International Journal of Security and Its Applications
Vol. 10, No. 8 (2016), pp.335-342
http://dx.doi.org/10.14257/ijsia.2016.10.8.29

Developing Data Mining Techniques for Intruder Detection in Network Traffic

Amar Agrawal, Sabah Mohammed and Jinan Fiaidhi

Department of Computer Science, Lakehead University,
Thunder Bay, ON, Canada
aagrawa1@lakeheadu.ca, mohammed@lakeheadu.ca, jfiaidhi@lakeheadu.ca

Abstract

In this paper we have proposed a hybrid intrusion detection system consisting of a misuse detection model based upon a Binary Tree of Classifiers as the first stage and an anomaly detection model based upon SVM Classifier as the second stage. The Binary Tree consists of several best known classifiers specialized in detecting specific attacks at a high level of accuracy. Combination of a Binary Tree and specialized classifiers will increase accuracy of the misuse detection model. The misuse detection model will detect only known attacks. In-order to detect unknown attacks, we have an anomaly detection model as the second stage. SVM has been used, since it's the best known classifier for anomaly detection which will detect patterns that deviate from normal behavior. The proposed hybrid intrusion detection has been tested and evaluated using KDD Cup '99, NSL-KDD and UNSW-NB15 dataset.

Keywords: *Intrusion Detection System, Data Mining, Hybrid IDS*

1. Introduction

With the advent of internet there has been an up rise in the amount of traffic being generated online. This phenomenon has led to the accumulation of immense amount of data containing sensitive info, which can be accessed by unauthorized parties using attacks that are constantly evolving overtime. Hence there is a need for an Intrusion Detection System that can also evolve and catch these malicious activities. To better understand the problem we first need to know what an IDS is which is explained below:

1.1. Intrusion Detection Systems (IDS)

An IDS is a Hardware/Software that can detect attacks or unauthorized access on a system, raise an alarm in response and maybe also stop them. The IDS analyses the data that can be network-based, wireless-based, host-based or application-based. Thus we can classify IDS based on its architecture into four categories as follows:

- *Network Based:* It is placed at any point within a network that needs to be monitored. It then analyzes the network traffic and compares it with normal traffic flow.

- *Host Based:* As the name suggests the Host-Based IDS resides on a Host machine and monitors incoming and outbound traffic of the host. They usually take snapshots of the file or configuration of the system and look for changes. If changes are found, it triggers an alarm. It can also monitor which program is accessing which resource.

- *Application Based:* It monitors events of a specific application by reading logs or measuring performance. It usually monitors data sources that the application is connected to.

ISSN: 1738-9976 IJSIA

- *Wireless Base:* These IDSs look for malicious nodes and prevent unauthorized access to a particular LAN.

This paper is focused on solving problems based upon Network Based Systems. Network Based systems consist of several attacks which can be categorized based on the KDD cup 99 dataset as follows:

- Denial of Service (DoS): In this kind of attack, the attacker will clog the traffic by occupying all the resources making the service unavailable to other genuine users. SYN flood attack is an example of such an attack.

- User to Root (U2R): The attacker will use a normal account and try to gain root or other unauthorized privileges by exploiting the system. Buffer overflow is an example of such an attack.

- Remote to User (R2L): Attacker will send request and try to gain access to a User account. Brute force attack to crack a user's password is an example of such an attack.

- Probing (PROBE): The attacker will scan the network packets in a network to find vulnerabilities and exploit the system. Port scanner is an example.

IDS uses several techniques for detecting if an intrusion/attack has taken place such as Statistical approaches, Predictive pattern generation, Expert systems, Keystroke monitoring, Model-based Intrusion detection, State transition analysis, Pattern matching and Data mining techniques. Our focus in this paper is the use of Data Mining techniques in IDS.

Data Mining is the technique of finding patterns in large data sets. Using data mining an IDS can be further classified as Misuse and Anomaly detection.

1. *Misuse detection:* This classical method uses signatures (Known patterns) of behaviour/activity that can be used to detect or even predict if an attack is going to take place. These kind of detections are very strong with very few false alarms.

Pros: Precise and accurate detection.
Cons: Limited to database of known attacks

2. *Anomaly detection:* These systems are used to detect behaviour that is not normal. The system can be trained using normal behaviour patterns and when some kind of activity shows behaviour that is deviating from normal, it will raise an alarm.

Pros: Can detect novel attacks with less information of known attacks
Cons: Cannot precisely identify an attack

There are Hybrid IDS that are a combination of the both a Misuse Detection as well as an Anomaly detection system, since they hold advantages of using both a Misuse and Anomaly based IDS. The IDS that we proposed is a Hybrid IDS.

2. Related Work

In paper [1] have proposed a signature based Intrusion Detection system which is a combination of two popular data mining algorithms, Apriori (Association) and K-means (Cluster). With several frequent item sets, large item sets, or very low minimum support, Apriori will suffer from the cost of generating a huge number of candidate sets as per survey of [2]. Also simply having a signature based IDS will only catch known attacks. Paper [3] have proposed an IDS on machine learning based data classification which is a combination of SVM as well as Ant Colony networks (CSOACNs). This kind of IDS on its own is Anomaly based and thus is not as fast and accurate as a misuse based IDS, it will have a high false alarm rate. The only advantage as stated by the author is that it can outperform traditional SVM and Clustering based on Self-Organized Ant Colony Network (CSOACN) algorithm when combined with ant colony.

The work of [4] demonstrates a Binary Tree consisting of the best classifiers for known attacks. Every level consists of respective classifier. If the classification fails in level one, then its passed on to the next best classifier for level 2 until none are found. This type of IDS is appropriate since it's quick and accurate. The downside for this technique is that if the attack is unknown then it will not be able to classify an attack and the attack can go through without detection. The characteristics proposed by this author have been considered in this paper for the proposal of hybrid intrusion detection system. Also [5] proposed a cloud IDS for CIDD dataset which contains instances of masquerade attacks. The data mining technique used is a modified version of Apriori as proposed by Agrawal. It is divided into three categories Apriori (normal) - large datasets, AprioriTID - datasets that can fit in memory and Apriori Hybrid - time consumption for switching. This is similar to the research of [1] who have used Apriori algorithm. Apriori will suffer from the cost of generating a huge number of candidate sets as per survey of (Wu *et. al.*, 2008) and thus it's not a good choice for an IDS where memory will be a key factor.

In [6] they have proposed a Hybrid IDS that uses a Misuse detection as its first stage since it's accurate and fast for detecting known attacks and then used an Anomaly detection as its second stage. As per performance test conducted by [6], the Hybrid IDS delivered a much better performance compared to conventional IDSs. But the use of a C4.5 decision tree for the misuse detection is not appropriate as they have mentioned that C4.5 decision tree does not consider clusters in normal data set and thus can divide a well formed normal cluster during decomposition and degrading its profiling ability. Based on the future scope of this author a new replacement for the tree first stage has been considered in this paper.

3. Methodology

There is a need for an IDS that can be trained on various sections of the dataset. This way we can have multiple IDSs that can specialize in various sections of the datasets. Example, we can divide the KDDCUP99 dataset based on its attack types into DOS, U2R, Probe and R2L. We can then choose algorithms that are specialized at classifying respective attack types. The more we fragment the dataset, the more accurate results we can obtain, but it will affect the speed of the algorithm. But, not all algorithms are trained for new attacks. There is a possibility for an unknown attack, that hasn't been trained to disguise as a normal attack and bypass the IDS. Thus to detect something that isn't normal we need to have an Anomaly based IDS as well. An anomaly based IDS will look for behaviour that isn't normal and will sound the alarm. Thus, we have built a hybrid IDS consisting of a Binary Tree of Classifiers as suggested in [4] which will be first stage as a misuse system. It will be then followed by an SVM Multi-Class Classifier anomaly based detection system as proposed by [6]. We have tested each classifier against DOS, R2L, U2R and Probe attacks and can conclude the best algorithms in accuracy and performance ratio for every such group of attack classes.

Table 1. Best Classifiers Group

Attack Class	Classifiers
DOS	Ripper
R2L	Ridor
Probe	J48
U2R	Random Tree
Normal	

The system diagram for the proposed method will be as shown in Figure 1.

3.1. Datasets

In order to evaluate our proposal and for benchmarking purposes we need to use a set of datasets. These datasets will provide us statistics of how our proposed algorithm has fared among others. The dataset should consist of attack and normal records. KDD cup 99 is the most widely used dataset since it has been used as a benchmark to compare other IDSs in several research papers. NSL KDD has also been used, which is a refined version of KDD Cup 99. KDD Cup 99 consists of several redundant records, which causes learning algorithms to be biased towards frequent records and thus producing a more accurate result. In such scenarios the classifiers aren't trained well enough with infrequent records. UNSW-NB15 Dataset is a fairly new dataset developed by Cyber Range Lab of the Australian Center for Cyber Security (ACCS). Even though we have used benchmark datasets like KDDCUP99 and NSLKDD datasets to evaluate our algorithm, we need an updated dataset that will reflect and simulate current network threat environments. For comparing UNSW-NB15 with NSL KDD and KDDCUP99 we ran a couple of tests on the algorithm as follows:

Table 2. Comparing UNSW-NB15 with NSL KKD, KDD Cup 99 and Proposed Model

Classifier	UNSW-NB15	NSL KDD	KDDCUP99
ZeroR	44.94%	53.39%	56.84%
Naïve Bayes	44.72%	89.66%	92.76%
J48	87.52%	98.47%	98.95%
Proposed Model	88.55%	98.80%	99.92%

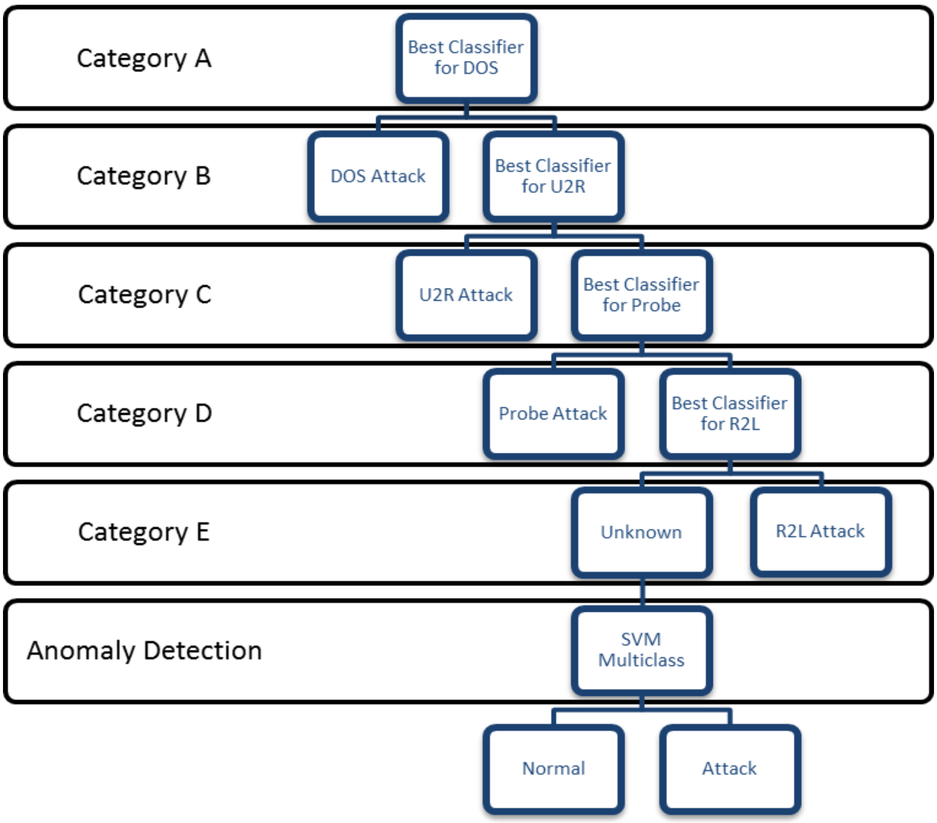

Figure 1. Proposed Hybrid IDS

4. Experiments and Results

4.1. Weka API and Pseudo Code

To carry out the test of our proposed research we made use of the Weka API. Weka API was used in the java program to design our custom classifier. Some code snippets have been included for better understanding the usage of Weka API in Java. The code below creates the data source which can be any input stream like file, database *etc*. We have used a request input stream for web based application. We then use Instances class to retrieve all instances from the above connected stream.

```
DataSource trainSource = new DataSource(inStream);
Instances totalInstances = trainSource.getDataSet();
```

We then created an instance of the Custom Classifier class which we have inherited and built the Classifier using all the instances. Since we are using cross validation we call crossValidateModel which is a custom method that will call cross validate function to evaluate our model. 5 here denotes the number of folds to be used. Results are then displayed by calling toSummaryString.

```
customClassifier.buildClassifier(totalInstances);
Evaluation eval = new Evaluation(totalInstances);
eval.crossValidateModel(customClassifier, totalInstances, 5, new Random(1));
String resultOfClassifier = eval.toSummaryString("\nResults\n ", false);
```

When we extend the Classifier abstract class, we need to override two functions namely buildClassifier and classifyInstance. The classify Instance class is the most important since this is where we define the logic to classify the instance. In the code below if the prediction is normal then the loop will continue until an attack is detected. If an attack is detected it will return the prediction value and exit the loop similar to Figure 1.

```
@Override
public double classifyInstance(Instance instance) throws Exception {
        double isNormal = -1.0;
        // Loop through all classifiers in the tree
        for (Classifier classifier : classifiers) {
                double pred = classifier.classifyInstance(instance);
                // If normal it might be another attack so continue else break
                if(("normal").equalsIgnoreCase(classAttribute.value((int) pred)))
                {
                        //store very first prediction
                        if (isNormal == -1.0)
                                isNormal = pred;
                }
                else
                {
                        if(!(classAttribute.value((int)instance.classValue()))
                        .equals(classAttribute.value((int) pred)))
                        return pred;
                }
```

After running the test, we obtained a set of results which are represented as follows. The diagram below gives us an overview on the accuracy of the algorithm. As we can see highest accuracy has been achieved for KDD Cup 99 (99.92%), NSL KDD (98.8%) which is fairly close while there's a quite difference in UNSW-NB15 due to the varying attacks in this fairly new dataset. In spite of the variations in attacks by UNSW-NB15 dataset, the classifier has fared pretty well with an accuracy of 88.5464%.

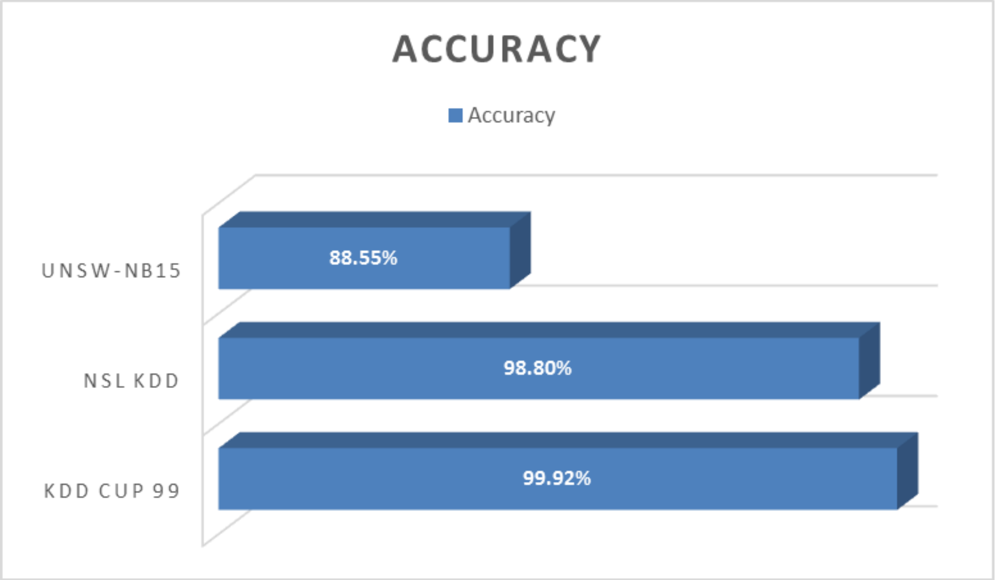

Figure 2. Dataset Accuracy Comparison

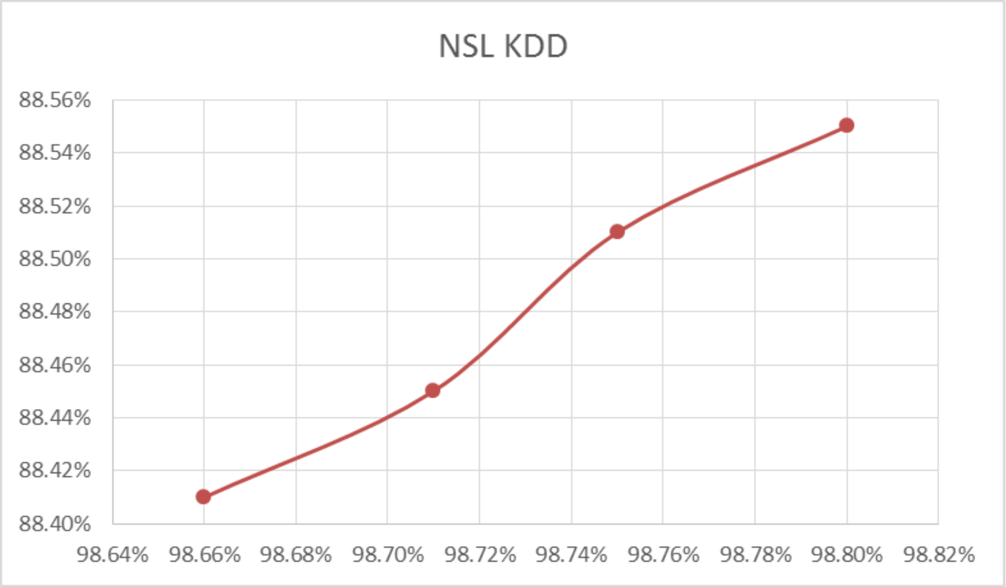

Figure 3. Correlation between NSL KDD and UNSW-NB15

From the figure above we can observe the correlation between UNSW-NB15 and NSL KDD dataset when the number of folds is increased. We have increased the number of folds in 2, 4, 8 and 10s order from left to right in an increasing order. When we compare our proposed classifier against other existing well known classifiers we obtained results which are represented in Figure 4.

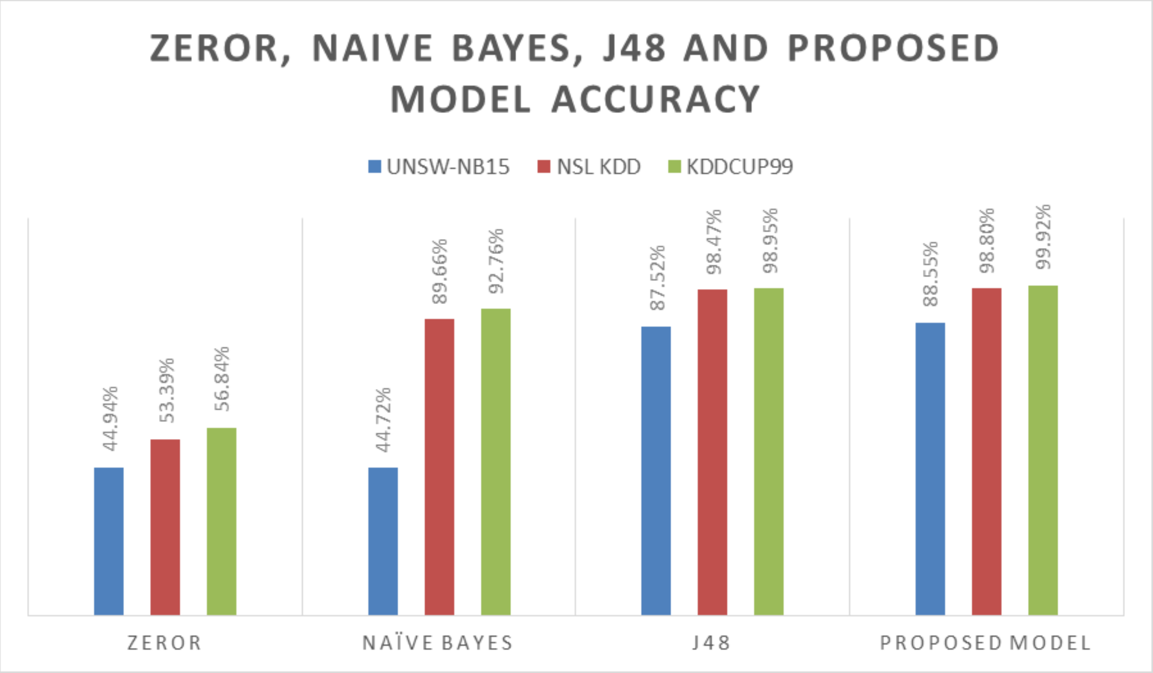

Figure 4. ZeroR, Naive Bayes, J48 and Proposed Model Accuracy Comparison

5. Conclusion and Future Work

The main goal of the paper was to prove that we can further reduce False Negative and False Positive rate in Network Intrusion Detection Systems. By creating a binary tree of classifiers and selecting the best classifier for every kind of attack increases the accuracy of the algorithm and reduces the number of false negatives and false positives that are generated. Furthermore, the use of anomaly detection by using SVM decreases the chances of an unknown attack to disguise as a normal attack and pass through. Since this kind of classifier has been designed based on a specific set of attack classes, it can be further improved by fragmenting the classifier even further to cover other attack classes using faster and efficient classifiers. There can be more experimentation done on other kind of datasets and classifiers.

References

[1] B. Sharma and H. Gupta, “A Design and Implementation of Intrusion Detection System by Using Data Mining”, 2014 Fourth International Conference on Communication Systems and Network Technologies., **(2014)**, pp. 700–704.

[2] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand and D. Steinberg, “Top 10 algorithms in data mining”, Knowl. Inf. Syst., vol. 14, no. 1, **(2008)**, pp. 1–37.

[3] W. Feng, Q. Zhang, G. Hu and J. X. Huang, “Mining network data for intrusion detection through combining SVMs with ant colony networks”, Futur. Gener. Comput. Syst., vol. 37, **(2014)**, pp. 127–140.

[4] A. Ahmim and N. Ghoualmi Zine, “A new hierarchical intrusion detection system based on a binary tree of classifiers”, **(2015)**.

[5] M. Kumar and R. Mathur, “Data Mining based CIDS: Cloud Intrusion Detection System for Masquerade Attacks [DCIDSM]”, International Conference for Convergence for Technology-2014., **(2014)**, pp. 1–4.

[6] G. Kim, S. Lee and S. Kim, “A novel hybrid intrusion detection method integrating anomaly detection with misuse detection”, Expert Syst. Appl., vol. 41, **(2014)**, pp. 1690–1700.