Identifying Relevant Features for Attacks in KDD 99 dataset

Deepti Bhatia
Department of Computer Science
Texas Tech University
Lubbock, Texas
Email: deepti.bhatia@ttu.edu

Nikitha Mahesh
Department of Computer Science
Texas Tech University
Lubbock, Texas
Email: nikitha.mahesh@ttu.edu

Priyanka Kumari
Department of Computer Science
Texas Tech University
Lubbock, Texas
Email: priyanka.kumari@ttu.edu

I. Introduction

Detection of actions that compromise the access to a resource plays an important role as the usage of internet has grown tremendously. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are the most common types of such attacks. This calls for the need of a software application which is known as a Intrusion Detection System (IDS) [1] that could monitor the system and network activities to identify if there happens to be any malicious operations. As some of the sectors such as business,security, financial, healthcare moved towards LAN and WAN applications, these applications made the network an attractive target for the intruder to bring a big vulnerability for the community. With the growth of internet, intruders have adopted different techniques to crack password, detect unencrypted text to cause vulnerabilities to the system. Thus, security is needed for the user to protect their system form the attackers. One of the popular prevention technique is Firewall system that is used to protect the private network from public network. IDS has been used in many different fields that includes network related activities, medical applications, credit card frauds.

Three types of Intrusion Detection Systems based on analyzed activity are Network based, Host based and Physical IDS. Network based IDS (NIDS) attempts to identify anomalous behavior on the network traffic. Being a detection system it does not block the network traffic but gathers information regarding the packets and also the logging information so as to identify any suspicious behavior in the network. Snort is an example of NIDS. Host based IDS (HIDS) attempts to identify unauthorized, anomalous behavior on a specific device. This involves an agent installed on each system that monitors local OS and application activity. In order to identify unauthorized activity it uses a combination of signatures and rules. Tripware and AIDE are examples of HIDS. Physical IDS identifies threats to physical systems. Firewalls are examples of Physical IDS.

There are two approaches to detect intrusion: Anomaly based intrusion detection and Signature based intrusion detection. Anomaly detection [9] observes normal behavior and any event that does not follow the normal behavior is considered to be suspicious. Activities far from normal are flagged as an intrusion. Signature based detection [8] (a.k.a. misuse detection) makes use of a dataset that has instances each of which is labeled as normal or intrusive. Machine learning approaches are used to train the data. The signature is retained in this technique that helps to detect the intrusion. The advantage of signature based IDS is that it is easy to implement, lightweight but it is impossible to detect new attacks when we do not have a signature for the new attack.

Weakness of signature-based IDSs in detecting novel attacks has attracted the attention of many researchers. KDD 99 intrusion detection dataset, based on DARPA 98 dataset, consists of TCP dump data from Lincoln Labs consisting of network activity grouped into five categories: normal activity and attacks such as DoS, U2R, R2L and Probe. It consists of 24 attack types in the training dataset and additional 14 attack types in test data which makes it interesting for evaluating a signature based IDS. [5] [7] A denial-of-service attack (DoS attack) is a type of cyber attack where the authorized, intended users are denied to use the network resources, this is achieved by flooding the targeted machine or resources with too many requests to overload the system. In user-to-root attack (U2R), the attacker tries to gain the root access which leads to several vulnerability such as sniffing password, a dictionary and social engineering attacks e.g. perl, xterm. In order to expose the machine vulnerabilities and exploit privileges, an attacker sends packets over the internet to a machine he/she does not have access to. This is a remote to user attack (R2L). In order to determine weakness or vulnerabilities that may later be exploited to compromise the system the scanner scans a machine or a networking device. This is known as Probing.

II. Motivation

It is infeasible to use humans to detect attacks in a network. Hence Machine Learning techniques have been used by researchers to eliminate human processes in intrusion detection and building an IDS. Such systems 'learn' or build models from historic data (normal/attacks) to detect attacks in future. Such systems are based heavily on statistical analysis of data. [2] [3] [4]

Raw real-time network activity logs usually contain a lot of information about the network such as packet information, host information, TCP connection information, etc. Researchers

have focused on building better Machine learning algorithms for classification of such data. The amount of data (instances) as well as the features being considered affect classification accuracy. A large dataset size makes it difficult for an algorithm to process the data on one system. This calls for a need to reduce the dataset for processing. The number of instances in data cannot be reduced because they contain important information about every activity on the network. However, we can reduce the feature set size. Large number of features provided by raw logs may also lead to overfitting of the classifier model. At the same time, we do not want to have very small number of features. Thus the feature set usually needs to be reduced down to retain important features and discard the ones that distract the classifier. Not only the size of the feature set, but also how well they classify the data is important.

Previous work has been focused on identifying relevant features for each attack (for e.g. smurf, perl, buffer_overflow, etc.) in the KDD dataset. Such an analysis is useful in the case when we wish to identify the exact type of attack. But in KDD 99, each such attack falls under one of the DOS, U2R, R2L and Probe categories. In order to understand more about the important similarities between attacks, it is important to identify the relevant features corresponding to each attack category. The focus of this project is selecting relevant features for each attack category to understand network patterns for each attack category.

III. Problem Definition

As we have seen, KDD 99 dataset consists of 41 features representing normal network traffic as well as attacks. Although there are 24 attack types present in the training dataset and 14 in the testing dataset, each attack belongs to one of the following categories: DoS, U2R, R2L and Probe. Thus we look at the dataset as representing normal traffic and 4 attack categories. There must be some unique attributes of these attacks such that they can be grouped into these categories. We make an attempt to answer the following research questions:

1) What are unique attributes for each attack category?
2) Are these attributes unique among attack categories? (For example, which features separate DoS from U2R or other attack categories?)
3) Which features are common among attacks of a particular category? (For example, smurf and neptune are both DoS attacks, do they have any features in common?)
4) Which features distinguish the categories from each other?

We wish to use Feature selection algorithms from Machine Learning to answer these questions and select features that make the attacks distinctive of each other. Thus, for normal traffic and for each attack category, we would be able to define the set of features which identify that attack and eliminate unimportant features. This would indirectly help in the following ways:

1) enable better data understanding, better representation of attacks
2) simplify the machine learning model and improve classification
3) reduce overfitting of the model
4) reduce storage and computational cost

An IDS can then detect attacks based on comparatively lesser and important features.

IV. Solution

As discussed above, the main objective of this project is to identify important features uniquely representing normal traffic and intrusions or attacks using Feature Selection methods. Filter based and Wrapper based models are traditional models used for feature subset selection[Wrappers for feature subset selection: Kohavi]. Wrapper methods evaluate the performance of all possible subsets of features from a dataset using a state space tree and choose the subset that gives the best performance. It relies on the correspondence between features to identify the class. However, this technique has large computational overhead when the number of features is large. On the other hand, filter methods assign relevance scores to variables and select ones that are most relevant towards predicting the class label irrespective of how they are related to each other. They tend to 'filter' out the irrelevant variables and thus the name of the method.

Due to the drawbacks of wrapper based methods mentioned earlier, this project focuses on a benchmark filter method of feature selection used in research, the Information Gain measure.

A. Information Gain

Information Gain assigns a score to each attribute based on the entropy i.e. information obtained from the attribute in classification. It begins by defining Entropy that measures the amount of uncertainty in the data. Entropy is given by [6]:

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i \tag{1}$$

where c is the number of class labels in the dataset, p_i is proportion of the dataset S belonging to class i.

Thus, when $Entropy(S) = 0$, the dataset is homogenous, i.e., all elements in the set belong to the same class and when $Entropy(S) = 1$, the dataset is equally divided among class labels. The next step is to measure how effective each attribute is in classifying the dataset. Information Gain measure attempts to reduce the uncertainty (entropy) caused by partitioning the data based on a certain attribute. The attribute which partitions the data while giving the least uncertainty (zero entropy) is given highest information gain score. The information gain $Gain(S, A)$ of an attribute A relative to a dataset S is defined as:

$$Gain(S, A) = Entropy(S) - \sum_{v \epsilon Values(A)} \frac{|S_0|}{|S|} Entropy(S_0) \tag{2}$$

where $Values(A)$ is the set of distinct values possessed by attribute A, S_0 is subset of S for which attribute A has value v. In simple words, $Gain(S, A)$ is the information gained about the class labels given the value of attribute A. If $Gain(S, A1) > Gain(S, A2)$ then attribute $A1$ is a better predictor of the class than attribute $A2$. Once we identify information gain for all attributes, they are ranked in the descending order of their gain measures and top n attributes can be chosen as the 'best predictors' among all attributes. Here, n is usually chosen by trial and error method depending on which value of n gives us the best classification rate.

B. Handling continuous valued attributes

As seen above, Information Gain focuses on distinct valued attributes. The KDD 99 dataset contains 22 continuous attributes out of 41 total attributes. Such attributes will have to be discretized into definite ranges in order to identify their information gain. Values of the attribute can be divided into bins and labels values based on the bins they fall under. For e.g. if an attribute can have values such as $1.5, 20.4, 10.3, 15.5, 17.9, 7.8$, we can define ranges to be $0-5, 6-10, 11-15, 16-20, 21-25$ such that the value 1.5 would be replaced with $0-5$, 7.8 would be replaced with 6-10 and so on. The continuous values are thus replaced with discrete values. We can then apply equations (1) and (2) and calculate information gain for the attributes.

C. One vs Rest Method

We wish to rank attributes based on their information gain with respect to each class label. Since information gain ranks attributes based on class labels, if we have binary classes, the ranked attribute order tells us how these attributes separate the two classes. Whereas if it is a multi class problem, we cannot map ranked order of attributes to each of the classes. Thus we transform the class attribute in a one-vs-rest method. For example, if we wish to identify the best predictors for DOS attacks, we keep the DOS labels intact and relabel all others (normal, U2R, R2L, probe) as 'others'. Thus we would be able to identify the ranked order of attributes which separates DOS attacks from the rest. We would repeat the same process for other categories of attacks.

V. Plan for Implementation and Evaluation

Following steps will be followed to implement the above approach:

1) Replace attack labels (smurf, pod, teardrop, etc.) with their corresponding attack categories (DOS, U2R, etc.)
2) Implement discretization algorithm and discretize continuous valued attributes
3) Implement information gain algorithm
4) For each attack category
    a) Label the classes in one-vs-rest method
    b) Calculate information gain for attributes and rank them

To evaluate the chosen attributes:

1) For each attack category
    a) Label the classes in one-vs-rest method
    b) Classify records
    c) Keep important attributes chosen by information gain and discard others
    d) Classify records

Classification accuracy with and without attribute selection will be noted to understand how it helped. Few of the benchmark classification algorithms that we plan to use are: J48 (Decision Tree), Naive Bayes, Random Forests. As we go ahead with the implementation, we might try using other algorithms for classification as well.

References

[1] Karen Scarfone, Peter Mell, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, NIST Special Publication 800-94, Recommendations of the National Institute of Standards and Technology.

[2] Nahla B. Amor, Salem Benferhat, Zied Elouedi, *Naive Bayes vs decision trees in intrusion detection systems*, SAC '04 Proceedings of the 2004 ACM symposium on Applied Computing, pp. 420-424.

[3] Zamani, Mahdi, and Mahnush Movahedi, *Machine Learning Techniques for Intrusion Detection*, arXiv preprint, arXiv:1312.2177 (2013).

[4] Swati Paliwal, Ravindra Gupta, *Denial-of-Service, Probing and Remote to User (R2L) Attack Detection using Genetic Algorithm*, International Journal of Computer Applications, Vol. 60, No. 19, Dec. 2012.

[5] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, *A Detailed Analysis of the KDD CUP 99 Data Set*, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).

[6] Tom Mitchell, *Machine Learning*, Mc Graw Hill, 1997.

[7] http://kdd.ics.uci.edu/databases/kddcup99/task.html

[8] Hugo Gascon, Agustin Orfila, Jorge Blasco, *Analysis of update delays in signature-based network intrusion detection systems*, COMPUTERS and SECURITY vol. 30, 2011, pp. 613-624

[9] Ming-Yang Su, *Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers*, Expert Systems with Applications vol. 38, 2011, pp. 4923498