Improving Network Intrusion Detection Through Classifier Combination

Lalindra De Silva, Ravindra Aditya Varma, Raghav Aggarwal
School of Computing, University of Utah
Salt Lake City, UT 84112

Abstract

Network intrusion detection is a problem that’s hardly being solved completely. Firewalls and other existing solutions do provide some resistance to the wide variety of attack types that can occur, but they suffer the drawback of not being able to generalize well into unseen attack types. Through this report, we propose a framework for addressing the problem of network intrusion by extracting ideas from the machine learning community. Specifically, we focus on ensemble methods where multiple schemes are joined to do a better job at predicting malicious connections. With this approach, we propose that malicious connections can be predicted with high precision and high confidence.

1 Introduction

As network systems get more complex by the day and new technologies emerge at a rapid pace, the vulnerabilities they are prone to also increases. The current systems/software in place for detecting network intrusions would not be able to prevent all types of attacks as new types appear. Present firewalls and other such mechanism rely on certain properties of a connection to determine if that connection is a normal or a malicious connection.

With that setting, it’s interesting to know if a predictive model can be trained to identify network intrusions using empirical methods. Having access to a large number of connection data, pertaining to both normal connections and malicious connections, it can be ascertained if it’s viable to produce models that don’t depend on certain properties as mentioned above. These models will take into account the patterns that appear in both normal and malicious network connections and by doing so, future attacks or malicious connections can be predicted by looking at the patterns in the data.

This concept has been utilized previously when researches have made an effort to incorporate machine learning methods into the field of network intrusion detection. However, most of those methods are based on frameworks surrounding single classifiers. The literature provides us only a handful of examples where multiple paradigms can be utilized to make this task more accurate. We propose, along those lines, that by combining classifiers, we could predict malicious attacks with high precision.

2 Related Work

For over a decade, there have been quite a number of research efforts in applying machine learning techniques to the problem of network intrusion detection problem. A large number of such efforts focus on the dataset provided by the KDD Cup 1999 competition (S. Stolfo et al., 1999). Most machine learning approaches to this problem of network intrusion detection focus on single classifier systems. (Behdad et al., 2010) and (Bernhard Pfahringer, 2000) provide two examples of such approaches. Also, there have been other methods such as clustering (Portnoy et al., 2001) and parzen windows (Yeung and Chow, 2002).

Genetic Algorithms have also played a role in this domain as shown by (Wei Li, 2004). A succinct, yet comprehensive review of how machine learning has been used for network intrusion detection is provided by (Tsai et al., 2009) and presents a few examples of classifier combination, or ensemble methods, used in this domain. One particular instance of such an ensemble method is presented by (Borji Ali, 2007).

3 Adversary Model

In our approach, we assume the adversary model to be comprised of the malicious data connections received by a node. These data connections are stored and certain features are either directly extracted or derived from them. With these features, we trained our ensemble model and as future connection come in, our model should be able
to identify each of them as being a malicious connection or not.

Figure 1: Framework

4 Datasets

For this task, we made use of the KDD cup 1999 dataset (S. Stolfo et al., 1999), which was made available through the Third International Knowledge Discovery and Data Mining Tools Competition in 1999. The dataset contains a wide variety of intrusions simulated in a military network environment. For simplicity, we have mapped all the different types of intrusions into one “attack” type class so that our models are able to carry out binary classification to determine if a connection is malicious or not.

The dataset accompanies 42 features with it. These 42 features are subcategorized using the properties of 1) being derived directly 2) content-based features and 3) traffic-based features.

5 Methodology

Our methodology (Figure. 1) begins, as explained in the previous section, with a data gathering phase where network data pertaining to a considerable time period are gathered. These data will be used as training data in the subsequent models that we describe later. Our framework makes use of four classifiers coming from four different paradigms. Namely...

- Margin-based Classifiers (eg: Support Vector Machines)
- Probabilistic Classifiers (eg: Logistic Regression)
- Decision Trees
- Genetic Algorithms

5.1 Margin-based Classifiers

Margin-based classifiers are a well known classification paradigm in the machine learning community with specific instances such as the Perceptron and Support Vector Machines (SVMs). Being one of the most widely used classification techniques, SVMs have been used in similar tasks before. The basic notion in SVMs is in creating a hyperplane separating the classes under consideration with the largest margin. It does so by optimizing w.r.t a weight vector $w$, usually through techniques such as Sequential Minimal Optimization (SMO). They also allow the use of kernel functions by which you can map your feature space into a higher dimension allowing better separation of classes. The SVM formulation is given as follows.

$$\text{Minimize } f(w, b) = \frac{||w||^2}{2}$$

Subject to $y_n (w^T x + b) \geq 1, n = 1, ..., N$

5.2 Probabilistic Classifiers

The advantage of using probabilistic classifiers instead of classifiers that make direct discrimination between classes is that, it allows the user to choose the selectivity of the classifier by adjusting some threshold value. Techniques such as Logistic Regression (LR) fall into this group. LRs also use training data to derive a weight vector which is ultimately used at prediction time in a sigmoid function, producing real valued outputs in the range $[0, 1]$. In the case of binary classification, as is in our case, a normal threshold of 0.5 is used to distinguish between the two classes.

$$P(y|x, w) = \sigma (w^T x) = \frac{1}{1 + \exp (-w^T x)}$$

5.3 Genetic Algorithms

Genetic Algorithms takes into account the process of natural evolution. It was originally presented as a methodology for selecting the best individuals for a population using a fitness function. Genetic Algorithms uses the processes of Crossover and Mutation. Crossover is the process of taking more than one parent solution and producing a child solution from them, whereas Mutation is the process which alters one or more gene values in a chromosome from its initial state. A fitness function interacts with these two to determine how closely fit a given chromosome is to population.

Genetic Algorithms have been extended to handle classification tasks, especially in the human language/text domain. We hypothesized that the same techniques can be applied to another domain, such as network intrusion detection, and be effectively used in that field too. In our scenario, a chromosome in the Genetic Algorithm corresponds to the $n - \text{length}$ features that are extracted from a given network connection.
5.4 Decision Trees

Decision trees is another classification technique that is widely used, especially when the number of features are small and are of discrete values. It starts by building a tree using the training data with the most informative features ending up at higher nodes of the tree. The way to select the most informative features is by means of entropy and information gain.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Precision</th>
<th>Recall</th>
<th>FScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>98.42 (500/508)</td>
<td>100.00 (500/500)</td>
<td>99.21</td>
</tr>
<tr>
<td>LR</td>
<td>98.62 (500/507)</td>
<td>100.00 (500/500)</td>
<td>99.30</td>
</tr>
<tr>
<td>GA</td>
<td>91.07 (500/549)</td>
<td>100.00 (500/500)</td>
<td>99.21</td>
</tr>
<tr>
<td>DT</td>
<td>99.60 (494/496)</td>
<td>98.80 (494/500)</td>
<td>99.20</td>
</tr>
<tr>
<td>SVM + LR</td>
<td>98.62 (500/507)</td>
<td>100.00 (500/500)</td>
<td>99.30</td>
</tr>
<tr>
<td>SVM + GA</td>
<td>100.00 (500/500)</td>
<td>100.00 (500/500)</td>
<td>100.00</td>
</tr>
<tr>
<td>SVM + DT</td>
<td>100.00 (494/494)</td>
<td>98.80 (494/500)</td>
<td>99.40</td>
</tr>
<tr>
<td>LR + DT</td>
<td>100.00 (500/500)</td>
<td>100.00 (500/500)</td>
<td>100.00</td>
</tr>
<tr>
<td>LR + GA</td>
<td>99.79 (494/495)</td>
<td>98.80 (494/500)</td>
<td>99.30</td>
</tr>
<tr>
<td>DT + GA</td>
<td>100.00 (494/494)</td>
<td>98.80 (494/500)</td>
<td>99.40</td>
</tr>
<tr>
<td>SVM + LR + DT</td>
<td>100.00 (494/494)</td>
<td>98.80 (494/500)</td>
<td>99.40</td>
</tr>
<tr>
<td>SVM + LR + GA</td>
<td>100.00 (494/494)</td>
<td>100.00 (500/500)</td>
<td>100.00</td>
</tr>
<tr>
<td>LR + DT + GA</td>
<td>100.00 (494/494)</td>
<td>98.80 (494/500)</td>
<td>99.40</td>
</tr>
<tr>
<td>SVM + LR + DT + GA</td>
<td>100.00 (494/494)</td>
<td>98.80 (494/500)</td>
<td>99.40</td>
</tr>
</tbody>
</table>

Table 1: Performance using different classifier combinations

<table>
<thead>
<tr>
<th>Attack:Normal</th>
<th>Precision</th>
<th>Recall</th>
<th>FScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:1 (500:500)</td>
<td>100.00 (494/494)</td>
<td>98.80 (494/500)</td>
<td>99.40</td>
</tr>
<tr>
<td>1:2 (500:1000)</td>
<td>100.00 (500/500)</td>
<td>100.00 (500/500)</td>
<td>100.00</td>
</tr>
<tr>
<td>1:3 (500:1500)</td>
<td>99.80 (483/486)</td>
<td>97.01 (485/500)</td>
<td>98.37</td>
</tr>
</tbody>
</table>

Table 2: Performance of the best combination in varying connection data distributions

5.5 Combining the classifiers

Having these four classifiers, we combine them to produce a model where the final decision for a particular network connection being normal or malicious is based on all classifiers having the same classification. i.e. a network connection is deemed malicious if all classifiers that are combined in the model ascertain that connection to be malicious. If even a single classifier disagrees, then the connection is deemed normal. This allows us to predict the malicious/attack network connections to a higher degree of precision albeit a small loss of recall.

6 Experiments

We carried out two types of experiments. First, we tried out different combinations of these classifiers to determine which combination performs best at this task. Surprisingly, almost all the classifiers, even in isolation, seemed to perform really well at this task. We attribute this high performance to the fact that the training and test data were drawn from the same distribution and that the 42 features which were used for this task were highly informative, allowing better discrimination. However, we noticed that as more and more classifiers were combined, the precision increased till, which suggested high confidence predictions of malicious/attack network connections. The second experiment was to determine if the combined classifiers are robust in identifying malicious connection in the midst of large number of normal connections. The result of the two experiments are provided in Tables 1 and 2.

7 Conclusions

We conclude that our experiments assure of our hypothesis that combining classifiers can lead to a high precision predictive model for the task of network intrusion detection. Having these models in practice would mean a certain lag between the time of receiving a connection and prediction due to the time it takes to derive features and execute these models. However, we believe this could be extended as a system given enough resources and methods to derive these features fast enough and we also believe these models will have some values at research level in identifying certain properties of these malicious connections and in analyzing them.

References

Behdad, Mohammad and French, Tim and Barone, Luigi and Bennamoun, Mohammed On the problems of using learning classifier systems for fraud detection, Proceedings of the 12th annual conference on Genetic and evolutionary computation
Bernhard Pfahringer Winning the KDD99 Classification Cup: Bagged Boosting. SIGKDD Explorations
Leonid Portnoy and Eleazar Eskin and Sal Stolfo Intrusion detection with unlabeled data using clustering, In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)
Tsai, Chih-Fong and Hsu, Yu-Feng and Lin, Chia-Ying and Lin, Wei-Yang Intrusion detection by machine learning: A review. In Expert Systems Applications, 2009