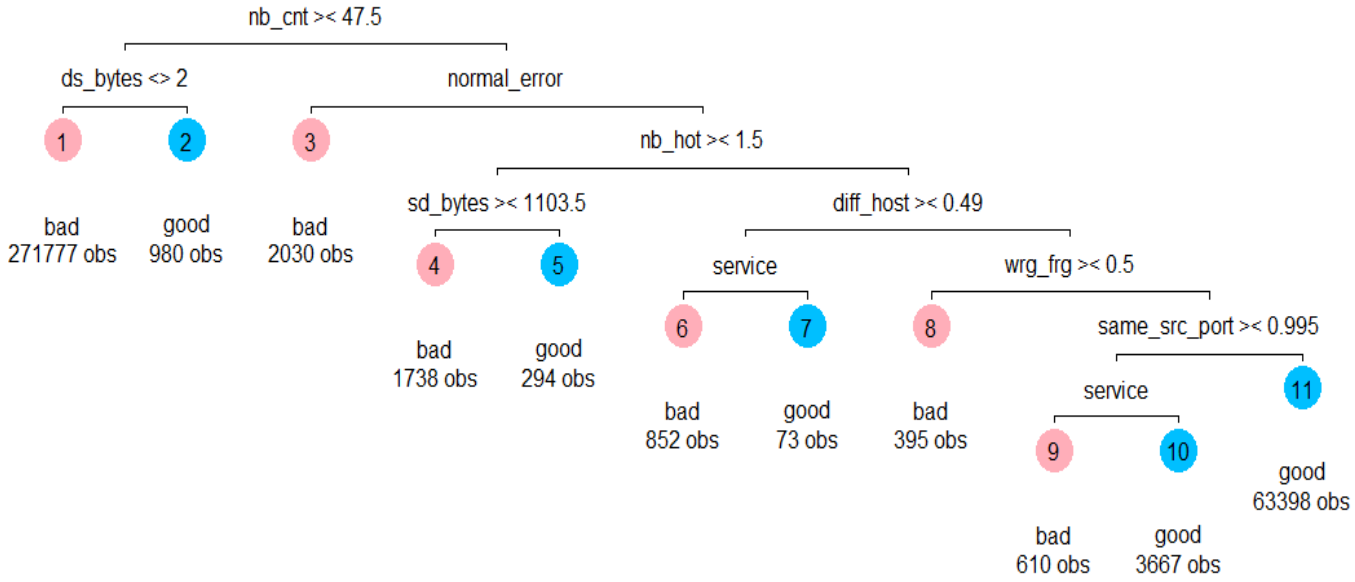
 ISSN: 2321-8169
317 – 320

selected during the construction of the tree. These variables are defined as follows:

- *nb_cnt*: is a continuous variable that describes the number of connections to the same host as the current connection in the past two seconds.
- *ds_bytes*: is a continuous variable describing the number of data bytes from destination to source.
- *normal_error*: is a discrete variable that indicates the normal or error status of the connection. This variable has 11 levels: OTH, REJ, RSTO, RSTOS0, RSTR, S0, S1, S2, S3, SF, SH.
- *nb_hot*: is a continuous variable that indicates the number of "hot" indicators
- *sd_bytes*: is a continuous variable describing number of data bytes from source to destination.
- *diff_host*: is a continuous variable that specifies the percentage of connections to different hosts.
- *service*: is a discrete variable indicating the network service on the destination. This variable has 66 levels including http, ftp, courier, http_443, nntp, etc.
- *wrg_frg*: is a continuous variable indicating the number of wrong fragments
- *same_src_port*: is a continuous variable that specifies the percentage of connections to the same destination host port.

The result of the optimal tree obtained when applied to the test set with $CP = 0.0008$ is shown in fig. 1, it consists of 11 leaves that classifies good and bad connections according to the values of the features mentioned above.

FIGURE I. OPTIMAL TREE OBTAINED



The tree starts with a root node that contains all data (100%), and then is decomposed in two daughter nodes according to values of the variable used to split the data and so on. For example, to use this tree to predict whether each packet is good or bad, if "*nb_cnt*" is less than 47.5, and *normal_error* = REJ,RSTO,S1,S2,S3, SF, and "*nb_hot*" is less than 1.5 and "*diff_host*" is less than 0.49 and "*wrg_frg*" is higher than 0.5 then it's an attack (leaf 8).

When dealing with a classification problem, sensitivity and specificity are two performance metrics that we should take into account. Considering the "*bad*" connections as the positive class and the "*good*" connection as the negative class. Sensitivity or true positive rate is the proportion of positives correctly classified as positives, and specificity or true negative rate is the proportion of negatives correctly classified as negatives, and they are defined as follows:

$$Sensitivity = \frac{number\ of\ true\ positives}{total\ number\ of\ actual\ bad\ connections} = \frac{TP}{TP + FN}$$

$$Specificity = \frac{number\ of\ true\ negatives}{total\ number\ of\ actual\ good\ connections} = \frac{TN}{TN + FP}$$

This tree achieves an accuracy rate of 99.8% corresponding to the following classifications shown in the confusion matrix (tab. 1), with a sensibility equal to 0.998 and specificity equal to 0.998.

TABLE I. *The confusion matrix*

		Actual	
		bad	***good***
Predicted	***bad***	118764 (True Positive TP)	55 (False Positive FP)
	good	218 (False Negative FN)	29170 (True Negative TN)

IV. CONCLUSION AND DISCUSSION

In this paper, we applied decision tree (CART algorithm) to classify *good* and *bad* connections, using data consist of records of packets, collected during an attack to the Faculty of Science's web servers. The data consisted of several variables. The most discriminating ones were selected during the construction of the tree. We showed that the accuracy of the classification reached 99% with high sensitivity and specificity rates.

However, the manipulated data was labeled using just two classes: *bad* or *good*. In the future, we tend not to label all the attacks as *bad*, we seek to broaden our classes and includes other types of intrusions like: buffer overflow, IP sweep, port sweep, root kit and many others.

REFERENCES

[1] Bace, R. and Mell, P. (2001). Intrusion Detection System, NIST Special Publications SP 800. November.

[2] Lee, W., Stolfo, S.J. and Mok, K. (1999). Data Mining in work flow environments: Experiments in intrusion detection. In Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining.

[3] Mukkamala, S., Janoski, G., Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In: Proceedings of IEEE International Joint Conference on Neural Networks, pp. 1702–1707.

[4] Byunghae, C., kyung, W.P. and Jaittyun, S. (2005) Neural Networks Techniques for Host Anomaly Intrusion Detection using Fixed Pattern Transformation in ICCSA. LNCS 3481. 254-263.

[5] Ajith, A., Ravi J., Johnson T. and Sang, Y.H. (2005). D-SCIDS: Distributed soft computing intrusion detection system, Journal of Network and Computer Applications, Elsevier, pp. 1-19.

[6] Susan M. B. and Rayford B.V. (2000). Intrusion detection via fuzzy data mining, Proceedings of the 12th Annual Canadian Information Technology Security Symposium, Ottawa, Canada, June 19-23, 2000, PP.109-122.

[7] Quinlan, J.L. (1993) C4.5 Program for Machine Learning, Morgan Kaufmam Publishers, Inc.

[8] Pavel L., Patrick D., Christia S. and Konrad R. (2005). Learning Intrusion Detection: Supervised or Unsupervised?, International Conference on image analysis and processing, (ICAP), Italie, 2005 (3617) pp. 50-57.

[9] Zhang, L., Zhang, G., YU, L., Zhang, J. and Bai, Y. (2004) Intrusion detection using Rough Set Classification, Journal of Zhejiang University SCIENCE, 5(9):1076-1086

[10] Byung-Joo, K., and Il-Kon, K. (2005) Machine Learning Approach to Real time Intrusion Detection System in Lecture Note in Artificial Intelligence, volume 3809 Edited by S.Zhang and R.Jarvis, Springer-verlag Berline, Heidelberg, pp. 153-163.

[11] Adetunmbi, A.O., Falaki, S.O., Adewale, O.S. and Alese, B.K. (2007a) A Rough Set Approach for Detecting known and novel Network ntrusion, Second International Conference on Application of Information and Communication Technologies to Teaching, Research and Administrations (AICTTRA, 2007) eds. Kehinde, L.O., Adagunodo, E.R. and Aderounmu, G.A., OAU, Ife, pp. 190 – 200.

[12] Adetunmbi, A.O., Alese, B.K., Ogundele, O.S and Falaki, S.O. (2007b). A Data Mining Approach to Network Intrusion Detection, Journal of Computer Science & Its Applications, Vol. 14 No. 2. pp 24 -37.

[13] Adetunmbi, A.O., Falaki, S.O., Adewale, O.S. and Alese, B.K. (2008) Intrusion Detection based on Rough Set and k-Nearest Neighbour, International Journal of Computing and ICT Research, vol. 2. pg 61-66

[14] Sanjay, R., Gulati, V.P. and Arun, K.P. (2005) A Fast Host-Based Intrusion Detection System Using Rough Set Theory in Transactions on Rough Sets IV, LNCS 3700, 2005, pp. 144 – 161.

[15] Axelsson, S. (1999). The Base –rate Fallacy and Its Implication for the Difficulty of Intrusion Detection, In the proceeding of the 6th ACM Conference on Computer and Communication Security. Pp. 127 -141.

[16] Amor, N.B., Beferhat, S. and Elouedi, Z.(2004) Naïve Bayes vs Decision Trees in Intrusion Detection Systems, ACM Symposium on Applied Computing, pp. 420 – 424.

[17] Sung, A.H. and Mukkamala, S. (2003) Identifying Important Features for Intrusion Detection using Support Vector Machines and Neural Networks, IEEE Proceedings of the 2003 Symposium on Applications and the Internet.

[18] Breiman L., Friedman J., Stone C., and Olshen R. (1984), "Classification and regression trees".