TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

THESIS NO.:

A

THESIS REPORT

ON

ANOMALY BASED – INTRUSION DETECTION SYSTEM USING USER PROFILE GENERATED FROM SYSTEM LOGS

SUBMITTED BY:

ROSHAN POKHREL
069/MSCS/664

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING AS A PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE MASTER'S DEGREE IN COMPUTER SYSTEM AND KNOWLEDGE ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

April, 2016

Anomaly Based – Intrusion Detection System using User Profile Generated from System Logs

by

Roshan Pokhrel
069/MSCS/664

Thesis Supervisor

Arun Timalsina

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Information and Communication Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Tribhuvan University

Lalitpur, Nepal

Apr, 2016

COPYRIGHT ©

The author has agreed that the library, Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus, may make this thesis freely available for inspection. Moreover the author has agreed that the permission for extensive copying of this thesis work for scholarly purpose may be granted by the professor(s), who supervised the thesis work recorded herein or, in their absence, by the Head of the Department, wherein this thesis was done. It is understood that the recognition will be given to the author of this thesis and to the Department of Electronics and Computer Engineering, Pulchowk Campus in any use of the material of this thesis. Copying of publication or other use of this thesis for financial gain without approval of the Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus and author's written permission is prohibited.

Request for permission to copy or to make any use of the material in this thesis in whole or part should be addressed to:

Head

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Pulchowk, Lalitpur, Nepal

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a thesis report entitled "**ANOMALY BASED – INTRUSION DETECTION SYSTEM USING USER PROFILE GENERATED FROM SYSTEM LOGS**" submitted by **Roshan Pokhrel** in partial fulfillment of the requirements for the degree of Master of Science in Knowledge Base and Computer Engineering.

Supervisor: **Arun Timalsina**

Faculty Member

Department of Electronics and Computer Eingineering

External Examiner: Om Bikram Thapa

Title:

Committee Chairperson:

Title:

Date:

Departmental Acceptance

The thesis entitled “Title of Thesis”, submitted by Name of Student in partial fulfillment of the requirement for the award of the degree of “Master of Science in Information and Communication Engineering” has been accepted as a bonafide record of work independently carried out by him in the department.

Dibakar Raj Pant

Head of the Department

Department of Electronics and Computer Engineering,
Pulchowk Campus,
Institute of Engineering,
Tribhuvan University,
Nepal.

Acknowledgement

I am grateful to my Thesis supervisor **Dr. Arun Timilsina** for his incessant cooperative support, guidance and suggestion at various stages of the Thesis, especially at times of dire straits. I deeply value his work as my mentor by providing me with constant counseling and precious feedbacks on Thesis progress.

I would also like to express my sincere thanks to **Prof. Dr. Sashidhar Ram Joshi, Dr. Sanjeeb Prasad Pandey** and **Dr. Basanta Joshi** for providing precious suggestions about thesis concept and its inception.

I would also like to thank **Prof. Dr. Subarna Shakya** and **Dr. Aman Shakya** for their constant support and encouragement on research activity during master's program.

I would also like to thank my friend Mr. Prabhat Pokharel and Umanga Bista for his valuable support, guidance and encouragement towards completion of this thesis.

Lastly, I would like to thank all who were directly or indirectly involved in this thesis.

Abstract

Intrusion Detection System (IDS) is a form of defense that aims to detect suspicious activities and attack against information systems in general. With new types of attacks appearing continuously, developing adaptive and flexible security oriented approaches is a severe challenge. In this scenario, this thesis presents an anomaly-based intrusion detection technique as a valuable technology to protect target system against malicious activities. This technique uses a semi-supervised learning model to identify and learn from past events as manifested in system logs and build a user behavior profile. The observed behavior of the user is analyzed to infer whether or not the normal profile supports the observed one. This is carried out using two class classifier. A new hybrid approach using SVM and NB is proposed that provides better accuracy and reduces the problem of high false alarm ratio. The comparison of the proposed approach is made with other SVM and NB techniques. Also, user profile training technique is enhanced by addition of new feature derived from the existing dataset. With these two proposed approaches detection rate is improved considerably. For the validation of the result cross validation is employed and the result is presented using ROC curve. The experimentation is implemented in two datasets from two different organizations.

Keywords: Intrusion Detection System (IDS), Anomaly detection, Security, SVM, NB, User profiling, Cross-Validation, Receiver Operating Curve (ROC)

Table of Contents

List of Figures

List of Tables

Abbreviations

ABIDS	Anomaly Based Intrusion Detection System
AD	Anomaly Detection
DARPA	Defense Advanced Research Projects Agency
HMM	Hidden Markov Model
IDS	Intrusion Detection System
Event ID	Event Identifier
NIDS	Network Intrusion Detection System
SIEM	Security Incident and Event Management
SVM	Support Vector Machine
SVC	Support Vector Classifier
OCSVM	One Class Support Vector Machine
RBF	Radial Basis Function
RBFSVM	Radial Basis Function Support Vector Machine
NB	Naïve Bayes
HIDS	Host Based Intrusion Detection System
NIDS	Network Based Intrusion Detection System
AIDS	Application-based Intrusion detection System
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
PPV	Positive Predictive Value

1. Introduction

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices [1]. An intrusion detection system (IDS) or *anomaly detection system* refers to the problem of finding patterns in data that do not conform to expected behavior. These nonconforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities, or contaminants in different application domains [2]. Of these, anomalies and outliers are two terms used most commonly in the context of anomaly detection; sometimes interchangeably.

Due to worldwide proliferation in network environments, a variety of faster services have become a reality. However, the higher the reliance on computers, the more important security problems become. An IDS identifies or monitors any kind of intrusion and notify immediately in the form of alert so that resources never get compromised. An IDS is also used in legal proceedings as forensic evidence against the intruder because it provides recording of any kind of intrusion involved in cybercrime. An IDS is deployed to cover unauthorized access to resources or data. It can be hardware and/or software. An IDS can be used to protect a single host or a whole computer network. IDS provides user friendly interface to non-expert staff for managing the systems easily [3].

1.1. Types of IDS

IDS uses different types of technologies and are classified as follows:

1. Host Based Intrusion Detection System (HIDS)
2. Network Based Intrusion Detection System (NIDS)
3. Application-based Intrusion detection System (AIDS)

1.1.1. Host Based Intrusion Detection System (HIDS)

HIDS is a intrusion detection that takes place on a single host system. Intrusion Detection System is installed on a host in the network and the system collects and analyzes the traffic that is originated or is intended to that host. It leverages their privileged access to monitor specific components of a host that are not readily accessible to other systems. Specific components of the operating system such as password files in UNIX and the Registry and Security Audit in Windows can be watched for misuse.

Although HIDS is far better than NIDS in detecting malicious activities for a particular host, they have limited view of entire network topology and they cannot detect attack that is targeted for a host in a network which does not have HIDS installed [4].

1.1.2. Network Based Intrusion Detection System (NIDS)

A network-based intrusion detection system monitors the traffic on its network segment as a data source. Network IDSs (NIDS) are placed in key areas of network infrastructure and monitors the traffic as it flows to other host. Unlike HIDS, NIDS have the capability of monitoring the network and detecting the malicious activities intended for that network. Monitoring criteria for a specific host in the network can be increased or decreased with relative ease.

NIDS should be capable of standing against large amount number of network traffic to remain effective. As network traffic increases exponentially NIDS must grab all the traffic and analyze in a timely manner [4].

1.1.3. Application Based Intrusion Detection System (AIDS)

It monitors the traffic on a network based on events that occur in some kind of specific applications. This may be Host based , Network based or hybrid. It monitors the traffic of a single host and the events happening within that host for malicious activities. It monitors the log file, various running processes, running, applications, file access and modification, system and application configuration changes on a single host. It is basically deployed on critical hosts like publicly accessible servers and those servers having critical information [5].

1.2. Approach to Intrusion Detection

In real-time detection, when an IDS monitor for any kind of intrusion in the computer system it immediately notifies in the form of alert and proper action is taken accordingly so that system security never breaches. The real-time IDS works in offline mode by using the previous or historical data collected from intrusion identified in the past [6]. Basically there are two approaches of intrusion detection system: signature based (also known as misuse based) and anomaly based [7].

1.2.1. Signature-Based Intrusion Detection System

Signature based systems rely on pattern recognition techniques where they maintain the database of signatures of previously known attacks and compare them with analyzed data. An alarm or intrusion signal is raised when the signatures are matched. Examples of signature-based systems are Snort [8] and Bro [9]. The advantages of signature-based IDS are commonly known to be their potential for low false alarm rates, and the information they often impart to a system security officer about a detected attack. Such information is often encoded in the rules or patterns central to the functionality of such systems. This information is often invaluable when initiating preventive or corrective actions. However, signature-based IDS have several disadvantages. Since the set of anomalous patterns are based on known attacks, new attacks cannot be discovered by these systems. Therefore, whenever a new attack is discovered, patterns corresponding to the attack have to be manually constructed. Moreover, a sophisticated and determined attacker cannot be expected to rely on known attacks – he will attempt to infiltrate a system with stealthy, sophisticated attacks using behavior that won’t be detected by static pattern-based schema [10]. For example, an attacker can “mix” normal activity with a real attack so that the trace does not match any of the pre-defined patterns.

1.2.2. Anomaly-Based Intrusion Detection System

On the other hand, anomaly based systems builds a statistical model of a subject (e.g. user, file, privileged program, host machine, and network) describing the normal behavior of network traffic; compare the observed behavior of the subject with its normal profile, and any abnormal behavior that deviates from normal model is identified and an alarm or intrusion signal is raised. In contrast to signature-based systems, anomaly-based systems have the advantage that they can detect zero-day attacks, since novel attacks can be detected as soon as they take place. However, anomaly based systems (unlike signature based systems) requires a training phase to develop the database of general attacks and a careful setting of threshold level of detection [10]. The major drawback of anomaly-based system is defining its rule set. The efficiency of the system depends on how well it is implemented and tested on all environments. For detection to occur correctly, the detailed knowledge about the accepted network behavior profile needs to be developed. But once the rules are defined, correct normal profile and test environment is built then anomaly detection systems works well [7].

1.3. Anomaly Detection Model

There are advantages and limitations of both the techniques. In signature based IDS there is a high accuracy of detecting known attack, but now-a-days security is the main concern in every field and every day a new type of attack is introduced. Therefore, signature based IDS are not able to detect that intrusion and system leads to zero-day attacks. To overcome this problem anomaly detection is the best method to detect the new type of attack on the basis of their behavior. There is some limitation in anomaly detection also but it protects the system from zero-day attacks.

Anomaly detection consist of two phases

a. Training phase
b. Testing phase

On the basis of that, behavior is differentiated. In a training phase, dataset is trained about the normal and/or abnormal traffic profile. After that this training profile is tested on the test dataset to check the accuracy of the detection approach.

To detect anomalies three different methods can be used [2]:

1.3.1. Supervised Anomaly Detection

Techniques trained in supervised mode assume the availability of a training data set, which has labeled instances for normal as well as anomaly class. Typical approach in such cases is to build a predictive model for normal vs. anomaly classes. Any unseen data instance is compared against the model to determine which class it belongs to. There are two major is- sues that arise in supervised anomaly detection. First, the anomalous instances are far fewer compared to the normal instances in the training data. Second, obtaining accurate and representative labels, especially for the anomaly class is usually challenging. Other than these two issues, the supervised anomaly detection problem is similar to building predictive models.

1.3.2. Semi-supervised Anomaly Detection

Techniques that operate in a semi-supervised mode, assume that the training data has labeled instances for only the normal class. Since they do not require labels for the anomaly class, they are more widely applicable than supervised techniques. For example, in space craft fault detection, an anomaly scenario would signify an accident, which is not easy to model. The typical approach used in such techniques is to build a model for the class corresponding to normal behavior, and use the model to identify anomalies in the test data.

1.3.3. Unsupervised Anomaly Detection

Techniques that operate in unsupervised mode do not require training data, and thus are most widely applicable. The techniques in this category make the implicit assumption that normal instances are far more frequent than anomalies in the test data. If this assumption is not true then such techniques suffer from high false alarm rate.

Many semi-supervised techniques can be adapted to operate in an unsupervised mode by using a sample of the unlabeled data set as training data. Such adaptation assumes that the test data contains very few anomalies and the model learnt during training is robust to these few anomalies.

1.4. Types of IDS Output

An important aspect for any intrusion detection technique is the manner in which the anomalies are reported. Typically, the outputs produced by anomaly detection techniques are one of the following two types:

1.4.1. Scores

Scoring techniques assign an anomaly score to each instance in the test data depending on the degree to which that instance is considered an anomaly. Thus the output of such techniques is a ranked list of anomalies. An analyst may choose to either analyze top few anomalies or use a cut-off threshold to select the anomalies.

1.4.2. Labels

Techniques in this category assign a label (normal or anomalous) to each test instance. The main motive of the IDS is to catch the intruder before a real and serious damage to computer network. An IDS notifies the administrator by using alerts about the unauthorized access to the computer system. IDS generates a huge number of alerts. IDS classify alerts into four categories [11]:

I. True Positive (TP): A real intrusion for which IDS generates an alert.

II. False Positive (FP): No intrusion, but IDS generate an alert.

III. False Negative (FN): A real intrusion, but IDS never generate any kind of alert.

IV. True Negative (TN): No intrusion and IDS never generate an alert.

2. Motivation

Due to the rapid growth of technology, defending against a number of threats to maintain the integrity, confidentiality and availability of the computer system is of prime importance and information security plays a very important role in the safety of computer devices. We need to detect intruder by applying proper techniques. But we cannot prevent all break-ins. There will always be new holes, new attacks, and new attackers. However, we need some way to cope with all these.

Every day a new type of attack has been introduced to a misuse based detection method. It is not possible to detect new attacks. So anomaly detection plays a vital role while detecting the unknown attack on the basis of their behavior.

3. Problem Definition

At an abstract level, an anomaly is defined as a pattern that does not conform to expected normal behavior. A straightforward anomaly detection approach, therefore, is to define a region representing normal behavior and declare any observation in the data that does not belong to this normal region as an anomaly [2]. But several factors make this apparently simple approach very challenging:

- Availability of data for training/validation of models used by anomaly detection techniques is usually a major issue.
- Defining a normal region that encompasses every possible normal behavior is very difficult.
- In addition, the boundary between normal and anomalous behavior is often not precise. Thus an anomalous observation that lies close to the boundary can actually be normal, and vice versa as shown in figure 2.1.
- In many domains normal behavior keeps evolving and a current notion of normal behavior might not be sufficiently representative in the future.

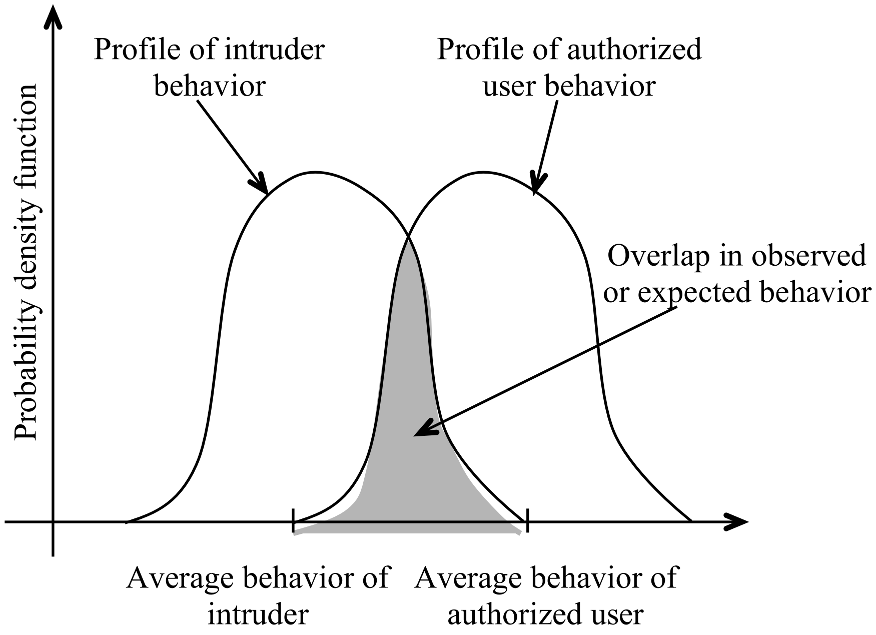

Probability density function
Profile of intruder behavior
Profile of authorized user behavior
Overlap in observed or expected behavior
Average behavior of intruder
Average behavior of authorized user

Figure 3.1: Behavior Profile of Intruder and Authorized User

4. Objective

The objective of this thesis proposal can be outlined as follow:

- To build normal behavior user profile based on system logs.
- To evaluate the performance of classification via outlier detection, and detect the anomalies with more accuracy
- To come up with a solution to detect anomalous events that may be an indicator of outlier, insider misuse or attack.

5. Scope of Work

Insider misuse can be defined as the performance of activities where computers and networks in an organization are deliberately misused by those who are not authorized to use them. Most of the models available are based on already trained dataset and they might not be able to find new attacks. Those models need to timely update their dataset to fulfill the same. Due to alarming needs of SIEM (Security Information and Event Management) in Organizations, monitoring and alerting administrators with new type of attacks and activities and incident has become a must in this growing technology. So, if we could make some sort of anomaly integration of real dataset with SIEM, it would add a new dimension to IT industry.

Therefore, this thesis proposal presents way to detect such incident categorized as:

- Anomalous behavior of user action
- Comply with security policies and regulation
- Develop systems for troubleshooting and problem diagnostics
- Providing evidence of accidental or deliberate security breaches for forensic investigations
- Zero-day attack

6. Research Methodology

6.1. Literature Review

Given the advantages that intrusion detection system may offer, more and more detection systems now rely based on anomaly. An intrusion detection system finds its application in all areas of information security and systems. Several research works have already been done and many research papers have been published regarding anomaly detection techniques. Some research also talks about support vector machine, Naïve Bayes process analysis with different approaches for process control. However, most of the researches done are using popular datasets like DARPA, KDD Cup, Kaggle datasets etc. Research using real dataset is very less heard of due to complexity attached with this.

Some of the related works that are closely related to proposed work are mentioned below along with their scope of research.

M. Corney et al. research titled "Detection of Anomalies from User Profiles Generated from System Logs" to identify anomalous events and event patterns manifested in computer system logs. Prototype software was developed with a capability that identifies anomalous events based on usage patterns or user profiles, and alerts administrators when such events are identified. More specifically the research attempted to detect unauthorized use of software applications by users from within an organization [12].

S. H. Paek et al. proposes the architecture of host-based intrusion detection model generation system in the paper titled "The Architecture of Host-based Intrusion Detection Model Generation System for the Frequency Per System Call". The architecture creates candidate models by various and popular existing data mining techniques and one new technique for the process behavior data set with the frequency feature per system call and then elects the best appropriate model according to user requirements after evaluating candidate models. The frequency feature per system call is simpler than the existing system call sequence feature in applying to intrusion detection system as the model [13].

J. P. Anderson introduced a term audit trail in a report titled "Computer security threat monitoring and surveillance" which includes information for tracking down the misuse and user behavior [14]. This paper basically introduced a misuse detection technique. This paper provides a base for IDS design and development.

A. H. Katherine from Department of Computer Science, Columbia University, New York researched on "One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses". This paper presents a new Host-based Intrusion Detection System (IDS) that monitors accesses to the Microsoft Windows Registry using Registry Anomaly Detection (RAD). The system uses a one class Support Vector Machine (OCSVM) to detect anomalous registry behavior and detect outliers in new (unclassified) data generated from the same system [15].

M. Zhang et al. research on "An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions" proposes an anomaly detection model based on One-class SVM to detect network intrusions. The one-class SVM adopts only normal network connection records as the training dataset. But after being trained, it is able to recognize normal from various attacks [16].

R. Chitrakar et al. researched on "Anomaly detection using Support Vector Machine classification with k-Medoids clustering". In this paper, a better combination is proposed to address problems of the previously proposed hybrid approach of combining k-Means/k-Medoids clustering technique with Naïve Bayes classification. In this approach, the need of large samples by the previous approach is reduced by using Support Vector Machine while maintaining the high quality clustering of k-Medoids [17].

N. B. Amor et al. research paper titled "Naive bayes vs decision trees in intrusion detection systems" performed a comparison between two classifiers native Bayes networks and decision tree using KDD Cup dataset 1999 [18]. Native Bayes and decision tree having their own decision capable to detect the intrusion. Both performed equally however, while detecting U2R and probe native bayes performed better and in normal, DOS and R2L decision tree performed better.

R. Jain et al. carried out a survey on “Network attacks, classification and models for anomaly based network intrusion detection system”. This paper presents a selective survey of incremental approaches for detecting anomaly in normal system and network traffic [19].

N. A. Durgin and P. Zhang researched on “Profile-Based Adaptive Anomaly Detection for Network Security”. The research focused on enhancing current IDS capabilities by addressing some of these shortcomings. They identified and evaluated promising techniques for data mining and machine-learning. The algorithms were "trained" by providing them with a series of data-points from "normal" network traffic. They also built a prototype anomaly detection tool that demonstrates how the techniques might be integrated into an operational intrusion detection framework [10].

S. S. Murtaza et al. research on “A host-based anomaly detection approach by representing system calls as states of kernel modules” attempts to reduce the false alarm rate and processing time while increasing the detection rate. The paper presents a novel anomaly detection technique based on semantic interactions of system calls which analyzes the state interactions, and identifies anomalies by comparing the probabilities of occurrences of states in normal and anomalous traces [20].

C. Manikopoulos and S. Papavassiliou paper titled “Network intrusion and fault detection: a statistical anomaly approach” applies neural network and SVM classifiers that was used to detect the anomalies [21]. The main objective of this paper was to create robust, effective and efficient classifiers which detects the intrusion in the real-time. The idea was to discover patterns or features that describe the user behavior. In this approach both neural network and SVM perform better rather than another technique of classifiers.

X. Yingchao et al. paper titled “Parameter Selection of Gaussian Kernel for One-Class SVM” proposes a novel method to solve the problem of kernel parameter selection in one class classifier, specifically, one-class SVM (OCSVM) [22]. D. P. Gaikwad et al. paper on “Intrusion Detection System Using Bagging Ensemble Method of Machine Learning” presents a novel intrusion detection technique based on ensemble method of machine learning is proposed. The Bagging method of ensemble with REPTree as base class is used to implement intrusion detection system [23].

A. J. Hoglund et al. paper titled “A computer host-based user anomaly detection system using the self-organizing map” aimed at designing a system that contains an automatic anomaly detection component [24]. A prototype UNIX anomaly detection system was constructed for anomaly detection attempts to recognize abnormal behavior to detect intrusions. The component for detection used a test based on the self-organizing map to test if user behavior is anomalous.

I. S. Thaseen paper titled “Intrusion detection model using fusion of PCA and optimized SVM” proposes a novel method of integrating principal component analysis (PCA) and support vector machine (SVM) by optimizing the kernel parameters using automatic parameter selection technique. This technique reduces the training and testing time to identify intrusions thereby improving the accuracy. The proposed method was tested on KDD data set [25].

L. Lin et al. research on “SVM ensemble for anomaly detection based on rotation forest” proposes a new intelligent intrusion detection system using SVM ensemble. The ensemble was made of two-layer, one is composed by five SVM network decided by winner-take-all, the other is a ensemble network composed of five classifier decided by majority voting [26].

S.Peddabachigari et al. paper on “Modeling intrusion detection system using hybrid intelligent systems” provides a new hybrid approach called DTSVM (Decision trees - SVM) in which two classifiers decision tree and SVM were used as an individual base classifier. The motive of this hybrid technique was to increase the detection accuracy and reduce the computational complexity. This hybrid approach was provided better accuracy than the individual classifier. The paper gives a great idea or a new concept of using multiple classifiers to improve the detection accuracy and reduce the computational complexity [27].

D. P. Gaikwad paper titled “Intrusion Detection System Using Bagging Ensemble Method of Machine Learning” presents a novel intrusion detection technique based on ensemble method of machine learning is proposed [28]. The Bagging method of ensemble with REPTree as base class was used to implement intrusion detection system. The relevant features from NSL_KDD dataset are selected to improve the classification accuracy and reduce the false positive rate. The performance of proposed ensemble method is evaluated in term of classification accuracy, model building time and False Positives. The experimental results show that the Bagging ensemble with REPTree base class exhibits highest classification accuracy.

P. Amudha et al. paper titled “Intrusion detection based on Core Vector Machine and ensemble classification methods” proposes a combined algorithm based on Principal Component Analysis (PCA) and Core Vector Machine (CVM), which is an extremely fast classifier, for intrusion detection [29]. PCA was used as feature extraction technique to select principal features from the intrusion detection KDDCup'99 dataset and an intrusion detection model was constructed by CVM algorithm. The effectiveness of the features selected was also tested on ensemble based classifiers and the results are compared with the standard classifiers.

P. Sornsuwit and S. Jaiyen research titled “Intrusion detection model based on ensemble learning for U2R and R2L attacks” concentrates on ensemble learning for detecting network intrusion data, which are difficult to detect. In addition, correlation-based algorithm was used for reducing some redundant features. Adaboost algorithm was adopted to create the ensemble of weak learners in order to create the model that can protect the security and improve the performance of classifiers. The U2R and R2L attacks in KDD Cup'99 intrusion detection dataset were used to train and test the ensemble classifiers. The experimental results show that reducing features can improve efficiency in attack detection of classifiers in many weak leaners [30].

L. Lin et al. research titled “SVM ensemble for anomaly detection based on rotation forest” presents a new intelligent intrusion detection system has been proposed using

SVM ensemble. The ensemble was made of two-layer, one was composed by five SVM network decided by winner-take-all, the other was a ensemble network composed of five classifier decided by majority voting. The KDD99 data sets was used to test which achieve a better performance [31].

6.2. System Design – Methodology

The assumption that underlies all user-based anomaly detection schemes for intrusion detection is that intrusive behavior is, by its very nature, anomalous [32]. Under such schemes, if it can be established that a given user is acting in an abnormal manner then the actions of that user (or someone who is masquerading as that user) can be classified as outlier or intrusive. In these approaches, behaviors can be determined to be abnormal through a comparison against a user profile that represents a user's typical behavior. This user profile, which can take on many forms, is based upon either an individual's behavior and/or the typical behavior of the individuals in a functional group. Some anomaly detection systems also maintain a model of typical system behavior. In an anomaly detection system, a model of system performance metric is maintained. Any time that the system is not operating in a normal manner there is an increased likelihood that an intruder is (or was) present on the system.

The aim of this thesis has been to develop prototype software that implements a capability to identify events that are anomalous and may be an indicative of computer misuse. The work has been carried out with the data from Windows security audit log from computer running the Windows Server 2008. When various audit controls are enabled, these logs record information about user log on sessions, failed login events, account lockout events, application or processes invoked or terminated b the user of computers and by the computer system itself.

The basic model contains two phases for anomaly detection: profile creating phase and detection phase as shown in figure 5.1.

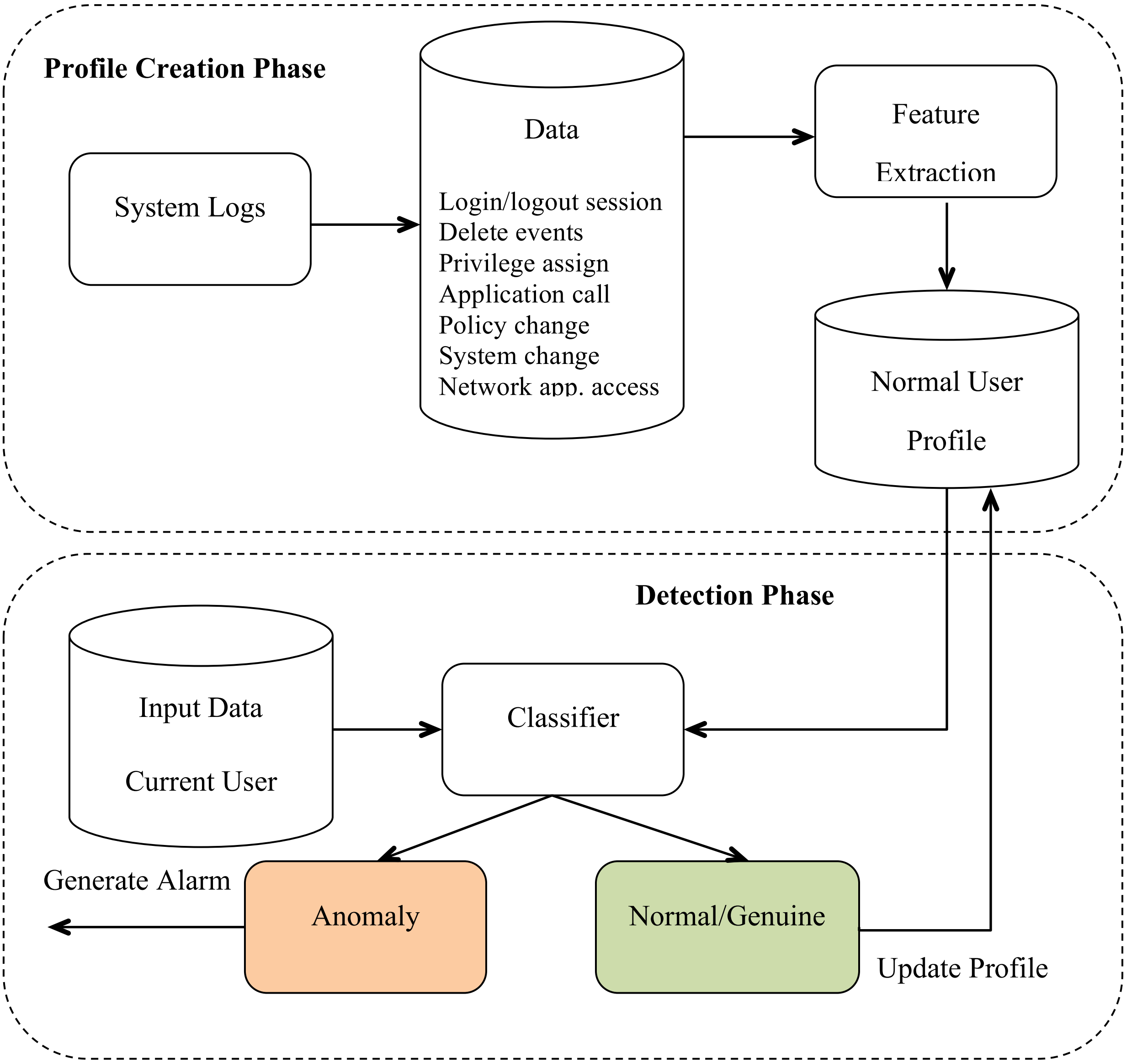


Figure 6.1: Model for ABIDS Using Profile Based Approach

6.2.1. Profile Creation

User profiles are generated so that users' normal or habitual or genuine use of system and applications can be determined. The approach, here, is to build user profiles from computer security audit logs which records user's activities as events.

Before a profile can be generated for a particular user we must have that person's usage data for a specific period of time. The training period should be selected so that most of the routine activities a user performs are included. This will likely be different for different users. For this thesis, static or constant window user profile is used. With this approach detection of outliers is carried out on the test datasets while the user profile remains same.

User profiles have been created from data recorded in the Windows Security log by identifying following:

- Duration of Logon Session and
- Frequency of occurances of the following:
    - Failed login
    - Account lockout
    - Process execution
    - Sessions during working hours

In particular, the security audit log form computers running the Windows Server 2008 has been used for this work.

6.2.2. Anomaly Detection

For the detection of anomalous behavior, current user behavior profile is compared with the existing normal profile. Certain threshold is maintained to calculate the similarity between current and normal profile. If current profile is similar to normal profile i.e. current profile does not cross threshold, than the user is genuine. Else, current user behavior is anomalous.

6.3. Data Collection

Windows Security log data is collected and examined during the course of this thesis were from desktop computers running Windows Server 2008. A great deal of information useful for extraction of a user's activities is recorded in the Windows Security log. Different windows events are assigned with different event-id. Based on this event-id we can extract the feature of our interest. All events contains common data such as date and timestamp, computer name, domain name, user name, event type, and further information specific to each type of event.

Security event logging was enabled and all available auditing options were set and data was collected for a period of nine consecutive months. Example of each audit log event collected and used in this thesis are as follows:

```
<8>Jan 5 11:29:51 test.logpoint.com MSWinEventLog   1   Service   277191969   Wed   Feb   25
06:42:50 2016   4624   Microsoft-Windows-Security-Auditing N/A   N/A   Success   Audit
        test.myaccount.comUser Logon   An account was successfully logged on. Subject: Security ID: S-1-0-0
Account Name: - Account Domain: - Logon ID: 0x0 Logon Type: 3 New Logon: Security ID: S-1-5-21-2187590103-
147294922-1584409417-33234 Account Name: INTPRDADMADWMI Account Domain: WinDomain Logon ID:
0x11e851098 Logon GUID: {00000000-0000-0000-0000-000000000000} Process Information: Process ID: 0x0
Process Name: - Network Information: Workstation Name: BEL-VAPP-RTMS2 Source Network Address: 1.1.1.1
Source Port: 48204 Detailed Authentication Information: Logon Process: NtLmSsp Authentication Package: NTLM
Transited Services: - Package Name (NTLM only): NTLM V1 Key Length: 128
```

```
<8>Jan 5 11:29:51 test.myaccount.com MSWinEventLog   1   Service   277191969   Wed   Feb   25
06:42:50 2016   4625   Microsoft-Windows-Security-Auditing N/A   N/A   Success   Audit
        test.myaccount.comUser Logon   An account failed to log on. Subject: Security ID: S-1-0-0 Account
Name: - Account Domain: - Logon ID: 0x0 Logon Type: 3 Account For Which Logon Failed: Security ID: S-1-0-0
Account Name: ABC Account Domain: DOMAINE_NT Failure Information: Failure Reason: %%2310 Status:
0xc000006e Sub Status: 0xc0000072 Process Information: Caller Process ID: 0x0 Caller Process Name: - Network
Information: Workstation Name: 99NB121 Source Network Address: 10.5.100.38 Source Port: 61780 Detailed
Authentication Information: Logon Process: NtLmSsp Authentication Package: NTLM Transited Services: - Package
Name (NTLM only): - Key Length: 0
```

```
<14>Mar 2 11:34:38 89.237.143.23 MSWinEventLog   1   Service   277191969   Wed   Feb   25
06:42:50 2016   4647   Microsoft-Windows-Security-Auditing N/A   N/A   Success   Audit
        test.myaccount.comUser Logoff   User initiated logoff: Subject: Security ID: WIN-
R9H529RIO4Y\Administrator Account Name: Administrator Account Domain: WIN-R9H529RIO4Y Logon ID:
0x19f4c
```

```
<14>Mar 2 11:34:38 89.237.143.23 MSWinEventLog   1   Service   277191969   Wed   Feb   25
06:42:50 2016   4740   Microsoft-Windows-Security-Auditing N/A   N/A   Success   Audit
        test.myaccount.comUser Logoff   A user account was locked out. Subject: Security ID: SYSTEM
Account Name: WIN-R9H529RIO4Y$ Account Domain: WORKGROUP Logon ID: 0x3e7 Account That Was Locked
Out: Security ID: WIN-R9H529RIO4Y\John Account Name: John Additional Information: Caller Computer Name:
WIN-R9H529RIO4Y
```

```
<13>Aug 6 16:23:44 SESOAWV2064.adroot.net MSWinEventLog   1   Service   277191969   Wed   Feb
25 06:42:50 2016   4688   Microsoft-Windows-Security-Auditing ADROOT\SESOAWV2064$ N/A   Success
Audit   SESOAWV2064.kb.myaccount.net   Process Creation   A new process has been created. Subject: Security
ID: S-1-5-18 Account Name: SESOAWV2064$ Account Domain: ADROOT Logon ID: 0x3e7 Process Information:
New Process ID: 0xa98 New Process Name: C:\Windows\System32\notepad.exe Token Elevation Type:
TokenElevationTypeDefault (1) Creator Process ID: 0x54c
```

```
<14>Mar 2 11:34:38 89.237.143.23 MSWinEventLog   1   Service   277191969   Wed   Feb   25
06:42:50 2016   4689   Microsoft-Windows-Security-Auditing N/A   N/A   Success   Audit
	test.myaccount.comUser Logoff   A process has exited. Subject: Security ID: WIN-
R9H529RIO4Y\Administrator Account Name: Administrator Account Domain: WIN-R9H529RIO4Y Logon ID:
0x1fd23 Process Information: Process ID: 0xed0 Process Name: C:\Windows\System32\notepad.exe Exit Status: 0x0
```

Table 1 below shows the summary of events.

Table 6.1: Summary of Windows Security Audit Event

Event ID	**Description**
4624	An account was successfully logged on
4625	An account failed to log on
4647	User initiated logoff
4740	A user account was locked out
4688	A new process has been created
4689	A process has exited

Figure 5.2 shows scatter plot of the collected data.

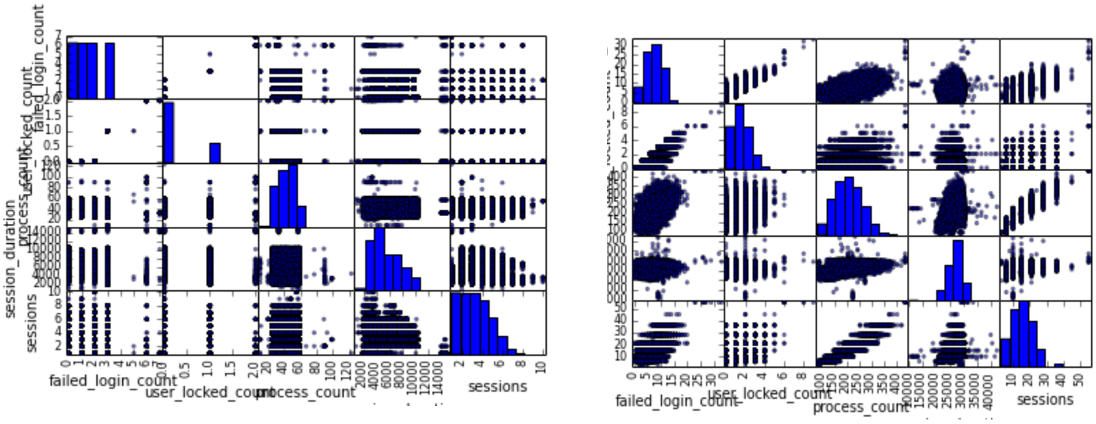


Figure 6.2: Data Summary. Left: per Sessions. Right: per Working Day

6.4. Data Preparation and Correlation

Windows has the ability to generate a detailed audit record of security events on each system. Windows logs events for the two types of security accounts: *Computer* and *User* for their logon and authentication. Computer account authentication events list computer name for the user name and is recognized with $ appended to the computer name. These computer accounts does not describe the actual user behavior. I, therefore, have excluded the affect of computer account in this thesis analysis.

For user profiling, the most useful event types include log on, log off, process start, process exited, failed login, account lockout. These events provide details of user's log-in sessions, interaction with application, account failed and locked.

A logon session is a computing session that begins when a user authentication is successful and ends when the user logs off of the system. It is necessary to correlate logon events (event IDs 4624 and 4634) to determine the duration of a user's log in session and the number of session during working hours.

Session duration and number of session in working hour is given by the formula below:

$$\begin{aligned} &Session\ Duration\ (for\ a\ given\ logon\ session) \\ &\qquad = Logoff\ time\ -\ Logon\ time \end{aligned} \quad 1$$

$$Number\ of\ Sessions = \ Count\ of\ Logon \qquad 2$$

Logon failure events are combined into one event ID 4625 with the proper status codes to identify different reason for logon failure.

$$Number\ of\ Failed\ Login\ = \ Count\ of\ event\ id\ 4625 \qquad 3$$

Account lockouts events are combined into one event ID 4740.

$$Number\ of\ Failed\ Login\ = \ Count\ of\ event\ id\ 4740 \qquad 4$$

Process executed and terminated are combined into event ID 4688 and 4689 respectively

$$Number\ of\ Failed\ Login\ = \ Count\ of\ event\ id\ 4688\ or\ 4689 \qquad 5$$

6.5. Algorithm Development

Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels. For example, in our problem, a classification model can be built to categorize events as outlier or normal. Such analysis can help provide us with a better understanding of the data at large.

Dataset classification is a two-step process, consisting of a *learning step* (where a classification model is constructed) and a *classification step* (where the model is used to predict class labels for a given data) [33].

In the first step, a classifier is built describing a predetermined set of data classes or concepts. This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or “learning from” a training set made up of database tuples and their associated class labels. A tuple, $\boldsymbol{X}$, is represented by an n-dimensional attribute vector, $\boldsymbol{X} = (x_1, x_2, \dots, x_n)$, depicting n measurements made on the tuple from n datasets attributes, respectively, $A_1, A_2, \dots, A_n$. Each tuple, X, is assumed to belong to a predefined class as determined by another database attribute called the class label attribute. The class label attribute is discrete-valued and unordered. It is categorical (or nominal) in that each value serves as a category or class. The individual tuples making up the training set are referred to as training tuples and are randomly sampled from the datasets under analysis.

This first step of the classification process can also be viewed as the learning of a mapping or function, $y = f(\boldsymbol{X})$, that can predict the associated class label y of a given tuple $\boldsymbol{X}$. In this view, it is wished to learn a mapping or function that separates the data classes. Typically, this mapping is represented in the form of classification rules, decision trees, or mathematical formulae. In this thesis, the mapping is represented as classification rules that identify datasets as being either $normal$ or anomaly. The rules can be used to categorize future data tuples, as well as provide deeper insight into the data contents.

In the second step, the model is used for classification. First, the predictive accuracy of the classifier is estimated. If it were to use the training set to measure the classifier's accuracy, this estimate would likely be optimistic, because the classifier tends to *overfit* the data (i.e., during learning it may incorporate some particular anomalies of the training data that are not present in the general data set overall). Therefore, a *test set* is used, made up of *test tuples* and their associated class labels. They are independent of the training tuples, meaning that they were not used to construct the classifier.

The *accuracy* of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. The associated class label of each test tuple is compared with the learned classifier's class prediction for that tuple. If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known [33].

Two classification algorithms are used in this thesis: Support Vector Machine (SVM) and Naïve Bayes (NB). To improve the accuracy of the classifier hybrid model is proposed that uses the advantage of both RBFSVM and NB.

6.5.1. Support Vector Machine (SVM)

Support vector machines (SVMs) is a method for the classification of both linear and nonlinear data. In a nutshell, an SVM is an algorithm that works as follows. It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane (i.e., a “decision boundary” separating the tuples of one class from another). With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. The SVM finds this hyperplane using support vectors (“essential” training tuples) and margins (defined by the support vectors) [33].

Although the training time of even the fastest SVMs can be extremely slow, they are highly accurate, owing to their ability to model complex nonlinear decision boundaries. They are much less prone to overfitting than other methods. The support vectors found also provide a compact description of the learned model. SVMs can be used for numeric prediction as well as classification. They have been applied to a number of areas, including handwritten digit recognition, object recognition, and speaker identification, as well as benchmark time-series prediction tests [33].

6.5.1.1. One Class Support Vector Machine (OCSVM)

OCSVM is a form of SVM. The OCSVM algorithm maps input data into a high dimensional feature space (via a kernel) and iteratively finds the maximal margin hyperplane which best separates the training data from the origin [15]. The OCSVM may be viewed as a regular two-class SVM where all the training data lies in the first class, and the origin is taken as the only member of the second class. Thus, the hyperplane (or linear decision boundary) corresponds to the classification rule:

$$f(x) =< \boldsymbol{w}, \boldsymbol{x} > \ +b \tag{6}$$

where $\mathbf{w}$ is the normal vector and b is a bias term. The OCSVM solves an optimization problem to find the rule f with maximal geometric margin. We can use this classification rule to assign a label to a test example $\mathbf{x}$. If $f(\boldsymbol{x}) < 0$ we label $\mathbf{x}$ as an anomaly, otherwise it is labeled normal. In practice there is a trade-off between maximizing the distance of the hyperplane from the origin and the number of training data points contained in the region separated from the origin by the hyperplane [15].

6.5.1.2. Kernels

Solving the OCSVM optimization problem is equivalent to solving the dual quadratic programming problem:

$$\min_{\alpha} \frac{1}{2} \sum_{ij} \alpha_i \propto_j K(x_i, x_j) \tag{7}$$

Subject to the constraints

$$0 \leq \propto_i \leq \frac{1}{vl} \tag{8}$$

and

$$\sum_{i} \propto_i = 1 \tag{9}$$

where $\propto_i$ is a lagrange multiplier (or "weight" on example i such that vectors associated with non-zero weights are called "support vectors" and solely determine the optimal hyperplane), v is a parameter that controls the trade-off between maximizing the distance of the hyperplane from the origin and the number of data points contained by the hyperplane, l is the number of points in the training dataset, and $K(x_i, y_i)$ is the kernel function. By using the kernel function to project input vectors into a feature space, it is allowed for nonlinear decision boundaries. Given a feature map:

$$\emptyset : X \rightarrow \mathbb{R}^N \tag{10}$$

where $\emptyset$ maps training vectors from input space X to a high dimensional feature space, we can define the kernel function as:

$$K(x,y) =< \emptyset(x), \emptyset(y) > \tag{11}$$

Feature vectors need not be computed explicitly, and in fact it greatly improves computational efficiency to directly compute kernel values $K(x,y)$. Three common kernels can be used in this experiments:

Linear kernel: $K(x,y) = (x.y)$

Polynomial kernel: $K(x,y) = (x.y + 1)^d$, where d is the degree of the polynomial

RBF or Gaussian kernel: $K(x,y) = e^{-||x-y||^2/2\sigma^2}$, where σ^2 is the variance.

6.5.2. Naïve Bayes

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem. Bayesian classifiers has exhibited high accuracy and speed when applied to large databases. Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class-conditional independence. It is made to simplify the computations involved and, in this sense, is considered "naïve" [33].

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

1. Let D be a training set of tuples and their associated class labels. Each tuple is represented by an n-dimensional attribute vector, $\boldsymbol{X} = x_1, x_2, \dots, x_n$, depicting n measurements made on the tuple from n attributes, respectively, $A_1, A_2, \dots, A_n$.
2. Suppose that there are m classes, $C_1, C_1, \dots, C_n$. Given a tuple, $\boldsymbol{X}$, the classifier will predict that $\boldsymbol{X}$ belongs to the class having the highest posterior probability, conditioned on $\boldsymbol{X}$. That is, the naïve Bayesian classifier predicts that tuple $\boldsymbol{X}$ belongs to the class C_i if and only if

$$P(C_i|\boldsymbol{X}) > P(C_j|\boldsymbol{X})\; for\; 1 \leq j \leq m, j \neq i \tag{12}$$

Thus, we maximize $P(C_i|\boldsymbol{X})$. The class C_i for which $P(C_i|\boldsymbol{X})$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})} \qquad 13$$

3. As $P(\boldsymbol{X})$ is constant for all classes, only $P(\boldsymbol{X}|C_i)P(C_i)$ needs to be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2)$ =…= $P(C_m)$, and it is therefore required to maximize $P(\boldsymbol{X}|C_i)$. Otherwise, maximize $P(\boldsymbol{X}|C_i)P(C_i)$.
4. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(\boldsymbol{X}|C_i)$. To reduce computation in evaluating $P(\boldsymbol{X}|C_i)$, the naïve assumption of class-conditional independence is made. This presumes that the attributes' values are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$P(\boldsymbol{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \qquad 14$$

5. To predict the class label of ***X***, $P(\boldsymbol{X}|C_i)P(C_i)$ is evaluated for each class C_i. The classifier predicts that the class label of tuple ***X*** is the class C_i if and only if

$$P(\boldsymbol{X}|C_i)P(C_i) > P\left(\boldsymbol{X}|C_j\right)P(C_i)\, for\ 1 \leq j \leq m, j \neq i \qquad 15$$

In other words, the predicted class label is the class C_i for which $P(\boldsymbol{X}|C_i)P(C_i)$ is the maximum.

Various empirical studies of this classifier in comparison to decision tree and neural network classifiers have found it to be comparable in some domains. In theory, Bayesian classifiers have the minimum error rate in comparison to all other classifiers. However, in practice this is not always the case, owing to inaccuracies in the assumptions made for its use, such as class-conditional independence, and the lack of available probability data.

Bayesian classifiers are also useful in that they provide a theoretical justification for other classifiers that do not explicitly use Bayes' theorem [33].

6.5.3. Proposed Hybrid Algorithm:

A proposed algorithm is based on hybrid model built using SVM and Naïve Bayes. Specific weights are assigned to each classifier. If all the stand-alone classifier developed for a given response happens to have the same level of accuracy, then an acceptable form for the combination would be a simple average of the models. However, this is not generally the case since some classifier tend to be more accurate than others. Hence, in attempting to enhance the accuracy of the ensemble, the stand-alone models (members of the ensemble) have to be multiplied by different weight factors. Using a weighted sum formulation an ensemble of models for approximation of response $y(\boldsymbol{x})$ is expressed as [34]:

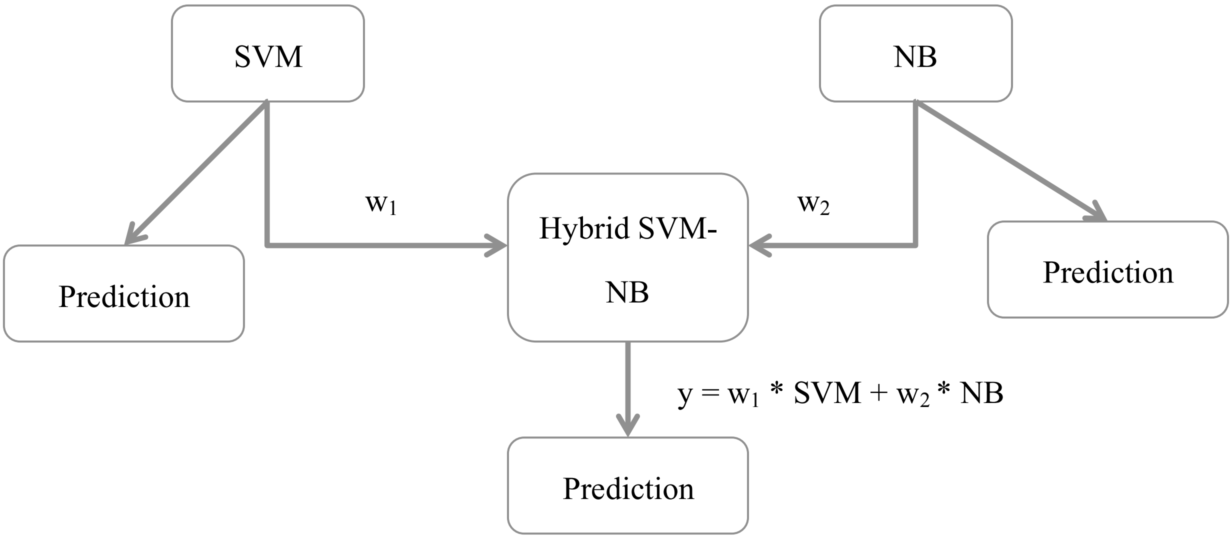


Figure 6.3: Hybrid Model using Weighted SVM and NB

$$y_h(\boldsymbol{x}) = \sum_{i=1}^{M} w_i(\boldsymbol{x})\, y_i(\boldsymbol{x}) \qquad 16$$

where $y_h(\boldsymbol{x})$ is the hybrid model predicted response, M is the number of classifiers in the hybrid model, w_i is the weight factor for the ith classifier, y_i is the response estimated by the ith classifier, and $\boldsymbol{x}$ is the vector of independent input variables.

To calculate weights of weighted sum model of different classifier (SVM and NB) for the optimization, the prediction variance is chosen as the error metric, and set the value of weight factor for each classifier to be inversely proportional to the point-wise estimate of the prediction variance as [35]

$$w_i = \frac{1}{V_i} \Big/ \sum_{j=1}^{M} \frac{1}{V_j} \qquad 17$$

where V_j is the prediction variance of the ith classifier. The selection of weights via equation 17 minimizes the prediction variance of the weighted sum model based on the assumption that the meta classifier predictions are unbiased and uncorrelated [34].

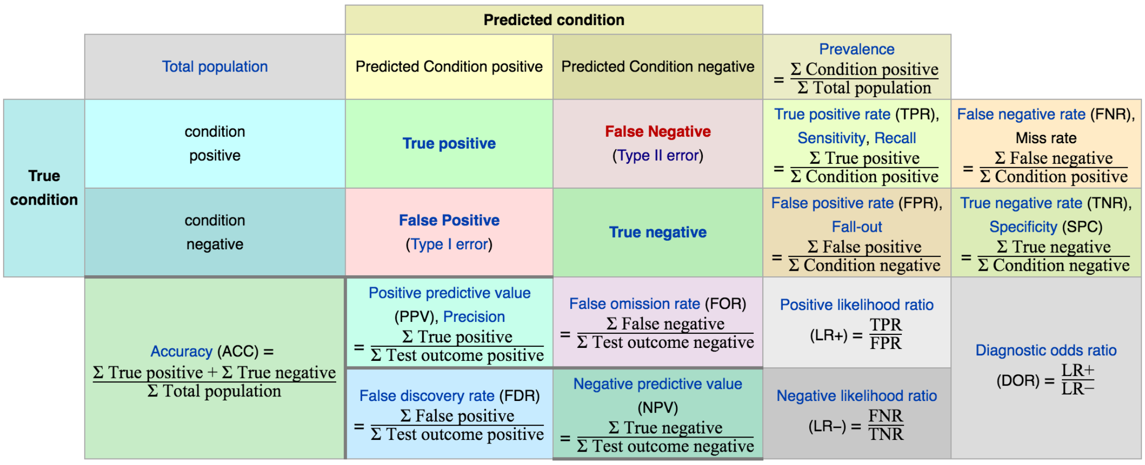
6.6. Validation

To estimate the area under curve (AUC) performance of a two-class classifier, a technique called cross validation is employed. In *k-fold cross-validation*, the initial data are randomly partitioned into *k* mutually exclusive subsets or “folds,” $D1, D2, \ldots, D_k$, each of approximately equal size. Training and testing is performed *k* times. In iteration *i*, partition D_i is reserved as the test set, and the remaining partitions are collectively used to train the model. That is, in the first iteration, subsets $D2, \ldots, D_k$ collectively serve as the training set to obtain a first model, which is tested on $D1$; the second iteration is trained on subsets $D1$, $D3$,..., D_k and tested on $D2$; and so on. Unlike the holdout and random subsampling methods, here each sample is used the same number of times for training and once for testing. For classification, the accuracy estimate is the overall number of correct classifications from the *k* iterations, divided by the total number of tuples in the initial data.

It is ensured that the cross-validation is stratified, which means each fold contains both normalities and anomalies with the same proportions as the original data set [36]. In stratified cross-validation, the folds are stratified so that the class distribution of the tuples in each fold is approximately the same as that in the initial data. The one-class classifier is trained and tested k times. Each time $t \epsilon 1,2,...,k$ the one-class classifier is trained on *D/D*$_t$ and tested on *D*$_t$. This results in k AUC performances.

7. Experiment and Results

For each user behavior, distinction between two classes, namely *normal* and *outlier* is made. In the process, half of the total data is split and used to build the user profile. Thus built user profile is considered as the normal behavior of the user and remaining datasets are predicted based on this user profile. In this hybrid approach, application of the Support Vector Machine, Naïve Bayes and ensemble of two algorithms is applied. To evaluate the proposed models ROC curve and accuracy measure is carried out. For the validation of the predicted datasets and to measure the accuracy k-fold cross validation is used, where k is an integer. In this this, 10 is used as a value for k. The samples or test datasets are randomly sampled into 10 subsamples. Out of these 10 subsamples, 1 sample used for testing, remaining 9 subsamples used for training and this is carried out for each combination of subsamples. The purpose of doing this is that, every subsample is used for both the training and testing.



		Predicted condition			
	Total population	Predicted Condition positive	Predicted Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
True condition	condition positive	**True positive**	**False Negative** (Type II error)	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$
	condition negative	**False Positive** (Type I error)	**True negative**	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$
	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	

Figure 7.1: Receiver Operating Characteristic Details

Source: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

The result of all these experiments is presented in the form of Receiver Operating Characteristics (ROC) curve. A ROC curve provides the possibility to observe, how the change of a free parameter influences the performance of the system. It represents the relation between the true positive rate and the false positive rate of a retrieval algorithm for each parameter value in one plot [37].

OCSVM was used to learn from the data i.e. build a normal user profile and use multiple algorithm to test the accuracy. Algorithms used in this thesis are as OCSVM, RBFSVM, Naïve Bayes (NB) and Hybrid model built from SVM and NB with weight calculated by brute force method.

7.1. Experimentation Approach

Windows Security log data collected and examined during the course of this thesis were from desktop computers running Windows Server 2008. Similar feature data were collected from two different organizations. Datasets from one organization is used to for analysis of different parameters and this was set as a benchmark. Second dataset was tested against the first datasets for proof of concept. Two level of experiment is carried out in this thesis.

7.1.1. Classification Enhancement Approach

In this experiment predefined feature set i.e. features defined in section 6.4. are used to train and test the dataset. SVM, NB and hybrid algorithm is used to classify the datasets and compare the accuracy to classification. This experiment is also known as classification enhancement approach.

7.1.2. User Profile Enhancement Approach

In this experiment, new sets of feature that describes the user is derived from existing features. The added features are next logon duration and the total office hour duration. Next logon duration is the duration between user logoff and next successful logon while total office hour duration is the duration of user between first successful login and last logoff. After addition of new features to a dataset, experiment similar to experiment – I is carried out. This experiment is also known as user profile enhancement approach.

7.2. σ Calculation for Support Vector Classifier

Initial result showed that the Naïve Bayes is accurate as compare to support vector classifier method used. Figure 7.1 shows the accuracy comparison between support vector machine and naïve bayes classifier.

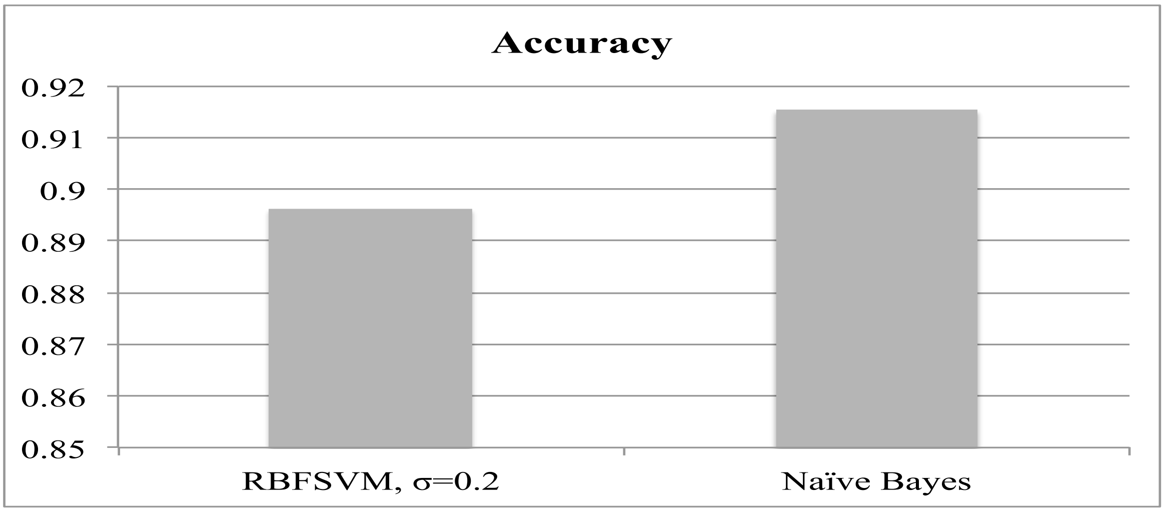


Figure 7.2: Accuracy Comparison between SVM and Naive Bayes Classifier

However, the classification performance of SVM is affected by its model parameters. For RBFSVM, the model parameters include the Gaussian width, σ, and the regularization parameter, *C*. The variation of either of them leads to the change of classification performance. It is to be noted that, although RBFSVM cannot learn well when a very low value of *C* is used, its performance largely depends on the σ value if a roughly suitable C is given [38]. Clearly, changing σ leads to larger variation on test error than changing *C*. This means that over a large range of *C,* the performance of RBFSVM can be adjusted by simply changing the value of σ [39]. It is known that, in a certain range, a large σ often leads to a reduction in classifier complexity but at the same lowers the classification performance. Also, a smaller σ often increases the learning complexity and leads to higher classification performance in general. A moderately accurate RBFSVM component classifiers is obtained by adaptively adjusting their σ values as shown in figure 7.2.

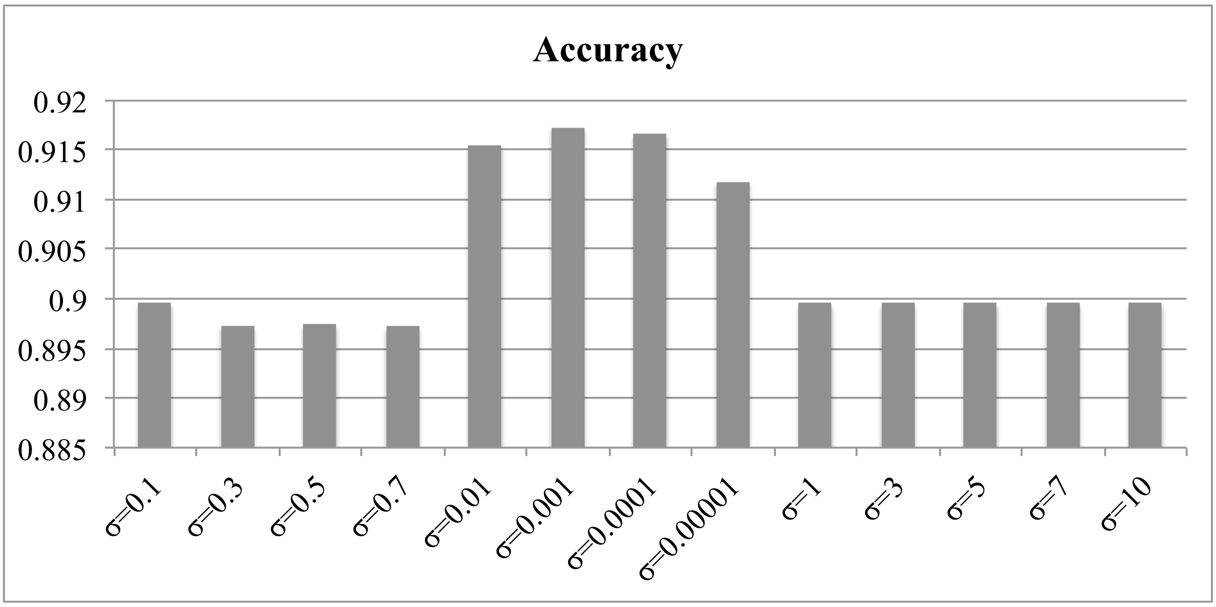


Figure 7.3: RBFSVM Accuracy Test with Varying σ

Figure 7.2 shows that the accuracy of SVM can be tuned by tuning the value of sigma, σ.

Accuracy

This refers to the ability of the classifiers to correctly measure the intrusions from the training dataset. This is defined as the ratio of correctly classified data to the total classified data.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{18}$$

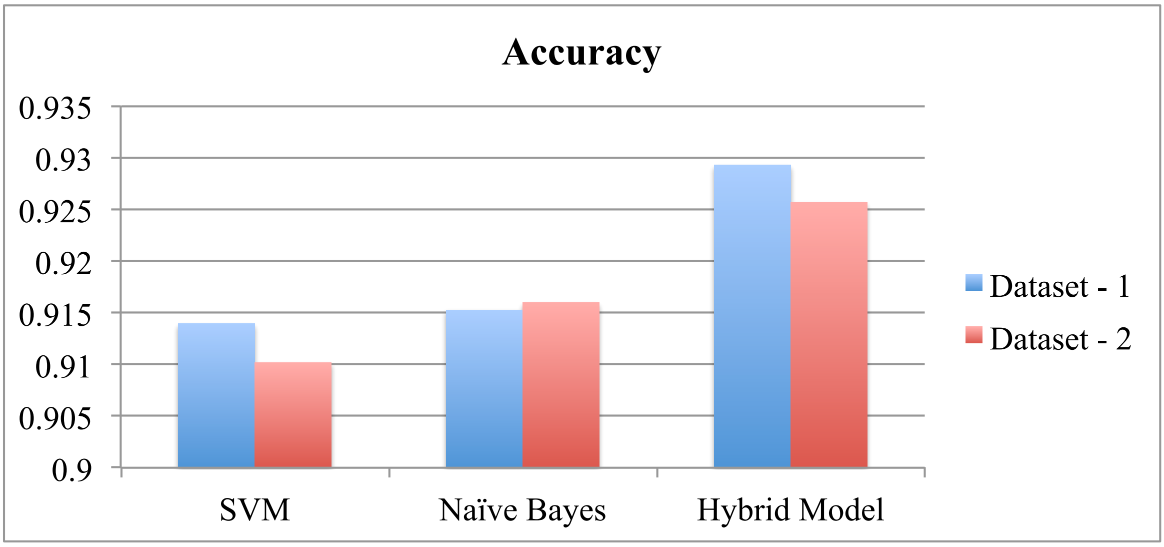


Figure 7.4: Comparison of Accuracy of Different Classifier with Hybrid Model

Experimentation result in figure 7.3 shows that the hybrid classifier is more accurate as compare to other classification techniques.

In above experimentation, the result shows the average performance of 10-Cross validation. These models are compared on the basis of each individual fold or rounds. To measure the robustness and effectiveness of any model, comparison of different parameters like Precision, Recall, F1-Score and ROC curve is computed and the performance of different models on the above parameters is evaluated.

Precision

It is also known as Positive Predictive Value (PPV). It measures the relevant instance that is retrieved after classification. A classifier that has high precision means that classifiers or algorithm returns more relevant results.

$$Precision = \frac{TP}{TP + FP} \qquad 19$$

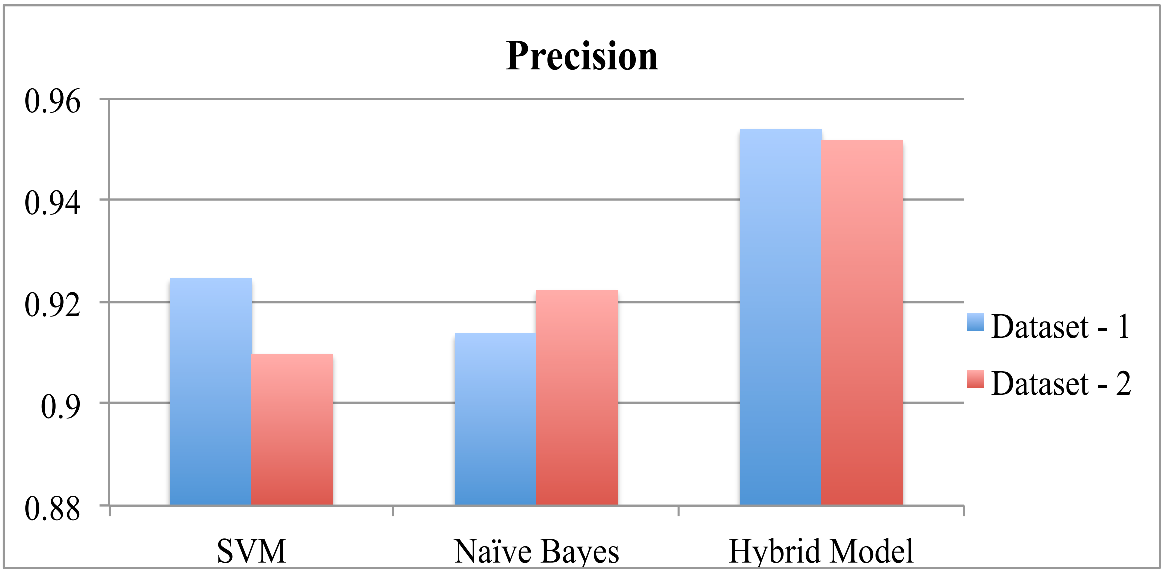


Figure 7.5: Comparison of Precision of Different Classifier with Hybrid Model

As shown in the figure 7.4, precision ratio of the hybrid classifier is high as compared to other models. It proved that hybrid classifier provides the less relevant results.

Recall/Sensitivity/Detection Ratio

It is defined as the ratio of detecting attacks to total no of attacks. This is the best parameter to measure the performance of the model. It is also known as sensitivity. This is also used to measure the relevant instance that is selected. The higher value of recall more the relevant data is selected for classification. It is defined as:

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

The figure 7.5 shows that the hybrid classifier is having a high recall or sensitivity. Hence, the most relevant data is selected as compared to other classifier.

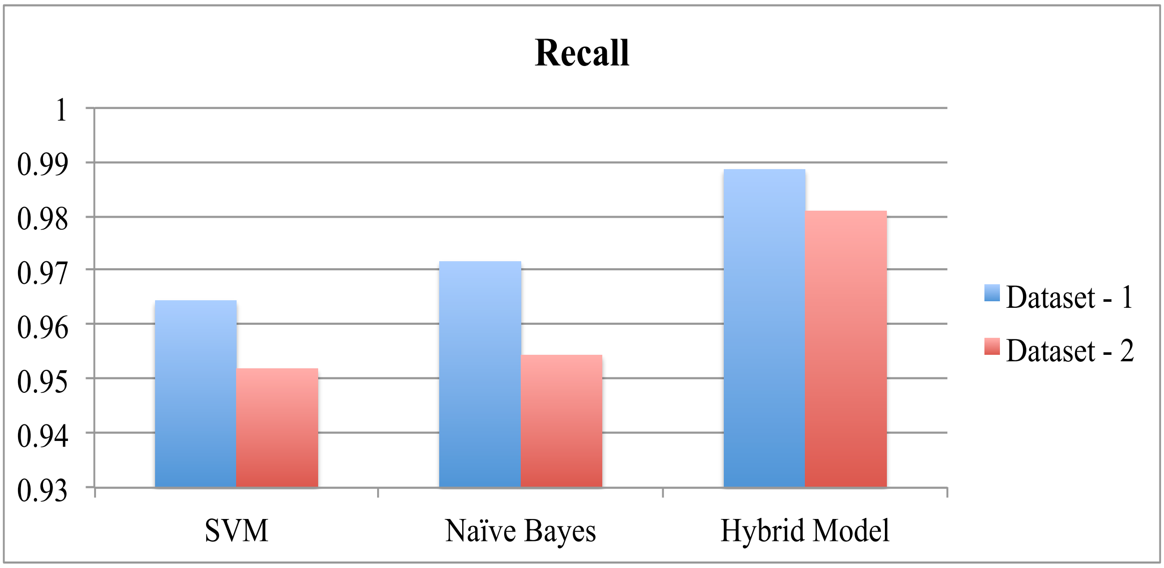


Figure 7.6: Comparison of Recall of Different Classifier with Hybrid Model

F1 – Score

It is basically used to measure the effectiveness of the classifiers. This is harmonic mean of precision and recall. It is also known as traditional F-measure or balanced F-score. It is defined as:

$$F1 - Score = 2\frac{Precision * Recall}{Precision + Recall} \tag{21}$$

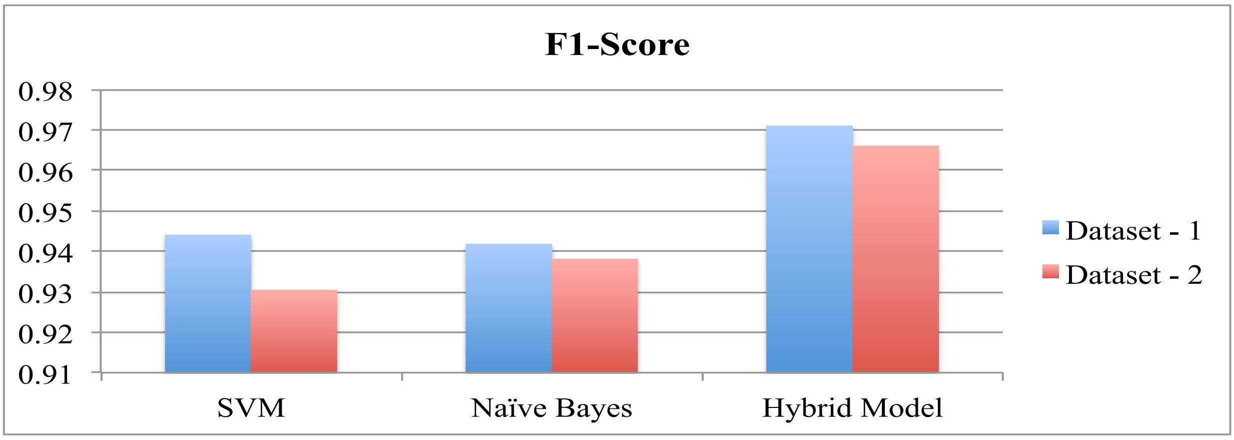


Figure 7.7: Comparison of F1-Score of Different Classifier with Hybrid Model

Figure 7.6 shows that the proposed model shows better result. Hybrid classifier technique performs much better in all aspects of the evaluation parameter of anomaly detection.

Area Under ROC Curve

It defined the correctness of the classifier that how a normal or abnormal dataset is separated by using training dataset. More the area under the ROC curve the more accurate the classifier is.

ROC Curve Dataset – I

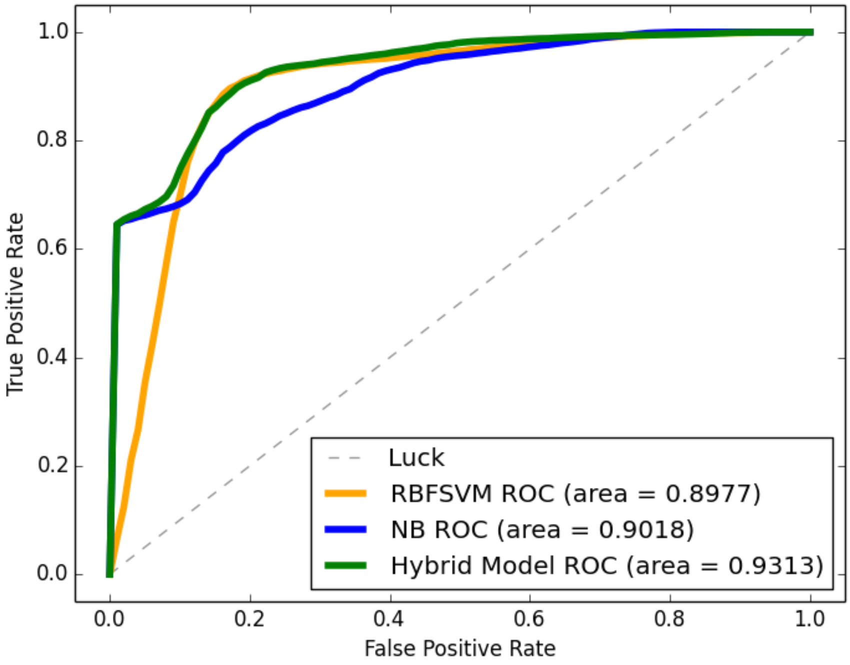


Figure 7.8: ROC Curve Comparison of SVM and NB with Hybrid Model: Dataset – I

Figure 7.8 and 7.9 shows the ROC curve comparison of SVM and NB with proposed hybrid model for dataset - I and dataset – II respectively.

ROC Curve Dataset - II

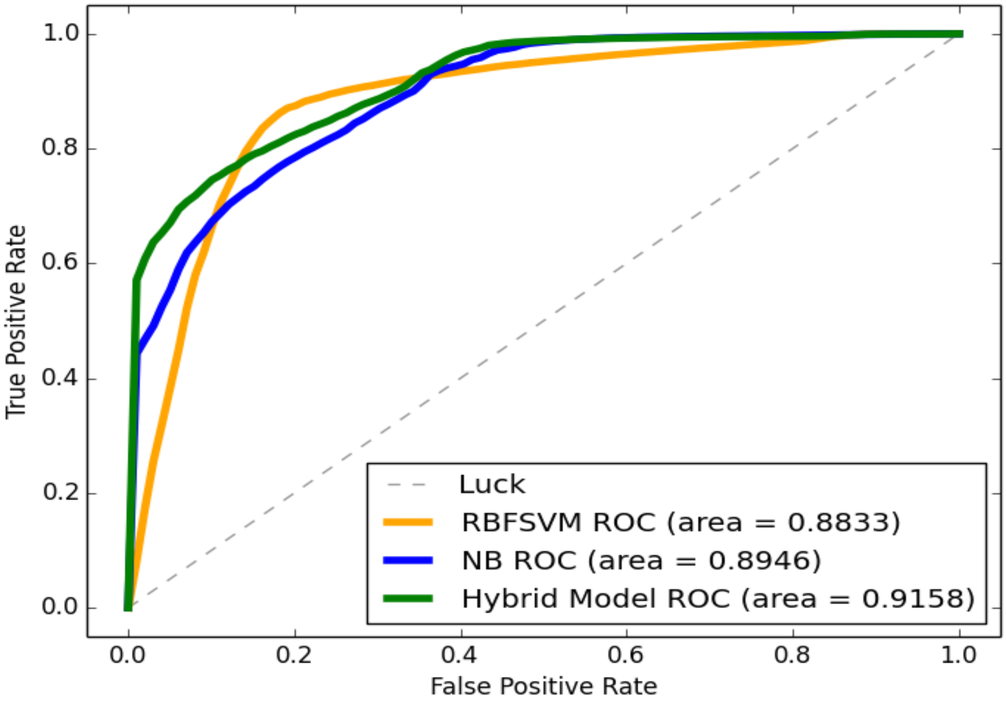


Figure 7.9: ROC Curve Comparison of SVM and NB with Hybrid Model: Dataset – I

From all the above experimentation results, it is shown that after applying all the evaluation parameters, hybrid model is found to be the best model in all scenarios. By applying the hybrid approach of on two datasets, the detection rate is improved for anomaly detection. So the main objective to improve the detection rate in anomaly detection has been met.

Further analysis is carried out to enhance user profile. In this experiment two more feature sets were derived from existing datasets namely, next logon and total duration. Next logon is defined as the duration of first logoff to the next login. Total duration is the duration of the sessions during working hour i.e. duration of first successful login to last successful logoff. The analysis is carried out the the result is presented in the form of accuracy plot and ROC curve.

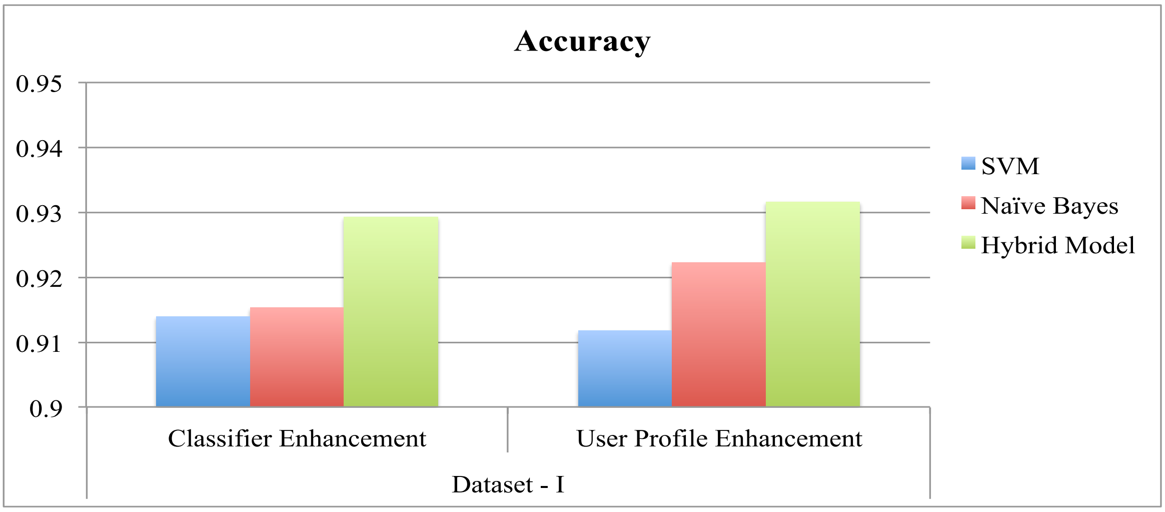


Figure 7.10: Accuracy Comparison of Classifier and User Profile Enhancement Technique

Figure 7.10 shows the comparison of accuracy of classification for two different approach. Classifier enhancement is the approach followed to increase accuracy using hybrid model while user profile enhancement is the approach followed to enhance user profile by addition of new feature set derived from existing dataset. The graph shows that the accuracy of classifier increases when important feature set that defines user profile is added. Figure 7.11 shows the ROC curve for user profile enhancement technique.

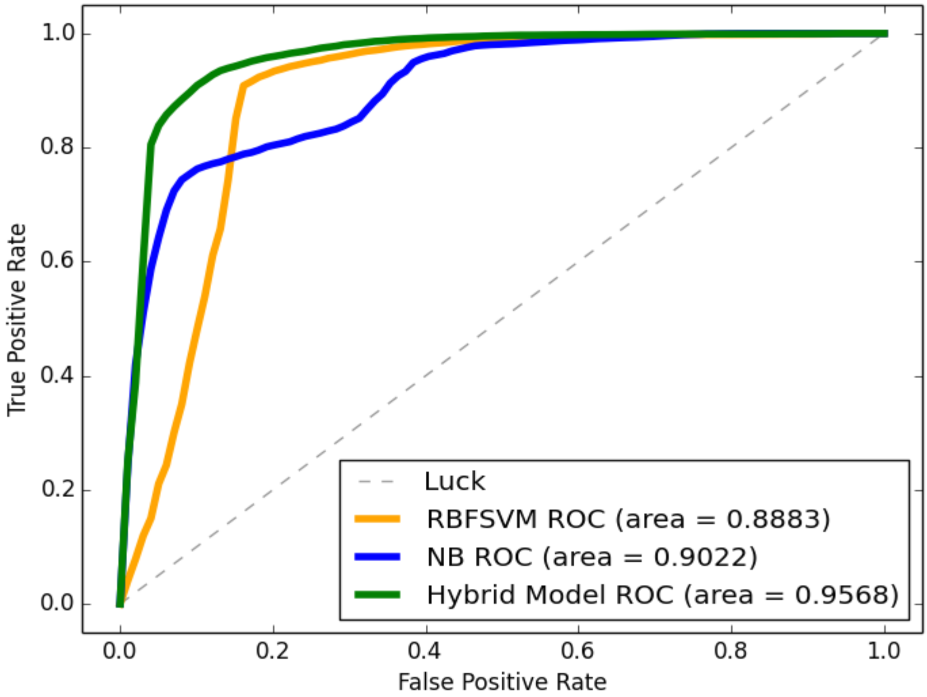


Figure 7.11: ROC Curve using User Profile Enhancement Technique Dataset - I

8. Conclusion

With proliferation of new technologies prevention of security breaches by the use of existing security technologies is simply unrealistic. As a result of this anomaly detection is an important component in security. To improve the accuracy rate of intrusion detection in anomaly based detection different data mining and machine learning technique is used. In this thesis a hybrid approach is implemented which is an amalgam of two different techniques namely support vector machine and naïve bayes. The results of the hybrid approach are compared with the results of other techniques and it is seen that hybrid approach outperforms them. Also, a method is proposed to enhance user profile by addition of new feature set derived from the existing dataset. The result shows, if new added feature properly defines the user then accuracy and detection ratio of the anomalous activity can be detected with high accuracy.

The new approach is effective during detection of attacks. The detection ratio of the hybrid algorithm is better than other techniques. The hybrid approach properly classifies the data either as normal or abnormal. Accuracy is also improved. It can be concluded that this hybrid approach is simple and efficient in terms of reducing the false alarm ratio.

9. Future Work

There are many possibilities to exploit and extend the learning approach used in this thesis. For example, this thesis includes five parameters to build a normal user profile. This could be extended to include more parameters that explain user profile like duration between logoff and next logon, number of process with their duration of use, name and types of application used, CPU and memory usage etc. Tracking types of application (like system tuning, programming tool, file sharing, messaging, networking, office package etc.) can help determine specific duties each user is assigned. A sudden departure from routine may be an indicator that the user is not carrying out their routine duties and helps to find out anomalous users. Also, tracking the exact time of when different processes and logon sessions were initiated could further enhance user profiling and extends the learning approach used in this thesis.

Also, different approach to user profile can be carried out. This thesis has included static approach to user profiling. With this approach detection of alerts is carried out on a after the training period and the user profile remains the same for each week of testing. Alerts are always generated based on that initial user profile. This approach is likely to generate many alerts as the person's usage changes due to their role changing within their organization or when software updates are applied. Two more approaches can be included e.g. growing window and sliding window. Growing window, where the profile training time period is continually extended by adding events to the user profile from the testing period. And sliding window, where the width of the time window remains constant. After training the user profile and anomaly detection based on the data from the testing period, the user profile is recalculated by removing the oldest time profile data and adding the new time period data that has had any events causing false alerts.

Moreover, hybrid model can be further tuned to improve accuracy with amalgamation of more classifier. Also, boosting algorithm can be used to boost the classifier, which could then be ensemble to increase the accuracy of the classifier.

10. References

[1] M. Peter S. Karen, "Intrusion Detection and Prevention Systems," in *Handbook of Information and Communication Security*, Peter Stavroulakis and Mark Stamp, Ed. New York, USA: Springer, 2010, pp. 177-192.

[2] A. Banerjee, V. Kumar V. Chandola, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, p. 58, July 2009.

[3] B. Priya, "A Hybrid Approach to improve the Anomaly Detection Rate Using Data Mining Techniques," COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, Thaper University, PATIALA, Msc Thesis 147004, 2015.

[4] D. Kom. (2008, April) Five Major Types of IDS. Blog.

[5] R. Kemmerer, and P. Porras K. Ilgun, "State transition analysis: A Rule-Based Intrusion Detection Approach," in *IEEE*, vol. 21, USA, 1995, pp. 181-199.

[6] S. Kumar, "Classification and Detection of Computer Intrusions," Department of Computer Science, Purdue University, -, PhD Dissertation -, 1995.

[7] V.V.R. Prasad, K.M. Prasad V. Jyothsna, "A Review of Anomaly based Intrusion Detection Systems," *International Journal of Computer Applications (0975 – 8887)*, vol. 28, no. 7, pp. 26-35, August 2011.

[8] M. Roesch, "Snort- lightweight intrusion detection for networks," in *Proceedings of LISA '99: 13th Systems Administration Conference*, Seattle, Washington, USA, November 1999, pp. 229-238.

[9] V. Paxon, "Bro: A system for detecting network intruders in real-time," in *In Proceedings of the 7-th USENIX Security Symposium*, San Antonio, Texas, 1998, p. 22.

[10] N. A. Durgin and P. Zhang, "Profile-Based Adaptive Anomaly Detection for Network Security," Sandia National Laboratories, Livermore, California, SANDIA REPORT SAND2005-7293, 2005.

[11] B. Khosravifar and J. Bentahar, "An experience improving intrusion detection systems false alarm ratio by using honeypot," in *22nd International Conference on Advanced Information Networking and Applications, AINA*, USA, 2008, pp. 997-1004.

[12] G. Mohay and A. Clark M. Corney, "Detection of Anomalies from User Profiles Generated from System Logs," in *9th Australasian Information Security Conference (AISC 2011),* , vol. 116, Perth, Australia, January 2011, pp. 23-32.

[13] S. h. Paek et al., "The Architecture of Host-based Intrusion Detection Model Generation System for the Frequency Per System Call," *IEEE*, vol. 2, no. -, pp. 277 - 283, Nov 2006.

[14] J. P. Anderson, "Computer security threat monitoring and surveillance," James P. Anderson Company, Pennsylvania, Technical report, -, 1980.

[15] M. S. Svore, and D.K. Angelos A.H. Katherine, "One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses," in -, Amsterdam, 2010, pp. -.

[16] M. Zhang, "An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions," *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, vol. -, no. -, pp. 102 - 107, December 2015.

[17] R. Chitrakar et al., "Anomaly detection using Support Vector Machine classification with k-Medoids clustering," *Internet (AH-ICI), 2012 Third Asian Himalayas International Conference on*, vol. -, no. -, pp. 1 - 5, November 2012.

[18] S. Benferhat, and Z. Elouedi N.B. Amor, "Naive bayes vs decision trees in intrusion detection systems," in *In Proceedings symposium on Applied computing of the ACM*, ACM, 2004, pp. 420-424.

[19] T. Singh, and A. Sinhai R. Jain, "A Survey on Network Attacks, Classification and models for Anomaly-based network intrusion detection systems," *International Journel of Engineering Research and Science & Technology*, vol. 2, no. 4, pp. 63-74, November 2013.

[20] S.S. Murtaza et al., "A host-based anomaly detection approach by representing system calls as states of kernel modules," *Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on*, vol. -, no. -, pp. 431-440, November 2013.

[21] C. Manikopoulos and S. Papavassiliou, "Network intrusion and fault detection: a statistical anomaly approach," in *IEEE Communications Magazine*, vol. 40, USA, 2002, pp. 76-82.

[22] X. Yingchao et al., "Parameter Selection of Gaussian Kernel for One-Class SVM," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 941 - 953, April 2015.

[23] D. P. Gaikwad, "Intrusion Detection System Using Bagging Ensemble Method of Machine Learning," *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference*, vol. -, no. -, pp. 291 - 295, February 2015.

[24] K. Hatonen, A. S. Sorvari A. J. Hoglund, "A computer host-based user anomaly detection system using the self-organizing map," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference*, vol. 5, Como, 2000, pp. 411 - 416.

[25] I. S. Thaseen, "Intrusion detection model using fusion of PCA and optimized SVM," in *Contemporary Computing and Informatics (IC3I), 2014 International Conference*, Mysore, 2014, pp. 27-29.

[26] L. Lin et al., "SVM ensemble for anomaly detection based on rotation forest," *Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference*, vol. -, no. -, pp. 150 - 153, July 2012.

[27] S. Peddabachigari et al., "Modeling Intrusion Detection System using Hybrid Intelligent Systems," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 114-132, 2007.

[28] R. C. Thool D. P. Gaikwad, "Intrusion Detection System Using Bagging Ensemble Method of Machine Learning," in *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference*, Pune, 2015, pp. 291 - 295.

[29] S. Karthik and S. Sivakumari P. Amudha, "« Prev | Back to Results | Next » Intrusion detection based on Core Vector Machine and ensemble classification methods," in *Soft-Computing and Networks Security (ICSNS), 2015 International Conference*, Coimbatore, 2015, pp. 1 - 5.

[30] P. Sornsuwit and S. Jaiyen, "Intrusion detection model based on ensemble learning for U2R and R2L attacks," in *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Chiang Mai, 2015, pp. 354 - 359.

[31] R. Zuo, S. Yang, Z. Zhang L. Lin, "SVM ensemble for anomaly detection based on rotation forest," in *Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference*, Dalian, 2012, pp. 150 - 153.

[32] D. Ragsdale, and J. Surdu J. Marin, "A Hybrid Approach to the Profile Creation and Intrusion Detection," in *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01*, vol. 1, Anaheim, CA, 2001, pp. 69-76.

[33] K. Micheline and P. Jian H. Jiawei, *Data Mining - Concepts and Techniques*, 3rd ed., -, Ed. Waltham, USA: Elsevier Inc., 2012.

[34] E. and Rais-Rohani, M. Acar, "Ensemble of Metamodels with Optimized Weight Factors," in *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials*, vol. 37, Schaumburg, 2008, pp. 279-294.

[35] L., Queipo, N.V., Pintos, S., Salager, J. Zerpa, "An optimization methodology of Alkaline-Surfactant-Polymer flooding processes using field scale numerical simulation and multiple surrogates," in *Journal of Petroleum Science and Engineering*, vol. 47, Chicago, 2005, pp. 197-208.

[36] J. H.M. Janssens, "Outlier Selection and One-Class Classification," Tilburg University, -, Thesis -, 2013.

[37] N. Stefanie, D. Peter L. Hanna, "USING ONE-CLASS SVM OUTLIERS DETECTION FOR VERIFICATION OF COLLABORATIVELY TAGGED IMAGE TRAINING SETS," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, New York, NY, Mar 2009, pp. 682 - 685.

[38] G. Valentini and T. G. Dietterich, "Bias-Variance Analysis of Support Vector Machines for the Development of SVM-based Ensemble Methods," in *Journal of*

Machine Learning Research 5, Chicago, 2004, pp. 725-775.

[39] Lei Wang, Eric Sung Xuchun Li□, "AdaBoost with SVM-based Component Classifiers," in *Engineering Applications of Artificial Intelligence 21*, 2008, pp. 785–795.

[40] K. Tan and R.A. Maxion S. Jha, "Markov Chains, Classifiers, and Intrusion Detection," *Computer Security Foundations Workshop*, no. 1063-6900, pp. 206-219, July 2001. [Online]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.1600&rep=rep1&type=pdf

[41] S. B Cho and H. J. Park, "Efficient anomaly detection by modeling privilege flows using hidden Markov model," *Elsevier Computers & Security*, vol. 22, no. 1, pp. 45-55, August 2003.

[42] J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson B. Scholkopf, "Estimating the support of a high-dimensional distribution," *Neural Computation*, pp. 1443–1471, 2001.