Comparison of Decision Tree Methods for Intrusion Detection

Fatih Ozturk,
fozturk@ibu.edu.ba

Abdulhamit Subasi
International Burch University, Faculty of Engineering and Information Technologies,
71000, Sarajevo, Bosnia and Herzegovina.
asubasi@ibu.edu.ba

Abstract: The popularity of using Internet contains some risks of network attacks, and attack methods differ each day, thus information security problem has become a significant issue all over the world. Intrusion detection is one major research problem in network security, whose aim is to identify unusual access or attacks to secure internal networks. At the moment, it is an urgent need to detect, identify and prevent such attacks effectively. In this work, we compared efficiency of decision tree methods in intrusion detection system. We compared the accuracy, detection rate, false alarm rate for different attack types.

Keywords: Decision Tree; CART (Classification and Regression Trees); ID3; C4.5; Random Forest; Internet attack; Intrusion detection system (IDS).

1. Introduction

In today's business oriented world information is very important. It is even considered as an intangible asset. The fastest way to bring the necessary information to end users is via Internet. Internet has been so deeply involved in our lives in a way that we have been dependent on it. It has even been used as an important component of business models (Shon & Moon, 2007). The internet has been used to bring the customers and the businesses together by applications such as websites and emails. This brings up a very important concern, the information security. Intrusion detection is one of the major research topics to prevent attacks from the internet for both companies and the end users. Firewalls may protect the networks but day by day the attacks became more complicated. Intrusion detection systems (IDSs) overcome this complexities providing ways to resist different types of suspicious network communications and computing habits. All of this is done assuming that the behaviour of intruders differs from an authenticated user (Stallings, 2006) (Tsai, et al. 2009).

Generally IDSs divide into two main categories based on their detection methods: anomaly and misuse (signature) detection (Anderson, 1995) (Rhodes, Mahaffey, & Cannady, 2000). Deviation from normal operation can be flagged as intrusion with anomaly detection. Meanwhile the use of well-known attacks to the system can be caught by misuse detection (Tsai, et al. 2009). Many issues should be considered when building an IDS, like data collection, data pre-processing, intrusion recognition, reporting and response. The most important of these components is intrusion recognition (Wu, Banzhaf, 2009).

In literature, since Denning first proposed the intrusion detection model in 1987 (Denning, 1987), different machine learning techniques are used to develop anomaly and misuse detection systems. Especially classifiers are used to detect whether a connection over internet is normal use or an attack (Wu, Banzhaf, 2009). Between late 80s and early 90s combination of expert systems and statistical methods were popular. Detection models have been acquired from field experts. From mid to late 90s normal abnormal detection moved to automatic modelling. Artificial intelligence (AI) and machine learning helped automation via a test data. Rule based induction, classification and data clustering highly referenced. Since IDS have huge network traffic volumes, highly diverse data distribution and difficult decision boundary to construct a good model is very challenging (Wu, Banzhaf, 2009). In this work, we compared different decision tree methods for intrusion detection using KDD'99 dataset.

This paper is organized as follows. Section 2 provides some literature review on IDS systems. Section 3 describes the data set and performance evaluation used in this paper. Section 4 gives brief theoretical background for different decision tree algorithms. Section 5 gives our analysis results and we conclude in Section 6.

2. Literature review on intrusion detection system

The IDS was first mentioned in Anderson's technical report (Anderson, 1980) where he mentioned the use of statistical methods to analyse users' behaviour to detect the misuse of the system. In 1987 Dorothy laid the foundations of Intrusion detection models. He proposed a prototype IDS: IDES (Intrusion Detection Expert Systems) (Denning, 1987). After this a number of IDS has been released such as Discovery, Haysack, MIDAS, NADIR, NSM, Wisdom and sense, DIDS, etc (Bace, 2002). (Wu, Yen, 2009).

We can classify IDSs into two major areas: Misuse Detection and Anomaly detection (Bace, 2002). In misuse detection we try to match incoming data with pre-defined intrusive behaviour (signature). So, the well known intrusions are detected very fast and accurately (low false alarm rates). For this reason this method is adopted by many commercial IDSs. However the intrusion methods are not trivial and will evolve continuously. If an unknown attack comes to the system it will fail. To overcome this shortage we have to update the signature database at the expense of our valuable time (Wu, Yen, 2009).

The second area, anomaly detection (Denning, 1987), may overcome this problem. The anomaly detection relies on the fact that most of the network communications are normal data and it tries to model this normal behaviour. Anything off this behaviour is marked as anomaly and a flag is raised by the system. Since the intrusions are rare and different from the normal data this method catches most of the abnormal behaviour even with unknown attacks. The difficulty here is the false alarm rates. The boundary between the normal and abnormal data is often too close (Wu, Banzhaf, 2009). Another problem faced is the constant changing of the normal usage statistics. For example we see more and more UDP connections since people use video sharing sites more frequently.
There is also a hybrid method introduced by MINDS (Ertoz et al., 2004), EMERALD, Prelude, etc. where they try to leverage the disadvantages of both of these methods (Wu, Yen, 2009).

3. Datasets and performance evaluation

The data we have used in the testing has been derived from well known KDD'99 datasets. This data is collected in 1998 by MIT Lincoln laboratory which has seven weeks of training and two weeks of test data. The test UNIX and Windows NT hosts and a Cisco router have faced more than 300 instances of 38 attack types. KDD'99 dataset is derived from this dataset in 1999 by assembling individual TCP packets into TCP connections. This data was the benchmark dataset used in the International Knowledge Discovery and Data Mining Tools competition (Tsai, et al. 2009).

Each TCP connection has 41 features with a label specifying the status of the connection. The features consist of 38 numeric and 3 symbolic features falling into one of the following categories:

1- Basic features: 9 basic features to describe each individual TCP connection.
2- Content features: 13 domain knowledge related feature to indicate suspicious behaviour without any sequential patterns.
3- Time-based traffic features: 9 features used to summarize the connections in the past 2 s that had the same destination host or the same service as the current connection.
4- Host-based traffic features: 10 features were constructed using a window of 100 connections to the same host instead of a time window to see the attacks that might take more than 2 s. (Tsai, et al. 2009).

In KDD'99 dataset we have 4,940,000 data instances covering the normal and 24 network attacks. The test set has 311,029 data instances with a total of 38 attacks, 14 of which do not appear in the training set. Because of this large training set, usually another dataset with the 10% of the data is used (Wu, Banzhaf, 2009).
The attacks contained in test information can be separated into the following categories:

Probe: These are not considered as real attacks but often used as preparation steps for a full scale attack. They are used to investigate the end system for future attacks.

Dos (Denial of service): This type of attack usually keeps server busy or uses the valuable bandwidth so that it will not provide the necessary service to the end users. Most common types are SYN Flooding, Ping Flooding and etc.
U2R (User to Root): In this type the attacker tries to take control of the admin user of the system from the leaks in the system. Buffer overflow is one of these methods.
R2L (Remote to Local): It is used to take advantage of the server providing the services in order to get sensitive security information on the server or user personal files. Unicode leak, SQL injection and etc. are examples of this attack type (Yen, Wu, 2009).

In our experiments we have selected a subset of 7000 instances to train our data. This subset has 6 different kinds of attacks and 1 normal data. The selected attack types are considered the most widely used types which include "port-sweep", "back-door", "IP-sweep", "nmap atack", "satan" and "smurf".

The best method to evaluate the effectiveness of an IDS is the correct prediction ability. There are four possible outcomes according to the real nature of a sample data compared to the outcome (prediction) from the IDS. This is also known as the truth matrix.

True negative (TN): The amount of normal data predicted when it is really normal,
True positive rate (TP): The amount of attack predicted when it is really attack
False negative rate (FN): The amount of normal predicted when it is really attack
False positive rate (FP): The amount of attack predicted when it is really normal.
TN/(TN+FP) gives specificity or False alarm rate (FAR),
TP/(TP+FN) gives sensitivity or detection rate (DR),

The most widely used performance evaluation are Detection Rate with False Alarm Rate. A good IDS must have high DR and a low or zero FAR (Wu, Yen, 2009) (Wu, Banzhaf, 2009).

4. Decision trees

In this paper we have compared four different Decision Tree methods. In a decision tree we try to classify a sample through some decisions leading us to a succeeding decision. The classification takes places from the root node to the leaf where the end leaf has the category information. Each node holds one attribute of the tree and the branches corresponds to the value of that attribute. (Mitchell, 1997). It is pretty much similar to flow chart structure; the test properties are internal nodes, test results corresponds to each branches, and distribution situation of various types are nodes of leaves. Each decision tree will belong to one of these categories: top-down tree construction or bottom-up pruning. The most essential and common method used for classification tree is CART (Classification and Regression Trees) (Breiman, Friedman, Olshen, & Stone, 1984), ID3 and C4.5 (Quinlan, 1993) and they are all top-down tree construction. The common algorithm can be described as such; (i) place all training set into the root of classification, (ii) check whether it contains all the same type or an empty set; if node contains more than one type of training set check each property of data according to certain function, and select a proper property. Divide training set into N parts, each constituting a new node to the root node according to the value of the property. This process is called as splitting node. (iii) Check whether each node is a leaf; if not, split them into new nodes as described in "ii". (iv) Proceed with splitting until each nodes turn into a leaf. This will construct our tree. In these methods we can totally classify any given training data into branches and leaves of the tree (Yen, Wu, 2009).

In ID3 method, by definition, all the parameters given should be discrete values. In our sample set we have 3 symbolic features. In order to overcome this problem we have converted these values into discrete valued properties with a pre-process before evaluating our sample data. Under certain conditions Decision Trees have advantage over other common supervised learning methods like discriminant analysis. Especially they do not suffer the same probability distribution restrictions. There is also no necessity to assume linear model and they are very useful with non-linear predictors (Wu, Banzhaf, 2009).

4.1 CART

In CART (Classification and Regression Trees) there are six general questions to be answered:

1- Should we allow properties to be restricted to binary values or let the multi valued properties exist?
2- Which property should be tested at a node?
3- When do we declare a node as a leaf

4- How to make trees smaller and simpler when it becomes too large? Prune?
5- When we induce an impure leaf node how shall we label it?
6- How can we handle the missing data?

According to these questions we create a tree giving good enough results with easy to compute and fast responses (Duda, Hart & Stork, 2002).

4.2 ID3

Since this is the third in a series of identification or "ID" process it is called ID3. The inputs were intended to be used with nominal (unordered) inputs only. When a real-valued variable is present it is first divided into intervals and then each interval applied as nominal input. Here each split has a branching factor Bj, where it is the number of discrete attribute container of variable j chosen for splitting. Usually these are not binary and a gain ratio impurity should be used. The number of levels they have is equal to the number input variables. It continues to run until all nodes are pure and no more possible splits exist. If necessary a common pruning technique can be applied to the algorithm (Duda, Hart & Stork, 2002).

4.3 C4.5

This is the successor and refinement of ID3 method, and very popular among classification tree methods. The real valued variables are handled as in CART. Nominal data is used to crate multi-way splits as in ID3 with a gain ratio impurity. The pruning is achieved with the statistical significance of the splits. The main difference between CART and C4.5 is the missing features. There are no substitute splits precomputed. If a defective test pattern with missing feature is present at branch N with branching factor B, C4.5 follows all possible B answers to the descendent nodes and at last to the B leaf nodes. The final decision is made according to the labels of B leaf nodes multiplied by the decision probability of N. Here, unlike CART, we don't exploit statistical correlations between different features of the training points. There is no extra computation and storage required for C4.5. So, it is much preferred where the storage is a major concern. As for the pruning, C4.5 generates the rules from the tree where each leaf node has one associated rule (the route from the root node to that leaf). Then it deletes the redundant antecedents in these rules (Duda, Hart & Stork, 2002).

4.4 Random forests

Decision Trees can be less accurate than methods like support vector machines and the structure of the trees can also be unstable (Breiman, 1996). Small changes in the training data may significantly result in changes to the tree, either to the identity of the split variable or to the value of the split. Even the structure changes significantly the resulting prediction may stay the same. RandomForest (a trademark of Saldorf Systems) algrithm is developed by Breiman (Breiman, 2001a,b) to overcome these shortcomings while possibly enhancing the interpretability. Some of the important features of RandomForest is: (Breiman and Cutler, 2004b)

- The accuracy that equals or tops many current classifiers without overfitting
- Fast on large databases and handle thousands of predictors without selector routines
- Each predictor importance is estimated
- Generalization error estimate is generated in an unbiased manner
- The missing data and error balance when a class proportion marked differently is dealt with robust algorithms.
- The proximities between the pairs of cases that can be used in clustering and identifying outliers are computed
- Variable interaction detection is done by an experimental method
- Generated forest can be saved to be used on other data.

The forest, generated by many trees, where each tree is different from each other at the generation of training cases and predictors used at each node. Each tree is generated by a subset (m) of available predictors drawn randomly, which is much less than the total available. This m value is the only adjustable parameter which the random forests are sensitive (Breiman and Cutler, 2004a). This parameter is the same for all the trees in forest. Each tree is allowed to grow to the most possible extend thus; there is no need for pruning. Besides increased prediction accuracy, they help to determine the importance of each variable and associations between each case (Fielding, 2007).

5. Results and Discussion

Over the past decade intrusion detection based on machine learning methods has been extensively studied topic, and they satisfy the growing demand of reliable and intelligent intrusion detection systems. In this study we compared the performance of different decision tree classifiers on solving intrusion detection problems. This research works were trained and tested on the KDD'99 dataset. The classification results are shown in Table 1. From this table, we can easily see that there is no significant difference between accuracy of these methods; however, random forest is better than others. According to the average, ID3 is worst in the classification. Accuracy refers to the proportion of attack detected among all attack data, namely, the situation of TP. In detection rate, the random forest has accuracy 99.94 % approximately. False alarm rate refers to the proportion that normal data is falsely detected as attack behaviour, namely, the situation of FP. In comparison of false alarm rate, random forest is 0.1 %, but it is worse than C4.5 and CART. According to the average value, false alarm rate of C4.5 and CART is 0 % and better than random forest.

Decision Tree Method	**Accuracy**
ID3	97.4%
CART	99.8286 %
C4.5	99.8857 %
Random Forest	99.9429 %

Table 1: Classification of Different Decision Tree Algorithms

We have considered the problem of comparing different decision tree classifiers, including ID3, CART, C4.5 and random forest. Here, rather than directly comparing typical implementations of CART, ID3, C4.5 and random forest methods, it is more useful to consider distinctions within the different component steps. Anybody can build a tree using any practical feature processing, impurity measure, stopping criterion or pruning method. Of course, if the designer has insight into feature pre-processing, this must be utilized. Generally, pruning can be preferred over stopped training and cross-validation, because it takes advantage of more of the information in the training set. On the other hand, pruning large training sets can be computationally expensive. The pruning of rules is less useful for problems that have high noise and are at base statistical in nature. Similar to the most classification methods, one gains expertise and insight through experimentation on a wide range of problems. Any single tree algorithm neither dominates nor is dominated by other classification methods. It can be seen that trees yield classifiers with accuracy as good as other classification methods. (Duda, Hart & Stork, 2002)

Even though some promising results have been accomplished by different decision tree classifier to IDSs, there are still challenges that lie ahead for researchers in this area. First and foremost, good benchmark datasets for network intrusion detection are needed. The KDD'99 is the most important benchmarks used to evaluate the performance of network intrusion detection systems. But, they are suffering from a serious drawback: failing to realistically simulate a real-world network (Brugger 2007)(Mahoney, Chan, 2003) (McHugh, 2000). An IDS working well on these datasets may demonstrate unacceptable performance in real environments. (Wu, Banzhaf, 2009)

These datasets possess some special characteristics, such as huge volume, high dimension and highly skewed data distribution. As a result, using only these datasets is not adequate to demonstrate the efficiency of a learning algorithm. It is also meaningful to note that the KDD'99 datasets were collected about 10 years ago. One of the important characteristics of intrusion detection is the capability of adaptation to continually changing environments. Not only the intrusive behaviour evolves continuously, but also the legitimate behaviour of users, systems or networks changes over time. If the IDS is not flexible enough to cope with behavioural changes, detection accuracy will dramatically decrease. A focus on adaptation in IDSs is highly recommended. Another challenge to confront in IDS is the huge volume of audit data that makes it difficult to build an effective IDS. Perhaps it is time to create a new and high-quality dataset for the intrusion detection task. (Wu, Banzhaf, 2009)

6. Conclusion

Intrusion detection based on computational intelligence is currently attracting considerable interest from the research community. This research compares accuracy, detection rate and false alarm rate of different attacks. KDD'99 dataset is current benchmark dataset in intrusion detection. For comparison results of decision tree

algorithms, we find that Random forest is superior to others in accuracy and detection; ID3 is the worst. In comparison of false alarm rate, C4.5 and CART are better than Random forest. Through test and comparison, the accuracy and detection rate of Random forest is higher than that of others, but false alarm rate of C4.5 and CART is better; if we combine the two methods, overall accuracy can be increased greatly. Dataset KDD'99 applied in the research is popularly used in current intrusion detection system; however, it is data of 1999, and network technology and attack methods changes greatly, it cannot reflect real network situation nowadays. Therefore, if newer information is got and tested and compared refresh, they can more accurately reflect current network situation.

References:

Anderson, James P. (1980). Computer security threat monitoring and surveillance, technical report, James P. Anderson Co., Fort Washington, Pennsylvania.

Anderson, J. (1995). An introduction to neural networks. Cambridge: MIT Press.

Bace, Rebecca G. (2002). NIST special publication on intrusion detection systems.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, P. J. (1984). Classification and regressing trees. California: Wadsworth International Group.

Breiman, L. (1996). Bagging predictors. Machine Learning, 24, 123_40.

Breiman, L. (2001a). Random forests. Machine Learning, 45, 5_32.

Breiman, L. (2001b). Statistical modelling: the two cultures. Statistical Science, 16, 199_215.

Breiman, L. and Cutler, A. (2004a). Interface Workshop: April 2004. Available at http://stat-www.berkeley.edu/users/breiman/RandomForests/interface04.pdf (accessed 1 February 2006).

Breiman, L. and Cutler, A. (2004b). Random Forests. Available at http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_home.htm (accessed 1 February 2006).

Brugger, T. (2007) KDD cup'99 dataset (network intrusion) considered harmful, 15 September 2007. Retrieved January 26, 2008, from http://www.kdnuggets.com/news/2007/n18/4i.html.

Dorothy, Denning. 1987. An intrusion detection model. IEEE Transaction on Software Engineering.
Intrusion detection by machine learning: A review

Duda R. O.; Hart, P. E. and Stork D. (2002). *Pattern Classification,* 2nd. Edition, John Wiley & Sons, 2002.

Rhodes, B., Mahaffey, J., & Cannady, J. (2000). Multiple self-organizing maps for intrusion detection. In Paper presented at the proceedings of the 23rd national information systems security conference. Baltimore, MD.

Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P., Srivastava, J., Kumar, V., et al. (2004). The MINDS – minnesota intrusion detection system. Next generation data mining. MIT Press.

Fielding, Alan H. (2007). Cluster and Classification Techniques for the Biosciences, Cambridge University Press The Edinburgh Building, Cambridge cb2 2ru, UK, 2007.

Mahoney, M.V.; Chan, P.K. (2003). An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection. Technical Report TR CS-2003-02, Computer Science Department, Florida Institute of Technology, 2003

McHugh, J. (2000). Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by Lincoln laboratory, ACM Transactions on Information and System Security 3 (4) (2000) 262–294.

Mitchell, T. (1997). Machine learning. New york: McGraw Hill.

Quinlan, J. R. (1993). C4.5: Programs for machine learning. Morgan Kaufmann Publishers.

Shon, T., & Moon, J. (2007). A hybrid machine learning approach to network anomaly detection. Information Sciences, 177, 3799–3821.

Stallings, W. (2006). Cryptography and network security principles and practices. USA: Prentice Hall.

Tsai, Chih-Fong; Hsu, Yu-Feng; Lin, Chia-Ying; Lin, Wei-Yang (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications,* 36 (2009) 11994–12000

Wu, Shelly Xiaonan; Banzhaf, Wolfgang (2009). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing,* 10 (2010) 1–35

Wu, Su-Yun; Yen, Ester (2009). Data mining-based intrusion detectors. *Expert Systems with Applications,* 36 (2009) 5605–5612