Fourth 21st CAF Conference in Harvard,
Boston, Massachusetts, USA,
March 2016, Vol. 9, No. 1
ISSN: 2330-1236

Constructing a Supervised Model for Network Intrusion Detection

Tigabu Dagne Akal
Addis Ababa Institute of Technology
Center of Information Technology and Scientific Computing
Addis Ababa, Ethiopia

Abstract

Computer networks are dynamic and continually evolving. Along with such evolution, it becomes harder to effectively communicate to human decision-makers the results of methods and metrics for monitoring networks, classifying traffic, and identifying malicious or abnormal events. Network administrators and security analysts require tools that help them understand, reason for, and make decisions about the information their analytic systems produce. Because of the dynamic change of the technology and the increasing number of hackers and crackers in the networking industry there should be a means to minimize or remove such challenges. Data mining is one of the technologies that are used for intrusion detection and prediction. In this study, attempts have been made to use data mining technology with the aim of detecting and predicting intrusions in the networking industry. The knowledge discovery in database process model designed by Fayyad et al. (1996) has been followed during the experimentation and discussion. The dataset used in this study has been taken from MIT Lincoln lab. After gathering the data, it has been preprocessed and prepared in a format suitable for the data mining tasks. This study proposed the supervised approach for IDS. The proposed model will offer the advantage of considering those unlabeled records. In this case there was a filling of only the top few most confident data points making empty the class of rest records. Supervised learning is more suitable for intrusion detection because they require a small quantity of labeled data while still taking advantage of the large quantities of unlabeled data. Both the J48 decision tree algorithm and the Naïve Bayes simple algorithm have been tested as a classification approach for building a predictive model for intrusion detection. By changing the training test options and the default parameter values of these algorithms, different models have been created. The model created using 10-fold cross validation using the J48 decision tree algorithm with the default parameter values showed the best prediction accuracy.

Keywords: Data Mining, security, intrusion detection, knowledge discovery in database

Introduction

As network-based computer systems play increasingly vital roles in modern society, they have become the targets of cyber criminals. The security of a computer system is compromised when an intrusion takes place. An intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource (Heady et.al, 1990). Lee et al (1999) defined intrusion as the act or attempted act of using a computer system or computer resources without the requisite privileges, causing willful or incidental damage. Intrusion Detection Systems (IDS) are computer programs that attempt to perform intrusion detection by comparing observable behavior against suspicious patterns, preferably in real-time. IDSs are systems that attempt to identify intrusions or abuses of computer systems by either authorized users or external perpetrators (Mukherjee et al., 1994). Some IDSs monitor a single computer, while others monitor several computers connected by a network. IDSs detect intrusions by analyzing information about user activities from sources such as audit records, log files, system tables, and network traffic summaries.

Literature Review

IDSs have been developed and used at several institutions. Some example of IDSs are National Security Agency's Multics Intrusion Detection and Alerting System (MIDAS), ATandT's Computer Watch (Dowell and Ramstedt, 1990), SRI International's Intrusion Detection Expert System (IDES) (Lunt,1990), Next-Generation Intrusion-Detection System (NIDES) (Anderson, 1994), UC Sanat Barbara's State Transition Analysis Tool for UNIX (USTAT) (Ilgun, 1993; Ilgun et al.,1995), Los Alamos National Laboratory's (LANL's) Network Anomaly detection and Intrusion Reporter (NADIR) (Hochberg, 1993), UC Davis' Network Security Monitor (NSM) (Heberlein et al.,1990) and Distributed Intrusion Detection System (DIDS) (Snapp, 1991). Intrusion prevention techniques such as user authentication (e.g. using password or biometrics), avoidance of programming errors, and information protection (e.g., encryption) have been used to protect computer systems as a first line of defense (Lee et al., 1999). Intrusion detection is needed as a wall to protect computer systems. The elements central to intrusion detection are: resources to be protected in a target system, i.e., user accounts, file systems, system kernels, etc; these are models that characterize the normal or legitimate behavior of these resources; techniques that compare the actual system activities with the established models, and identify those that are abnormal or intrusive (Lee et al., 1999). According to Mukherjee et al (1994) the goal of intrusion detection is to identify, preferably in real time, unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrators.

Generally, an intrusion would cause loss of integrity, confidentiality, denial of resources, or unauthorized use of resources. According to Eric et al (2002), some specific examples of intrusions that concern system administrators include:

- ✓ Unauthorized modifications of system files so as to facilitate illegal access to either system or user information;

- ✓ Unauthorized access or modification of user files or information;
- ✓ Unauthorized modifications of tables or other system information in network components (e.g. modifications of router tables in an internet to deny use of the network);

The most widely used and commercially available IDSs are signature-based systems (William, 1995). A signature based system matches features observed from the audit stream to a set of signatures handcrafted by experts and stored in a signature database. Signature-based methods have some inherent limitations. What is significant is that a signature-based method is designed to only detect attacks for which it contains a signature in the database. In addition to the expense in time and human expertise of manually encoding a signature for each and every known attack, the signature-based methods therefore cannot detect unknown attacks since there is no signature in the database for them. It is these unknown attacks that are typically the most dangerous because the system is completely vulnerable to them (William, 1995).

Data mining (DM) -based methods are another paradigm for building intrusion detection systems. The main advantage of these methods is that they leverage the generalization ability of data mining methods in order to detect new and unknown attacks. A data mining-based IDS uses machine learning and data mining algorithms on a large set of system audit data to build detection modes. These models have been proven to be very effective (Lee et al., 1999). These algorithms are generally classified as either misuse detection or anomaly detection. Misuse detection algorithms learn how to classify normal and attack data from a set of training data which contains both labeled attack and normal data (Lee et al., 1999). Anomaly detection algorithms learn a model of normal activity by training on a set of normal data. Anomaly detection algorithms then classify as an attack activity that diverges from this normal pattern based on the assumption that attacks have much different patterns than normal activity.

According to Christos and Aikaterini (2004) a DM intrusion detection system (IDS) inspects all inbound and outbound network activities and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. According to Christos and Aikaterini (2004) data mining has been known to aid the process of Intrusion Detection and the ways in which the various techniques have been applied to enhance the security of the network.

Generating patterns and knowledge is vital for IDSs to differentiate standard behaviors from strange behavior by examining the dataset which is a list of tasks created by the operating system that are registered into a file in historical sorted order (Dewan & Mohammad, 2010). According to Pachghare et al (2011) IDS can be implemented using unsupervised, supervised and semi-supervised machine learning algorithms. Unsupervised learning uses unlabeled data. This method can detect the intrusions that have not been previously learned. Examples of unsupervised learning for intrusion detection include K-means-based approaches and self-organizing map (SOM)-based approaches. In supervised learning for intrusion detection, the labeled data is needed for training. These are mainly neural network (NN)-based approaches, and support vector machine (SVM)-based approaches for IDS. The third method is semi-supervised learning in which both the labeled and unlabeled data are used for training.

Statement of the Problem

Intrusion detection is becoming a challenging task due to the proliferation of heterogeneous computer networks since the increased connectivity of computer systems gives greater access to outsiders and makes it easier for intruders to avoid identification (Helali, 2010). Hence, there is a need for an effective and efficient system which allows the protection of the network from the intruders. To develop such kind of system there is a need to use methods like feature selection which is a growing field of interest involving selecting proper features from many features. This is because it is expensive to carry out the entire process and degrading to the classification performance of data mining algorithms. Therefore, feature selection approaches reduce the complexity of the overall process by allowing the data mining system to focus on the really important features.

Many researchers proposed different models for network intrusion detection systems (NIDS). Adamu (2010) tried to study a machine learning IDS that investigated the application of cost sensitive learning by applying decision tree algorithm. He did not compare the result with other predictive model techniques like neural network, Naïve Bayes and other techniques. Zewdie (2011) proposed an optimal feature selection for Network Intrusion Detection using an indirect cost sensitive feature selection approach. The latter is a DM approach system that tried to investigate jointly cost sensitive learning and feature selection to advance the classification performance of algorithms that incorporate cost. In his study, Information Gain Ratio (IGR) and Correlation Feature Selection (CFS) are investigated for ranking and selecting features using the proposed cost sensitive approach. Zewide tried to investigate decision tree classification algorithms that used indirect cost sensitive feature ranking and selection algorithms. Zewdie used in his study only those records which are labeled. He did not consider those records which are not labeled. Both Adamu (2010) and Zewdie (2011) conducted the NIDS on a supervised approach. As described by Pachghare et al (2011), a traditional intrusion detection algorithm is based on supervised learning and non-supervised learning. These two algorithms have some limitations; the supervised learning process cannot use a lot of unlabeled data while non-supervised learning often results in a high false alarm rate.

Pachghare et al (2011) evaluated the performance of the supervised intrusion detection model using labeled data. They concluded that labeling the training data for real-world applications is difficult, expensive, and time consuming, as it requires the effort of human sometimes with specific domain experience and training. There are implicit costs associated with obtaining these labels from domain experts, such as limited time and resources. This is especially true for applications that involve learning with a large number of class labels and sometimes with similarities among them. Therefore, this research intends to get answers to the following research questions.

- ✓ Which Data Mining algorithm can be more suitable for the purpose of predicting Network Intrusions?
- ✓ To what degree can the NIDS correctly classify intrusions? Can the system correctly classify intrusion to such a degree that it can be trusted to respond actively to them?
- ✓ What is the pattern that describes whether given networks signal is a normal packet or an intrusion?
- ✓ What can be done to design an IDS model which is based on feature selection?

This research is basically the extension of the thesis work of Adamu (2010) and Zewdie (2011) in the area of NIDS. On top of their work features which were not addressed by both of them have been addressed in this paper, namely, the result of the J48 decision tree algorithm compared with other predictive model techniques in developing an IDS model. For feature selection CfsSubsetEval is used as attribute evaluator and Best First as a search method is also used in this study. This research addressed the supervised modeling of an intrusion detection system that considers both labeled and unlabeled records which were indicated as a future research direction by Pachghare et al (2011). It is not easy to classify network packets whether attack or normal that always need domain experts for applying only supervised modeling. At the same time, there is the labeling of the class of network packets consumed resources. Because datasets were taken from the Massachusetts Institute of Technology (MIT) Lincoln lab, this research did not include data from network security organizations in Ethiopia. So, further research needs to be conducted including data from these organizations. Because of time and financial limitations this research focused mainly on how to effectively detect attacks, not to prevent them. The IDS model constructed in this thesis just notify network administrators after detecting an attack and administrators to manually take proper actions.

System Design and Data Preprocessing

Data processing, a critical initial step in data mining work, is often used to improve the quality of training data set. To do so data cleaning and preparation is the core task of data mining which is dependent on the software chosen and algorithms used (Mahbod et al., 2009). The IDS models in this study are developed on full training Network Simulation Language- Knowledge discovery in Database (NSL-KDD) dataset using a powerful machine learning and data mining WEKA tool. The data mining model used in this study is the KDD process. The KDD process refers to the whole process of changing low level data into high level knowledge whose automated discovery of patterns and relationships in large databases and data mining is one of the core steps in the KDD process. The goal of KDD and DM is to find interesting patterns and/or models that exist in databases but are hidden among the volumes of data (Fayyad et al., 1996). The KDD process as described by Fayyad et al (1996) consists of five major phases. Data were collected using appropriate algorithms then mined patterns were modeled. Figure 1 showed the KDD process model used in this thesis.

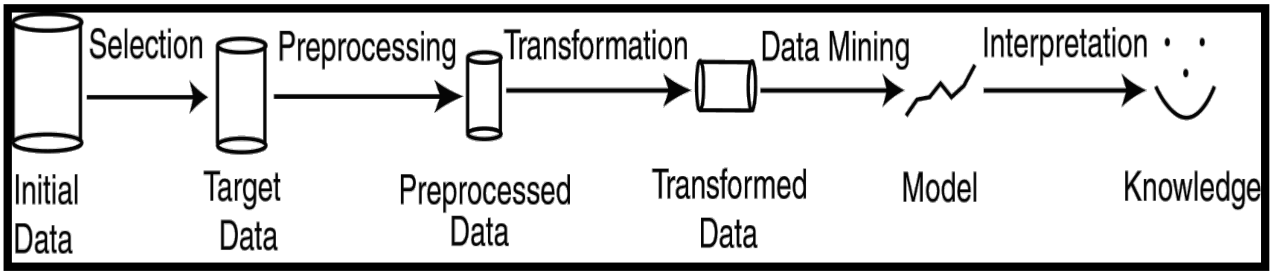

Figure 1: An overview of the steps that compose the KDD process (Fayyad et al., 1996).

Initial Data Selection: NSL_KDD dataset (Helali, 2010) most widely used and publicly available for IDS is used for the experiment purpose. The KDD (Knowledge discovery in Database) Cup 1999 Intrusion detection contest data (KDD cup 99 Intrusion detection data sets) has been used in this study. This data was prepared by the 1998 DARPA (Defense Advanced research Project Agency) Intrusion Detection Evaluation program by MIT Lincoln Labs (MIT Lincoln Laboratory). Data Preprocessing*:* The data preprocessing step in this study includes basic operations, such as removing noise or outliers if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time sequence information and known changes, deciding on database management system issues, such as data types, schema, and mapping of missing and unknown values. Also, since a predictor can exploit only certain data features, it is important to detect which data preprocessing works best (Meera et al., 2003). For this study preprocessing of NSL-KDD dataset contains the following processes: assigning attack names to one of the five classes Normal, Probe, DOS (Denial of Service), U2R (User to Root) and R2L (Remote to Local). To identify and label each attack, different literatures are consulted and Microsoft Excel helps to filter and name easily using fill handle.

There are records which don't have attributes and are removed from the dataset and there is also a mismatch in the KDD 99 winner cost matrix and the confusion matrix; as a result arrangements are made to match the cost matrix and confusion matrix. The NSL-KDD dataset is available in text format; so to be read for Waikato Environment for Knowledge Analysis (WEKA) tool it has to be changed into ARFF format. For WEKA Data can be imported from a file in various formats: CSV, C4.5, binary (Chang et al., 2005).

Data Transformation: the data transformation step includes finding useful features to represent the data, depending on the goal of the task, and using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration.

Choosing Data mining tasks*:* In this step the DM methods used for the thesis are decided. DM methods have been successfully applied for solving classification problems in many applications (Pradeep, 2005). In DM, algorithms (learners) try to automatically filter the knowledge from example data (datasets).This knowledge can be used to make predictions about original data in the future and provide insight into the nature of the target concept(s). According to Pradeep (2005) example data typically consists of a number of input patterns or examples to be learned. DM systems typically attempt to discover regularities and relationships between features and classes in the learning or training phase. To analyze the data and classify of network attacks from a network environment, the three machine learning algorithms (Eibe & Witten, 2005), the J48 decision tree classifier, Naïve Bayes Classifier and simple k-means clustering are used in this thesis.

Decision Tree: Decision tree is a predictive modeling technique most often used for classification in DM. The Classification algorithm is inductively learned to construct a model from the pre-classified dataset. Each data item is defined by values of the attributes. Classification may be viewed as mapping from a set of attributes to a particular class. The Decision tree classifies the given data item using the values of its attributes. The decision tree is initially constructed from a set of pre-classified data. The main approach is to select the attributes, which best divides the data items into their classes (Kruegel & Toth, 2003). In this study the J48 decision tree algorithms was used. It is an implementation of the C4.5 decision tree learner. This implementation produces decision tree models. It recursively splits a dataset according to tests on attribute values in order to separate the possible predictions. A decision-tree model is built by analyzing the training data and the model is used to classify the trained data.

The node of the J48 decision trees evaluates the existence and the significance of every individual feature. Considering a set A of case objects, J48 initially grows a tree and uses divide-and-conquer algorithm as follows: (i) if all the cases in A belong to the same class or if the set is a small one, the tree is leaf labeled with the most frequent occurring class in A. (ii) or, a test is selected based on a single attribute with two or more outcomes. This test is made with the root of the tree with each branch as one outcome of the test. Further the same procedure is applied recursively for each subset. Naive Bayes: the other supervised approach used in this thesis is the Naïve Bayes classifier which is based on probabilistic model for assigning the most likely class to a given instance. Probabilistic model (approach) in classification field allows (model or looks for) the estimation of conditional probability of classes given instance, p(C/A1…, AN) where C∈ {C1…CM} the classes and Ai, i=1...N, a set of features describing dataset examples (Shekhar et al., 2007). Given a valued example, the most appropriate class to be assigned to is the class with the upper posterior probability,

$$\text{Argmaxc } p(C=c/A1=a1\ldots, AN=aN)\ldots\ldots\ldots\ (1)$$

The Bayesian approach splits a posterior distribution into a priori distribution and likelihood,

$$\text{Argmaxc } p(C=c/A1=a1\ldots, AN=aN) = \text{Argmaxc } \alpha\ p\ (A1=a1\ldots AN=aN/C=c)\ p(C=c)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (2)$$

Where α is normalization factor to ensure that sums of conditional probabilities over class labels are equal to 1. The distribution of features given class label is more complex to estimate. Its estimation is exponential in an attribute number and requires a complete training dataset with sufficient examples for each class. Such problem can be avoided, assuming the independence of features of given class, and likelihood estimation uses the following formula.

$$P\ (A1=a1\ldots AN=aN\ /C=c) = \Pi i\ p\ (Ai=ai\ /C=c)\ldots\ldots\ldots\ldots\ldots\ldots\ (3)$$

Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning mode for this study.

K-means Clustering*:* in this study for semi-supervised modeling, the researcher used k-means clustering. The k-means clustering algorithm is used for clustering those unlabeled records into their appropriate classes. After clustering, classification techniques are applied. In K-means clustering, the assignments of the data points to clusters depend on the distance between cluster centroids.

Architecture of the study

Supervised intrusion detection approaches use only labeled data for training. To label the data however is often difficult, expensive, or time consuming as it requires the efforts of experienced domain experts. Semi-supervised learning addresses this problem by using a large amount of unlabeled data, together with the labeled data, to build better classifiers. Semi-supervised learning requires less human effort. The architecture used for this thesis is shown in figure 2. This architecture was proposed by Pachghare et al. (2011) for the semi-supervised approach for intrusion detection systems. As shown in figure 2, labeled data used for training the system as supervised approach. After training, the system test used unlabeled data. The tested data will add to the training data so as to implement a semi- supervised approach.

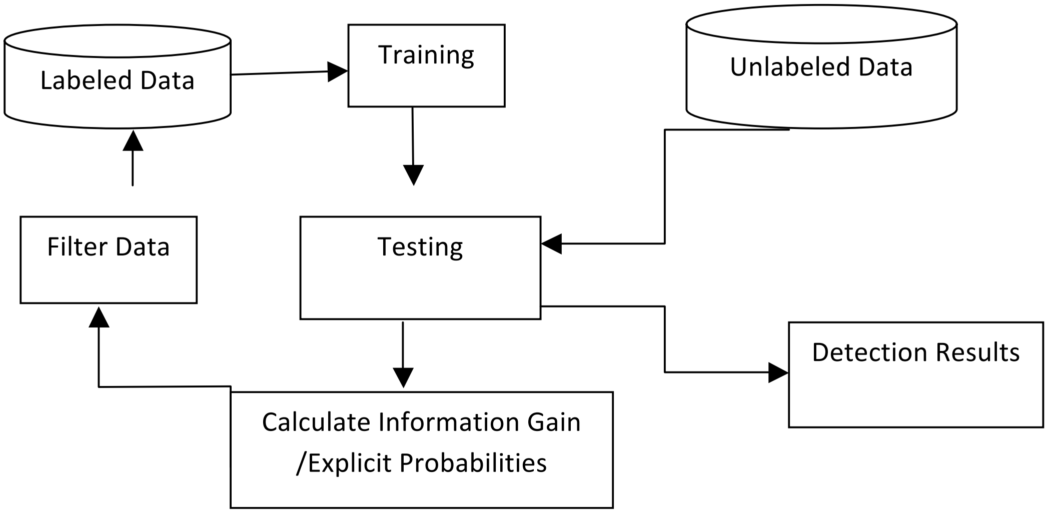

Figure 2: Architecture proposed for Semi-supervised IDS (Pachghare et al., 2011)

Implementation results and Comparison of Supervised Approaches: J48 decision tree and Naive Bayes models

Comparing different classification techniques and selecting the best model for predicting the network intrusions is one of the aims of this study. Accordingly the decision trees particularly the J48 algorithm and the Naïve Bayes classification approaches were used for conducting experiments. A summary of experimental result for the two classification algorithms is presented in table 1 below:

		Accuracy			
		Input ares all Features		With feature Selection	
Classifier/ Model	**Test Mode**	Correctly Classified	Incorrectly Classified	Correctly Classified	Incorrectly Classified
Naïve Bayes: Semi-Supervised	10-fold cross validation and Other default values	94.82 %	5.18 %	94.02%	5.98 %
	Percentage Split	94.67%	5.33%	94.02%	5.98%
J48 : Semi-Supervised	10-fold cross validation and Other default values	96.11 %	3.89%		
	Percentage Split	95.95%	4.05%		

Table 1: Comparison of Semi-Supervised Approaches

For comparison of the selected models summarized experimental results are shown in the table 2 below.

Features used	Classifiers	Accuracy	Classes									
			Normal		DOS		Probe		R2L		U2R	
			TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
17	**J48 with 10 fold cross validation**	96.11 %	0.95	0.01	1	0	0.97	0.01	0.98	0.02	0.53	0.01
17	**J48 with percentage split (set to 75%)**	95.95%	0.95	0.01	1	0	0.97	0.01	0.98	0.02	0.44	0.01
41	**Naïve Bayes with 10 fold cross validation**	94.82 %	0.92	0	1	0	0.87	0	0.92	0.02	0.93	0.03
41	**Naïve Bayes with percentage split (set to 75%)**	94.67%	0.93	0	1	0	0.85	0	0.91	0.02	0.91	0.03
11	**Naïve Bayes with 10 fold cross validation**	94.02%	0.92	0	1	0	1	0.03	0.92	0.02	0.35	0.02
11	**Naïve Bayes with percentage split (set to 75%)**	94.02%	0.93	0	1	0	1	0.03	0.91	0	0.35	0.03

Table 2: Comparison of the confusion matrix results for J48, and Naïve Bayes Algorithms

In this thesis J48 and Naïve Bayes algorithms performed different prediction accuracies. As shown in figure 3 from all six experiments, the J48 with 10-fold cross validation performed better classification accuracy in identifying intrusions whether normal or attack (DOS, U2R, R2L and Probe).

The reason for the J48 decision tree performing better than Naïve Bayes is because of the linearity of the dataset. This means there is a comprehensible segregation point that can be defined by the algorithm to predict the class of a particular network intrusion.

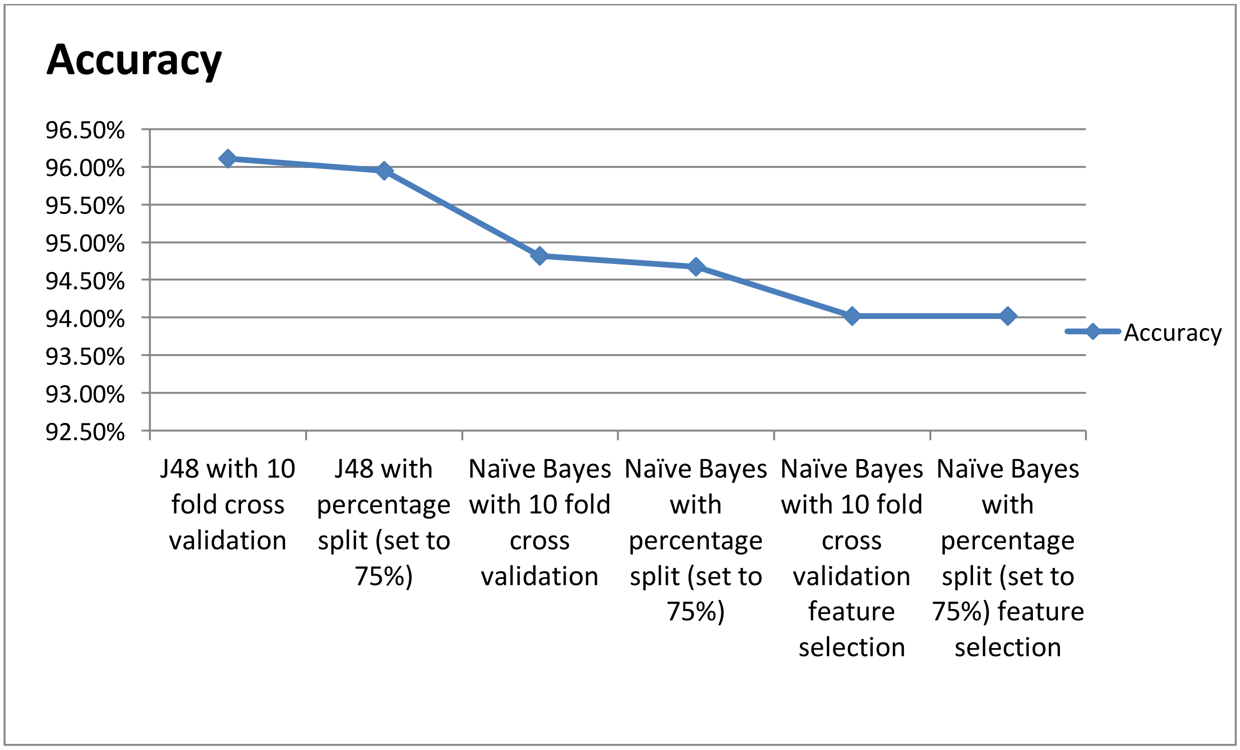

Figure 3: Comparison of Accuracy the J48 and Naïve Bayes Algorithms

The other reason for the Naïve Bayes, scoring a lower accuracy than the J48 decision tree is because class conditional independence assumption may not hold for some attributes, therefore causing a loss of accuracy. In addition, the ease of interpreting and implementing the J48 decision tree is more self-explanatory. It can a handle large number of features and generate rules that can be converted to simple and easy to understand classification if-then-else rules. The average TP and FP rates for all experiments conducted in this study are shown in table 3.

Algorithms	TP	FP
J48 with 10 fold cross validation	0.96	0.008
J48 with percentage split (set to 75%)	0.96	0.008
Naïve Bayes with 10 fold cross validation	0.95	0.004
Naïve Bayes with percentage split (set to 75%)	0.95	0.004
Naïve Bayes with 10 fold cross validation feature selection	0.94	0.006
Naïve Bayes with percentage split (set to 75%) feature selection	0.94	0.006

Table 3: Average TP and FP Rates

As shown in the following figure 4, the TP rate of the J48 algorithm is higher in most classes when compared with other algorithms.

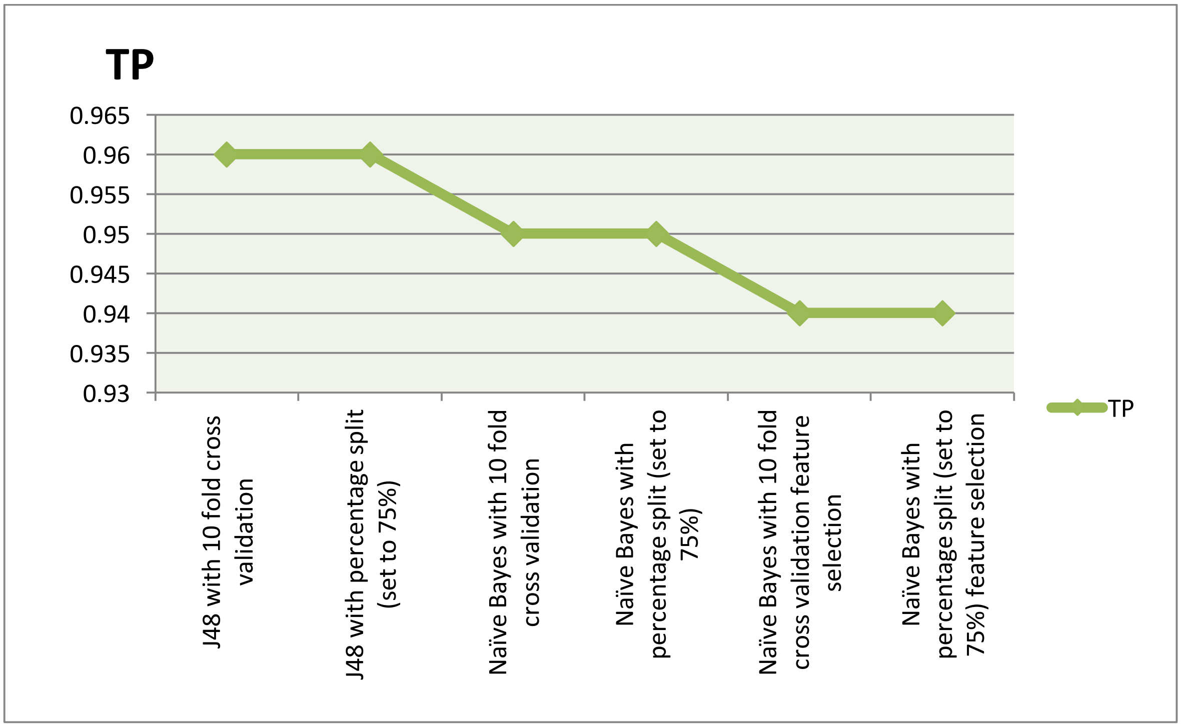

Figure 4: True Positive (TP) rate comparison of the J48 and Naïve Bayes Algorithms

A good IDS FP rate should be low. As shown in figure 4 the FP rate of the J48 algorithm for both cases (10 fold-cross validation and percentage split mechanism) is higher when compared with the Navie Bayes algorithms. From all experiments the Naïve Bayes algorithms with all features have the lowest FP rates.

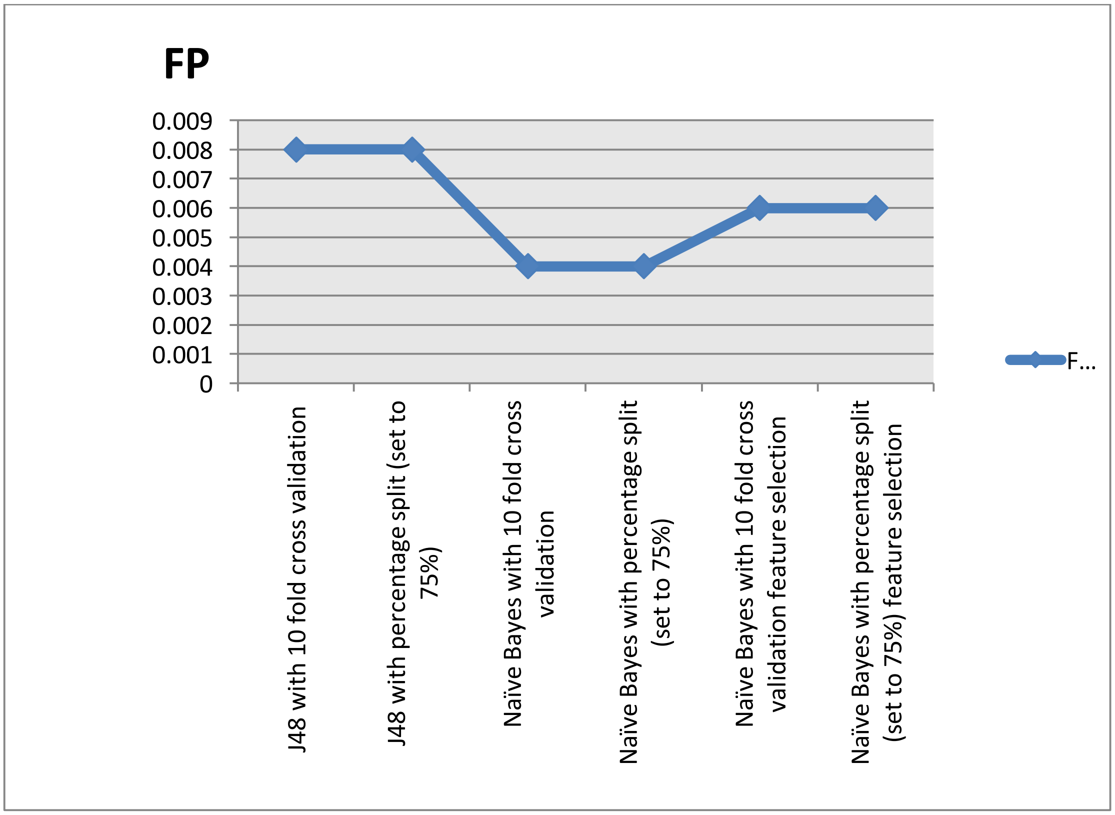

Figure 5: False Positive (FP) rate comparison of the J48 and Naïve Bayes Algorithms

The gap of the TP rate between the J48 decision tree and Naïve Bayes from figure 4.2 is 0.01. This means the TP rate of J48 decision tree is greater by 0.01. The gap of the FP rate between the J48 decision tree and Naïve Bayes algorithm from figure 5 is 0.004. This means the FP rate of Naïve Bayes algorithm is lower by 0.004. So, the greater TP rate of J48 decision tree will lead to more effectiveness than Naïve Bayes algorithm.

In summary, from figure 3 and 5, it is clear that J48 algorithm with 10-fold cross validation accuracy and the TP rate is better than with other algorithms. As a result, it is reasonable to conclude that the J48 algorithm is better than Naïve Bayes method for this study. Therefore, the model which is developed with the J48 decision tree with 10-fold cross validation classification techniques is considered as the selected working model for this thesis.

An Evaluation of the Discovered Knowledge

From all the experiments in this study, one model has achieved better classification performance. The J48 decision tree algorithm with the 10-fold cross validation model gives a better classification accuracy of predicting newly arriving intrusions in their respective class categories. Some of the rules generated from the selected model are the following:

Rule 1: If protocol_type=tcp and rerror_rate='(-inf-0.1]' and logged_in = '(-inf-0.5]' and flag = SF and Duration = '(-inf-0.5]' and num_failed_logins ='(-inf-0.5]' and src_bytes = '(3-6.5]' then normal (the intrusion is normal traffic)

Rule 2: Ifprotocol_type=udp andsame_srv_rate='(-inf-0.005]'and dst_host_same_src_port_rate = '(-inf-0.965]' then probe (the attack type is probe)

Rule 3: If protocol_type=icmp and service = telnet or http or private or domain_u or smtp or finger or ftp or pop_3 or X11 or ftp_data then DOS (the attack type is DOS)

Rule 4: If protocol_type=icmp and service = ecr_i and src_bytes ='(27.5-38.5]'and dst_host_count = '(-inf-5.5]' then normal (the intrusion is normal traffic)

Rule 5: If protocol_type=tcp and rerror_rate='(-inf-0.1]' and logged_in = '(-inf-0.5]' and flag = SF and Duration = '(-inf-0.5]' and Duration = '(-inf-0.5]' and num_failed_logins ='(0.5- inf)' and dst_host_diff_srv_rate = '(-inf-0.005]' then R2L (the attack type is R2L)

Rule 6: If protocol_type=udp and same_srv_rate = '(0.995-inf)'and service = telnet or http then R2L (the attack type is R2L)

Rule 7: If protocol_type=tcp and rerror_rate='(-inf-0.1]' and logged_in = '(-inf-0.5]' and flag = SF and Duration = '(-inf-0.5]' and num_failed_logins ='(-inf-0.5]'and dst_bytes = '(36.5-41.5]'then U2R (the attack type is U2R)

Rule 8: If protocol_type=icmp and service = eco_i and dst_host_srv_count = '(-inf-1.5]' then U2R (the attack type is U2R)

Rule 9: If protocol_type=tcp and rerror_rate='(-inf-0.1]' and logged_in = '(-inf-0.5]' and flag = SF and Duration = '(-inf-0.5]' and num_failed_logins = '(-inf-0.5]' and src_bytes = '(-inf- 0.5]' then U2R (the attack type is U2R)

Rule 10: If protocol_type=udp and same_srv_rate = '(0.995-inf)'and service = domain_u and dst_host_count = '(-inf-51.5]' then normal (the intrusion is normal traffic)

Rule 11: If protocol_type=tcp and rerror_rate='(-inf-0.1]' and logged_in = '(-inf-0.5]' and flag = SF and Duration = '(- inf-0.5]' and num_failed_logins ='(- inf-0.5]'and dst_bytes = '(34.5-35.5]'then U2R (the attack type is U2R)

Rule 12: If protocol_type=icmp and service = ecr_i and src_bytes = '(-inf-27.5]' then DOS (the attack type is DOS)

Rule 13: If protocol_type=tcp and rerror_rate='(-inf-0.1]' and logged_in = '(-inf-0.5]' and flag = SF and Duration = '(- inf-0.5]'and Duration = '(-inf-0.5]' and num_failed_logins ='(inf- 0.5]' and src_bytes ='(6.5-11.5]' and dst_bytes= '(-inf-16]' then normal (the intrusion is normal traffic)

Rule 14: If protocol_type=tcp and rerror_rate='(-inf-0.1]' and logged_in = '(-inf-0.5]' and flag = SF and Duration= '(-inf-0.5]' and Duration = '(-inf-0.5]' and num_failed_logins ='(-inf-

0.5]' and src_bytes = '(6.5-11.5]' and ' dst_bytes = '(16-34.5]'then R2L (the attack type is R2L)

Rule 15: If protocol_type=udp and same_srv_rate = '(0.005-0.19]' and count = '(-inf- 44.5]' then U2R (the attack type is U2R)

Rule 16: If protocol_type=tcp and rerror_rate='(-inf-0.1]' and logged_in = '(-inf-0.5]' and flag = SF and Duration = '(- inf-0.5]'and Duration = '(-inf-0.5]' and num_failed_logins ='(inf-0.5]' and dst_bytes ='(35.5-36.5]'then normal (the intrusion is normal traffic)

Rule 17: If protocol_type=icmp and service = eco_i and dst_host_srv_count='(14.5-57.5]' then normal (the intrusion is normal traffic)

The selected model for this study is the J48 decision tree algorithm with a default value which scores the highest classification accuracy of 96.11%. This model is tested with 3,397 testing dataset and scored a prediction accuracy of 93.2%. The selected model for this study is validated by real-life data. The real life data is with unlabeled classes. The prediction performance of this model is tested using a java code by using either Disk Operating System (DOS) or Simple Command line Interface (SCLI) on WEKA.

Conclusions and Recommendations

In summary, the results from this study can contribute to an improvement in the networking security. The study has shown that it is promising to identify those network intrusions, whether normal or attacks (DOS, U2R, Probe and R2L) and put forward tangible mechanisms to detect and prevent them, using the appropriate Data mining approaches. The result of the study has shown that the J48 decision tree algorithm with cross-validation test mode and other default values is appropriate in the area of intrusion detection. Hence, based on the findings of this study, the following are recommended as future research directions:

The Network Intrusion predictive model, which is developed in this study, generated various patterns and rules. To use this model effectively in the real world Network Security environment, designing a knowledgebase system which will add adaptability and extensibility features to the IDS and connect those to the DM model is one of the future research directions.

Constructing an IDS which will have both high intrusion detection (that is true positive) rate and low false alarm (that is false positive) rate is recommended.

To use the selected models there is a need to visualize the patterns, as visualization methods enhance network intrusion detection and anomaly detection. Information visualization techniques help network administrators and security analysts to quickly recognize patterns and anomalies; visually integrate heterogeneous data sources; and provide context for critical events. Information visualization and visual analytics hold great promise for making the information accessible, usable, and actionable by taking advantage of the human perceptual abilities. Visualization methods are also employed in the classification of network traffic and its analysis.

So, designing and integrating computer network visualization and visual analytics with the predictive intrusion detection model is one of the future research directions.

This study is conducted on the dataset taken from the MIT Lincoln lab. Future research should be conducted on real-life datasets from organizations that have their own network by combining the problem domain and the domain expert on the study process. This study was carried out using a clustering technique of simple K-means and classification algorithms such as J48 decision tree and Naïve Bayes algorithms. So, further investigation needs to be done using other classification algorithms such as Neural Networks and Support Vector Machine, in addition to the association rule discovery.

References

Adamu T. (2010), Computer Network Intrusion Detection: Machine Learning Approach, M.Sc Thesis, School of Information Science, Addis Ababa University, Addis Ababa, Ethiopia.

Adem K. and Julio P. (2009), Data Mining and Knowledge Discovery in Real Life Applications, In-Teh is Croatian branch of I-Tech Education and Publishing KG, Vienna.

Agrawal S., Mannila N., Srikant Y., Toivonen M. and Verkamo L (1996), Fast Discovery of Association Rules, In Advances in Knowledge Discovery and Data Mining, American Association for Artificial Intelligence Press

Alan B., Chandrika P., Rasheda S. & Boleslaw S. (2002), Network-Based Intrusion Detection Using Neural Networks, in Proceedings of the Intelligent Engineering Systems Through Artificial Neural Networks, St. Louis, Vol. 12, p. 579-584.

Amir A., Ahmad H. and Hadi B (2011)., A New System for Clustering and Classification of Intrusion Detection System Alerts Using SOM, International Journal of Computer Science and Security, Vol. 4, No. 6,.

Anand S., Patrick A., Hughes J., and Bell D (1998)., Data Mining Methodology for Cross- sales, Knowledge Based Systems Journal., Vol.10, PP.449–461

Anderson J. (1980), Computer Security Threat Monitoring and Surveillance, Technical Report, Washington,

Anirut S. and Nualsawat H. (2011). Euclidean-based Feature Selection for Network Intrusion Detection, International Conference on Machine Learning and Computing, IACSIT Press, Vol.3, Singapore.

Berson A., Smith S. and Thearling K. (2000), Building Data Mining Applications for CRM, McGraw-Hill Professional Publishing.

Blum A. and Rivest L (1992).Training 3-node neural networks is NP-complete, Neural Networks, Issue 5.

Brachman R. & Anand T. (1996). The process of knowledge discovery in databases, p. 37–57,

Bro V. (1998), System for detecting network intruders in real-time, In Proceedings of the 7thUSENIX Security Symposium, San Antonio, TX.

Cabena P., Hadjinian P., Stadler R., Verhees J., and Zanasi A. (1998). Discovering Data Mining: From Concepts to Implementation, Prentice Hall.

Chapman P., Clinton J., Kerber R., Khabaza T., Reinartz T., Shearer C., & Wirth R. (2003). CRISPDM 1.0 step-by-step data mining guide, Technical report, CRISP-DM.

Carl F. (2012). *Intrusion Detection and Prevention*. McGraw-Hill, Osborne Media.

Cisco System. (2012). Configuring Cisco IOS Firewall Intrusion Detection System, Cisco Security Guide, USA

Charles E. (2001). The foundations of cost-sensitive learning. In Proceedings of the Seventeenth International Joint Conference of Artificial Intelligence, Morgan Kaufmann, Seattle, Washington, p. 973–978.

Crothers M. (2002), Implementing Intrusion Detection Systems, a Hands-On Guide for Securing the Network, USA.

Chaudhuri S. (1998). Data Mining and Database Systems: Where is the Intersection? IEEE Bulletin of the Technical Committee on Data Engineering, Vol. 21(1), p. 4-8.

Cheng J., Greiner R., Kelly J., Bell D., & Liu W. (2002), Learning Bayesian networks from data: An information-theory based approach, Artificial Intelligence 137, PP. 43–90

Cheng S. (2000). Knowledge discovery in databases: an information retrieval perspective. *Malaysian Journal of Computer Science, 13*(2), p. 54-63.

Christos D. and Aikaterini M. (2004). DDoS attacks and defense mechanisms: classification and state

Of-the-art of Computer Networks. *The International Journal of Computer and Telecommunications Networking, Vol. 44*(5), p. 643 – 666.

Cios K. & Kurgan L. (2000). *Trends in Data Mining and Knowledge Discovery*. Springer-Verlag, London.

Dash M. & Liu H. (1997). Feature selection for classification: Intelligent Data Analysis. *An International Journal*, p. 131–156.

Denning D. (1987). An Intrusion Detection Model: Transactions on Software Engineering. *IEEE Communication Magazine*, SE-13, p. 222-232,

Dewan M. & Mohammed Z. (2010). Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm. *Journal of Computers, 5*, p. 23-31.

Doak J. (1992). An evaluation of feature selection methods and their application to computer security, Technical report, Davis CA: University of California, Department of Computer Science.

Dowell C. and Ramstedt P. (1990). The computer watch Data reduction Tool, Proc., 13th Natioanl Computer Security Conference, Washington, D.C., p.99-108.

Dunham H. (2003). *Data mining introductory and advanced topics*. Upper Saddle River, NJ: Pearson Education, Inc.

Eibe F. & Witten I. (2005). Data Mining–Practical Machine Learning Tools and Techniques (2nd ed). Elsevier

Eric B., Alan D. & Christiansen W. (2002). Data Mining for Network Intrusion Detection: How to Get Started? The MITRE Corporation.

Fayyad U., Piatetsky G., & Smyth P. (1996). The KDD process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM, 39*, p. 27-34.

Ferri C., Flach P. & Henrandez-Orallo J. (2002). Learning Decision Trees Using Area under the ROC Curve, Proceedings of the 19th International Conference on Machine Learning, Morgan Kaufmann, p. 139-146.

Frank J. (1994). Artificial intelligence and intrusion detection: Current and future directions, In Proc. of the 17th National Computer Security Conference, Baltimore, MD. National Institute of Standards and Technology (NIST)

Heady R., Luger G., Maccabe A., & Servilla M. (1990). The architecture of a network level intrusion detection system, Technical report, Computer Science Department, University of New Mexico.

Heberlein L., Dias G., Levitt K., Mukherjiee B., Wood J., & Wolber D. (1990), A Network Security Monitor, Proc., 1990 IEEE Symposium on Research in Security and Privacy, Oakland, CA, p.196-304.

Helali M. (2010). Data Mining Based Network Intrusion Detection System: A Survey in Novel Algorithms and Techniques in Telecommunications and Networking, p. 501-505.

Hershkop S., Apap F., Eli G., Tania D., Eskin E., & Stolfo S. (2007). A data mining approach to host based intrusion detection, technical reports, CUCS Technical Report.

Lee W. , Stolfo S., and Mok K. (1999), A Data Mining Framework for Building Intrusion Detection Model, In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, p. 120-132.

Liu H. and Motoda H. (1998). Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic Publishers.

Lunt T. (1990). IDES: A progress Report, Proc. 6th Annual Computer Security Applications Conference, Tucson, AZ.

Mahbod T., Ebrahim B., Wei L., & Ali A. (2009). A detailed analysis of the KDD CPU 99 Dataset, proceedings of 2009 of the IEEE Symposium on computational Intelligence in Security and Defense Applications, National Research Council, p.1-6.

Manago K., & Auriol S. (1996), Mining for OR.ORMS Today (Special Issue on Data Mining), American Association for Artificial Intelligence Press, p. 28-32.

Marin J., Ragsdale D. & Surdu J. (2001). A hybrid approach to profile creation and intrusion detection, In Proc. of DARPA Information Survivability Conference and Exposition, Anaheim, CA. IEEE Computer Society.

Meera G., Gandhi & Srivatsa S. (2010). Adaptive Machine Learning Algorithm (AMLA) Using J48 Classifier for an NIDS Environment. *Advances in Computational Sciences and Technology, 3*, p. 291–304.

Mukherjee B., Todd, L., & Karl N. (1994). Network Intrusion Detection, *IEEE, 12*(4), p. 132-143.

Mukkamala H. & Sung A. (2003), Comparative study of techniques for intrusion detection. Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03).

Nanda A (2010), Data Mining and Knowledge Discovery in Database: An AI perspective, Proceedings of national Seminar on Future Trends in Data Mining.

Pachghare V., Vaibhav K., & Parag K. (2011). Performance Analysis of Supervised Intrusion Detection System, IJCA Special Issue on Network Security and Cryptography, NSC

Pradeep S (2005). Comparing the Effectiveness of Machine Learning Algorithms for Defect Prediction. *International Journal of Information Technology and Knowledge Management, 2*(2), p.481-483.

Sterry B. (2004). Data Mining Methods for Network Intrusion Detection. University of California

Thair P. (2009). Survey of Classification Techniques in Data Mining. Proceedings of the International Multi-Conference of Engineers and Computer Scientists, Hong Kong.

William W. (1995). Fast effective rule induction. In Internal conference on Machine learning, *IEEE*, p. 115-123.

Yeung D. & Ding Y. (2003). Host-Based Intrusion Detection Using Dynamic and Static Behavioral Models. *Journal of Pattern Recognition, 36*, p. 229-243.

Zewdie M. (2011). Optimal feature selection for Network Intrusion Detection, a Data Mining Approach. M.Sc thesis. School of Information Science: Addis Ababa University.