© 2014,
Scienceline Publication
www.science-line.com

ISSN: 2322-5114
Journal of World's Electrical Engineering and Technology
J. World. Elect. Eng. Tech. 3(2): 47-52, 2014

JWEET

Comparison of Bayesian Networks and Lazy Model Algorithms in Intrusion Detection Systems Based on Data Mining

Hasan Fazli-Maghsoudi[1*], Hossein Momeni[2]

[1] Master student in Information Technology, Department of Information Technology, University of Science and Technology of Mazandaran, Babol, Iran
[2] Assistant Professor of Golestan University, Gorgan City, Golestan, Iran

*Corresponding author's Email: Fazli.m099@gmail.com

Abstract *– By development of information technology, network security is considered as one of the main issues and great challenges. Intrusion detection systems are a major component of a secure network. Traditional intrusion detection systems cannot adapt themselves to new attacks thus today's Intrusion detection systems have been introduced based on data mining. Identifying patterns in large volumes of data is a great help to us. Data mining techniques by identifying a binary label (normal packet, abnormal packet) and specifying attributes by classification algorithms can recognize the abnormal data Therefore, the precision and accuracy of intrusion detection systems will increase, thereby increasing network security. In this paper, we compare the performance of the different lazy model-based algorithms and Bayesian networks on their data sets. Obtained results show that the HNB algorithm has the highest precision of 83.29% for the intrusion detection system.*

Keywords*: Intrusion Detection, Data Mining, Lazy Model, Normal, Abnormal, Bayesian Networks*

ORIGINAL ARTICLE
Received 18 Jan. 2014
Accepted 14 Feb. 2014

INTRODUCTION

By rapid growth of computer networks and growth of the internet, network attacks, especially on the Internet has increased. Intrusion detection systems have been developed to ensure secure storage and processing of data on the network. Denial of service attacks on the Web is one of the most important attacks [1]. A secure network must have features like data integrity, data availability, and data accuracy. The availability of data is the same as addressing and prevention of service denial attacks. Considering renovation of the attacks we must use learning systems in intrusion detection system which has capability of mining pattern of previous attacks, and can detect new attacks [2].

In this paper we provide a model based on data mining. Initially dataset preprocessing is conducted and then using lazy model algorithms and Bayesian network with respect to the accuracy parameters we propose the best algorithm. In the rest of this paper, at first the fundamental principles and concepts are explained, related works are presented in Section III, in Section IV the simulation and data sets are explained. Section V includes the simulation results, and finally, the last section concludes the paper.

PRINCIPLES AND FUNDAMENTAL CONCEPTS

Intrusion detection is the process of monitoring the events occurring on a network or computer system in order to detect the deviation of security policies. Intrusion detection systems are an application with the ability to identify, detect and respond to unauthorized or abnormal activities associated with the system [1]. Intrusion detection process is shown below:

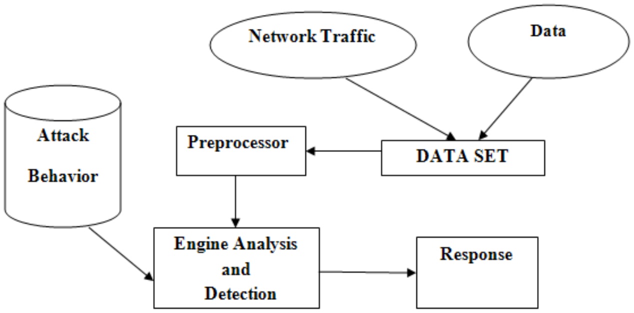

Fig. 1- Block diagram of Intrusion detection

Two general approaches exist to implement intrusion detection test as

- Misuse detection

To cite this paper: Fazli-Maghsoudi H, Momeni H. 2014. Comparison of Bayesian Networks and Lazy Model Algorithms in Intrusion Detection Systems Based on Data Mining. *J World Elec. Eng. Tech.*, 3 (2): 47-52.
Journal homepages: http://www.jweet.science-line.com/

The performance of this technique to identify the attack and define a model for analyzing engine and search for a series of events which correspond with predetermined pattern.

- Anomaly Detection

The performance of this technique is to identify the normal operation of the system and provision of a profile of normal behavior for analysis engine and search for operating abnormalities [1]. Data mining is the process of finding knowledge from vast amounts of data stored in databases, data warehouses or other information repositories[2].

A. *Lazy model*

In an overall view the classification can be divided into two groups, eager and lazy. In the keen type a model data in the training phase is constructed. Decision tree is an example of this model. In a lazy model training samples received and stored and only are used for classification. In fact a model of data is constructed and learning is delayed until classification. We call this type of classification, sampled-based learning. The difference between these two models is that the keen type spends a long time to build the model and acts fast in classification and lazy kind spends much time on classification[2].

*A.*1 *Review of several lazy model algorithms*

The K-nearest neighbor (KNN) algorithm consists of the following three steps:

- Calculate the distance between the input sample with all the training samples
- Arrange training samples based on the distance and select *K* nearest neighbor
- Using the class which has majority of those in the next-door neighbors, as an estimate for the input sample

In the first step in *K*NN method, the distance between the input sample and all the training samples is computed. To do this we define the distance between the two samples. Euclidean distance between the two samples with *n* feature is calculated as

$$D\left(x_1,x_2\right)=\sqrt{\sum_{i=1}^{n}\left(x_1^i-x_2^i\right)^2} \qquad (1)$$

K nearest neighbors are selected, and new data belongs to the group that has maximum number of training data [2].

B. *Algorithms to ensure the absence of distorted data*

In algorithm which we have discussed about it, if *K* is sufficiently large so the distorted data cannot have a large impact on the outcome. But the big challenge is to find the appropriate *K*. Following we introduce the algorithms described which are based on the assumption that samples with better performance in categorizing are kept in the training set.

*B.1 Algorithm IB*3

This is actually a pre-processing algorithm on training data, indeed if *T* is the training set, we actually keep a subset of its *s*. The algorithm is as follows:

```
For each instance $t$ in $T$
Let a be the nearest acceptable instance in $S$ to $t$
(if there are no acceptable instance in $S$, let a be a random instance in $S$)
If class($a$)!=class($t$) then add to $S$
For each instance s in $S$
    If $s$ in least as close to $t$ as a $i$
    Then update the classification record of $s$
    And remove $s$ from $S$ if its classification record is significantly poor.
Remove all non-acceptable instances from $S$.
```

Adding and removing elements of S with respect to the sample success rate and the success rate of default occurs. The success rate of sample is defined as follows:

$$p=\frac{\left(f+\frac{z^2}{2N}+z\sqrt{\frac{f}{N}-\frac{f^2}{N}-\frac{z^2}{4N^2}}\right)}{\left(1+\frac{z^2}{N}\right)} \qquad (2)$$

In this relation, the value of z is obtained from the table of normal distribution. Variable f is classifier precision in N times test[2].

*B.*2 *K-D tree method*

Low speed is the problem of above mentioned algorithms that directly correlates with the number of training samples, in other words has complexity of order *O*(D)[2]. If D is the size of the training set, to solve this problem we use the method of K-D tree. This method builds tree based on training samples which its nodes are samples and K is the number of qualities. In fact samples are considered as points in k-dimensional space. This binary tree partitions the input space into sections. The general pattern is that in each step a feature is selected and segmentation based on that is re-done. All divisions are parallel and eventually every region has at most one point [2]. The algorithm is as follows:

```
Function kdtree (list of points point list, in $t$ depth)
P
{if point list is empty
    Return null
Else
{
//select aixs based on depth so that axis cycles through all valid values
Var int axis=depth mod k
//sort point list and choose median as pivot element
Select median from point list
```

```
    //create node and construct sub tree
    Var tree_node node;
    node.location=median
    node.leftchild=kdtree(points in pointlist before median ,depth+1)
    node.rightchild=kdtree(points in pointlist after median ,depth+1)
    return node;
    }
    }
```

In the recursive algorithm, at each step a feature is alternatively and based on the depth selected. The median is calculated and finally Wall recursively for the left and right parts of median and with increasing depth is recalled. Indeed, this is an index for quick searching.

C. Method of Bayesian network classification

At first consider the simple Bayes. Suppose A_1 to A_n are attributes with discrete values, these values are used to predict a class of discrete C. Our goal is to predict and select the category that following equation becomes the maximum.

$$P\left(C = c \mid A_1 \cup A_2 \cup \ldots \cup A_n\right) \tag{3}$$

Using the Bayesian rule, we have:

$$P\left(C = c \mid A_1 \cup A_2 \cup \ldots \cup A_n\right) = \frac{P\left(A_1 = a_1 \cup \ldots \cup A_n = a_n\right) \times P\left(C = c\right)}{P\left(A_1 = a_1 \cup \ldots \cup A_n = a_n\right)} \tag{4}$$

The denominator is ineffective for making decisions. Because C is the same for all values. On the other hand due to our independent feature set, we have

$$P\left(A_1 = a_1 \cup \ldots \cup A_n = a_n \mid C = c\right) = P\left(A_1 = a_1 \mid C = c\right) \times \ldots \times P\left(A_n = a_n \mid C = c\right) \tag{5}$$

In general, if we consider C as indicator for classification issues, the goal is to maximize the value of $P\left(X \mid C\_i\right) \times P(C\_i)$ which x is other characters. The advantages of simple Bayes are easy implementation and good results for many applications. The disadvantages one can say that perhaps all the features are not independent of each other and there is dependence, in this case the model is poor[2].

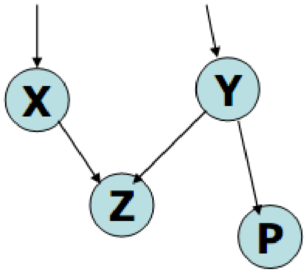

Fig. 2- Bayesian networks

Bayesian networks describe the conditional dependencies between variables (attributes). By using this network, prior knowledge in the field of dependencies between variables with training data classification model are combined [2].

On Bayesian networks, nodes are variables that each of them has a specific set of conditions which are incompatible pairwise. Arcs (edges) indicate variable dependencies to each other. There are local distribution probabilities for each node which depends on the node and is independent from parents' status [2]. An important assumption of a simple Bayes is conditional independence of classes from each other. But in practice, there are dependencies between variables. Possible Bayesian networks explore these possibilities. A Bayesian network consists of two parts: the non-cyclic graph and conditional probabilities. If an arc is connected to the Y from Z, it means that Y is the father Z.

Each arc shows cause and effect relationship between associated variables. For each variable A with parents B_1, B_2, ..., B_n, there is a conditional probability table. In this table, for each variable its relationship considered with its parents. Suppose given x with attributes x_1, x_2, ...,x_n, in this case:

$$P\left(x_1, x_2, \ldots, x_n\right) = \prod_{i=1}^{n} P\left(x_i \mid parents\left(y_i\right)\right) \tag{6}$$

To learn these networks several scenarios exist: One is giving experts to fill the conditional probability table and drawing the related graph. Another method is to use heuristic methods such as hill climbing.

Related works

In this area, a lot of work has been done. Of course, data sets and algorithms used in this paper is somewhat different.

A. Lazy model

A lot of work have been done in this case but the data sets and algorithms used in this paper are somewhat different. These algorithms are mostly applicable in medical issues. In [3] its application to retrieve data from a medical information system is presented. In study of Merschmann [4] an algorithm is proposed for classification of proteins using this model. In study of Zhang [5] algorithm KNN is used for tagging multiple attributes for processing text. In study of Zhang [5] lazy algorithms are used for data mining so that they are labeled as having several traits by considering separateness of labeled attributes.

B. Bayesian networks

In study of Kruegel [7] Bayesian model has been used for intrusion detection in a way that simple Bayesian model a little has been modified and the answer is more accurate. In[8] the method of game theory with Bayesian model for intrusion detection in wireless networks has been used. In this issue it is tried to build a model that can prevent energy loss in case of an attack. The two game modes are used in static and dynamic form in which the dynamic mode is closer to reality. In [9], the Bayesian network approach for intrusion detection in wireless networks has been used. The purpose of using signature detection is to detect abnormal packets. When this signatures does not conform to attack packages, Package is discovered. The main problem with this method is updating attacks. In study of Auld et al. [10] Bayesian neural network approach has been used to classify network traffic. Technique is a method of supervised classification by using data and features which are derived from the contents of the pack and the strength of method is based on this characteristics. In study of Gowadia, and Farkas [11] an intrusion detection system based on a probabilistic model that uses Bayesian networks is suggested. This method is more focused on IP Spoofing attacks and characteristics and features of the package contents are used in. In this method agent-based architecture is used so that agents communicate and collaborate with each other and they have the ability to update.

Simulations and Data Sets

In this model, we first consider the data set. These datasets are related to NSL-KDD which contain 42 features [12]. That feature Num_outbound_cmd is always zero in which this amount will be eliminated. The rest of the features, depending on the algorithm used for data pre-processing operations conducted and the data appropriately modified to work with the decision trees algorithms and Bayesian networks. The modeling on Rapid miner software, adding to this setting WEKA [13] algorithms, and preprocessing data has been done. The main task which has been done is to provide a number of data sets extracted from the original sets and also modeling of a lazy model that so far has not been used for intrusion detection systems. This model, like any other model which is based on data mining is as follows:

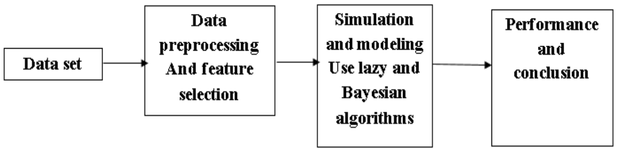
Data set
Data preprocessing And feature selection
Simulation and modeling Use lazy and Bayesian algorithms
Performance and conclusion

Fig. 3- Suggested architecture for the evaluation of intrusion detection using lazy and Bayesian algorithms

Data set is considered and according to the nature of the algorithms is divided into two categories and perform preprocessing and discretization.

In algorithms based on the lazy model regarding repeated tests the best data set is that all the data to be entered by database format and data are in nominal state. In this case, considering nature of the algorithms the best solution is obtained.

Due to the nature of Bayesian network algorithms the data sets are divided into three categories as follows and we do preprocessing and discretization. Three categories are: Integer and Real data set according to Table 1, Numerical datasets except Index, and Discrete datasets.

TABLE 1.
CONVERTED INDICATORS

Attribute name	Attribute type
Protocal_Type	Nomial
Flag	Nomial
Service	Nomial
Land	Binomial
Wrong_Fragment	Nomial
Num_Failed_Logins	Nomial
Root_Shell	Binomial
Su_Attempted	Nomial
Num_Shells	Nomial
Num_Access_File	Nomial
Is_Host_Login	Binomial
Is_Guest_Login	Binomial

RESULTS

In Rapidminer software we use precision parameter (accuracy) that is defined as the percentage of the correct classified samples [12].

A. *Lazy model based algorithms*

IBI algorithm [14]uses normalized Euclidean distance to find the nearest neighbor, its accuracy is equal to 80.10%. IBK algorithm [14] uses normalized Euclidean distance to find the nearest neighbor and K values is achieved based on cross-validation, its accuracy is equal to 80.47%. LWL algorithm [15] uses normalized Euclidean distance to find the nearest neighbor. So that weight has been attributed to characteristics with accuracy is equal to 78.02%. KSTAR [16] is distance-based algorithm that uses the entropy function. In fact, these algorithms take similarities with the training data into consideration, its accuracy is equal to 80.77% KNN algorithm[16] is type three, and KD-tree is used to build the model, its accuracy is equal to 80.10%.

B. *Bayesian network algorithms*

Kernel naive Bayesian algorithm makes use of probability theory of simple Bayes with kernel density. Weight function used in kernel uses estimation techniques without parameter. This model uses the data set type 1, its accuracy is equal to 79.50%. Naive Bayesian algorithm [17] uses simple probability theory of Bayes and variables should be independent which uses data set type2, its accuracy is equal to 77.24%. Waode[18] is model based on the average estimate of weigh make dependency among the parameters which uses data set type 3,its accuracy is equal to 81.88% Aode [19] achieve average classification accuracy by replacing a simple Bayesian approach which does not have simple Bayesian terms. Like independence of variables which uses data set type 3, its accuracy is equal to 82.09%. Aodesr [18] is the same as Aode, with these features at classification time that specification between the two features are detected and removed at the time of Generalization. The data set type 3 is used. And its accuracy is equal to 80.70%, its accuracy is equal to 81.99%. Bayesenet model uses different kinds of algorithms to enhance the precision accuracy. This algorithm does not use ADT TREE data structure and uses type three of data sets. And its accuracy is equal to 81.73%. In HNB[20], a hidden parent is created for each attribute by combining its effects on other traits. This algorithm is known as HNB. It uses the weight of the inter-dependencies with its accuracy is equal to 83.29%. Dmnbtext[21] is a model for making a simple Bayesian using polynomial division (Discriminative Multinomial). Data set Type 1 is used. And its accuracy is equal to 75.35% [12]. Bayesian logic regression implements Bayesian function by Laplace and Gauss estimation, and uses type 3 datasets, its accuracy is equal to 76.65%. In Fig. 4, obtained results for different algorithms are shown.

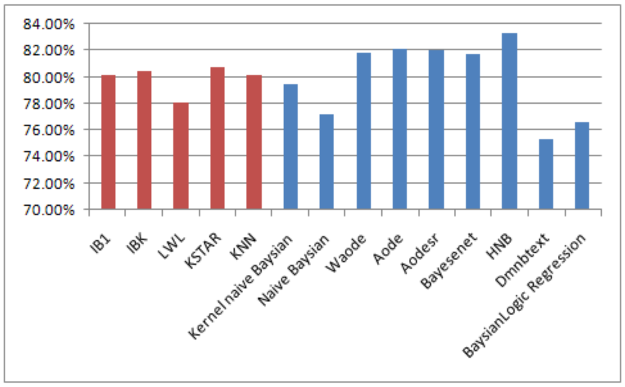

Fig. 4 - The results of comparison between lazy model algorithms and Bayesian networks (red bars are lazy model algorithms and blue bars are Bayesian network).

CONCLUSION

Due to the increasing development of Internet and Internet attacks, the existence of intelligent intrusion detection system becomes necessary. Many systems were proposed to perform the intrusion detection approach, such as lazy model and Bayesian network-based algorithms. In this paper, we provide the comprehensive study on the performance of the above mentioned algorithms, where several algorithms based on the data mining on training data and the test was evaluated. Test results show that the HNB algorithm has the highest accuracy equal to 83.29%.

REFERENCES

[1] M. Bishop, "*Introduction to Computer Security*," Prentice Hall, 2004.

[2] J. Han and M. Kamber, "*Data Mining: Concepts and Techniques*," San Diego, Academic Press, 2001.

[3] N. Lavrač, "Selected techniques for data mining in medicine," *Data mining techniques and applications in medicine*, vol. 16, pp. 3–23, 1999.

[4] L. Merschmann, "A Lazy Data Mining Approach for Protein Classification," *Nano Bioscience, IEEE Transactions on*, vol. 6, pp. 36-42, 2007

[5] M.L. Zhang, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, pp. 2038–2048, 2007.

[6] I. Vlahavas, I. Katakis, and G. Tsoumakas, "Mining Multi-label Data," *Data Mining and Knowledge Discovery Handbook*, pp 667-685, 2010.

[7] C. Kruegel, D. Mutz, and W. Robertson, F. Valeur, "Bayesian event classification for intrusion detection," in *Proc. IEEE Computer Security Applications Conference*, pp. 14–23, 2003.

[8] Y. Liu, C. Comaniciu, and H. Man, "Game theory for communications and networks," in Proc. workshop ACM New York, NY, USA, ISBN: 1-59593-507-pp. 34-43, 2006.

[9] F. Jemili, M. Zaghdoud, and A. Ben, "A Framework for an adaptive intrusion detection system using Bayesian network," in *Proc. of IEEE intelligence and Security Informatics conference*, pp. 66–70, 2007.

[10] T. Auld, A.W. Moore, and S.F. Gull, "Bayesian Neural Networks for Internet Traffic Classification," *IEEE Trans. on* Neural Networks, vol. 18, pp. 223-239, 2007.

[11] V. Gowadia, and C. Farkas, "PAID: A Probabilistic Agent-Based Intrusion Detection system," *Computers & Security*, vol. 24, pp. 529–545, 2005.

[12] *http://nsl.cs.unb.ca/NSL-KDD/*

[13] *http://www.rapid-i.com/content/view/181/*

[14] D. Aha and D. Kibler ''Instance-based learning algorithms. Machine Learning," *Journal Machine Learning*, vol. 6, pp. 37–66, 1991.

[15] E. Frank, M. Hall, B. Pfahringer, "Locally Weighted Naive Bayes," in *Proc. of Conference in Uncertainty in Artificial Intelligence*, pp. 249-256, 2003.

[16] J.G. Cleary and L.E. Trigg, "K*: An Instance-based Learner Using an Entropic Distance Measure," in *Proc. of the 12th International Conference on Machine Learning*, pp. 130-138, 1995.

[17] G. Webb, J. Boughton, and Z. Wang, "Not So Naive Bayes: Aggregating One-Dependence Estimators," *Machine learning*, vol. 58, pp. 5-24, 2005.

[18] L. Jiang, H. Zhang, "Weightily Averaged One-Dependence Estimators," in *Proc. of biennial pacific rim international conference on artificial intelligence*, pp. 970-974, 2006.

[19] F. Zheng and I. Geoffrey, "Webb: Efficient lazy elimination for averaged-one dependence estimators," in *Proc. of the international conference on Machine Learning*, pp. 1113-1120, 2006.

[20] H. Zhang, L. Jiang, and J. Su, “Hidden Naive Bayes,” in *Proc. of conference on artificial intelligence*, pp. 919-924, 2005.

[21] J. Su, H. Zhang, C.X. Ling, and S. Matwin, “Discriminative parameter learning for Bayesian networks,” in *Proc. of international conference on Machine learning*, pp. 1016-1023, 2008.