An Efficient Intrusion Detection Model Using Ensemble Methods

Debachudamani Prusti

Roll. No. 213CS2167

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

An Efficient Intrusion Detection Model Using Ensemble Methods

Dissertation submitted in

May 2015

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

by

Debachudamani Prusti

(Roll No. 213CS2167)

under the supervision of

Prof. Sanjay Kumar Jena

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

May 30, 2015

Certificate

This is to certify that the work in the thesis entitled ***An Efficient Intrusion Detection Model Using Ensemble Methods*** by ***Debachudamani Prusti***, bearing roll number **213CS2167**, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering Department*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Sanjay Kumar Jena
Department of Computer Science

Declaration By Student

I certify that, I have complied with all the benchmark and criteria set by NIT Rourkela Ethical code of conduct. The work done in this project is carried out by me under the supervision of my mentor. This project has not been submitted to any other institute other than NIT Rourkela. I have given due credit and references for any figure, data, table which was being used to carry out this project.

Signature of the Student

Place: Rourkela Date: 30/05/2015

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Sanjay Kumar Jena, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Character Recognition and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I thank our H.O.D. Prof. Santanu Kumar Rath, for his constant support in my thesis work. He has been great sources of inspiration to me and I thank him from the bottom of my heart.

I would also like to thank all faculty members, PhD scholars, my seniors and juniors and all colleagues to provide me their regular suggestions and encouragements during the whole work.

At last but not the least I am in debt to my family to support me regularly during my hard times.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

Debachudamani Prusti

Abstract

Ensemble method or any combination model train multiple learners to solve the classification or regression problems, not by simply ordinary learning approaches that can able to construct one learner from training data rather construct a set of learners and combine them. Boosting algorithm is one of the most important recent developments in the area of classification methodology. Boosting belongs to a family of algorithms that has the capability to convert a group of weak learners to strong learners. Boosting works in a sequential manner by adding a classification algorithm to the next updated weight of the training samples by doing the majority voting technique of the sequence of classifiers. The boosting method combines the weak models to produce a powerful one and reduces the bias of the combined model. AdaBoost algorithm is the most influential algorithm that efficiently combines the weak learners to generate a strong classifier that could be able to classify a training data with better accuracy. AdaBoost differs from the current existing boosting methods in detection accuracy, error cost minimization, computational time and detection rate. Detection accuracy and computational cost are the two main metrics used to analyze the performance of AdaBoost classification algorithm. From the simulation result, it is evident that AdaBoost algorithm could able to achieve high detection accuracy with less computational time, and minimum cost compared to a single classifier. We have proposed a predictive model to classify normal class and attack class and an online inference engine is being imposed, either to allow or deny access to a network.

Keywords: Ensemble Methods, Boosting, Weak learners, Strong Learners, AdaBoost

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The internet plays an increasingly important role in worldwide with a large betterment in E-commerce digital government, social networking, etc. But now internet is unsecured because of threats, criminal activities, cyber-attacks and have started devising and launching a pretty sophisticated attacks motivated with some destructive objectives. We must be sure with the security, i.e., confidentiality, integrity, and availability, of our network infrastructures and devices [1] [2]. Intrusion detection is the process to identify and respond to the malicious activity intended at compromising computer and network security [1]. It is a critical and sensitive component of the defence-in-depth security mechanisms, which includes: security policy, vulnerability, patching and scanning , access control and authentication, encryption, firewalls, program wrappers and intrusion tolerance.

Protecting critical resources in the internet must need security to prevent the damages from the unseen class of attacks. Intrusion detection systems require to adapt efficiently and accurately to exploit new knowledge of previously unseen classes of attacks that likely to be constantly invented [3]. This task can be resolved either by doing the hand coding new attack patterns and inserting it into existing models that are then broadly distributed by some automatic means of learning about new classes of attacks or incorporated into existing deployed models. The key benefits of intrusion detections are the efficiency of both training and deployment using

inference engine. Intrusion prevention systems and detection systems are installed in our networks, will always be attempting to develop and launch the new attacks. Once the knowledge about new attacks are collected through the detection method, they need to be quickly incorporated into existing detection systems to prevent any further damage of the new attacks as quickly as possible. But, re-training a model for both existing and new attacks is usually very slow due to the learning complexity and large size of the dataset. By the mean time a new detection model is ready, the new type of attack may have already caused significant damages.

1.1 Intrusion Detection System (IDS)

Intrusion is an illegal access to the concealed resources or banned domains. It is a way the attackers enter into a network or to a confidential property forcefully without having an authorized permission. Intrusions are the suspicious or malicious components that are found in the network or in a computer system. This is the act of intruding to the others property beyond the legal limit. In the background, the intruder tries to know the vulnerability before attacking in the security system.

To detect the unwanted actions that compromise the security components such as confidentiality, integrity and availability. The massive growth in technology over the internet creates a critical issue in computer security. Various Machine learning algorithms, data mining, and artificial intelligence are subjected to modern and advanced research in intrusion detection with an intention to improve the accuracy of detection. There are two types of detection techniques such as static detection technique (offline) and dynamic detection technique (online). These are the methods used to detect various suspected objects in the network instantly. The dynamic detection technique is very much efficient, reliable and effective in comparison to static technique.

It is a software application or a physical device that monitors the malicious activities in the network or system and also inspects the incoming, and outgoing

network traffics and suspicious patterns of attacks. It is an immune model to detect the attacks with more accuracy and extracts the important features. IDS mainly two types such as Network-based Intrusion Detection System (NIDS) and Host Based Intrusion Detection System (HIDS)

1.2 Types of IDS

The Network Intrusion Detection Systems (NIDs) are located at a specific point or points inside the network to control the traffic of all the systems in the network. It attempts to detect the malicious activities such as denial of service attacks, port scans or tries to crack into computers by monitoring network traffic. An NIDS scans all the incoming packets and tries to find the suspicious patterns known as signatures or rules. For example, if a very large number of TCP connection requests to a large number of different ports are observed, one might assume that there is someone conducting a port scan of some or all of the computer(s) in the communication network. It also (mostly) tries to detect incoming shellcodes in the network in the same manner that an ordinary intrusion detection system does.

Host Intrusion Detection Systems (HIDs) run on single systems or devices in the network. It monitors incoming and outgoing packets or data from the system or device and alerts the administrator if any suspicious activity identified.

All the types of IDS use either signature-based detection technique or anomaly-based detection technique. Signature-based detection technique compares the incoming packets against the database of signatures or attributes from the known suspicious threats. It Checks every request for access to the network against a set of existing attack in to detect possible attacks and trigger an appropriate response. It is by Rule-Based IDS, Knowledge-Based IDS.

It checks each and every request for accessing against patterns of network traffic to detect possible deviations from the normal behavior (anomaly) which may indicate an attack and trigger the appropriate responses by Anomaly-Based IDS.

For Example, a company uses a public network for its applications and it is exposed to security threats, especially a variety of unknown attacks, their business could be in jeopardy if their customers realize the fact that their system is not secure enough.

1.3 Learning Techniques

The process of generating models from data is called learning or training. There are different learning techniques and two standard learning techniques are supervised learning and unsupervised learning. In supervised learning, the goal is to predict the target feature on unseen instances. Unsupervised learning does not rely on label information, and its goal is to discover some inherent distribution information in data.

In other aspects, the classifiers are single classifier or multiple classifiers. The multiple classifiers are also known as ensemble of classifiers. While designing the model, feature selection in learning process leads to a reduction of computational cost, model size, overfitting and accuracy.

In the single classifier system, the classifier sometimes treated as weak classifier and cannot classify the objects with more accuracy. The detection accuracy is very close to the random guess. But in multiple classifiers, a set of weak classifiers are combined to form a strong classifier, and that can able to classify with a high detection rate [4].

Several existing learning algorithms are there for the classification of the instances or samples in a dataset. Some efficient algorithms are linear discriminant analysis, decision trees, neural networks, Navie Bayes classifier, k-nearest neighbor and support vector machine [5].

1.4 Ensemble Classification Methods

Ensemble method is nothing but the model combination technique where a number of weak learners are grouped iteratively to yield a better learner that can classify the training samples efficiently. It can able to generate a good performable predictive model where classification accuracy is very close to the exact or correct value [6].

Basically, an ensemble classification technique is a supervised learning algorithm as it can train itself and make the prediction accuracy. For a particular problem, it can able to find the suitable hypothesis to make a good prediction. The main idea behind the ensemble technique is to use the same base learner to generate multiple hypotheses [7].

1.5 Motivation

The detection accuracy of a single classifier is not good always. To enhance the detection rate and reduce training and generalization error, sophisticated techniques should be adopted which motivate us to use ensemble methods for detect novel attacks.

1.6 Objective

The objective is to achieve a high detection rate of the network traffics and less false positive with minimum cost. The aim is to design an ensemble classifier for the intrusion detection model that can efficiently combines the weak learners to form a strong learner. The strong learner can classify the online network traffics by extracting the desired features since statistical features are used to identify and classify the network traffics.

From some empirical studies, it is shown that a given algorithm may outperform all others for a particular subset of problems, but there is no single algorithm that achieves best accuracy for all situations. Therefore, there is growing research interest

in combining a set of learning algorithms into one system, which is the so-called ensemble method. The ensemble of classifiers improves the scalability to classify the both existing and new classes of intrusions. We proposed a predictive model to classify the attack class and normal class after doing the model training. Both online and offline phases are considered to train the model for unidentified traffics in the network and to know their behavior.

Chapter 2

Literature Review

2.1 Introduction

The philosophy behind the ensemble classifier is that one base classifier compensates the error made by another classifier. But simply training the base classifier may not solve the desired problem as the base classifiers are uncorrelated. Base classifiers are the individual classifiers mainly used to construct the ensemble of classifiers. We can consider various weak learners such as support vector machine, neural network, and k-nearest neighbor to construct the ensemble classifier. The base learners are basically generated sequentially such as boosting and in parallel such as bagging.

In boosting the weak learners are iteratively added to form a strong learner to yield a good result in prediction accuracy. Boosting algorithm is originated from a theoretical question asked by Kearns and Valiant [1989], i.e. whether the two complexity classes, weakly learnable and strongly learnable problems are equal. This question is of some fundamental importance because if the answer is positive, any weak learner is potentially able to be boosted to a strong learner. But in real practice it is generally easy to obtain weak learners but difficult to find strong learners. Schapire [1990] proved that the answer is positive, and that led to the construction of boosting algorithm [8].

Bagging is abbreviated term for Bootstrap AGGregatING [7]. Bagging has two components such as bootstrap and aggregating. Bagging adopts the bootstrap distribution for generating different base learners. It applies bootstrap sampling to obtain the data subsets for training the base learners. It also adopts the most popular strategies to aggregate the outputs of base learners, that is, voting for classification and averaging for regression. Bagging feeds the instances to its base classifiers and collects all of their outputs and then votes the label to predict the winner label. Bagging deals with both binary class and multi-class classification.

2.2 Ensemble Model Formation

An enormous amount of work has been done by researchers in the area of ensemble methods. Ensemble techniques are conventionally used in very early age, but nowadays in the field machine learning, it has a very good impact on the classification techniques with good prediction accuracy. Ensemble methods have two components, according to its operation such as empirical study and theoretical study.

In the empirical study [Hansen and Salamon, 1990], the predictions made by the combinations of a set of classifiers are often more accurate than the prediction made by a single best classifier. But in the theoretical study [schapire, 1990], it is proved that a set of weak learners can be boosted to strong learners. Although strong learners are desirable, but it is difficult to get, while weak learners are easy to obtain. This led to the generation of strong learners by ensemble methods. Ensemble methods have two fundamental steps, that is, generating the base learners and combining them. Rather than getting good prediction accuracy, also the computational cost is not much larger than generating a single classifier.

Freund (1995) proposed a model Boost by Majority which combines many weak learners simultaneously and improves the performance of the simple boosting procedure.

Table 2.1: DataSet used for Ensemble Methods

Sl. No	Author	Ensemble Approach	DataSet used	Year
1	*Freund et al*	*AdaBoost algorithm*	*Business analytic dataset*	1997,2001
2	*Friedman et al*	*LogitBoost algorithm*	*KDD Cup*	2009
3	*Zhu et al*	*AdaBoost algoritm*	*NSLKDD*	1999,2009
4	*Leo Breiman*	*Bagging algorithm*	*CAIDA*	2007

According to Friedman et al, (2000) adaboost is just an optimization process which tries to fit an additive model and developed logitboost algorithm which is a log loss function.

Leo Breiman (1996) proposed the bootstrap aggregating or bagging algorithm where the combination of independent base learners leads to the decrease of errors. Bagging is also known as parallel ensemble process.

2.3 Model Combination

By combining the models ensemble methods achieves a strong generalization ability. Rather than finding the best single classifier ensemble method combines the set of base learners and the benefits of combination have three fundamental reasons such as statistical issue, computational issue, and representational issue.

Statistical issue is often the case that the hypothesis space is too large to explore for limited training data, and there may be several different hypotheses giving the same accuracy on the training data. If the learning algorithm chooses one of these hypotheses, there is a risk that a mistakenly chosen hypotheses could not predict the future data well. By combining the hypotheses, the risk of choosing the wrong hypothesis can be reduced.

For computational issue, even if there are enough training data, it may still be difficult to find the best hypothesis. But by running the local search from many different starting points, the combination may provide a better approximation to

the true unknown hypothesis.

For representational issue, the true unknown hypothesis could not be represented by any hypothesis in the hypothesis space. By combining the hypotheses, it may be possible to expand the space of representable functions.

2.3.1 Averaging

It is a most popular and fundamental combination method for numeric outputs. It is two types such as simple averaging and weighted averaging.

In simple averaging the combined output is obtained by averaging the outputs of individual learners directly. The simple averaging gives the combined output H(x) as

$$H(x) = \frac{1}{T}\sum_{i=1}^{T} h_i(x) \tag{2.1}$$

In weighted averaging the combined output is obtained by averaging the output of individual learners with different weights. The weighted averaging gives the combined output H(x) as

$$H(x) = \sum_{i=1}^{T} w_i \times h_i(x) \tag{2.2}$$

2.3.2 Voting

It is a combination method for nominal outputs. Suppose we have given a T individual classifiers (h1....hT) and we have to combine the classifiers to predict the class label from a set of l possible class labels (c1....cl).

Majority voting is the most popular voting method where every classifier votes for one class label and the final output class label is the one that receives more than half of the votes. If none of the class labels receives more than half of the votes, a rejection option will be given, and the combined classifier makes no prediction.

Plurality voting takes the class label that receives the largest number of votes as the final winner. Plurality voting does not have a rejection option since it can always find a label receiving the largest number of votes. In case of binary classification, plurality voting coincides with majority voting.

By weighted voting, individual classifiers providing unequal performance are ensemble to give a strong learner. In practical applications, the weights are normalized for all the learners.

Soft voting is opted when the individual classifiers produce class probability outputs. If all the individual classifiers are treated equally, the simple soft voting method generates the combined outputs by simply averaging the individual outputs.

Chapter 3

Ensemble Methods

3.1 Introduction to Ensemble classification algorithms

Ensemble methods or ensemble learning train multiple learners instead of a single learner and combine the result of different learners to yield a better result than individual weak learners. Hence, it is also called multiple classifier system [6]. Ensemble methods always combines multiple hypotheses to form a better hypothesis. In other words, an ensemble is a technique to combine a large number of weak learners in an attempt to produce a strong learner. The term ensemble is usually reserved for methods that create multiple hypotheses by using the same base learner. The broader term of explicit multiple classifier systems also covers hybridization of hypotheses that are not induced by the same base learner.

Fig. 3.1 Shows a common ensemble architecture where n number of weak learners are combined to form a strong learner. The weak learners are also called the base learners which are usually generated from base learning algorithms that can be decision tree, neural network or any kind of learning algorithms. The major goal of ensemble methods is to combine the prediction of several models that is built with a learning algorithm to improve the generalizability or robustness over a single model.

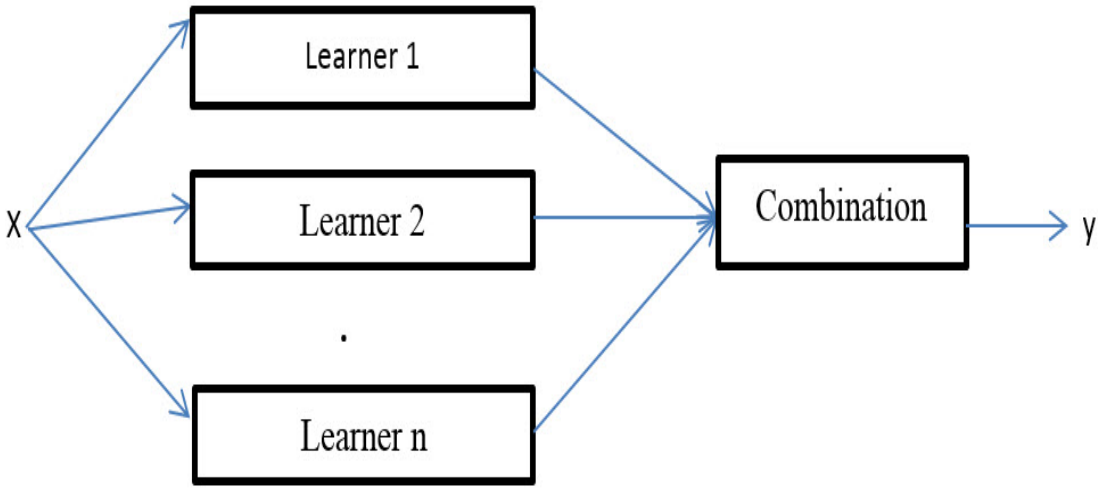

Figure 3.1: Simple ensemble architecture

Basically, there are two types of ensemble learners: homogeneous learners and heterogeneous learners. Ensemble methods use a single base learning algorithm produces homogeneous learners, i.e., learners of the same type, leading to homogeneous ensembles [8]. If learners are of different types that lead to heterogeneous ensembles, and use multiple learning algorithms. The generalization ability of an ensemble is often much stronger than that of base learners. Actually, ensemble methods are able to draw the interest mainly because they can boost weak learners to strong learners. Weak learners are even just slightly better than a random prediction but, strong learners can make very accurate predictions.

There are three threads of early contributions that led to the current area of ensemble methods; i.e., combining classifiers, ensembles of weak learners and mixture of experts.

- Combining classifiers was usually studied in the pattern recognition community. In this thread, researchers generally work on strong classifiers, and try to design powerful combining rules to get stronger combined classifiers [2]. As a result, this thread of work has accumulated with deep understanding

on the design and use of many different combining rules.

- Ensemble process of the weak learners was mostly studied in the machine learning community. In this thread, researchers often work on weak learners and try to design efficient and powerful algorithms to boost the performance from a weak zone to strong zone. This thread of work has led to the birth of famous ensemble methods such as boosting, bagging, etc.

- Mixture of experts was usually studied in the neural networks community. In this method, researchers generally suppose to consider a divide-and-conquer strategy, try to learn a mixture of parametric models jointly and use combining rules to get an overall solution.

Ensemble methods have become a measure learning paradigm since the 1990s, with great promotion by two pieces of pioneering work such as empirical and theoretical work.

- In empirical work, it was found that predictions made by the combinations of a set of classifiers are often more accurate than predictions made by the best single classifier.

- In theoretical work, it was proved that weak learners can be boosted to strong learners. Since strong learners are desirable but difficult to get, while weak learners are easy to obtain in real practice, this consequence forwards an actual direction of generating strong learners by ensemble methods.

Generally, an ensemble is constructed in two steps; i.e. generating the base learners and then combining them. To get a good ensemble the base learners should be as accurate as possible.

In general, the computational cost of constructing an ensemble is not much larger than creating a single learner [5]. This because when we want to use a single learner, we usually need to generate multiple versions of the learners for model selection or

parameter tuning; which is expensive. But the computational cost for combining base learners is small since most combination strategies are simple.

Application of ensemble methods

1. For detection, recognition and tracking of objects.

    - A general object detection framework was done by Viola and Johns [2001, 2004] by combining AdaBoost with a cascade architecture. They reported that the face detector spent only .067 seconds for a 384X288 size image, whose detection accuracy is 15 times faster than the normal face detector.
    - A general object detection framework was done by Viola and Johns [2001, 2004] by combining AdaBoost with a cascade architecture. They reported that the face detector spent only .067 seconds for a 384X288 size image, whose detection accuracy is 15 times faster than the normal face detector.
    - Object tracking aims to assign consistent labels to the target objects in consecutive frames of a video. Ensemble tracking trains an ensemble online to distinguish between the object and the background. The ensemble tracking framework can work in a large variety of videos with various object size and runs very efficiently without any optimization, hence can be used in online applications.

2. To diverse real-world task Ensemble methods have been successfully applied to different real-world task since they have been found useful in almost all places, where learning techniques are exploited.

3. Characterize computer security and problem. It is appropriate to characterize computer security problems because each activity performed on computer systems can be observed at multiple abstraction levels, and the relevant information may be collected from multiple information sources.

4. Network intrusion detection

- Giacinto [2003] reported that when detecting known attacks ensemble methods leads to the best performance. Later in 2008, he again proposed an ensemble method for anomaly-based intrusion detection that can detect intrusions never seen before.
- Schultz [2001] proposed an ensemble method to detect previously unseen malicious executable codes automatically. This method also able to detect virus, worms and Trojan horses automatically.

Apart from this ensemble methods have also been applied to many other domains and tasks such as credit card fraud detection, bankruptcy prediction, weather forecasting, aircraft engine fault diagnosis, etc.

In another aspect, ensemble methods are of two types such as averaging method and boosting method [9].

- Averaging method builds several models independently and then takes the takes the average prediction to select the best one. Examples of these methods are bagging, random forest, etc.
- In boosting method, the second model depends on the first one. Models are built sequentially. Its primary goal is to combine weak model to produce a powerful one, and it reduces the bias of combined model. Examples of these methods are AdaBoost, LogitBoost, etc.

3.2 Bagging

It is also known as bootstrap aggregating. It is nothing but the parallel ensemble process, where base learners are generated in parallel [10]. It maintains the independence between the base learners and combines them, by which errors can be reduced. Bagging can also accelerate the training speed using parallel computers. Another benefit of bagging is that they are inherently favorable to parallel computing as multi-core processors are commonly available nowadays. Bagging adopts the

bootstrap distribution for generating different base learners. In other words, it applies bootstrap sampling to obtain the data subsets for training the base learners. It also adopts the most popular strategies for aggregating the outputs of base learners, i.e. voting for classification and averaging for regression. To predict a test instance, taking classification for example, bagging feeds the instance to its base classifiers and collects their outputs, and then votes the labels and take winner label as prediction, where ties are broken arbitrarily.

3.3 Boosting Algorithm

Boosting algorithm or simply boosting is one of most important recent developments in classification methodology. It is similar in overall structure to bagging, with a exception that one keeps track of the performance of the learning algorithm and forces it to concentrate all of its efforts on instances that have not been correctly learned or trained [10]. Instead of choosing the t number of training instances randomly using a uniform distribution, one may choose the training instances in such a manner that, as to favour the instances that have not been accurately learned. After several number of cycles, the prediction for it is performed by taking the weighted vote of the predictions of each individual classifier, with the weights being proportional to each classifier's accuracy on its training set [9].

Boosting algorithms may be considered stronger than bagging on noise-free data. However, there are some strong empirical indications that bagging algorithm is much more robust than boosting in the case of noisy settings. For this reason, Kotsiantis and Pintelas (2004) built an ensemble model using a voting methodology of bagging and boosting ensembles that give better classification accuracy as comparison to other. Boosting involves incrementally building an ensemble iteratively by training each new model instances to emphasize the training instances that previous models misclassified. In some cases, boosting has been shown to yield better accuracy than bagging, but it also tends to be more likely to over-fit the training instances.

By definition, boosting belongs to a family of algorithms that are able to convert the weak learners to strong learners [1]. A weak learner is just slightly better than random guess, while a strong learner is very close to perfect performance. In boosting algorithm, models are built in a sequential manner by exploiting a classification algorithm to the reweighted versions of the training instances. It combines the performance of many weak learners to produce a powerful strong learner. Boosting was proposed by Schapire in 1990 and Freund 1995.

3.4 General Boosting Procedure

The general boosting procedure is simple technique, where weak learners work on any given data distribution process and take the binary classification task to classify the training instances as positive and negative instances. The training instances in the space X are drawn from distribution D and the ground truth function is f. X is composed of three parts x1, x2, x3 and each takes the 1/3rd amount of distribution. Let we have a weak learner that works by random guess with 50% classification error. If we want to get an accurate classifier (zero error) on the problem and let we have a weak classifier which can classify correctly x1 and x2 but not x3, then there will be 33.33% classification error. Hence the weak classifier (say h1) is not desired.

The idea of boosting is to correct the mistake made by h1. For this we have to derive a new distribution D' from D and we have to give more focus on x3. Now we can train a weak classifier h2 from D'. Let h2 can correctly classify x1 and x3 and wrongly classify x2. By combining both h1 and h2, there will be correct classifications in x1 and might be some errors in x2 and x3 but the error is less. Again, by deriving a new distribution D'' and training a classifier h3 which can correctly classify x2 and x3 and wrongly classify to x1. By combining all the classifiers we can get a perfect classifier that can correctly classify the instances. In brief, the idea of boosting is to train a set of learners sequentially and to rectify the mistakes made by the weak classifiers [10].

Chapter 4

Performance Evaluation of Ensemble Methods

4.1 Introduction

To evaluate the performance of existing ensemble methods, we have implemented the following well known methods.

Adaboost Algorithm LogitBoost Algorithm Bagging Algorithm

4.2 Ensemble Methods

Various ensemble methods are proposed by many researchers for the classification techniques and developed the algorithms to achieve a strong generalization ability.

4.2.1 Adaboost Algorithm

AdaBoost is one of the efficient boosting algorithms that can able to classify the objects or instances with a strong learner which is generated by the aggregation of weak learners. AdaBoost is called adaptive boosting because it uses multiple iterations to generate a single composite strong learner. AdaBoost generates the

strong learners iteratively by adding weak learners in asequential manner [11]. In each round of training, a new weak learner is added to the ensemble process and a weight vector is adjusted to focus on the misclassified instances in previous rounds. AdaBoost algorithm is the adaptive boosting algorithm in which instances can be either classified either by correctly or incorrectly. If the instances are classified correctly the weight of these instances are reduced drastically such that the instances will not be considered in next round iteration of training process. But, in case If the instances are incorrectly classified there are two possibilities such as normal traffic classified as attack or attack traffic classified as normal. And the objective is to reduce the false positive and to increase the detection accuracy.

The general boosting procedure is not a real algorithm since because of some unspecified components. The components like adjust distribution and combine outputs are not specified properly. AdaBoost is the most influential and real representation of boosting algorithm.

AdaBoost is the predictive algorithm used for both classification and regression. Classification is used to analyze the entire input data and to develop an accurate description on the model for each class using the features present in the data, whereas regression estimates the relationships among the variables [7]. It is a prediction method based on an assumed or known numerical output data. The primary difference between classification and regression is that in classification the dependent variables are categorical and in regression the dependent variables are numerical.

The AdaBoost algorithm generates a set of hypotheses, and they are combined with weighted majority voting technique of the classes predicted by the each individual hypotheses. To generate the above said hypotheses by training a weak classifier, instances are drawn from an iteratively updated distribution of the training instances are used. This distribution is updated in such a manner, so that instances misclassified by the previous hypothesis are more likely to be included in the training data of the next classifier. Consequently, the consecutive hypotheses training data

are organized toward increasingly hard-to-classify instances. It was proven that a weak learner can algorithm which generates classifiers that can merely do better than random guessing can be turned into a strong learner using boosting.

To evaluate the performance of four parameters are taken into account. These are detection accuracy, computation time, cost minimization and detection rate [6].

- Detection accuracy: To achieve high detection accuracy with less false positive. When an intrusion detection system (IDS) identifies a normal traffic as an attack and gives the alarm, then that case is called false positive.
- Computational time: The computational complexity should be less and take less time to classify.
- Cost minimization: Single classifier may not classify the training sample correctly and error cost increases. But multi-class classifier reduces the cost due incorrect classification. Error cost reduces during the training and also during validation (post-training). In a classification system, using ensemble of classifiers, each class uses the classification error cost minimization technique in the intrusion detection problems.
- Detection rate: It is defined as the number of intrusion instances detected by the system divided by the total number of instances present in the test set. High detection rate is always desired to identify an intrusion in real time.

The accuracy result for the AdaBoost algorithm is shown in the confusion matrix. Here in the dataset 38645 instances are there. And using this algorithm it could able to classify 37658 instances correctly and the prediction accuracy is 97.4%. In this, we got the misclassification error is 2.6%.

4.2.2 Bagging Algorithm

Bagging uses the bootstrapping to generate the training sets. It trains the base learners using an unstable learning process, but during the testing it takes the

Input:

for *each Dataset Di, (Where i = 1 and 2 that represents normal class and attack class)* **do**

Number of training data of size '$N' = TD_i$

Adaptive Boosting learning algorithm as supervised base classifier

Number of iterations/ learning rounds/classifiers = T

Number of classes $L = \alpha_1, \alpha_2$

end

Initialization:

$\mu = 0.5$: Threshold value for false alarm

$L = 2$: Number of classes (Normal and Attack)

$dt(e) = 1/n$: Weighted sum of the instances in the data set

Training Process:

for $i = 1 \ldots L$ **do**

Select training samples or instances from class i, and from dataset Di

Split dataset Di into j subsets. (s1, s2 . . . sj)

for *Do for each k = 1, 2 . . . j: Number of subsets* **do**

for *Do for t = 1 . . . T: Number of classifiers* **do**

Train sk by AdaBoost and obtain hypothesis ht

Compute the error of ht : hypothesis ht

$\epsilon = \sum_{e=1}^{n} \frac{[h_t(x) \neq f(x)]}{n}$

If $t >$, Then drop the hypothesis and go to step B

Else if $t = 0$, the weak classifier is not weak and run with a weak classifier

Else add the classifier ht to the ensemble 'Ek'.

Compute normalized error

$t = t/(1-t)$ Where 0¡t¡1

Assign Weight to classifier ht

$t = ln((1)/t)$

Update Weight distribution

$d_{t+1}(e) = \frac{d_t(e)}{Z_t} \times \begin{cases} e^{-wt} & if\ h_t(x) = f(x) \\ e^{wt} & if\ h_t(x) \neq f(x) \end{cases}$ Where Zt is normalization factor which enables dt+1(e) to be a distribution.

So that $_(e = 1)^n dt(e) = 1$

end

end

The additive weighted combination of Weak learners:

$H(x) = _(t = 1)^T tht(x)$

end

Algorithm 1: Adaboost algorithm

Table 4.1: Confusion Matrix for Adaboost

Adaboost	PC	1	-1
AC	1	23523	554
	-1	433	14135

average. Generally Bagging adopts the bootstrap distribution to generate different base learners. It applies bootstrap sampling to find the data subsets to train the base learners. It adopts the most popular strategies for aggregating the outputs of base learners, i.e. voting for classification and averaging for regression. To predict a test instance, taking classification for example, bagging feeds the instance to its base classifiers and collects their outputs, and then votes the labels and take winner label as prediction, where ties are broken arbitrarily.

Input: Dataset D ={$(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$};
Base Learning algorithm ε ;
Number of learning rounds T;
Process:
$y_0(x)$ = f(x) : initialize target
$H_0(x) = 0$: initialize function
for *t=1.....T* **do**
| $h_t = \varepsilon(D, d_{bs})$: d_{bs} *is the bootstrap distribution*
end
output :$H(x) = argmax(\sum_{t=1}^{T} I(h_t(x) = y))$

Algorithm 2: Bagging algorithm

Table 4.2: Confusion Matrix for Bagging

Bagging	PC	1	-1
AC	1	22856	1
	-1	1100	14688

Result

The accuracy result for the Bagging algorithm is shown in the confusion matrix. Here in the dataset 38645 instances are there. And using this algorithm it could

able to classify 37544 instances correctly and the prediction accuracy is 97.2%. In this, we got the misclassification error is 2.8%.

4.2.3 LogitBoost Algorithm

In logitboost algorithm, the weight update is done without using the logarithmic function. Adaboost algorithm uses a log function for the weight update, and it is an additive logistic regression model, in which the estimation procedure is stage wise. In the logitboost algorithm, the model is used by optimizing the log loss function.

Input: Dataset D $=\{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$;
Least Square Base Learning algorithm ε ;Number of learning rounds T;
Process:
$y_0(x)$ = f(x) : initialize target
$H_0(x) = 0$: initialize function
for *t=1.....T* **do**
  $p_t(x) = \frac{1}{1+e^{-2H_t-1(x)}}$: Calculate probability
  $y_t(x) = \frac{y_t-1(x)-p_t(x)}{p_t(x)\times(1-p_t(x))}$: Update target
  $d_t(x) = p_t(x)(1-p_t(x))$: Update weight
  $h_t = \varepsilon(D, y_t, d_t)$: Train a classifier h_t to fit y_t in data set D under distribution
  d_t$H_t(x) = H_t - 1(x) + \frac{1}{2}H_t(x)$: update combine classifier
end
output :$H(x) = sign(\sum_{t=1}^{T} h_t(x))$

Algorithm 3: LogitBoost algorithm

Table 4.3: Confusion Matrix for Logitboost

Logitboost	PC	1	-1
AC	1	23307	371
	-1	649	14318

Result

The accuracy result for the logitboost algorithm is shown in the confusion matrix. Here in the dataset 38645 instances are there. And using this algorithm it could

able to classify 37658 instances correctly and the prediction accuracy is 97.4%. In this, we got the misclassification error is 2.6%.

The Table 4.4 showing that the AdaBoost ensemble method gives better result in comparison of LogitBoost and Bagging algorithm. Hence, AdaBoost algorithm is considered as an ensemble framework for our proposed model.

Table 4.4: Comparison of Ensemble Methods

Algorithms	TPR	TNR	FNR	FPR	Accuracy	PPV	NPV	MCC	F1 Score
AdaBoost	0.977	0.97	0.023	0.029	**97.44%**	0.981	0.962	0.945	0.979
LogitBoost	1	0.93	0.023	0.069	97.15%	0.954	0.999	0.942	0.976
Bagging	0.984	0.956	0.015	0.043	97.36%	0.972	0.974	0.944	0.978

Specifications of Confusion Matrix Components:

TP= True Positive

TN= True Negative

FN= False Negative

FP= False Positive

True Positive Rate $(TPR) = TP/(TP + FN)$

True Negative Rate $(TNR) = TN/(FP + TN)$

False Negative Rate $(FNR) = FN/(FN + TP)$

False Positive Rate $(FPR) = FP/(FP + TN)$

Accuracy $= (TP + TN)/(TP + TN + FN + FP)$

Positive Predictive Value $(PPV) = TP/(TP + FP)$

Negative Predictive Value $(NPV) = TN/(TN + FN)$

False Discovery Rate $(FDR) = FP/(FP + TP) = 1 - PPV$

F1 Score $= 2TP/(2TP + FP + FN)$

Chapter 5

Ensemble Based Intrusion Detection Model

5.1 Introduction

We have proposed a predictive model for the training process to detect the attacks in real time. The algorithm for the proposed model is implemented. For the proposed model we used support vector machine, decision tree and neural network as weak learners and combined them by majority voting we got the final predicted class. We have proposed a predictive model to detect the attacks in real time. The algorithm for the proposed model is implemented.

5.1.1 Support Vector Machine

Support Vector Machine (SVM) [Cristianini and Shawe-Taylor, 2000] was originally designed for binary classification, in which the margin is large and that try to separate the instances of various classes [12]. The margin is the minimum distance from instances of different classes to the classification hyperplane. The larger margin minimizes the generalization error of the classifier. The support vectors are the data points that are closest to the separating hyperplane [13]; these points are on the

boundary of the slab. SVM is associated with analysing the data and recognizing the patterns used both for classification and regression model.

5.1.2 Decision Tree

A decision tree consists of a set of tree-structured decision tests that works in a divide and conquer way. Each non-leaf node is associated with a feature test called split. The data in the nodes split into different subsets according to the values on the feature test. Each node is associated with a label, which will be assigned to instances falling into this node. A series of featured test is conducted starting from the root node and the result is obtained when a leaf node is reached.

In decision tree each internal node represents test on an attribute and each branch represents the result of test. The leaf represents the class label and the path from root to leaf represents the classification rule.

5.1.3 Neural Network

Neural networks, also known as artificial neural networks, are determined by the model of neuron, the network structure and the learning algorithm. Neuron is a unit, which is the basic computational component in neural networks [14]. Neurons are linked by weighted connections to form a network. There are many possible network structures, among which the most popular one is multi-layer feed-forward network. Here the neurons are connected layer-by-layer, and there are neither in-layer connections nor cross-layer connections. There is an input layer which receives the input feature vectors, where each neuron is usually corresponds to one element of the feature vector. There is an output layer, where each neuron usually corresponds to a possible label, or an element of a label vector. The layers between the input and output layers are called hidden layers. The goal of training the neural network is to determine the values of the connection weights and the bias of the neurons. Once these values are decided, the function computed by the neural

network is decided.

5.1.4 Majority Voting

Voting is the most popular and fundamental combination method for nominal outputs. Suppose we have given a T individual classifiers (h1....hT) and we have to combine the classifiers to predict the class label from a set of l possible class labels (c1....cl).

Majority voting is the most popular voting method where every classifier votes for one class label and the final output class label is the one that receives more than half of the votes [15]. If none of the class labels receives more than half of the votes, a rejection option will be given and the combined classifier makes no prediction.

5.2 Proposed predictive model

Our proposed predictive model is given in Fig. 5.1 and the detail about the training and testing process of the model is described in Fig. 5.2. The training and testing process has two phases, such as online and offline phase. In the offline phase network traffic repository is there to store the training instances and to extract the features with proper data pre-processing and to match this with the network online traffic behavior with proper model training. Our model has three main components like data pre-processing, model training and inference engine.

5.2.1 Implementation

The Proposed model is implemented using Matlab 2015a. The system configuration is Intel i7 CPU with 3.40 GHZ, 14GB RAM and Windows 8.1 64x Operating System.

As per the proposed architecture, the datasets are fed the ensemble framework. The three learning algortihm trained using the dataset. The detailed training process is given in Fig. 5.1

The training process first the data are preprocessed and normalized before the training process. Then the processed data is used for training the learning models. The AdaBoost concept is used to reduce the training error during training process. After successfully trained the model, we deploy the model for prediction. As per Fig. 5.1 the model the individual predicted class are combined using majority voting process. The final predicted class is taken for performance evaluation of the model.

To evaluate the model, two datasets are used namely NSLKDD Full and KDDCorrected. During Training the model total 25192 instances for NSLKDD and 77291 instances of KDDCorrected are used. In testing process, 125973 instances of NSLKDD and 311029 instances of KDDCorrected are predicted. The details of confusion matrix along with the model's performance parameters are given in Table 5.3

The model gives better result in NSLKDD and GureKDD as given in Table 5.3. The number of false alarms are reduced. The model gives better result in comparison to single classifier approach.

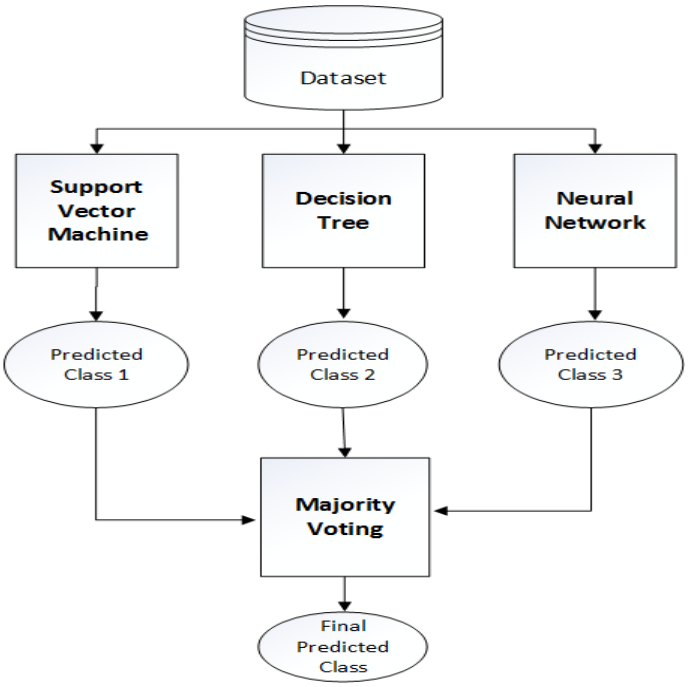

Figure 5.1: Proposed architectural diagram

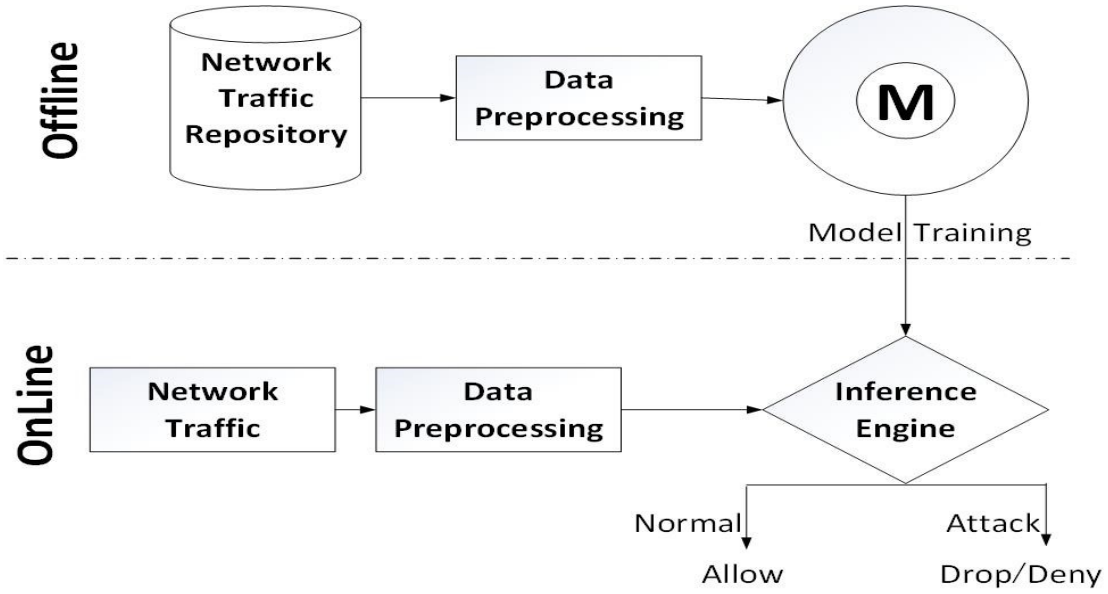

Figure 5.2: Model training to classify the network traffic

Table 5.1: Confusion Matrix for NSLKDD Dataset

Logitboost	PC	1	-1
AC	1	57993	637
	-1	450	66893

Table 5.2: Confusion Matrix for KDDCorrected Dataset

Logitboost	PC	1	-1
AC	1	244476	655
	-1	5960	58938

Table 5.3: Performance Evaluation of the proposed model

DataSet	TPR	TNR	FNR	FPR	Accuracy	PPV	NPV	MCC	F1 Score
KDDCorrected	0.97	0.97	0.023	0.027	97.55	0.99	0.9	0.924	0.984
NSLKDD	.989	0.993	0.01	0.006	99.13	0.992	0.99	0.982	0.99

Chapter 6

Conclusion

To discriminate the normal traffic and the attack traffic by using the Ensemble approach improves the detection accuracy with a less computational time and minimum cost compared to a single classifier. AdaBoost is an efficient false positive detection technique to minimize the false alarms. For the proposed model when we are using same dataset for training and testing, the accuracy percentage is high and error is less. But with different dataset for training and testing, the accuracy rate is comparatively less. Three weak classifiers such as support vector machine, decision tree and neural network are combined and their preformance accuracy is better than the individuals. We also concluded that with adding more number of learners, to the combination model, the detection accuracy increases and probability of misclassified instances reduces in each iteration.

Bibliography

[1] Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

[2] William Stallings. *Network and internetwork security: principles and practice*, volume 1. Prentice Hall Englewood Cliffs, 1995.

[3] Edward Amoroso. Intrusion detection: an introduction to internet surveillance, correlation, trace back, traps, and response. *Intrusion. Net Book*, 1999.

[4] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[5] Wray Buntine and Tim Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8(1):75–85, 1992.

[6] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[7] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[8] Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 325–332. ACM, 1996.

[9] Érico N de Souza and Stan Matwin. Extending adaboost to iteratively vary its base classifiers. In *Advances in Artificial Intelligence*, pages 384–389. Springer, 2011.

[10] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999.

[11] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[12] Giorgio Valentini and Thomas G Dietterich. Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *The Journal of Machine Learning Research*, 5:725–775, 2004.

[13] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001.

[14] Simon Haykin and Neural Network. A comprehensive foundation. *Neural Networks*, 2(2004), 2004.

[15] David H Wolpert and William G Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997.