Selection of Relevant Feature for Intrusion Attack Classification by Analyzing KDD Cup 99

Prof. N.S. Chandolikar  
Department of Computer Engineering  
Vishwakarma Institute of Technology  
Pune-411037  
Email: neelam.chandolikar@vit.edu

Prof. (Dr.) V.D. Nandavadekar  
Director  
Department of MCA  
Sinhgad Institute of Management, Pune  
Email: vilasn73@yahoo.com

ABSTRACT

Security is one of most important issue in network management and detection of Intrusion based security attacks play vital role in it. To have a holistic picture of the network intrusion detection, selection of appropriate feature is very important; it reduces analysis effort and time too. Identification of most astute feature for attack classification plays significant role in intrusion detection. Data mining can be very fruitful for feature selection and intrusion detection. In this paper, KDD ’99 intrusion detection dataset is evaluated to find out most important and relevant features.

I. INTRODUCTION

Intrusion Detection System (IDS) can detect, prevent and more than that IDS react to the attack. Therefore, the main objective of IDS is to at first detect all intrusions at first effectively. This leads to the use of an intelligence technique known as data mining/machine learning technique as an alternative to expensive and strenuous human input. These techniques automatically learn from data or extract useful pattern from data as a reference for normal/attack traffic behavior profile from existing data for subsequent classification of network traffic.

Data Mining

The process of extracting useful and previously unnoticed models or patterns from large data set. Data mining [1] [2] is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data.

Data mining [3][4][5] can be used for solving the problem of network intrusion based security attack because of following reasons:

• Ability to process large amount of data.
• Ability to reduce data and by extracting specific data.

Data Reduction

Data mining can significantly reduce data overload through its capability to extract specific amounts of data for identification and analysis. This helps the system to determine which data is most relevant and breaks it down so anomaly detection is easier to spot.

• Easy data summarization and visualization that help the security analysis.
• Data mining offers Code Variants, it scans for abnormal activity.
• It Filter out Valid Network Activity.

II. OVERVIEW OF INTRUSION DETECTION SYSTEMS

Detection method in IDS [6][7] can be divided into two categories: anomaly detection and misuse detection categories.

A. Signature-based IDS

Network traffic is examined for preconfigured and predetermined attack patterns known as signatures. It is widely available, it uses known patterns as it is easy to implement but they cannot detect attacks for which it has no signature and they are also prone to false positives since they are commonly based on regular expressions and string matching. Since they are based on pattern match, signatures usually don’t work that great against attacks with self-modifying behavior.

B. Anomaly-Based IDS

Anomaly-based IDS works on a performance baseline based on normal network traffic evaluations. It sample current network traffic activity to this baseline in order to detect whether or not it is within baseline parameters. If the sampled traffic is outside baseline parameters, an alarm will be triggered. They Can detect attempts to exploit new and unforeseen vulnerabilities...
and have Lower false positives most important is that they are Very scalable, due to its architecture and method of operation. No need to create new signatures for every attack and variant.

III. INTRUSION DETECTION DATASETS

A. KDDCup’99 Data Set

The data set used to perform the experiment is taken from KDD Cup’99 [8][9][10], which is widely accepted as a benchmark dataset and referred by many researchers. “10% of KDD Cup’99” from KDD Cup ‘99 data set was chosen to evaluate rules and testing data sets to detect intrusion. The entire KDD Cup’99 data set contains 41 features. Connections are labeled as normal or attacks fall into four main categories.

1. DOS – Denial of Service
2. Probe – e.g. Port Scanning
3. U2R – unauthorized access to root privileges,
4. R2L – unauthorized remote login to machine.

In this dataset there are 3 groups of features: Basic, content based, time based features.

Training set consists 5 million connections.

• 10% training set – 494,021 connections
• Test set have – 311,029 connections

Test data has attack types that are not present in the training data. Problem is more realistic:

− Train set contains 22 attack types.
− Test data contains additional 17 new attack types that belong to one of four main categories.

IV. FEATURE SELECTION

Feature selection [11] [12] [13] is one of the common terms used in data mining. It is used to reduce inputs to a manageable size for processing and analysis. Many tools and techniques are available for the same. Feature selection is used for imposing an arbitrary or predefined cutoff on the number of attributes that can be considered when building a model, and also the choice of attributes, meaning that either the analyst or the modeling tool actively selects or discards attributes based on their usefulness for analysis.

Feature selection for intrusion detection is most important factor for the success of intrusion detection system. In this paper, KDD 99 dataset is investigated to identify the relevance of each feature in intrusion detection. Rule induction method is used to determine the most discriminating features for each class. Therefore, the relevance of the forty one (41) features with respect to dataset labels was investigated.

V. DISCRTIZATION BASED ON RULE INDUCTION

A. Rule Induction

Rule induction is[1] one of the major forms of data mining and is perhaps the most common form of knowledge discovery in unsupervised learning systems. Rule induction on a data base can be a massive undertaking where all possible patterns are systematically pulled out of the data and then an accuracy and significance are added to them that tell the user how strong the pattern is and how likely it is to occur again. In rule induction systems the rule itself is of a simple form of “if this and this and this then this”.

In order for the rules to be useful there are two pieces of information that must be supplied as well as the actual rule:

− Accuracy – How often is the rule correct?
− Coverage – How often does the rule apply?

B. Rule Induction Using a Sequential Covering Algorithm

Using a sequential covering algorithm [1] IF-THEN rules can be extracted directly from the training data (i.e., without having to generate a decision tree first)

Sequential covering accepts D, a data set class-labeled tuples and f, the set of all features and their possible values. and generates A set of IF-THEN rules.

Algorithm: Sequential covering (d)

1. Rulelist<- {};
2. for each class c do
3. repeat
4. Rule <-Learn One Rule(D, f, c);
5. remove tuples covered by Rule from D;
6. until terminating condition;
7. Rulelist <-Rulelist +Rule;
8. End for
9. return Rulelist;

C. Learn one Rule Algorithm

Learn One Rule adopts a greedy depth-first strategy. Each time it is faced with adding a new attribute test (conjunct) to the current rule, it picks the one that most improves the rule quality, based on the training samples.

− Learn one rule at a time, sequentially.
− After a rule is learned, the training examples covered by the rule are removed.
− Only the remaining data are used to find subsequent rules.
− The process repeats until some stopping criteria are met.
Other than rule induction one more popular method is decision trees, which is used for feature selection and classification, but it can sometimes be quite difficult to comprehend when the tree size is too big. To offset the drawback of decision trees, rule sets generated by rule induction can be used efficiently. Rule sets consist of unordered collections of simple if-then rules. Rule sets are generally easier to understand compared to the trees since each rule describes a specific context associated with a class or an attribute. Plus, rule set is easier to understand than decision tree.

D. Classifying and Detecting Anomalies

The rules are applied as SQL query to the database. The initial training set contained 22 attack types. Since the objective of our project is not to detect each of these attack types but rather to detect the major categories into which these attacks fall, we aggregated the 22 attack types into four more generic categories as follows:

- DOS (denial-of-service)
  - Back, land, Neptune pod smurf teardrop.
- Probe:
  - Ipsweep, nmap portsweep, llt_port_attack.
- R2L (Remote-to-Local)
  - Imap, ftp_write, guess_passwd, multihop, phf, spy, warezclient, warezmaster.
- U2R (User-to-Root)
  - buffer_overflow, Rootkit, Perl, Loadmodule.

VI. PROPOSED SYSTEM DESCRIPTION

Rule induction on a KDD cup dataset is applied to get appropriate rule for intrusion detection. In rule induction systems the rule itself is of a simple form of “if this and this and this then this”. For experimentation rule induction is used in spite of decision tree [14] which is also one of popular technique because of following reason:

- Rule set are highly expressive as decision trees.
- Rule set are easy to interpret.
- Rule set are easy to generate.
- Rule set can classify new instances rapidly.
- Rule set can easily handle missing values and numeric Attributes.

Figure 1 specifies the process steps used in our project.

A. The Algorithm

While experimenting sequential covering method is used. In this rule building stopped when:

- The rule is perfect, i.e. Accuracy = 1.
- Increase in accuracy gets below a given Threshold.
- The training set cannot be split any further.

To evaluate our system, there are two major indicators of performance: the detection rate for each attack class, false positive and false negative rate. The detection rate (true attack alarms) is defined as the number of intrusion instances detected by the system divided by the total number of intrusion instances present in the test set. The false positive rate (false attack alarms) represents the total number of normal instance that were classified as intrusion divided by the total number of normal instances.

ONE R algorithm steps, used for experiment can be written in following way.

Let T be the training set.

For each class C.

1. Initialize set S with T.
2. While S contains instances in class C.
3. Learn one rule R for class C.
4. Remove training records covered by the rule R.

AIM: to create rules that cover many examples of a class C and none (or very few) of other classes.

In sequential covering algorithm each new test reduces rule’s coverage:

Let

- T-total number of instances covered by rule C.
- P positive examples of the class predicted by rule.
- T-P number of errors made by rule.
- Therefore Rules accuracy = p/t.

In this algorithm total space for example taken and per iteration it is reduced as per rules generated. Figure 1 specify this process.
To assess the effectiveness of proposed intrusion detection approaches, the series of experiments were performed in Weka. The java heap size was set to 1024 MB for weka-3-6. Experiment performed on KDD cup dataset.

B. Weka

Weka is a collection of machine learning algorithms for data mining tasks. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. WEKA consists of Explorer, Experimenter, Knowledge flow, Simple Command Line Interface, Java interface.

The WEKA tool is used in experiment to perform following task:

- To investigate and pre-process the features in the database and assessing the correctness of the data.
- To define the class attributes which divide the set of instances into the appropriate classes.
- To mine potential features to be used for classification and select subset of features to be used in the learning process.
- To examine possible imbalance in the selected data set and how it may be counteracted.
- To select subset of the instances, i.e. the records that learning is to be based on.
- To Apply Rule Induction Algorithm (One R) for the learning process.
- To make Decision on a testing method to estimate the performance of the algorithm.

C. Performance Measurement Terms

Detection of attack can be measured by following metrics:

- False Positive (FP): Or false alarm, Corresponds to the number of detected attacks but it is in fact normal.
- False Negative (FN): Corresponds to the number of detected normal instances but it is actually attacks, in other words these attacks are the target of intrusion detection systems.
- True Positive (TP): Corresponds to the number of detected attacks and it is in fact attack.
- True Negative (TN): Corresponds to the number of detected normal instances and it is actually normal.
- The accuracy of an intrusion detection system is measured regarding to detection rate and false alarm rate.

True positive is an instance which is normal and classified normal, so it should be high. Whereas false positive means no attack but IDS detects attack, it should be low.

VII. THE EXPERIMENTAL RESULTS

Proposed intrusion detection approaches are implemented to detect 5 different classes of attacks from the dataset including Dos, U2R, Probe, U2L and normal. The distribution of an attack and normal records are 80%-20%, based on the experiment association of any feature with attack class is analyzed.

Experiment shows high true positive rate (TP) and low false positive (FP) using ONER algorithm.

<table>
<thead>
<tr>
<th>Attack type</th>
<th>True Positive</th>
<th>False Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMAL</td>
<td>.96</td>
<td>.002</td>
</tr>
<tr>
<td>DOS</td>
<td>.99</td>
<td>.034</td>
</tr>
<tr>
<td>PROBE</td>
<td>.99</td>
<td>.006</td>
</tr>
<tr>
<td>R2L</td>
<td>.79</td>
<td>0</td>
</tr>
<tr>
<td>U2R</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Experimental Result

Table 2 one shows the result shows the Performance of ONE R algorithm based on Correctly classified instance, Incorrectly classified instance, Kappa statistics, Mean absolute error, Root mean squared error.

The correctly and incorrectly classified instances show the percentage of test instances that were correctly and incorrectly classified.

The percentage of correctly classified instances is often called accuracy or sample accuracy.

Kappa is a chance-corrected measure of agreement between the classifications and the true classes. It’s calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum possible agreement. A value greater than 0 means that your classifier is doing better than chance (it really should be!).

The error rates are used for numeric prediction rather than classification. In numeric prediction, predictions aren’t just right or wrong, the error has a magnitude, and these measures reflect that.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Performance in Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Correctly classified instance</td>
<td>96.18</td>
</tr>
<tr>
<td>2 Incorrectly classified instance</td>
<td>3.81</td>
</tr>
<tr>
<td>3 Kappa statistics</td>
<td>.92</td>
</tr>
<tr>
<td>4 Mean absolute error</td>
<td>.03</td>
</tr>
<tr>
<td>5 Root mean squared error</td>
<td>.19</td>
</tr>
</tbody>
</table>

Table 2: Performance of One R Algorithm
To present different classes are shown which achieved good levels of discrimination from others in the training set and the analysis of feature relevancy in the training set. Table 3 details the most relevant features for each class.

Based on experiment we can say that normal, Neptune and SMURF classes are highly related to certain features that make their classification easier. Since these three classes make up 98% of the training data, it is very easy for a intrusion detection system to achieve good results. There are few features which are not relevant in terms of intrusion detection and there are some which are highly relevant.

Table 3: Most Relevant Feature per Class Label

<table>
<thead>
<tr>
<th>Feature No</th>
<th>Feature Name</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Duration,</td>
<td>Normal</td>
</tr>
<tr>
<td>6</td>
<td>Dst_Bytes,</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Su_Attempted,</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Num_Root,</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Num_File_Creations,</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Num_Shells,</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Num_Access_Files,</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Srv_Diff_Host_Count,</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Dst_Host_Srv_Count,</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Dst_Host_Diff_Srv_Count,</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Dst_Host_Srv_Count,</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Dst_Host_Srv_Serror_Rate,</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Flag,</td>
<td>SMURF</td>
</tr>
<tr>
<td>25</td>
<td>Srv_Error_Rate,</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Same_Srv_Rate,</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Diff_Srv_Rate,</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Dst_Host_Srv_Count,</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Dst_Host_Same_Srv_Rate,</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Dst_Host_Diff_Srv_Rate,</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Dst_Host_Srv_Serror_Rate,</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Dst_Host_Srv_Serror_Rate,</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Protocol_Type,</td>
<td>Neptune</td>
</tr>
<tr>
<td>3</td>
<td>Service,</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Src_Bytes,</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Count,</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Srv_Count,</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Rrror_Rate,</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Srv_Rrror_Rate,</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Dst_Host_Same_Src_Port_Rate,</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Dst_Host_Rrror_Rate,</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Dst_Host_Srv_Serror_Rate,</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Hot,</td>
<td>Back</td>
</tr>
<tr>
<td>13</td>
<td>Num_Compromised,</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Land,</td>
<td>Land</td>
</tr>
<tr>
<td>8</td>
<td>Wrong_Fragment,</td>
<td>Teardrop</td>
</tr>
<tr>
<td>9</td>
<td>Urgent,</td>
<td>Ftp_Write</td>
</tr>
<tr>
<td>11</td>
<td>Num_Failed_Logins,</td>
<td>Guess_Pwd</td>
</tr>
<tr>
<td>14</td>
<td>Root_Shell,</td>
<td>Buffer_Overflow</td>
</tr>
<tr>
<td>22</td>
<td>Is_Guest_Login,</td>
<td>Warezclient</td>
</tr>
</tbody>
</table>

VIII. CONCLUSION

Data mining can improve intrusion based security attacks detection system by adding a new level of observation to detection of network data indifferences. It is highly required to identify appropriate features to categorize into different types of attack. Feature selection abbreviate the size of network data which improve finally performance of intrusion detection system. One R algorithm which is used for experimentation is an efficient algorithm of feature selection in KDD cup dataset. This feature identification helps to improve efficiency of intrusion detection system.

REFERENCES


Created with