8-2007

Differentiated Intrusion Detection and SVDD-based Feature Selection for Anomaly Detection

Inho Kang

University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Inho Kang entitled "Differentiated Intrusion Detection and SVDD-based Feature Selection for Anomaly Detection." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Dongjoon Kong, Myong K. Jeong, Major Professor

We have read this dissertation and recommend its acceptance:

Xueping Li, Frank Guess

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Inho Kang entitled "Differentiated Intrusion Detection and SVDD-based Feature Selection for Anomaly Detection." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Dongjoon Kong
Major Professor

Myong K. Jeong
Co-Advisor

We have read this dissertation
and recommend its acceptance:

Xueping Li

Frank Guess

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and
Dean of the Graduate School

(Original signatures are on file with official student records.)

Differentiated Intrusion Detection and SVDD-based Feature Selection for Anomaly Detection

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Inho Kang
August 2007

ACKNOWLEDGEMENTS

I would like to give all praises to the God for directing, teaching, and always embracing me with great and endless love.

I am deeply indebted to my advisors Prof. Drs. D. J. Kong and M. K. Jeong for their consistent supports, stimulating suggestions, and thoughtful directions during the doctoral program. I am also grateful to Prof. Dr. F. Guess for his special encouragement and Prof. Dr. X. Li for his assistance. I have furthermore to thank all the professors and teachers for teaching and inspiring me.

My supervisors and colleagues in the Korea Institute for Defense Analyses supported me to pursue my doctoral degree. I would like to thank them for all their invaluable supports and concerns throughout the program. Especially I am grateful to Dr. J. K. Noh, Dr. N. S. Han, and Dr. S. B. Choi for their help.

Finally, I would like to thank my family for their unconditional supports and earnest prayers. I am obliged to my parent and parent-in-law, my grace wife Yongsook, and my lovely daughters, Yunjin and Jieun for their patient love. Also I gratefully acknowledge the support of all my friends and fellow students.

Inho Kang

June 2007

Abstract

Most of existing intrusion detection techniques treat all types of attacks equally without any differentiation of the risk they pose to the information system. However, certain types of attacks are more harmful than others and their detection is critical to protection of the system. This study proposes a novel differentiated anomaly detection method that can more precisely detect intrusions of specific types of attacks.

Although many researchers have been developed many efficient intrusion detection methods, fewer efforts have been made to extract effective features for host-based intrusion detection. In this study, we propose a new framework based on new viewpoints about system activities to extract host-based features, which can guide further exploration for new features.

There are few feature selection methods for anomaly detections although lots of studies have been done for the feature selection both in classification and regression problems. This study proposes new support vector data description (SVDD)-based feature selection methods such as SVDD-R2-recursive feature elimination (RFE), SVDD-RFE and SVDD-Gradient method. Concrete experiments with both simulated and the Defense advanced research projects agency (DARPA) datasets shows promising performance of the proposed methods.

These achievements in this dissertation could significantly contribute to anomaly detection field. In addition, the proposed differentiated detection and SVDD-based feature selection methods would benefit even other application areas beyond intrusion detection

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

This chapter provides an introduction to the research. Section 1.1 presents the motivation for the research. The contributions of the research are presented in Section 1.2. The organization of the rest of this dissertation is outlined in Section 1.3.

1.1 Motivation

As Internet and computer networks play an increasingly vital role in modern society, intrusions into information systems have become a significant threat to our society with potentially severe consequences. To protect information systems from external attackers and disgruntled employees, effective and efficient intrusion detection techniques are required. As one of defense layers, intrusion detection has been widely studied and operated. However, there is still enough room to improve the performance of intrusion detection system (IDS) toward perfect detection accuracy and zero false alarm rate. Since a typical intrusion detection system first gathers information from a computer of interest and attempts to detect intrusions based on the information, more effective information and more accurate detection technique are required for better intrusion detection.

Compared to many researches on detection technique of IDS, fewer studies on host-based features have been carried out although features as input for IDS are as important as the techniques. Due to poor interest on feature development, existing host-based features are not diverse and are based on only system event type. Since a feature represents one viewpoint for user behavior in the information system, more features can help IDS produce more reliable and accurate results. Moreover, immensely increased computing power has made it significantly easier task than the before that IDS processes high-dimensional data. Therefore, we need to explore new viewpoints about system activities and widen searching range for new host-based features in order to get as many effective features as possible.

There exists a type of attack which causes more severe consequence than other attack types when it penetrates the defense layers of information system. In response, the system administrator wants to more strictly detect this destructive attack among other attack types. However, existing anomaly intrusion detection techniques do not support this task since they treat all attacks with equal importance. To more precisely detect intrusions of specific attack type, a new approach is required to perform tighter detection on the type and ordinary detection on the other attack type.

Feature selection contributes cost and time reduction in obtaining and processing data by identifying, removing unnecessary features and selecting most predictive ones among whole features. There are few literatures on feature selection for the anomaly detection problem although feature selection has been deeply studied in the classification problem. The classification feature selection methods are not applicable directly to the anomaly

detection and there is no feature selection method solely dedicated to anomaly detection. Novel feature selection methods for anomaly detection techniques are required to take advantages of feature selection in the field of anomaly detection.

1.2 Contributions of the Dissertation

Based on the motivations in Section 1.1, the contributions of this dissertation are as follows:

1. A new approach is proposed to generate features for host-based intrusion detection system. The proposed approach has been applied to Defense Advanced Research Projects Agency (DARPA) and MIT Lincoln Lab (MITLL) 1998 BSM data set to extract features for anomaly intrusion detection.
2. A new differentiated intrusion detection method is developed to more precisely detect more harmful attack type to information system than ordinary attacks. Mathematical formulation has been derived for the developed method. Based on the formulation, a lemma has been drawn to underpin theoretical base of the differentiated detection.
3. Novel support vector data description(SVDD)-based feature selection methods such as SVDD-R2-RFE, SVDD-RFE and SVDD-Gradient are proposed to take advantages of feature selection in anomaly detection area. Mathematical formulations for criterion functions of both methods are developed in cases of kernel functions and executable algorithms are provided for the proposed methods.

1.3 Outlines of the Dissertation

The remainder of this dissertation is organized as follows:

Chapter 2 introduces intrusion detection systems on how to collect information and what detection techniques are used. Anomaly detection is presented in detail since it is related to this dissertation topic. In addition, recent literatures on system features and detection techniques are reviewed in Chapter 2.

In Chapter 3, the concept of new feature framework for host-based intrusion detection system is presented. Length, intensity, and event type related features are described under the proposed framework. The process of feature extraction from the DARPA 98 BSM dataset is also presented. Furthermore, the results of the experiment of new features from the DARPA are discussed.

A differentiated intrusion detection methodology is presented in Chapter 4. Introduction to SVDD, formulation for differentiated intrusion detection, and differentiated anomaly intrusion detection are also presented. The performance of the proposed method was examined with simulated data and the DARPA data. The results on the experiment are discussed.

Chapter 5 presents the motivation of SVDD-based feature selection and introduction to feature selection for anomaly detection. Formulations and algorithms for SVDD-R2-RFE,

SVDD-RFE and SVDD-Gradient feature selection methods are also presented. Experiments with the proposed methods and their results are discussed.

In Chapter 6, finally, conclusions and future research are presented.

Chapter 2 Intrusion Detection System

This chapter provides literature reviews for intrusion detection system. Section 2.1 presents a background for intrusion detection system. General introduction to intrusion detection system is described in Section 2.2. Section 2.3 presents the summary of recent research for intrusion detection system.

2.1 Background

We now live in the information age. It is nearly impossible to imagine our lives without the Internet and information systems. We increasingly rely on information systems in banking, stock trading, telecommunication, broadcasting, transportation, and many other systems which are operated on the computer networks. While the possibilities and opportunities afforded by computer information systems are steadily expanding, the risk of malicious intrusions such as computer viruses or the theft of data, also, is growing. Damage of information systems due to system attacks has been increasing. In 2002, companies lost roughly $20 billion to $30 billion from the virus attacks according to a ZDNet Security News Article dated January 2004. This figure went up from about $13

billion in 2001. In response for these destructive system attacks, financial services companies are spending about 6% of their IT budgets for security of their information system according to global security survey (Tohmatsu, 2003). Intrusions into information systems have become a significant threat to our society with potentially severe consequences.

An intrusion into an information system is defined as compromising its security such as availability, integrity and confidentiality through a series of events in the information system (Ye & Chen, 2001). In its broadest definition, a computer attack is any malicious activity directed at a computer system or the services it provides (Kendall, 1999). There are several types of intrusions as follows (Kendall, 1999):

- viruses,
- use of a system by an unauthorized individual,
- denial-of-service by exploitation of a bug or abuse of a feature,
- probing of a system to gather information, or
- a physical attack against computer hardware.

Also, there are categories for attack techniques as in table 2.1. For example of social engineering, an attacker can call an individual on the telephone impersonating a network administrator in an attempt to convince the individual to reveal confidential information including passwords, file names and details about security policies. Specific examples of implementation bugs are buffer overflows, race conditions, and mishandled

Table 2.1 Categories for attack techniques

Categories	Description
Social engineering	Gaining access to a system by fooling an authorized user into providing information that can be used to break into a system.
Implementation bug	Bugs in trusted programs can be exploited by an attacker to gain unauthorized access to a computer system.
Abuse of feature	Legitimate actions that one can perform that when taken to the extreme can lead to system failure.
System misconfiguration	An attacker can gain access because of an error in the configuration of a system.
Masquerading	In some cases it is possible to fool a system into giving access by misrepresenting oneself.

temporary files. Examples for abuse of feature include opening hundreds of telnet connections to a machine to fill its process table, or filling up a mail spool with junk e-mail. An example of system misconfiguration is that the default configuration of some systems includes a "guest" account that is not protected with a password. Masquerading example is sending a TCP packet that has a forged source address that makes the packet appear to come from a trusted host.

Defense measures are required to protect computers and networks from unauthorized use or malicious attack. Layered defense measures are generally used to reduce the possibility of intrusions as possible. Prevention is the first measure to be used. They are firewalls and guards, authentication, and encryption. The second measure is intrusion detection to identify intrusions being leaked through the fence of prevention. The last one is reaction to minimize damage due to intrusions penetrating the defense layers. Among

those defense measures intrusion detection has been attracting more attention as backup of not-robust prevention. Moreover it is said that intrusion detection has become an indispensable defense line in the information security infrastructure (Li et al., 2005).

2.2 General Introduction to Intrusion Detection System

Generally an intrusion detection system (IDS) detects a possible intrusion and notifies a system administrator of its presence (Kendall, 1999) as shown figure 2.1. An IDS consists of two functioning parts, information collection and decision. Information collection part is to gather data from a computer or network of computers of interest. It is important for this part to get more representative features well describing user's activities. Decision part is to attempt to detect an intrusion based on the obtained information. Main interest in this part is to develop more effective decision rule to reduce decision errors. Key elements for good IDS are to acquire representative features and to apply effective detection technique.

2.2.1 Information collection

There are two questions to be answered in order to find more representative features:

- What observable subjects should be selected for monitoring and analyzing user's behavior?
- What attributes should be considered for characterizing these related subjects?

Although there are many observable subjects, most intrusion detection systems in existence today use one or more of these three types of data such as sniffed network traffic, host-level audit files, and file-system state. The first subject is traffic sent over the network. All data that is transmitted over an Ethernet network is visible to any machine that is present on the local network segment. Because this data is visible to every machine on the network, one machine connected to this Ethernet can be used to monitor traffic for all the hosts on the network. Network traffic can be sniffed using a single machine running the tcpdump program to save the network traffic. The second object for an intrusion detection system is host-level audit data. Most operating systems offer some level of auditing of operating system events. An example is Basic Security Module (BSM) data from a Solaris operating system. The third object is information about file system state. Daily file system dumps is collected from each machine. An intrusion detection system that examines this file system data can alert an administrator whenever a system binary file such as the ps, login, or ls program is modified. Normal users have no legitimate reason to alter these files, so a change to a system binary file indicates that the system has been compromised. Usually network traffic data and host-level audit data are frequently used in IDS. Therefore an IDS is categorized into host IDS or network IDS by where its data is collected.

Attributes are data or a group of data describing observable subjects. Attributes for host-level audit data are command line strings, system call traces, and resource consumption patterns while attributes for network traffic data are intrinsic features, traffic features, and

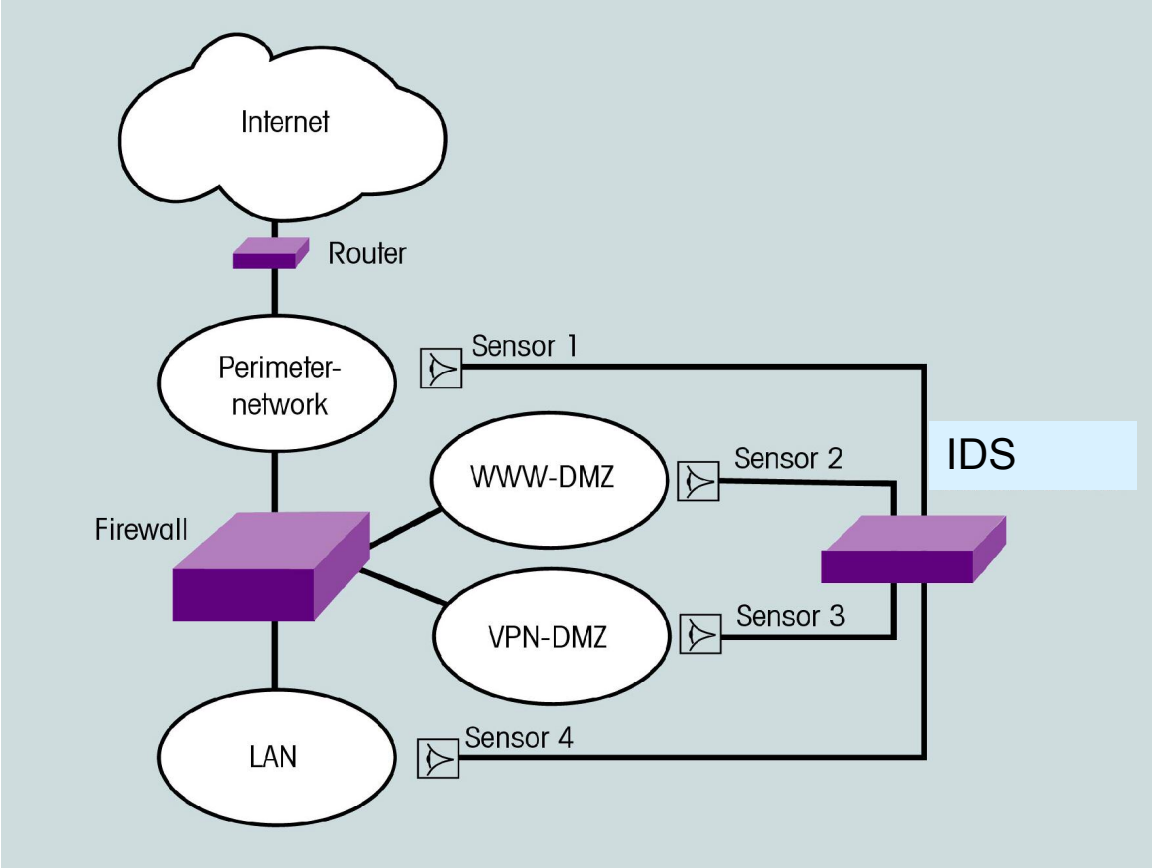


Figure 2.1 IDS's role in an information system
Source: www.genua.de/dateien/gd-installation-en.jpg

content features. Information collection is performed by making data or features from those attributes. Many researched are still seeking more practical and effective attributes.

2.2.2 Detection techniques

Various detection techniques have been applied into IDS. Those detection techniques are in figure 2.2. Signature recognition techniques can find known types of attack while bottleneck verification, specification-based detection and anomaly detection techniques can find new types of intrusion. Anomaly detection technique requires more computation efforts and memories since it is the most sophisticated among detection techniques.

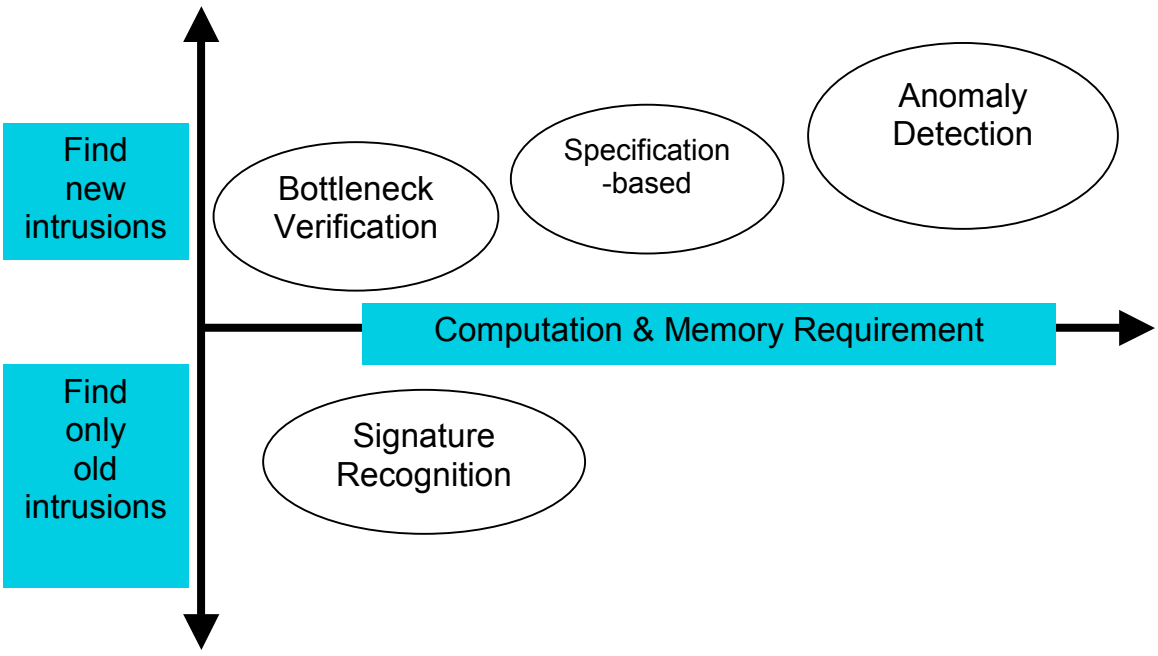


Figure 2.2 Detection Techniques for intrusion detection
Source: Kendall, 1999

Bottleneck verification technique detects illegal transitions between two groups of states. However, this technique applies to only situations where there are only a few well defined ways to transition between two groups of states. One example of such a well-defined transition is transitions from a normal user to super-user within a shell. If an individual is in the user state, the only way to legally gain root privileges is by using the su command and entering the root password. Thus, if a bottleneck verification system can detect a shell being launched, determine the permissions of the new shell, and detect the successful use of the su command to gain root access, then illegal transitions from normal user to root user can be detected (Kendall, 1999).

Specification-based detection technique detects behavior that violates the security specifications. Before monitoring user's activities, this approach requires written security specifications that describe the normal behavior of programs. Then host-based audit records are then monitored to detect behavior that violates the security specifications. However, there is a limitation to apply because writing security specifications for all monitored program which are constantly updated.

Signature recognition and anomaly detection techniques are popular since bottleneck verification specification-based detection techniques are applied only to specific cases. Currently existing intrusion detection techniques fall in two major categories: signature recognition and anomaly detection. Signature recognition technique looks for an invariant sequence of events that match a known type of attack. There are three steps: collection the signatures of known intrusion scenarios, matching the observed behavior with these intrusion signatures, and notifying signal an intrusion when there is a match (Ye & Chen,

2001). An example of signature recognition technique is network security monitor (NSM), an early signature-based intrusion detection system that find attacks by searching for keywords in network traffic captured using a sniffer (Lippmann et al., 2000; Anderson et al., 1995). The advantage of signature recognition technique is that the computation required to reconstruct network sessions and search for keywords is not excessive. However, the limitation is that it cannot detect novel attacks whose signatures are unknown. The limitation of signature recognition techniques can be overcome by using anomaly detection techniques as a complement (Ye & Chen, 2001).

Anomaly detection technique is one of the most frequently suggested approaches to detect novel new attacks. Basic idea is that intrusive behavior often shows anomalies from normal behavior in an information system and anomalies can be used to detect possible intrusions (Ye & Chen, 2001). It first establishes a statistical model of the subject's normal behavior and then issue warnings when it observes actions that deviate significantly from those models. Examples of anomaly detection technique are NIDES and EMERALD. NIDES is one of the first statistical-based anomaly detection systems used to detect unusual user and unusual program behavior. It forms a model of a user, system, or network activity (Kendall, 1999). EMERALD combines statistical anomaly detection from NIDES with signature recognition (Kendall, 1999). Anomaly detection technique has advantage to detect both known and novel intrusions if they demonstrate departures from a norm profile. Also there are disadvantages for anomaly detection such as careful tuning and large computation. Since anomalous behavior does not always mean an intrusion, anomaly detection systems need to be carefully tuned to avoid high false

alarm rates. A second disadvantage of anomaly detection schemes is the large computation and memory resources required to maintain the statistical model (Kendall, 1999).

2.2.3 Kinds of anomaly intrusion detection techniques

Existing anomaly detection techniques are strings, formal logic, production rules and statistical-based, stochastic, and data mining as seen in table 2.2.

In strings approach, a set of detector strings is constructed for a set of normal strings so that detector strings do not match self strings. If an incoming string matches any of the detector strings for at least the r number of contiguous bits, the detection of an anomaly is declared (Forrest et al., 1997). However, strings approach become infeasible when there exist normal strings for which it is impossible to generate detector strings.

Table 2.2 Various kinds of anomaly intrusion detection techniques

Strings		
Formal logic		
Rule-based / Production rules	RIPPER	
Statistical-based	SPC	EWMA
	Chi square	
	Factor analysis	
	Mahalanobis distance	
Stochastic	Markov process(first-order)	
	Partial high-order	
Data mining	Classification	SVM
		ANN
	Clustering	
		Nearest neighbor clustering
	Logistic regression	
	Neural network model	Self-Organizing Map (SOM)

The logic-based anomaly detection technique has been applied to routers, Domain Name System and some privileged programs. However, formal logic is difficult for most system administrators to understand and use for specifying a norm profile (Ko et al., 1997). In contrast, production rules in expert systems are more natural and understandable than formal logic for most system administrators to specify and update a norm profile (Anderson et al., 1995). However, it is difficult to enumerate and specify all possibilities of normal behavior, especially when multiple subjects are involved. Moreover, the behavior of a subject such as a user is generally not fixed but dynamically changing. The limitation in using formal logic or production rules is the difficulty to specify the dynamically changing behavior in advance (Ye & Chen, 2001).

Statistical-based anomaly detection approach represents well the expected normal behavior of a user and variance due to noises, thereby overcomes the problems with the string-based, logic-based and rule-based technique (Jou et al., 2000). However, there is a limitation that the computationally intensive procedure of the multivariate techniques cannot meet the demands - minimum delay of processing. Many researchers have tried to find a multivariate technique with a low computation cost.

As more advanced statistical tools, data mining techniques are able to deal with huge data. They can satisfy the demands for modern intrusion detection technique that should deal with large volumes of high-dimensional process data due to a large number of behavior measures and process rapidly to ensure an early indication and warning of intrusions. Also there is limitation that improper parameter selection might cause the over-fitting problem.

2.3 Recent Researches

In recent years, there have been a lot of researches on intrusion detection system (IDS) which differentiates an intrusive behavior from ordinary activities on the information systems. Those researches have focused on what attributes and data are most suitable for the user behavior and what technique is most effective in detecting suspicious activities. There has been a ground presumption related to data acquisition that normalcy and anomaly of a system be accurately manifested in selected system features (Lee & Stolfo, 1998).

2.3.1 System features

System features or data are collected at a host computer or network linking hosts. According to data source location, IDS are categorized into host-based and network-based IDS's. Host-based IDS detects an intrusion on the system by monitoring activities of only host computer. Various features such as sequence of system events (Forrest et al., 1996; Lee et al., 1997), event sequential order (Ye et al., 2002), the number of system events (Li & Ye, 2002; Ye et al., 2003) and frequency of each system events (Oh & Lee 2003; Zhang & Shen 2005) have been used for host-based IDS's.

Ye and Chen (2001) used intensity of each event for host-based anomaly detection. Audit data was obtained from a UNIX-based host machine, specifically a Sun SPARC 10 workstation with the Solaris operating system. Since there were about 284 different types

of BSM audit events on the host machine, 284 event types were considered in this study. Intensity of each event was measured from event type with occurring time as follows:

$$X_i(t) = \lambda \times 1 + (1 - \lambda) \times X_i(t-1) \tag{2.1}$$

when the audit event at time t falls into the i^{th} event type and

$$X_i(t) = \lambda \times 0 + (1 - \lambda) \times X_i(t-1) \tag{2.2}$$

when the audit event at time t is different from the i^{th} event type where $X_i(t)$ is the observed value of the i^{th} variable in the vector of observation at time t, λ is a smoothing constant that determines k or the decay rate.

Chen et al. (2005) introduced *tf* (term frequency) × *idf* (inverse document frequency) scheme, a common method in text categorization, based on frequency of system events. Each system call was treated as a "word" in a document and the set of system calls generated by a process was treated as the "document". This analogy made it possible to bring the full spectrum of well-developed text processing methods to apply to the intrusion detection problem. In order to apply text categorization, each process was first represented as a vector where each entry represents the occurrence of a specific system call during the process execution. Frequency-based encoding method was used to characterize program behavior. It requires to aggregate system call information over the entire execution of a process. Frequency-based encoding technique reduces the system overhead compared to sequence-based encoding techniques which require building a profile for each program and checking for attacks at every time frame. Since Frequency-

based encoding techniques build a profile only for each process and not for each program and check for attack instances at the end of the process.

Network-based IDS monitors traffic data traveling on the communication links and uses connection information among hosts as system features. Depren et al. (2005) used only six basic features of TCP/IP data while Wang (2005) used 46 variables. Wang (2005) reduced the number of independent variables by identifying the risk factors associated with individual major attacks. Most previous studies used all possible independent variables. Statistically, a model with a large number of independent variables does not necessarily have high predictive ability. Unnecessary variables can create bias and lead the model either to overestimate or underestimate predicting values but information about an individual risk factor associated with the attacks remains unclear. 46 risk factors, that is independent variables, with all features summarizing each connection information were used. Wu and Zhang (2006) used association rules to get more representative data from TCP/IP data. As more and more useful system features as possible are available for IDS, the classifier based on the features would be more effective.

2.3.2 Detection techniques

Detection techniques are broadly categorized into misuse detection (signature recognition) and anomaly detection according to their ideas on detecting intrusions. Misuse detection techniques signal an intrusion when an observed behavior matches a known attack. Anomaly detection techniques regard anomalies from normal behavior as

intrusions. Generally anomaly detection performs better to detect new attacks than misuse detection techniques.

Various detection techniques have been applied into IDS. Ye and Chen (2001) applied a multivariate anomaly detection technique based on the chi-square statistics. Many intrusions involve multiple subjects and multiple actions having impact on multiple behavior measures. Hence, a multivariate anomaly detection technique is needed for intrusion detection. However, the computationally intensive procedure of multivariate techniques cannot meet the demands of intrusion detection that can process large volumes of high-dimensional data within a short processing time. They selected chi-square as a statistics for multivariate anomaly detection technique since it has a low computation cost. Also, specification-based detection (Sekar et al., 2002), stochastic model (Ye et al., 2002) and factor analysis (Wu & Zhang, 2006) have been used as anomaly detection techniques.

Data mining techniques have become popular in intrusion detection research field since Lee and Stolfo (1998) proposed using data mining techniques for IDS. Data mining can relatively easily extract structural information and insights from huge datasets. Such an advantage of data mining techniques is also very useful to IDS. Li and Ye (2002), Oh and Lee (2003), Liu et al. (2004), and Li and Ye (2006) developed clustering methods based intrusion detection systems.

Jiang et al. (2006) proposed a clustering-based method for unsupervised intrusion detection (CBUID) to overcome shortages in the all existing unsupervised methods. Existing unsupervised methods cannot deal with categorical attributes or their solutions

are very complicated, the result of detection is sensitive to the parameters which are difficult to be determined and it is not reasonable to assume that the smaller size clusters of objects have, the more possible they are anomalous. In CBUID the data classification is performed by an improved nearest neighbor method and its time complexity is linear with the size of dataset and the number of attributes.

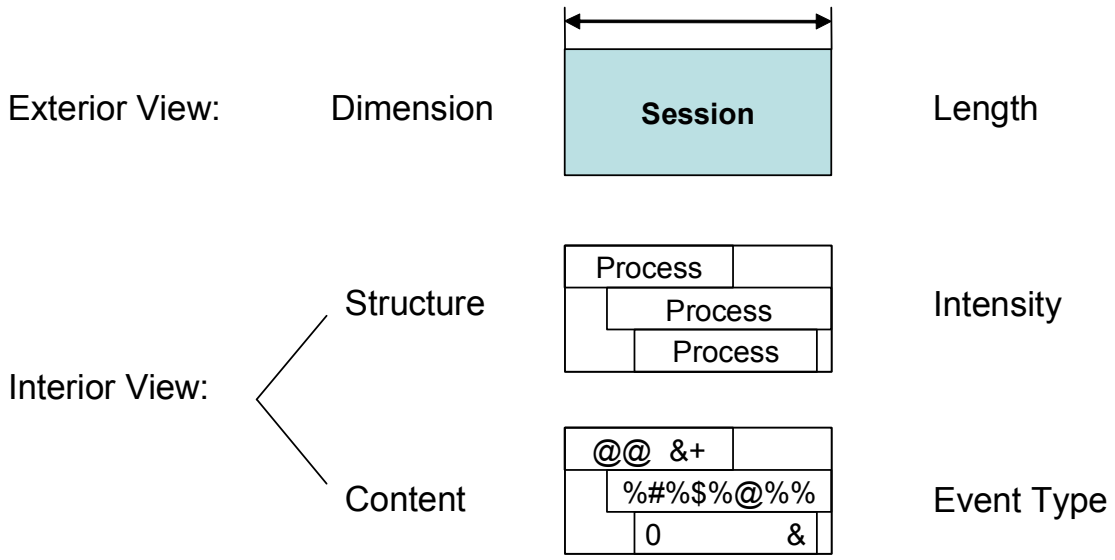
While Wang (2005) applied multinomial logistic regression modeling approach for anomaly intrusion detection. Previous studies focused on a signal binary outcome, that is, normal or abnormal, to detect potential attacks. This multinomial logistic regression can identify multi-type attacks as an outcome. Zhang and Shen (2005) presented the use of support vector machine (SVM) for IDS. SVM is a new technique for solving a variety of learning, classification and prediction problems. It is originated as an implementation of Vapnik's structural risk minimization (SRM) principle, which minimizes the generalization error, i.e., true error on unseen examples. One remarkable property of SVM is being independent of the feature space dimensionality. This means that SVM can generalize well in the presence of many features. Chen et al. (2005) proposed application of SVM and artificial neural network (ANN) for intrusion detection. ANN is a biologically inspired form of distributed computation. It is composed of simple processing units, or nodes, and connections between them. The connection has some weight, which is used to determine how much one unit will affect the other. The result has shown that the SVM performance is better than that for the ANN.

Chapter 3 New Framework for Host-based Feature Extraction

This chapter provides new framework for host-based feature extraction. Section 3.1 explains concept of new feature framework. New features developed under the proposed framework are discussed in Section 3.2. Section 3.3 presents the process of feature extraction from the DARPA 98 BSM dataset. Finally, experiment with the new features and the results are presented in Section 3.4.

3.1 New Framework Concept

Host-based IDS requires information for users' activities to detect an intrusion into a system of interest. The required information is data representing system users' activities and data is a collection of values for features or variables which are defined by specific descriptions or equations with output values from system monitoring. For example, session length can be a feature defined as session duration time measured in seconds. Since a feature represents a kind of sight of system administrator for users' behavior, diverse features mean layered and different points of view for an activity in the system.



* @&+%#$0: example symbols representing system event types

Figure 3.1 New feature framework development concept

Therefore, more features guarantee more reliable performance of IDS. Researchers have tried to develop more useful system features as possible for more effective results of IDS.

Two kinds of general view point for an abject are exterior and interior. Exterior view point concerns the shape or size while interior view examines the contents in an object and inside shape. Applying this concept into IDS feature development, session dimension features come from exterior view whereas session structure and content features come from interior view point as in figure 3.1. In the figure a session has three processes with four, eight, and two system events, respectively.

Session dimension features relate with size of a session which means session length and the length of its processes. For instance, it is about how long a session last in terms of time or the number of system events. Session structure features measure what the session looks like, for example, how many processes the session has and how the processes relate with one another. Session content features identify the event types while session dimension features do not distinguish the event types of a session. The content features deal with the occurrence of a specific event type and how many kinds of events appear on a session.

Various features have been used for Host-based IDS. They are sequence of system events (Forrest et al., 1996; Lee et al., 1997; Ye et al., 2002), the number of system events (Li & Ye, 2002; Ye et al., 2003) and frequency of each system events (Ye & Chen, 2001; Oh & Lee, 2003; Zhang & Shen, 2005). Chen et al. (2005) introduced *tf*(term frequency) × *idf* (inverse document frequency) scheme, a common method in text categorization, based on frequency of system events. Existing features can be categorized into three categories based on the concept as seen table 3.1. Frequency of each system events and *tf* × *idf* scheme are a form mixed with session structure and content features. However, there is no literature for using session dimension features and all the features from three feature categories. To get as many features as possible, we need to more thoroughly explore features in the three points of view such as session dimension, structure and content.

Table 3.1 Categorizing existing features

Categories	Literatures
Session dimension	-
Session structure	Forrest et al. (1996)*, Lee et al. (1997)*, Ye & Chen (2001)* Li & Ye (2002)*, Ye et al. (2002)*, Ye et al. (2003)* Chen et al. (2005)*, Zhang & Shen (2005)*
Session content	Forrest et al. (1996)*, Lee et al. (1997)*, Ye & Chen (2001)* Li & Ye (2002)*, Ye et al. (2002)*, Ye et al. (2003)* Oh & Lee (2003), Chen et al. (2005)*, Zhang & Shen (2005)*

* Features combined with session structure and content

3.2 New Feature Development under the Framework

Three kinds of view for a session can be well described as more practical terms. Since session dimension is measured by length of a session, length features are for session dimension. Session structure concerns mainly its intensity and intensity features represents session structure. As session consists of system events, event features mean session content related features. Length and intensity features ignore system event type and just count the number of events while event features distinguish a type of event with other event types.

3.2.1 Length-related features

Statistics related to length are based on session length and process length. There are three subgroups for length-related features such as overall length features, features by process duration time, features by number of events in processes. The length of a session is measured by duration time in seconds, the number of system event occurrences and the number of processes in the session. Also, the length of a process is measured by duration time in seconds and the number of system event occurrences in the process. First, last, longest, shortest process in a session are considered to have important information for user's behavior. Average values and ratios among features are introduced to carry potential information. Table 3.2 shows all possible features related to length and their range of values. Five features came from overall length feature subgroup, twelve features from feature subgroup by process duration time and twelve features from feature subgroup by number of events in processes. There are 29 features related to length in total.

3.2.2 Intensity-related features

Intensity of a session can be understood by three points of view such as overall intensity, process level and system event level. Overall intensity is calculated by average number of processes and system events over system length time which gives information for how many processes and system events occurs per second. The intensity of process level is about how many processes intersect at a given time period and how long they intersect

Table 3.2 Length-related features

Subgroup	Description	Name	Range
Overall length	Session duration time in seconds	"Dur"	$value \geq 0$
	Logarithm Dur	"lnDur"	$value \geq 0$
	Number of events	"NumEv"	$value > 0$
	Logarithm NumEv	"ln NumEv"	$value \geq 0$
	Number of processes	"NumPr"	$value \geq 1$
By process duration time	Average of all processes' duration time in seconds	"avDurPr"	$value \geq 0$
	Longest duration time among all processes	"longDurPr"	$value \geq 0$
	Shortest duration time among all processes	"shoDurPr"	$value \geq 0$
	First process's duration	"DurFirPr"	$value \geq 0$
	Last process's duration time	"DurLasPr"	$value \geq 0$
	$\frac{Average}{SessionDurationTime}$	"avDurPr%"	$0 \leq value \leq 1$
	$\frac{Longest}{SessionDurationTime}$	"longDurPr%"	$0 \leq value \leq 1$
	$\frac{Shortest}{SessionDurationTime}$	"shoDurPr%"	$0 \leq value \leq 1$
	$\frac{First}{SessionDurationTime}$	"DurFirPr%"	$0 \leq value \leq 1$
	$\frac{Last}{SessionDurationTime}$	"DurLasPr%"	$0 \leq value \leq 1$
	$\frac{Shortest}{Longest}$	"S/L_DurPr%"	$0 \leq value \leq 1$
	$\frac{First}{Last}$	"F/L_DurPr%"	$value \geq 0$

Table 3.2 Continued

Subgroup	Description	Name	Range
By number of events in processes	Average of all processes' number of events	"av#EvPr"	*value*>0
	Largest number of events among all processes	"larg#EvPr"	*value*>0
	Smallest number of events among all processes	"smal#EvPr"	*value*>0
	First process's number of events	"FirPr#Ev"	*value*>0
	Last process's number of events	"LasPr#Ev"	*value*>0
	$\frac{Average}{\#\ of\ \ Events\ \ in\ \ Session}$	"av#Ev Pr%"	0<*value*≤1
	$\frac{Largest}{\#\ of\ \ Events\ \ in\ \ Session}$	"larg#Ev Pr%"	0<*value*≤1
	$\frac{Smallest}{\#\ of\ \ Events\ \ in\ \ Session}$	"sma#Ev Pr%"	0<*value*≤1
	$\frac{First}{\#\ of\ \ Events\ \ in\ \ Session}$	"FirPr#Ev %"	0<*value*≤1
	$\frac{Last}{\#\ of\ \ Events\ \ in\ \ Session}$	"LasPr#Ev %"	0<*value*≤1
	$\frac{Smallest}{Largest}$	"S/L_#Ev Pr%"	0<*value*≤1
	$\frac{First}{Last}$	"F/L_#Ev Pr%"	*value*>0

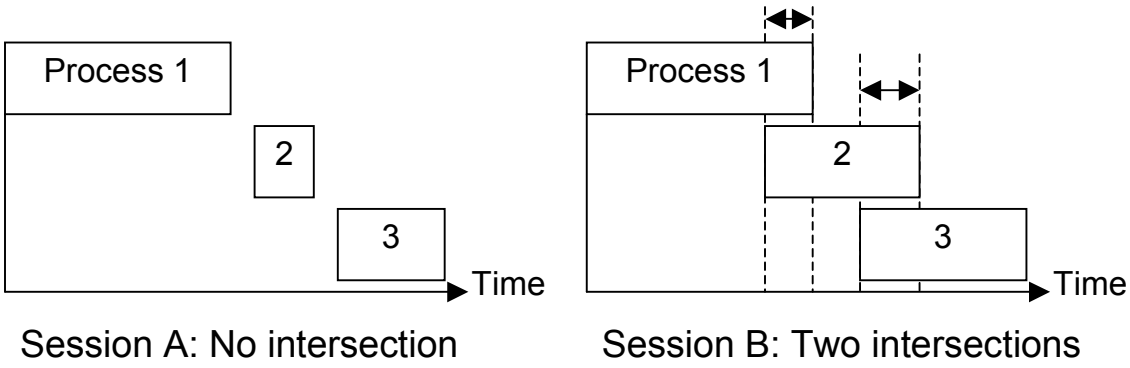


Figure 3.2 Example for process level intensity based on the intersection

together. Figure 3.2 shows how to measure process level intensity based on the process intersection. Session B has higher intensity of process level than session A although both of sessions have three processes in the figure. 46 Various features related to process level intensity are developed from intersection, based on the number of intersecting, intersecting time, the number of processes related to an intersection and the number of intersections related to a process.

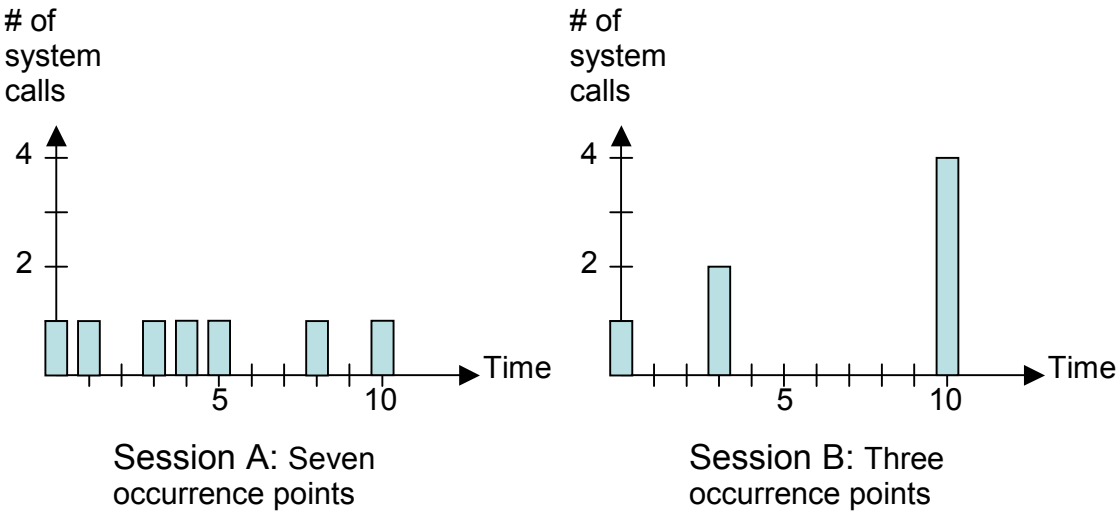


Figure 3.3 Example for system event level intensity in a session

The intensity of system event level is based on how many event occurrence points are in a session. Event occurrence point is the time when a system event or system events appear. Given same number of system events and same session duration time, the number of event occurrence points can be different. In figure 3.3, session A has seven occurrence points and session B has only three occurrence points even though both sessions have seven system events with ten second duration time. Session A is denser than session B in terms of occurrence points while session B shows more number of system events per a occurrence point than session A. 16 features based on the occurrence points are developed. Table 3.3 shows 64 features related to intensity.

3.2.3 Event-related features

Features related to event types are created by using the number of event types which depends on the operating systems. Event-related features come from two subgroups such as event diversity and event frequency. Event diversity is about how many diverse events happen in a session. Event frequency is measured for each event type as how many times a specific event appears in a session. Table 3.4 shows event-related features.

Table 3.3 Intensity-related features

Subgroup	Description	Name	Range
Overall intensity	Average number of events per second	"AvgNumEv"	*Value*>0
	Average number of processes per second	"AvgNumPr"	*Value*>0
Process level intensity	$\frac{\sum processDurationTime}{SessionDurationTime}$	"IntTime/Ses"	*value*≥1
	$\frac{Number\ of\ Intersections}{SessionDurationTime}$	"#Int/Time"	*value*≥0
	$\frac{Number\ of\ Intersections}{\#\_of\_processes}$	"#Int/Pro"	*value*≥0
	$\frac{Number\ of\ Intersections}{\#\_of\_Events\_in\_Session}$	"#Int/Ev"	*value*≥0
	Maximum time of Intersection	"MaxIntT"	*value*≥0
	Minimum time of Intersection	"MinIntT"	*value*≥0
	Average time of Intersection	"AvgIntT"	*value*≥0
	Summation of all Intersection duration times	"SumIntT"	*value*≥0
	$\frac{Maximum\ time\ of\ an\ Intersection}{SessionDurationTime}$	"MaxIntT/Time"	*value*≥0
	$\frac{Maximum\ time\ of\ an\ Intersection}{\#\_of\_processes}$	"MaxIntT/Pro"	*value*≥0
	$\frac{Maximum\ time\ of\ an\ Intersection}{\#\_of\_Events\_in\_Session}$	"MaxIntT/Ev"	*value*≥0
	$\frac{Minimum\ time\ of\ an\ Intersection}{SessionDurationTime}$	"MinIntT/Time"	*value*≥0
	$\frac{Minimum\ time\ of\ an\ Intersection}{\#\_of\_processes}$	"MinIntT/Pro"	*value*≥0
	$\frac{Minimum\ time\ of\ an\ Intersection}{\#\_of\_Events\_in\_Session}$	"MinIntT/Ev"	*value*≥0
	$\frac{Average\ time\ of\ an\ Intersection}{SessionDurationTime}$	"AvgIntT/Time"	*value*≥0
	$\frac{Average\ time\ of\ an\ Intersection}{\#\_of\_processes}$	"AvgIntT/Pro"	*value*≥0

Table 3.3 Continued

Subgroup	Description	Name	Range
Process level intensity	$\frac{Average\ time\ of\ an\ Intersection}{\#\_of\_Events\_in\_Session}$	"AvgIntT/Ev"	*value*≥0
	$\frac{Sum\ of\ all\ Intersection\ duration}{SessionDurationTime}$	"SumIntT/Time"	*value*≥0
	$\frac{Sum\ of\ all\ Intersection\ duration}{\#\_of\_processes}$	"SumIntT/Pro"	*value*≥0
	$\frac{Sum\ of\ all\ Intersection\ duration}{\#\_of\_Events\_in\_Session}$	"SumIntT/Ev"	*value*≥0
	$\frac{\#of\ processes\ involved\ in\ any\ Intersection}{\#\_of\_processes}$	"IntPro/Pro"	*value*≥0
	Maximum number of processes involved in an Intersection	"Max#ProInt"	*value*≥0
	Minimum number of processes involved in an Intersection	"Min#ProInt"	*value*≥0
	Average number of processes involved in an Intersection	"Avg#ProInt"	*value*≥0
	$\frac{Max.\#of\ processes\ involved\ in\ an\ Intersection}{SessionDurationTime}$	"Max#ProInt/Time"	*value*≥0
	$\frac{Max.\#of\ processes\ involved\ in\ an\ Intersection}{\#\_of\_processes}$	"Max#ProInt/Pro"	*value*≥0
	$\frac{Max.\#of\ processes\ involved\ in\ an\ Intersection}{\#\_of\_Events\_in\_Session}$	"Max#ProInt/Ev"	*value*≥0
	$\frac{Min.\#of\ processes\ involved\ in\ an\ Intersection}{SessionDurationTime}$	"Min#ProInt/Time"	*value*≥0
	$\frac{Min.\#of\ processes\ involved\ in\ an\ Intersection}{\#\_of\_processes}$	"Min#ProInt/Pro"	*value*≥0
	$\frac{Min.\#of\ processes\ involved\ in\ an\ Intersection}{\#\_of\_Events\_in\_Session}$	"Min#ProInt/Ev"	*value*≥0
	$\frac{Avg.\#of\ processes\ involved\ in\ an\ Intersection}{SessionDurationTime}$	"Avg#ProInt/Time"	*value*≥0
	$\frac{Avg.\#of\ processes\ involved\ in\ an\ Intersection}{\#\_of\_processes}$	"Avg#ProInt/Pro"	*value*≥0
	$\frac{Avg.\#of\ processes\ involved\ in\ an\ Intersection}{\#\_of\_Events\_in\_Session}$	"Avg#ProInt/Ev"	*value*≥0

Table 3.3 Continued

Subgroup	Description	Name	Range
Process level intensity	Maximum number of intersections involved in a process	"Max#IntPro"	*value*≥0
	Minimum number of intersections involved in a process	"Min#IntPro"	*value*≥0
	Average number of intersections involved in a process	"Avg#IntPro"	*value*≥0
	$\frac{Max.\#\ of\ intersections\ involved\ in\ a\ process}{SessionDurationTime}$	"Max#IntPro/Time"	*value*≥0
	$\frac{Max.\#\ of\ intersections\ involved\ in\ a\ process}{\#\_of\_processes}$	"Max#IntPro/Pro"	*value*≥0
	$\frac{Max.\#\ of\ intersections\ involved\ in\ a\ process}{\#\_of\_Events\_in\_Session}$	"Max#IntPro/Ev"	*value*≥0
	$\frac{Min.\#\ of\ intersections\ involved\ in\ a\ process}{SessionDurationTime}$	"Min#IntPro/Time"	*value*≥0
	$\frac{Min.\#\ of\ intersections\ involved\ in\ a\ process}{\#\_of\_processes}$	"Min#IntPro/Pro"	*value*≥0
	$\frac{Min.\#\ of\ intersections\ involved\ in\ a\ process}{\#\_of\_Events\_in\_Session}$	"Min#IntPro/Ev"	*value*≥0
	$\frac{Avg.\#\ of\ intersections\ involved\ in\ a\ process}{SessionDurationTime}$	"Avg#IntPro/Time"	*value*≥0
	$\frac{Avg.\#\ of\ intersections\ involved\ in\ a\ process}{\#\_of\_processes}$	"Avg#IntPro/Pro"	*value*≥0
	$\frac{Avg.\#\ of\ intersections\ involved\ in\ a\ process}{\#\_of\_Events\_in\_Session}$	"Avg#IntPro/Ev"	*value*≥0
System event level intensity	Number of event-occurring times in a session	"NumPoints"	*value*≥1
	$\frac{Number\_of\_Event-Occurring\_times}{Session\_Duration\_Time}$	"Points/Time"	*value*>0
	$\frac{Number\_of\_Event-Occurring\_times}{\#\_of\_Events\_in\_a\_Session}$	"Points/Event"	*value*>0
	$\frac{Number\_of\_Event-Occurring\_times}{\#\_of\_processes\_in\_a\_Session}$	"Points/Process"	*value*>0
	Maximum Number of events in a Occurring point of a session	"MaxEvPts"	*value*≥1
	Minimum Number of events in a Occurring point of a session	"MinEvPts"	*value*≥1
	Average Number of events of Occurring points in a session	"AvgEvPts"	*value*≥1

Table 3.3 Continued

Subgroup	Description	Name	Range
System event level intensity	$\frac{Maximum\ Number\ of\ events\ among\ Occurring\ points}{Session\ \_\ Duration\ \_\ Time}$	"MaxPts/Time"	*value*>0
	$\frac{Maximum\ Number\ of\ events\ among\ Occurring\ points}{\#\ \_\ of\ \_\ Events\ \_\ in\ \_\ a\ \_\ Session}$	"MaxPts/Event"	*value*>0
	$\frac{Maximum\ Number\ of\ events\ among\ Occurring\ points}{\#\ \_\ of\ \_\ processes\ \_\ in\ \_\ a\ \_\ Session}$	"MaxPts/Process"	*value*>0
	$\frac{Minimum\ Number\ of\ events\ among\ Occurring\ points}{Session\ \_\ Duration\ \_\ Time}$	"MinPts/Time"	*value*>0
	$\frac{Minimum\ Number\ of\ events\ among\ Occurring\ points}{\#\ \_\ of\ \_\ Events\ \_\ in\ \_\ a\ \_\ Session}$	"MinPts/Event"	*value*>0
	$\frac{Minimum\ Number\ of\ events\ among\ Occurring\ points}{\#\ \_\ of\ \_\ processes\ \_\ in\ \_\ a\ \_\ Session}$	"MinPts/Process"	*value*>0
	$\frac{Average\ Number\ of\ events\ among\ Occurring\ points}{Session\ \_\ Duration\ \_\ Time}$	"AvgPts/Time"	*value*>0
	$\frac{Average\ Number\ of\ events\ among\ Occurring\ points}{\#\ \_\ of\ \_\ Events\ \_\ in\ \_\ a\ \_\ Session}$	"AvgPts/Event"	*value*>0
	$\frac{Average\ Number\ of\ events\ among\ Occurring\ points}{\#\ \_\ of\ \_\ processes\ \_\ in\ \_\ a\ \_\ Session}$	"AvgPts/Process"	*value*>0

Table 3.4 Event-related features

Subgroup	Description	Name	Range
Event diversity	Number of event types occurred in a session	"#typeEv"	*value*≥1
	$\frac{\#\ of\ \ event\ types\ occurred\ in\ a\ Session}{SessionDurationTime}$	"#typeEv/T"	*value*>0
	$\frac{\#\ of\ \ event\ types\ occurred\ in\ a\ Session}{\#\_of\_\ Events\_in\_a\_Session}$	"#typeEv/Ev"	*value*>0
Event frequency	Number of occurrences of event *i*	"NumEv(*i*)"	*value*≥0
	$\frac{Number\ of\ \ occurrences\ of\ \ event\ i}{\#\_of\_\ Events\_in\_a\_Session}$	"NumEv(*i*)%"	*value*≥0

3.3 Feature Extraction from the DARPA 98 BSM Data

3.3.1 Data source

The Defense Advanced Research Projects Agency (DARPA) wished to evaluate competing algorithms and systems for computer intrusion detection. MIT Lincoln Lab (MITLL) built a simulation network at an Air Force base which consisted of models for different types of users including secretaries and managers and known attacks or their variants such as DOS, R2L, U2R and PROBE. Figure 3.4 shows the simulation network comprising about 100 users and 1000 host computers.

Simulated traffic of an air force local area network was collected into two types of files; Transmission Control Protocol (TCP) dump data file and Basic Security Module (BSM) data file as seen in figure 3.5. TCP dump data files record information for traffic sent over the network by sniffing the network at a machine connected to it. BSM data files

record system events made in a victim machine for host-level audit. This simulation had been performed for two years beginning in 1998.

There are two data sets as the result of this DARPA intrusion detection evaluation project; 1998 and 1999 data sets. The 1998 data set consists of seven-week training data and two-week testing data. The 1999 set contains three-week training data and two-week test data. Those data sets have been widely used to evaluate many intrusion detection systems newly proposed in the literatures. We used host-based BSM audit data from the seven-week training data of the 1998 data set to evaluate our method.

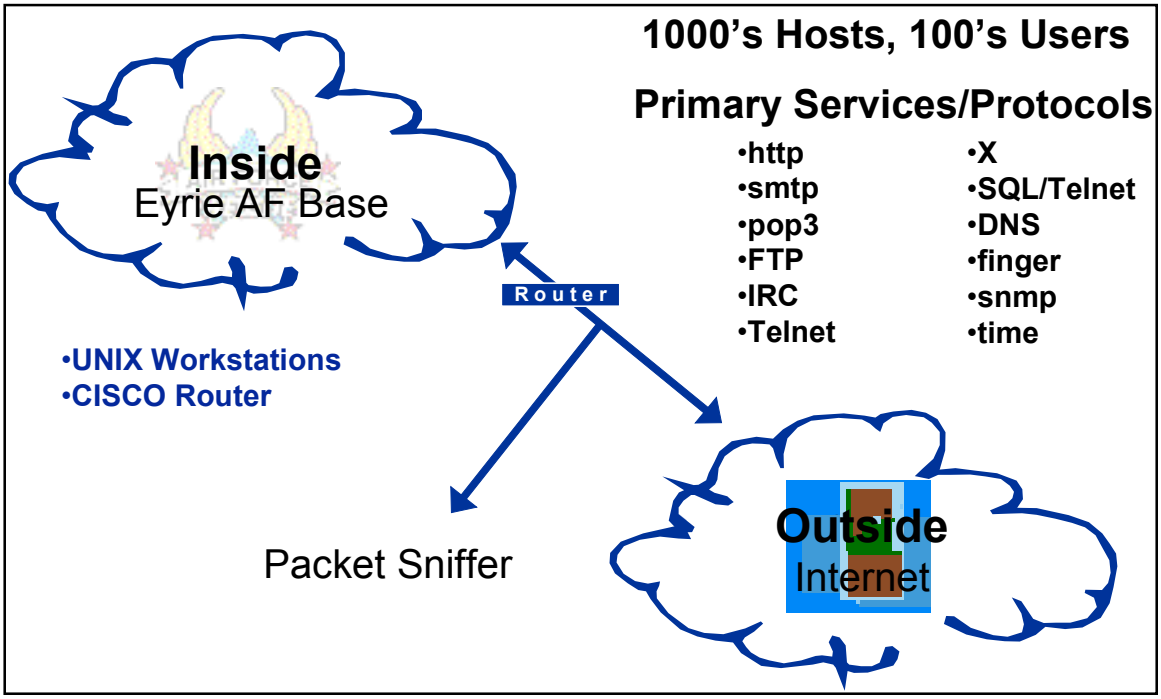


Figure 3.4 DARPA –MITLL simulation network

Source: http://www.ll.mit.edu/IST/ideval/docs/1998/introduction/index.htm

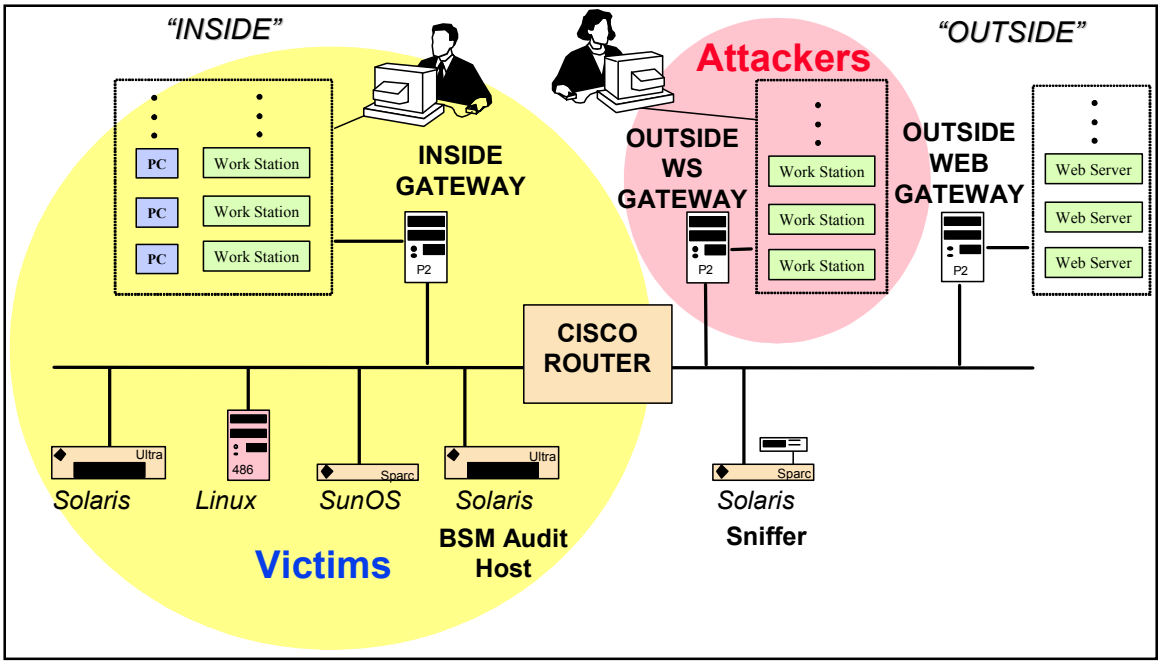


Figure 3.5 Data acquisition and simulation network detail

Source: http://www.ll.mit.edu/IST/ideval/docs/1998/introduction/index.htm

3.3.2 Data preprocessing

BSM records security-relevant events to monitor activities in a host machine. Two files are given by BSM; BSM list file and BSM audit data file. BSM list files show information for sessions comprising one or more system events of all hosts in the network whereas BSM audit files provide information for only system events of a specific host. Since a session is a set of system events, BSM list file is a brief summary of activity in the network and BSM audit file is a large raw-data file recording the details of activity in a certain machine. Table 3.5 shows list and size of the DARPA 98 BSM list files and audit

Table 3.5 List and size of the DARPA 98 BSM list files and audit files

Date		BSM audit file & size	BSM list file & size
First week	Monday	bsm.audit 174,618KB	bsm.list 26KB
	Tuesday	pascal.praudit 180,142KB	bsm.list 23KB
	Wednesday	bsm.audit 256,927KB	bsm.list 59KB
	Thursday	bsm.audit 179,754KB	bsm.list 42KB
	Friday	bsm.audit 193,231KB	bsm.list 33KB
Second week	Monday	pascal.praudit 154,870KB	bsmout.list 28KB
	Tuesday	pascal.praudit 139,984KB	bsmout.list 28KB
	Wednesday	pascal.praudit 187,613KB	bsmout.list 32KB
	Thursday	pascal.praudit 182,420KB	bsmout.list 29KB
	Friday	pascal.praudit 213,860KB	bsmout.list 32KB
Third week	Monday	pascal.praudit.gz 552,565KB	bsm.list.gz 46KB
	Tuesday	pascal.praudit.gz 257,125KB	bsm.list.gz 40KB
	Wednesday	pascal.praudit.gz 642,188KB	bsm.list.gz 38KB
	Thursday	pascal.praudit.gz 400,464KB	bsm.list.gz 40KB
	Friday	pascal.praudit.gz 212,560KB	bsm.list.gz 31KB
Fourth week	Monday	pascal.praudit.gz 221,202KB	bsm.list.gz 36KB
	Tuesday	pascal.praudit.gz 305,277KB	bsm.list.gz 54KB
	Wednesday	pascal.praudit.gz 177,834KB	bsm.list.gz 24KB
	Thursday	pascal.praudit.gz 208,413KB	bsm.list.gz 47KB
	Friday	pascal.praudit.gz 528,592KB	bsm.list.gz 33KB
Fifth week	Monday	pascal.praudit.gz 447,703KB	bsm.list.gz 43KB
	Tuesday	pascal.praudit.gz 255,600KB	bsm.list.gz 47KB
	Wednesday	pascal.praudit.gz 195,566KB	bsm.list.gz 41KB
	Thursday	pascal.praudit.gz 439,921KB	bsm.list.gz 49KB
	Friday	pascal.praudit.gz 205,868KB	bsm.list.gz 44KB

Table 3.5 Continued

Date		BSM audit file & size	BSM list file & size
Sixth week	Monday	pascal.praudit.gz 418,010KB	bsm.list.gz 35KB
	Tuesday	pascal.praudit.gz 257,561KB	bsm.list.gz 39KB
	Wednesday	pascal.praudit.gz 197,389KB	bsm.list.gz 34KB
	Thursday	pascal.praudit.gz 214,170KB	bsm.list.gz 50KB
	Friday	pascal.praudit.gz 194,590KB	bsm.list.gz 43KB
Seventh week	Monday	pascal.praudit.gz 227,808KB	bsm.list.gz 39KB
	Tuesday	pascal.praudit 195,461KB	bsm.list.gz 5KB
	Wednesday	pascal.praudit.gz 195,461KB	bsm.list.gz 35KB
	Thursday	pascal.praudit.gz 415,893KB	bsm.list.gz 32KB
	Friday	pascal.praudit.gz 226,327KB	bsm.list.gz 42KB

```
1799  06/01/1998 08:07:47 00:01:09 telnet 1814 23 172.016.114.168
172.016.112.050 0 -
```

Figure 3.6 Example of session information on BSM list file of Monday, first week.

```
header,182,2,ioctl(2),,Mon Jun 01 07:56:56 1998, + 788290611 msec
path,/devices/pseudo/cn@0:console
attribute,20620,2122,tty,8388608,11409,0
argument,2,0x7415,cmd
argument,3,0xeffff2b0,arg
argument,2,0x501cd434,strioctl:vnode
subject,2122,root,other,root,other,273,258,0 0 pascal.eyrie.af.mil
return,success,0
trailer,182
```

Figure 3.7 Example of description for a system event in BSM audit file of Monday, first week.

files. In the case of Monday, first week 1998 data set, for example, the size of the BSM list file is 26 KB and one of the BSM audit file 174,618 KB.

BSM list file serves information such as index, start time, duration, source and destination IP address, and the nature, normal or attack, of each session in a host machine. Figure 3.6 shows that session 1799 started at 08:07:47 June 1, 1998 and lasted for one minute and nine seconds and it was a normal session with source IP, 172.016.114.168, and destination IP, 172.016.112.050.

A BSM audit file contains information corresponding to each system event made by a host. The information such as event type, session ID, process ID, IP address, occurrence time and so on for the event is stored in the audit file whenever a system event occurs in a host machine. Figure 3.7 shows the information for a system event stored in the audit file of Monday, first week. The event was “ioctl(2)” type, occurred on Monday the first of June 1998 and belonged to process 273 of session 258 according to the information on Figure 3.7.

Statistics of features from a session were required for our method to judge if the session was an attack or a normal session. More possible kinds of statistics for a session extracted from BSM files make more useful information available in intrusion detection system. Most research has tried to get diverse statistics of a session from BSM audit files and to only identify attack or normal sessions from BSM list files. In this study, there were three steps to get session statistics for our analysis. We used Microsoft Visual Basic 6.0

programming language to process BSM files and obtain statistics of interest from BSM audit files and list files. Figure 3.8 shows the procedure of data preprocessing.

First, events in a BSM audit file were categorized into sessions according to session ID of the event. Session-named files, corresponding to each session respectively, were created to contain information of system events belonged to the session. Table 3.6 shows how many sessions and how many system events appeared in a day from the DARPA 98 BSM audit files. For example, 744,085 system events from the BSM audit file of Monday first week 1998 data set were allocated into 182 session files.

Second, various features were extracted from a session file according to the feature descriptions in table 3.2, 3.3 and 3.4. Before extracting features related to event types, we

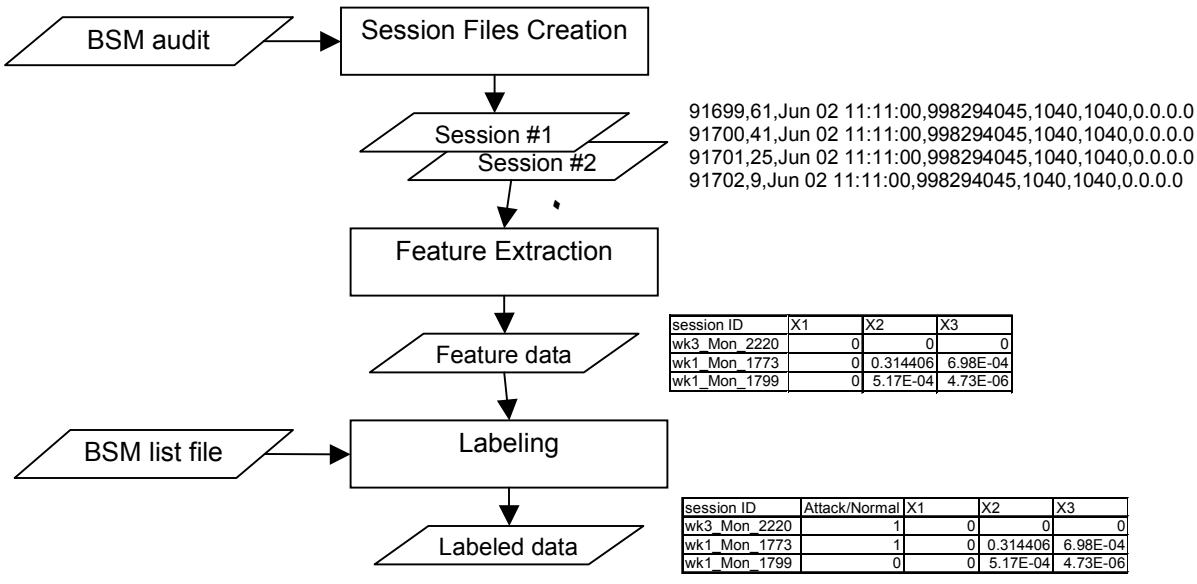


Figure 3.8 Procedure of data preprocessing

needed to know what kinds of system events exist in the DARPA 98 BSM data set. It turned out that there are 75 types of system events in the BSM files by searching all audit data files. Figure 3.9 shows the list of 75 system event types. As a result of feature extraction, 246 features were obtained from each session file as seen in table 3.7.

Third, sessions were labeled with normal or attack by referring a BSM list file. Unfortunately, session indices in a BSM list file are not consistent with the session ID's in a BSM audit file and the number of sessions of both BSM files, also, are significantly different. Because BSM list file records sequentially session information of all hosts in the network whereas BSM audit file stores information for only a specific host. For example of Monday first week 1998 data set, there are 308 sessions in the BSM list file, compared to 182 sessions in the BSM audit file. We used start time, duration and IP address of sessions as matching criteria to relate a session in BSM audit file with a session in BSM list file instead of session index. By performing the matching procedure on first week Monday 1998 data set, for example, 178 session files were labeled with normal and two sessions with attack. We applied our data pre-processing method on seven-week training data of 1998 data with about 40,495,000 system events to obtain a data set to be ready for our analysis. As a result of the pre-processing, we got data with 246 variables for 7,632 normal sessions and 456 attack sessions as in table 3.8.

Table 3.6 Numbers of sessions and system events in the DARPA 98 BSM audit files

Date		Number of sessions	Number of system events
First week	Monday	182	744,085
	Tuesday	148	778,781
	Wednesday	277	1,094,935
	Thursday	194	767,926
	Friday	184	819,471
Second week	Monday	174	661,129
	Tuesday	150	595,198
	Wednesday	196	800,938
	Thursday	179	780,361
	Friday	197	906,479
Third week	Monday	431	2,397,774
	Tuesday	205	1,121,967
	Wednesday	204	2,759,945
	Thursday	363	1,713,695
	Friday	210	891,696
Fourth week	Monday	330	941,820
	Tuesday	339	1,320,478
	Wednesday	163	768,152
	Thursday	322	903,596
	Friday	184	2,249,503
Fifth week	Monday	364	1,938,514
	Tuesday	212	1,116,098
	Wednesday	192	845,300
	Thursday	226	1,829,613
	Friday	196	875,700

Table 3.6 Continued

Date		Number of sessions	Number of system events
Sixth week	Monday	337	1,815,737
	Tuesday	209	1,108,912
	Wednesday	202	850,350
	Thursday	239	925,079
	Friday	213	840,072
Seventh week	Monday	208	948,058
	Tuesday	191	829,151
	Wednesday	191	829,151
	Thursday	331	1,775,358
	Friday	215	950,400

Table 3.7 Features extracted from the DARPA 98 BSM dataset

Groups		Variable number
Length	Overall length	1-5
	Length by process duration time	6-17
	Length by number of system events in a process	18-29
Intensity	Overall intensity	30-31
	Process level intensity	32-77
	System event level intensity	78-93
Events	Event diversity	94-96
	Event frequency	97-246

accept(2)	kill(2)	pathdonf(2)
access(2)	link(2)	pipe(2)
audit(2)	login - local	putmsg(2)
auditon(2) - get audit state	login - telnet	putmsg-connect
bind(2)	logout	putpmsg(2)
chdir(2)	lstat(2)	readlink(2)
chmod(2)	memcntl(2)	recvfrom(2)
chown(2)	mkdir(2)	rename(2)
close(2)	mknod(2)	rmdir(2)
connect(2)	mmap(2)	sendto(2)
creat(2)	munmap(2)	setaudit(2)
doorfs(2) - DOOR_CALL	old nice(2)	setegid(2)
doorfs(2) - DOOR_CREATE	old setgid(2)	seteuid(2)
execve(2)	old setuid(2)	setgroups(2)
exit(2)	old utime(2)	setpgrp(2)
fchdir(2)	open(2) - read	setrlimit(2)
fchmod(2)	open(2) - read,write	setsockopt(2)
fchown(2)	open(2) - read,write,creat	socket(2)
fcntl(2)	open(2) - read,write,creat,trunc	stat(2)
fork(2)	open(2) - read,write,trunc	statvfs(2)
fork1(2)	open(2) - write	su
getaudit(2)	open(2) - write,creat	symlink(2)
getmsg(2)	open(2) - write,creat,trunc	sysinfo(2)
inetd	open(2) - write,trunc	unlink(2)
ioctl(2)	pathconf(2)	vfork(2)

Figure 3.9 List of 75 system event types

Table 3.8 Data resulted from pre-processing procedure

	Normal	Attack	Total
Number of sessions	7,632	456	8,088

3.4 Experiment based on the DARPA 98 BSM Data

3.4.1 Experimental setup

We made a smaller set of data from the larger pre-processed data for our experiment. We extracted data of 3051 normal sessions for the last five weeks from seven-week training data in the 1998 data set which is considered as more representative for ordinary activities than the first two weeks. Data for 456 attacks from the whole seven weeks were used in our analysis without any omission of the data since our experiment needed much attack data as possible to get more reliable results. Our data set for the experiment comprises 3,051 normal session and 456 attack session data.

We randomly divided the data set into the training data and the testing data for our experiment as shown in table 3.9. About 16% of the normal data and 20% of the attack data came to be a training data and the remaining of the data became the testing data. Table 3.9 shows the number of samples in the training and testing data.

Table 3.9 Number of samples in data set

Training data		Testing data	
Normal sessions	Attack sessions	Normal sessions	Attack sessions
500	91	2551	365

Support vector data description (SVDD) was used as anomaly detection technique for the performance experiment of the proposed feature framework. SVDD is a one-class classification originated from the support vector machines (SVMs). One-class classification method tries to detect which sample is similar to training data based on a description of this training data set. This method is able to detect outliers which have different characteristics with training data as well. The basic idea of the SVDD method is to find a spherically-shaped small boundary that envelops most of data of interest. The hypersphere should have minimum volume as possible and simultaneously contain as many data as possible in order to minimize the possibility of accepting outlier data. The more details on SVDD are in Tax and Duin (2004) and the application of SVDD in intrusion detection is explained in Tao et al. (2004) and Yang et al. (2004).

An evaluation of intrusion detection system requires the estimation of two quantities: false alarm rate and attack detection rate. False alarm occurs when a normal session is assigned to attack by the trained SVDD boundary. False alarm rate is calculated by the number of false alarm over the number of normal sessions in testing data. Attack detection rate is the probability of correctly detecting the presence of the attack session

by the boundary. The performance of our method was measured by calculating false alarm rates and detection rates for the data set.

3.4.2 New feature framework results

Table 3.10 shows the performance results of new feature framework for the data set from the 1998 DARPA BSM data compared with the existing features. The results of performance are 99.73% of detection rate with 2.63% of false alarm rate for the testing data. Chen et al. (2005) applied Support vector machine (SVM) with *tf×idf* scheme and Artificial neural networks (ANN) with frequency scheme for the 1998 DARPA BSM data. Compared to the best results in Chen et al., SVDD with the features from the proposed new framework showed higher detection rate and lower false alarm rate.

3.4.3 Performance comparison among three feature groups

Table 3.11 shows the performances of three individual feature categories and combined two feature categories. Among the individual categories, the event type feature group achieved the best performance, showing 100% detection rate and reasonable false alarm

Table 3.10 Detection performance comparison between two existing features and the proposed features extracted by the new framework

Data	Attack detection rate	False alarm rate
SVDD with proposed features	99.73%	2.63%
SVM with *tf×idf* scheme*	99.60%	2.87%
ANN with frequency scheme*	99.20%	4.94%

* Best results of Chen et al. (2005)

Table 3.11 Performance comparison among individual feature categories, combined two categories, and all three categories

Feature groups		Number of features used	Attack detection rate (%)	False alarm rate (%)
Individual category	Length	29	4.93	1.61
	Intensity	64	97.26	46.18
	Event type*	153	100.00	3.88
Combined two categories	Length + Intensity*	93	99.73	2.78
	Length + Event type	182	3.29	1.22
	Intensity + Event type	217	100.00	20.38
All three categories*		246	99.73	2.63

* Three feature categories showing best performances.

rate. The intensity feature group brought good detection rate, over 97%, whereas its false alarm rate is too high, over 46%. The worst performance among the three groups came from the length feature group which showed very low detection rate, under 5%. Both the length and intensity categories turned out to be not practical for intrusion detection because of their unreasonable low detection rate and high false alarm rate, respectively. However, features combined with the two categories showed surprisingly good performance, 99.73% detection rates and 2.78% false alarm rate. Individual event type category and features combined with length and intensity were as good as all 246 features in terms of performance.

Chapter 4 Differentiated Intrusion Detection

This chapter provides differentiated intrusion detection methodology. Section 4.1 presents motivations of the proposed method. Introduction to SVDD is presented in Section 4.2. Section 4.3 presents mathematical formulation for differentiated intrusion detection. The proposed differentiated anomaly intrusion detection is explained in detail in Section 4.4. Finally, experiment with simulated data and the DARPA data and the results are presented in Section 4.5.

4.1 Motivation

There exists more harmful type of attack to an information system among intrusion types. According to 2006 CSI/FBI computer crime and security survey, the most common attack type was “computer virus” and the attack type causing the biggest loss per case was “unauthorized access to information” in the United States as seen in table 4.1. The unauthorized access to information is the most critical attack type especially in an organization with confidential information on its computer network systems which

Table 4.1 Most frequent top seven attack types and their loss amounts

Types of attacks	Percent of respondents*	Losses* (Dollars)	Losses per case (Dollars)
Virus	65	15,691,460	241,407
Laptop/mobile theft	47	6,642,660	141,333
Insider abuse of Net access	42	1,849,810	44,043
Unauthorized access to information	32	10,617,000	331,781
Denial of service	25	2,922,010	116,880
System penetration	15	758,000	50,533
Abuse of wireless network	14	469,010	33,501

* Source: 2006 CSI/FBI computer crime and security survey

should not be released to the public. Because it is expected to bring huge negative consequences such as operational trouble, financial loss and reputation damage of the organization when the attack passes through its defense layers including authentication, encryption, firewall, and intrusion detection system. Therefore, a system administrator needs to more strictly detect intrusions of the worst attack type to her organization while detecting other ordinary attacks. Existing anomaly intrusion detection techniques do not support this task.

In the existing anomaly detection methods, all attacks are treated with equal importance regardless of their types. For example, a denial of service attack which blocks victim system's whole operation and brings severe loss is regarded as only one of anomalies as

same as a probe attack which merely searches weak points of the system without disturbing the system operation. There is no anomaly detection method which differentiates attack types by their harmfulness when training its classifier.

Although there are anomaly detection techniques which identify the type of attack, they still can not perform intrusion detection with different weights on intrusion types. KDD '99 classifier learning contest was to create a predictive model to distinguish five categories such as normal, probe, denial of service, user-to-root and remote-to-local (Levin, 2000). Wang (2005) proposed a multinomial logistic regression approach for anomaly intrusion detection in which one record is assigned one of the above five categories based on 13 risk factors. Since those approaches aim at classifying intrusion types, they are not useful in differentiated detection on attack types.

This dissertation proposes a novel differentiated detection approach for anomaly intrusion detection to perform tighter detection on a targeted attack type and ordinary detection on nontargeted attack types. To the best of my knowledge, this is the first such approach for anomaly intrusion detection. The main idea is to use regularization parameter in support vector data description (SVDD) as a weight factor for a targeted type of attack on how strictly it is detected compared to nontargeted types. The higher weight for a targeted attack type means that the type is more harmful and needs to be more strictly detected than the nontargeted types. SVDD is a one-class classifier which was developed by Tax and Duin (2004) and Tao et al. (2004) introduced it as anomaly intrusion detection method to the field of intrusion detection for the first time.

4.2 Introduction to SVDD

One-class classification method tries to detect which sample is similar to training data based on a description of this training data set. This method is able to detect outliers which have different characteristics with training data. It is quite useful in solving a classification problem in which samples for one of the classes are plentiful and samples for the others are very few. SVDD is a one-class classification originated from the support vector machines (SVMs).

The basic idea of the SVDD method is to find a spherically-shaped small boundary that envelops most of data of interest. The hypersphere should have minimum volume as possible and simultaneously contain as many data as possible in order to minimize the possibility of accepting outlier data. Given N observations of normal data with p variables $\{\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_n\}$, the hypersphere of SVDD with a radius R and a center $\boldsymbol{\mu}$ is subject to

$$\|\mathbf{x}_i - \boldsymbol{\mu}\|^2 \leq R^2, \forall i \tag{4.1}$$

to envelop all normal data. Minimizing the volume of the hypersphere is represented with minimizing R^2 with respect to R and $\boldsymbol{\mu}$. It is possible that there exist a few outliers in training data set and we can not distinguish them from normal data, thereby bigger sphere may be obtained. To prevent this consequence we need to penalize outliers' participation

in constructing the hypersphere. Therefore, slack variables $\xi_i (\geq 0)$ are introduced to penalize larger distance between $\mathbf{x}_i$ and $\boldsymbol{\mu}$. The minimization problem is modified with

$$Min. \quad R^2 + C\sum_{i=1}^{N}\xi_i \tag{4.2}$$

where the parameter C gives the trade-off between the volume of the sphere and the number of observations outside. Equation (4.1) changes into the following constraints that almost all observations are within the sphere:

$$\left\|\mathbf{x}_i - \boldsymbol{\mu}\right\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad \forall i. \tag{4.3}$$

By introducing the Lagrange multipliers for inequality conditions of equation (4.3), we can obtain the Lagrangian function:

$$L(R, \boldsymbol{\mu}, \alpha_i, \beta_i, \xi_i) = R^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\{R^2 + \xi_i - (\left\|\mathbf{x}_i\right\|^2 - 2\boldsymbol{\mu}\cdot\mathbf{x}_i + \left\|\boldsymbol{\mu}\right\|^2)\} - \sum_{i=1}^{N}\beta_i\xi_i \tag{4.4}$$

with $\alpha_i \geq 0$ and $\beta_i \geq 0$. The solution of equation (4.4) is obtained by setting partial derivatives R, $\boldsymbol{\mu}$, ξ_i of $L(R, \boldsymbol{\mu}, \alpha_i, \beta_i, \xi_i)$ to zero:

$$\frac{\partial L}{\partial R} = 0 : \sum_{i=1}^{N} \alpha_i = 1 \tag{4.5}$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}} = 0 : \boldsymbol{\mu} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\sum_i \alpha_i} = \sum_i \alpha_i \mathbf{x}_i \tag{4.6}$$

$$\frac{\partial L}{\partial \xi_i} = 0 : C - \alpha_i - \beta_i = 0 \tag{4.7}$$

Since $\alpha_i = C - \beta_i$ from equation (4.7), $\alpha_i \geq 0$, and $\beta_i \geq 0$, we can remove the Lagrange multipliers β_i by producing $0 \leq \alpha_i \leq C$. Applying equations (4.5-7) into equation (4.4) results in:

$$L = \sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i,j=1}^{N} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) . \tag{4.8}$$

A set of α_i is obtained by maximizing of equation (4.8) with $0 \leq \alpha_i \leq C$. According to the Kuhn-Tucker complementarity's condition, the following equation should be true at the optimal solution (Park et al., 2005):

$$\alpha_i(\|\mathbf{x}_i - \boldsymbol{\mu}\|^2 - R^2 - \xi_i) = 0, \forall i \tag{4.9}$$

When a sample $\mathbf{x}_i$ satisfies the inequality $\|\mathbf{x}_i - \boldsymbol{\mu}\|^2 < R^2 + \xi_i$, that is, sample $\mathbf{x}_i$ is within the hypersphere, the corresponding Lagrange multiplier is zero, i.e., $\alpha_i = 0$ from equation (4.9). For samples satisfying the equality $\|\mathbf{x}_i - \boldsymbol{\mu}\|^2 = R^2 + \xi_i$ that are on the boundary or outside the sphere, the corresponding Lagrange multipliers are not zero, i.e., $\alpha_i > 0$. The center of the sphere $\boldsymbol{\mu}$ is a linear combination of the samples according to equation (4.6). SVDD needs only samples $\mathbf{x}_i$ with $\alpha_i>0$ which are called support vectors of the SVDD. Once the SVDD is constructed on the training data, we need to decide whether a given test sample z is normal or outlier. The criterion for the decision can be stated as:

$$f(z) = I(\|z - \boldsymbol{\mu}\|^2 \leq R^2) \tag{4.10}$$

where *I(condition)* equals to one if the condition is true and zero otherwise.

We can get a better description of normal data by incorporating attack samples in the SVDD training when they are available. Considering M attacks available in the training set, the problem in equation (4.2) with constraint (4.1) changes into:

$$
\begin{aligned}
\textit{Min.} \quad & R^2 + C_1 \sum_{i=1}^{N} \xi_i + C_2 \sum_{k=1}^{M} \xi_k \\
\textit{s.t.} \quad & \left\| \mathbf{x}_i - \boldsymbol{\mu} \right\|^2 \le R^2 + \xi_i, \qquad \xi_i \ge 0, \quad \forall i \\
& \left\| \mathbf{x}_k - \boldsymbol{\mu} \right\|^2 \ge R^2 - \xi_k, \qquad \xi_k \ge 0, \quad \forall k
\end{aligned}
\tag{4.11}
$$

where ξ_i and ξ_k are slack variables for normal data $\mathbf{x}_i$ and attack data $\mathbf{x}_k$.

The boundary of the hypersphere around the data is not flexible and often not a good description. For more flexible boundaries, inner products of samples $(\mathbf{x}_i \cdot \mathbf{x}_j)$ as shown in equation (4.8) is replaced by a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, where $K(\mathbf{x}_i, \mathbf{x}_j)$ satisfies Mercer's theorem (Schölkopf et al., 1998). This kernel trick implicitly carries out mapping samples into a nonlinear feature space to obtain a tighter boundary. By introducing kernel function instead of inner products, equation (4.8) can be expressed as:

$$
L = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j=1}^{N} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)
\tag{4.12}
$$

with constraints $0 \le \alpha_i \le C$ and $\sum_i \alpha_i = 1$. By the use of a kernel function, the execution of the nonlinear mappings and the dot products in a nonlinear feature space becomes unnecessary (Cortes & Vapnik, 1995). The most commonly used kernel functions are the Gaussian function as in equation (4.13) and the polynomial functions as in equation (4.14).

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / \sigma\right) \tag{4.13}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d \tag{4.14}$$

A boundary of the SVDD depends on which kernel function is used to train the description. Gaussian kernel produces tighter description than the polynomial function in SVDD according to Tax and Duin (2004).

The value of the regularization parameter C can be determined by using the false alarm rate, FA, for the target data and the number of observations, N:

$$C \leq \frac{1}{FA * N} \tag{4.15}$$

When C is set to 1, it requests the boundary which should accept all target data and reject all outlier data. Figure 4.1 shows the description obtained using Gaussian kernel for a simple two dimensional data set with outliers. The closed solid curve is the boundary which distinguishes normal data indicated with star symbols from attack data represented with plus signs. Normal data on the boundary indicate the support vectors of the description.

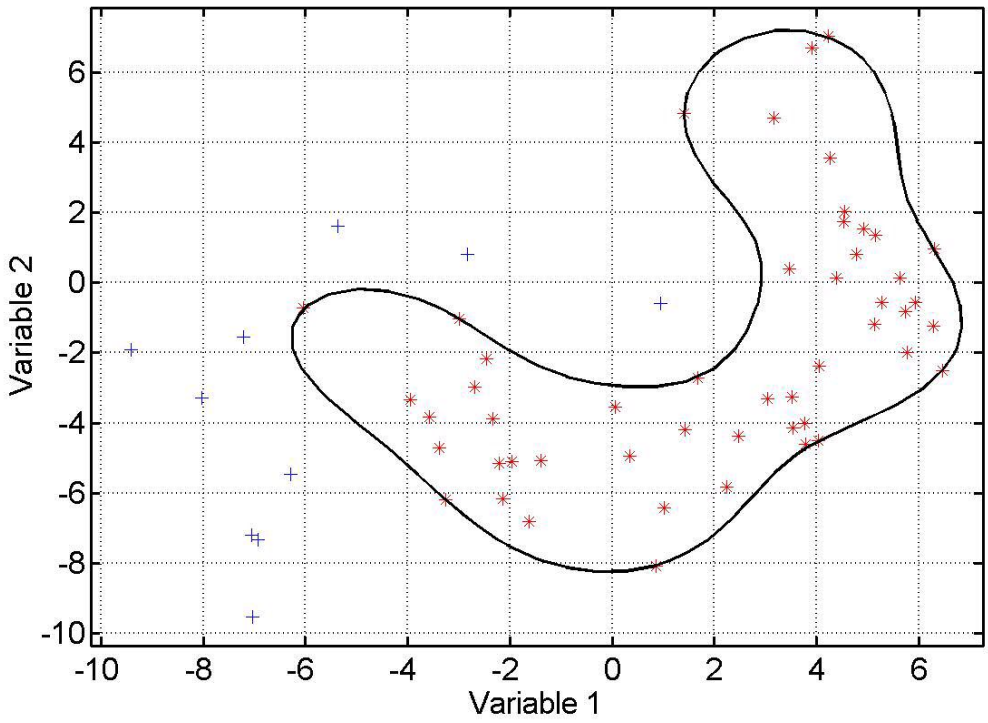


Figure 4.1 Example of data description trained with outliers

4.3 Formulation for Differentiated Intrusion Detection

Let us consider a training data set with N samples of normal data, L of targeted attack type and M of nontargeted attack type. When the targeted attack type needs to be more strictly detected than the nontargeted types, the problem in equation (4.11) is represented into:

$$
\begin{aligned}
Min. \quad & R^2 + C_1 \sum_{i=1}^{N} \xi_i + C_2 \sum_{j=1}^{L} \xi_j^* + C_3 \sum_{k=1}^{M} \xi_k^{**} \\
s.t \quad & \left\| \mathbf{x}_i - \boldsymbol{\mu} \right\|^2 \le R^2 + \xi_i, \qquad \xi_i \ge 0, \qquad \forall i \\
& \left\| \mathbf{x}_j^* - \boldsymbol{\mu} \right\|^2 \ge R^2 - \xi_j^*, \qquad \xi_j^* \ge 0, \qquad \forall j \\
& \left\| \mathbf{x}_k^{**} - \boldsymbol{\mu} \right\|^2 \ge R^2 - \xi_k^{**}, \qquad \xi_k^{**} \ge 0, \quad \forall k
\end{aligned}
\tag{4.16}
$$

where R and $\boldsymbol{\mu}$ are a radius and a center of the hypersphere, and C_1, C_2, C_3, ξ_i, ξ_j^*, ξ_k^{**} are regularization parameters and slack variables, respectively, for normal data $\mathbf{x}_i$, targeted attack type data $\mathbf{x}_j^*$ and nontargeted attack types data $\mathbf{x}_k^{**}$. Equation (4.16) changes into the Lagrangian function by using the Lagrange multipliers for its inequality conditions:

$$
\begin{aligned}
L(R, \boldsymbol{\mu}, \xi_i, \xi_j^*, \xi_k^{**}) = {} & R^2 + C_1 \sum_{i=1}^{N} \xi_i + C_2 \sum_{j=1}^{L} \xi_j^* + C_3 \sum_{k=1}^{M} \xi_k^{**} \\
& - \sum_{i=1}^{N} \alpha_i \{ R^2 + \xi_i - (\left\| \mathbf{x}_i \right\|^2 - 2\boldsymbol{\mu} \cdot \mathbf{x}_i + \left\| \boldsymbol{\mu} \right\|^2) \} - \sum_{i=1}^{N} \beta_i \xi_i \\
& - \sum_{j=1}^{L} \alpha_j^* \{ (\left\| \mathbf{x}_j^* \right\|^2 - 2\boldsymbol{\mu} \cdot \mathbf{x}_j^* + \left\| \boldsymbol{\mu} \right\|^2) - R^2 + \xi_j^* \} - \sum_{j=1}^{L} \beta_j^* \xi_j^* \\
& - \sum_{k=1}^{M} \alpha_k^{**} \{ (\left\| \mathbf{x}_k^{**} \right\|^2 - 2\boldsymbol{\mu} \cdot \mathbf{x}_k^{**} + \left\| \boldsymbol{\mu} \right\|^2) - R^2 + \xi_k^{**} \} - \sum_{k=1}^{M} \beta_k^{**} \xi_k^{**}
\end{aligned}
\tag{4.17}
$$

with the Lagrange multipliers $\alpha_i \ge 0$, $\beta_i \ge 0$, $\alpha_j^* \ge 0$, $\beta_j^* \ge 0$, $\alpha_k^{**} \ge 0$ and $\beta_k^{**} \ge 0$. The dual of equation (4.16) is:

$$
\begin{aligned}
&Max. \quad L(R, \boldsymbol{\mu}, \xi_i, \xi_j^*, \xi_k^{**}) \\
&s.t. \quad \frac{\partial L(R, \boldsymbol{\mu}, \xi_i, \xi_j^*, \xi_k^{**})}{\partial R} = 0 \\
&\frac{\partial L(R, \boldsymbol{\mu}, \xi_i, \xi_j^*, \xi_k^{**})}{\partial \boldsymbol{\mu}} = 0 \\
&\frac{\partial L(R, \boldsymbol{\mu}, \xi_i, \xi_j^*, \xi_k^{**})}{\partial \xi_i} = 0 \\
&\frac{\partial L(R, \boldsymbol{\mu}, \xi_i, \xi_j^*, \xi_k^{**})}{\partial \xi_j^*} = 0 \\
&\frac{\partial L(R, \boldsymbol{\mu}, \xi_i, \xi_j, \xi_k^{**})}{\partial \xi_k^{**}} = 0 \\
&\alpha_i \geq 0, \quad \beta_i \geq 0 \qquad \forall i \\
&\alpha_j^* \geq 0, \quad \beta_j^* \geq 0 \qquad \forall j \\
&\alpha_k^{**} \geq 0, \quad \beta_k^{**} \geq 0 \qquad \forall k
\end{aligned}
\tag{4.18}
$$

The terms with partial derivatives set to zero in equation (4.18) produce simplified forms. For the partial derivative with respect to R we get equation (4.19):

$$
\begin{aligned}
\frac{\partial L}{\partial R} &= 2R - 2R\sum_{i=1}^{N}\alpha_i + 2R\sum_{j=1}^{L}\alpha_j^* + 2R\sum_{k=1}^{M}\alpha_k^{**} = 0 \\
2R(1 &- \sum_{i=1}^{N}\alpha_i + \sum_{j=1}^{L}\alpha_j^* + \sum_{k=1}^{M}\alpha_k^{**}) = 0 \\
1 &- \sum_{i=1}^{N}\alpha_i + \sum_{j=1}^{L}\alpha_k^{**} + \sum_{k=1}^{M}\alpha_k^{**} = 0
\end{aligned}
\tag{4.19}
$$

$$
\sum_{i=1}^{N}\alpha_i - \left(\sum_{j=1}^{L}\alpha_j^* + \sum_{k=1}^{M}\alpha_k^{**}\right) = 1 \tag{4.20}
$$

We obtain equation (4.22) from the partial derivative with respect to $\boldsymbol{\mu}$:

$$\frac{\partial L}{\partial \boldsymbol{\mu}} = \frac{\partial}{\partial \boldsymbol{\mu}}\left(-\sum_{i=1}^{N}\alpha_i\left(2\boldsymbol{\mu}\cdot\mathbf{x}_i - \|\boldsymbol{\mu}\|^2\right) + \sum_{j=1}^{L}\alpha_j^*\left(2\boldsymbol{\mu}\cdot\mathbf{x}_j^* - \|\boldsymbol{\mu}\|^2\right) + \sum_{k=1}^{M}\alpha_k^{**}\left(2\boldsymbol{\mu}\cdot\mathbf{x}_k^{**} - \|\boldsymbol{\mu}\|^2\right)\right) = 0$$

$$-\sum_{i=1}^{N}\alpha_i\left(2\mathbf{x}_i - 2\boldsymbol{\mu}\right) + \sum_{j=1}^{L}\alpha_j^*\left(2\mathbf{x}_j^* - 2\boldsymbol{\mu}\right) + \sum_{k=1}^{M}\alpha_k^{**}\left(2\mathbf{x}_k^{**} - 2\boldsymbol{\mu}\right) = 0$$

$$-2\sum_{i=1}^{N}\alpha_i\mathbf{x}_i + 2\sum_{j=1}^{L}\alpha_j^*\mathbf{x}_j^* + 2\sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**} + \left(2\sum_{i=1}^{N}\alpha_i - 2\sum_{j=1}^{L}\alpha_j^* - 2\sum_{k=1}^{M}\alpha_k^{**}\right)\boldsymbol{\mu} = 0$$

$$2\left(\sum_{i=1}^{N}\alpha_i - \sum_{j=1}^{L}\alpha_j^* - \sum_{k=1}^{M}\alpha_k^{**}\right)\boldsymbol{\mu} = 2\left(\sum_{i=1}^{N}\alpha_i\mathbf{x}_i - \sum_{j=1}^{L}\alpha_j^*\mathbf{x}_j^* - \sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**}\right)$$

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{N}\alpha_i\mathbf{x}_i - \sum_{j=1}^{L}\alpha_j^*\mathbf{x}_j^* - \sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**}}{\sum_{i=1}^{N}\alpha_i - \sum_{j=1}^{L}\alpha_j^* - \sum_{k=1}^{M}\alpha_k^{**}} \tag{4.21}$$

$$\boldsymbol{\mu} = \sum_{i=1}^{N}\alpha_i\mathbf{x}_i - \left(\sum_{j=1}^{L}\alpha_j^*\mathbf{x}_j^* + \sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**}\right) \tag{4.22}$$

For the partial derivative with respect to ξ_i, ξ_j^* and ξ_k^{**}:

$$\frac{\partial L}{\partial \xi_i} = C_1 - \alpha_i - \beta_i = 0 : \qquad C_1 - \alpha_i - \beta_i = 0$$

$$\frac{\partial L}{\partial \xi_j^*} = C_2 - \alpha_j^* - \beta_j^* = 0 : \qquad C_2 - \alpha_j^* - \beta_j^* = 0 \tag{4.23}$$

$$\frac{\partial L}{\partial \xi_k^{**}} = C_3 - \alpha_k^{**} - \beta_k^{**} = 0 : \qquad C_3 - \alpha_k^{**} - \beta_k^{**} = 0$$

Since $\alpha_i = C_1 - \beta_i$ from equation (4.23), $\alpha_i \geq 0$, and $\beta_i \geq 0$, we can remove the Lagrange multipliers β_i by producing $0 \leq \alpha_i \leq C_1$. By doing same processes for α_j^*, β_j^*, α_k^{**} and β_k^{**}, $0 \leq \alpha_j^* \leq C_2$, $0 \leq \alpha_k^{**} \leq C_3$.

By incorporating equations (4.20), (4.22) and (4.23) into $L(R, \mathbf{\mu}, \xi_i, \xi_j^*, \xi_k^{**})$ of equation (4.18), L results in the function of only α_i, α_j^*, α_k^{**}, $\mathbf{x}_i$, $\mathbf{x}_j^*$ and $\mathbf{x}_k^{**}$. The terms of R in L reduce to zero:

$$R^2 - \sum_{i=1}^{N} \alpha_i R^2 + \sum_{j=1}^{L} \alpha_j^* R^2 + \sum_{k=1}^{M} \alpha_k^{**} R^2 = R^2 \left(1 - \sum_{i=1}^{N} \alpha_i + \sum_{j=1}^{L} \alpha_j^* + \sum_{k=1}^{M} \alpha_k^{**} \right) = 0 \tag{4.24}$$

Also, the terms of ξ_i, ξ_j^* and ξ_k^{**} in L become to zero:

$$C_1\xi_i - \alpha_i\xi_i - \beta_i\xi_i = (C_1 - \alpha_i - \beta_i)\xi_i = 0$$

$$C_2\xi_j^* - \alpha_j^*\xi_j^* - \beta_j^*\xi_j^* = (C_2 - \alpha_j^* - \beta_j^*)\xi_j^* = 0 \tag{4.25}$$

$$C_3\xi_k^{**} - \alpha_k^{**}\xi_k^{**} - \beta_k^{**}\xi_k^{**} = (C_3 - \alpha_k^{**} - \beta_k^{**})\xi_k^{**} = 0$$

After removing the terms of R, ξ_i, ξ_j^* and ξ_k^{**} in $L(R, \boldsymbol{\mu}, \xi_i, \xi_j^*, \xi_k^{**})$ of equation (4.18), it is as follows then:

$$\begin{aligned} L = &\sum_{i=1}^{N}\alpha_i(\|\mathbf{x}_i\|^2 - 2\boldsymbol{\mu}\cdot\mathbf{x}_i + \|\boldsymbol{\mu}\|^2) - \sum_{j=1}^{L}\alpha_j^*(\|\mathbf{x}_j^*\|^2 - 2\boldsymbol{\mu}\cdot\mathbf{x}_j^* + \|\boldsymbol{\mu}\|^2) \\ &- \sum_{k=1}^{M}\alpha_k^{**}(\|\mathbf{x}_k^{**}\|^2 - 2\boldsymbol{\mu}\cdot\mathbf{x}_k^{**} + \|\boldsymbol{\mu}\|^2) \end{aligned} \tag{4.26}$$

$$\begin{aligned} L = &\sum_{i=1}^{N}\alpha_i\|\mathbf{x}_i\|^2 - \sum_{j=1}^{L}\alpha_j^*\|\mathbf{x}_j^*\|^2 - \sum_{k=1}^{M}\alpha_k^{**}\|\mathbf{x}_k^{**}\|^2 \\ &- 2\sum_{i=1}^{N}\alpha_i\boldsymbol{\mu}\cdot\mathbf{x}_i + 2\sum_{j=1}^{L}\alpha_j^*\boldsymbol{\mu}\cdot\mathbf{x}_j^* + 2\sum_{k=1}^{M}\alpha_k^{**}\boldsymbol{\mu}\cdot\mathbf{x}_k^{**} \\ &+ \left(\sum_{i=1}^{N}\alpha_i - \sum_{j=1}^{L}\alpha_j^* - \sum_{k=1}^{M}\alpha_k^{**}\right)\|\boldsymbol{\mu}\|^2 \end{aligned} \tag{4.27}$$

After replacing $\boldsymbol{\mu}$ with equation (4.22) the second line on equation (4.27) is:

$$
\begin{aligned}
& -2\sum_{i=1}^{N}\alpha_i\boldsymbol{\mu}\cdot\mathbf{x}_i+2\sum_{j=1}^{L}\alpha_j^{*}\boldsymbol{\mu}\cdot\mathbf{x}_j^{*}+2\sum_{k=1}^{M}\alpha_k^{**}\boldsymbol{\mu}\cdot\mathbf{x}_k^{**} \\
= \; & -2\sum_{i=1}^{N}\alpha_i\left(\sum_{i=1}^{N}\alpha_i\mathbf{x}_i-\sum_{j=1}^{L}\alpha_j^{*}\mathbf{x}_j^{*}-\sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**}\right)\mathbf{x}_i+2\sum_{j=1}^{L}\alpha_j^{*}\left(\sum_{i=1}^{N}\alpha_i\mathbf{x}_i-\sum_{j=1}^{L}\alpha_j^{*}\mathbf{x}_j^{*}-\sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**}\right)\mathbf{x}_j^{*} \\
& +2\sum_{k=1}^{M}\alpha_k^{**}\left(\sum_{i=1}^{N}\alpha_i\mathbf{x}_i-\sum_{j=1}^{L}\alpha_j^{*}\mathbf{x}_j^{*}-\sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**}\right)\mathbf{x}_k^{**} \\
= \; & -2\sum_{i=1}^{N}\sum_{p=1}^{N}\alpha_i\alpha_p\mathbf{x}_i\cdot\mathbf{x}_p+2\sum_{i=1}^{N}\sum_{j=1}^{L}\alpha_i\alpha_j^{*}\mathbf{x}_i\cdot\mathbf{x}_j^{*}+2\sum_{i=1}^{N}\sum_{k=1}^{M}\alpha_i\alpha_k^{**}\mathbf{x}_i\cdot\mathbf{x}_k^{**}+2\sum_{j=1}^{L}\sum_{i=1}^{N}\alpha_j^{*}\alpha_i\mathbf{x}_j^{*}\cdot\mathbf{x}_i \\
& -2\sum_{j=1}^{L}\sum_{q=1}^{L}\alpha_j^{*}\alpha_q^{*}\mathbf{x}_j^{*}\cdot\mathbf{x}_q^{*}-2\sum_{j=1}^{L}\sum_{k=1}^{M}\alpha_j^{*}\alpha_k^{**}\mathbf{x}_j^{*}\cdot\mathbf{x}_k^{**}+2\sum_{k=1}^{M}\sum_{i=1}^{N}\alpha_k^{**}\alpha_i\mathbf{x}_k^{**}\cdot\mathbf{x}_i-2\sum_{k=1}^{M}\sum_{j=1}^{L}\alpha_k^{**}\alpha_j^{*}\mathbf{x}_k^{**}\cdot\mathbf{x}_j^{*} \\
& -2\sum_{k=1}^{M}\sum_{r=1}^{M}\alpha_k^{**}\alpha_r^{**}\mathbf{x}_k^{**}\cdot\mathbf{x}_r^{**}
\end{aligned}
\tag{4.28}
$$

And the third line on equation (4.27) is reduced to $\|\boldsymbol{\mu}\|^2$ by equation (4.20). The remaining $\|\boldsymbol{\mu}\|^2$ is as the following:

$$
\begin{aligned}
\|\boldsymbol{\mu}\|^2 &= \boldsymbol{\mu}\cdot\boldsymbol{\mu} \\
&= \left(\sum_{i=1}^{N}\alpha_i\mathbf{x}_i-\sum_{j=1}^{L}\alpha_j^{*}\mathbf{x}_j^{*}-\sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**}\right)\cdot\left(\sum_{i=1}^{N}\alpha_i\mathbf{x}_i-\sum_{j=1}^{L}\alpha_j^{*}\mathbf{x}_j^{*}-\sum_{k=1}^{M}\alpha_k^{**}\mathbf{x}_k^{**}\right) \\
&= \sum_{i=1}^{N}\sum_{p=1}^{N}\alpha_i\alpha_p\mathbf{x}_i\cdot\mathbf{x}_p+\sum_{j=1}^{L}\sum_{q=1}^{L}\alpha_j^{*}\alpha_q^{*}\mathbf{x}_j^{*}\cdot\mathbf{x}_q^{*}+\sum_{k=1}^{M}\sum_{r=1}^{M}\alpha_k^{**}\alpha_r^{**}\mathbf{x}_k^{**}\cdot\mathbf{x}_r^{**} \\
&\quad -2\sum_{i=1}^{N}\sum_{j=1}^{L}\alpha_i\alpha_j^{*}\mathbf{x}_i\cdot\mathbf{x}_j^{*}-2\sum_{i=1}^{N}\sum_{k=1}^{M}\alpha_i\alpha_k^{**}\mathbf{x}_i\cdot\mathbf{x}_k^{**}+2\sum_{j=1}^{L}\sum_{k=1}^{M}\alpha_j^{*}\alpha_k^{**}\mathbf{x}_j^{*}\cdot\mathbf{x}_k^{**}
\end{aligned}
\tag{4.29}
$$

L in equation (4.27) is rewritten as:

$$
\begin{aligned}
L = & \sum_{i=1}^{N} \alpha_i \left\| \mathbf{x}_i \right\|^2 - \sum_{j=1}^{L} \alpha_j^* \left\| \mathbf{x}_j^* \right\|^2 - \sum_{k=1}^{M} \alpha_k^{**} \left\| \mathbf{x}_k^{**} \right\|^2 \\
& - \sum_{i=1}^{N} \sum_{p=1}^{N} \alpha_i \alpha_p \mathbf{x}_i \cdot \mathbf{x}_p - \sum_{j=1}^{L} \sum_{q=1}^{L} \alpha_j^* \alpha_q^* \mathbf{x}_j^* \cdot \mathbf{x}_q^* - \sum_{k=1}^{M} \sum_{r=1}^{M} \alpha_k^{**} \alpha_r^{**} \mathbf{x}_k^{**} \cdot \mathbf{x}_r^{**} \\
& + 2 \sum_{j=1}^{L} \sum_{i=1}^{N} \alpha_j^* \alpha_i \mathbf{x}_j^* \cdot \mathbf{x}_i + 2 \sum_{k=1}^{M} \sum_{i=1}^{N} \alpha_k^{**} \alpha_i \mathbf{x}_k^{**} \cdot \mathbf{x}_i - 2 \sum_{k=1}^{M} \sum_{j=1}^{L} \alpha_k^{**} \alpha_j^* \mathbf{x}_k^{**} \cdot \mathbf{x}_j^* \\
= & \sum_{i=1}^{N} \alpha_i \left(\mathbf{x}_i \cdot \mathbf{x}_i \right) - \sum_{j=1}^{L} \alpha_j^* \left(\mathbf{x}_j^* \cdot \mathbf{x}_j^* \right) - \sum_{k=1}^{M} \alpha_k^{**} \left(\mathbf{x}_k^{**} \cdot \mathbf{x}_k^{**} \right) - \sum_{i=1}^{N} \sum_{p=1}^{N} \alpha_i \alpha_p \left(\mathbf{x}_i \cdot \mathbf{x}_p \right) \\
& - \sum_{j=1}^{L} \sum_{q=1}^{L} \alpha_j^* \alpha_q^* \left(\mathbf{x}_j^* \cdot \mathbf{x}_q^* \right) - \sum_{k=1}^{M} \sum_{r=1}^{M} \alpha_k^{**} \alpha_r^{**} \left(\mathbf{x}_k^{**} \cdot \mathbf{x}_r^{**} \right) + 2 \sum_{j=1}^{L} \sum_{i=1}^{N} \alpha_j^* \alpha_i \left(\mathbf{x}_j^* \cdot \mathbf{x}_i \right) \\
& + 2 \sum_{k=1}^{M} \sum_{i=1}^{N} \alpha_k^{**} \alpha_i \left(\mathbf{x}_k^{**} \cdot \mathbf{x}_i \right) - 2 \sum_{k=1}^{M} \sum_{j=1}^{L} \alpha_k^{**} \alpha_j^* \left(\mathbf{x}_k^{**} \cdot \mathbf{x}_j^* \right)
\end{aligned}
\tag{4.30}
$$

The equation (4.18) is transformed into:

$$
\begin{aligned}
\underset{\alpha_i, \alpha_j, \alpha_k}{Max.} \quad & \sum_{i=1}^{N} \alpha_i \left(\mathbf{x}_i \cdot \mathbf{x}_i \right) - \sum_{j=1}^{L} \alpha_j^* \left(\mathbf{x}_j^* \cdot \mathbf{x}_j^* \right) - \sum_{k=1}^{M} \alpha_k^{**} \left(\mathbf{x}_k^{**} \cdot \mathbf{x}_k^{**} \right) - \sum_{i=1}^{N} \sum_{p=1}^{N} \alpha_i \alpha_p \left(\mathbf{x}_i \cdot \mathbf{x}_p \right) \\
& - \sum_{j=1}^{L} \sum_{q=1}^{L} \alpha_j^* \alpha_q^* \left(\mathbf{x}_j^* \cdot \mathbf{x}_q^* \right) - \sum_{k=1}^{M} \sum_{r=1}^{M} \alpha_k^{**} \alpha_r^{**} \left(\mathbf{x}_k^{**} \cdot \mathbf{x}_r^{**} \right) + 2 \sum_{j=1}^{L} \sum_{i=1}^{N} \alpha_j^* \alpha_i \left(\mathbf{x}_j^* \cdot \mathbf{x}_i \right) \\
& + 2 \sum_{k=1}^{M} \sum_{i=1}^{N} \alpha_k^{**} \alpha_i \left(\mathbf{x}_k^{**} \cdot \mathbf{x}_i \right) - 2 \sum_{k=1}^{M} \sum_{j=1}^{L} \alpha_k^{**} \alpha_j^* \left(\mathbf{x}_k^{**} \cdot \mathbf{x}_j^* \right) \\
s.t. \quad & 0 \le \alpha_i \le C_1, \forall i, \qquad 0 \le \alpha_j^* \le C_2, \forall j, \qquad 0 \le \alpha_k^{**} \le C_3, \forall k
\end{aligned}
\tag{4.31}
$$

The solution of equation (4.31) is a set of values for $\alpha_i, \alpha_j^*, \alpha_k^{**}$. The following relation between the Lagrange multipliers $\alpha_i, \alpha_j^*, \alpha_k^{**}$ and the constraints of equation (4.16) are always true according to the Kuhn-Tucker condition:

$$\begin{aligned}
&\alpha_i(\|\mathbf{x}_i - \boldsymbol{\mu}\|^2 - R^2 - \xi_i) = 0, \forall i \\
&\alpha_j^*(\|\mathbf{x}_j^* - \boldsymbol{\mu}\|^2 - R^2 + \xi_j^*) = 0, \forall j \\
&\alpha_k^{**}(\|\mathbf{x}_k^{**} - \boldsymbol{\mu}\|^2 - R^2 + \xi_k^{**}) = 0, \forall k
\end{aligned} \tag{4.32}$$

From equation (4.32), it is true that a normal observation $\mathbf{x}_i$ satisfies the constraint of equation (4.16) when the corresponding Lagrange multiplier α_i=0. $\mathbf{x}_i$ with α_i=0 is inside the boundary, $\mathbf{x}_i$ with $0 < \alpha_i < C_1$ is on the boundary, and $\mathbf{x}_i$ with $\alpha_i = C_1$ is outside the boundary:

$$\begin{aligned}
&\alpha_i = 0 \quad \rightarrow \quad \|\mathbf{x}_i - \boldsymbol{\mu}\|^2 < R^2, \quad \xi_i = 0 \\
&0 < \alpha_i < C_1 \quad \rightarrow \quad \|\mathbf{x}_i - \boldsymbol{\mu}\|^2 = R^2, \quad \xi_i = 0 \\
&\alpha_i = C_1 \quad \rightarrow \quad \|\mathbf{x}_i - \boldsymbol{\mu}\|^2 > R^2, \quad \xi_i > 0
\end{aligned} \tag{4.33}$$

For the targeted attack samples, $\mathbf{x}_j^*$, an observation with α_j^*=0 is outside the boundary, one with $0 < \alpha_j^* < C_2$ on the boundary, and the other with $\alpha_j^* = C_2$ inside the boundary:

$$\begin{aligned}
&\alpha_j^* = 0 \quad \rightarrow \quad \|\mathbf{x}_j^* - \boldsymbol{\mu}\|^2 > R^2, \quad \xi_j^* = 0 \\
&0 < \alpha_j^* < C_2 \quad \rightarrow \quad \|\mathbf{x}_j^* - \boldsymbol{\mu}\|^2 = R^2, \quad \xi_j^* = 0 \\
&\alpha_j^* = C_2 \quad \rightarrow \quad \|\mathbf{x}_j^* - \boldsymbol{\mu}\|^2 < R^2, \quad \xi_j^* > 0
\end{aligned} \tag{4.34}$$

The same relation holds for the nontargeted attack type, $\mathbf{x}_j^*$:

$$
\begin{aligned}
&\alpha_k^{**} = 0 \quad \rightarrow \quad \left\|\mathbf{x}_k^{**} - \boldsymbol{\mu}\right\|^2 > R^2, \quad \xi_k^{**} = 0 \\
&0 < \alpha_k^{**} < C_3 \quad \rightarrow \quad \left\|\mathbf{x}_k^{**} - \boldsymbol{\mu}\right\|^2 = R^2, \quad \xi_k^{**} = 0 \\
&\alpha_k^{**} = C_3 \quad \rightarrow \quad \left\|\mathbf{x}_k^{**} - \boldsymbol{\mu}\right\|^2 < R^2, \quad \xi_k^{**} > 0
\end{aligned}
\tag{4.35}
$$

The following lemma is derived to explain the relationship between the number of detected targeted attacks and its regularization parameter value. This lemma can be used as a guideline to determine the appropriate value of regularization parameters for a given level of detection for the targeted attack type.

Lemma 1: Let N_f, L_u and M_u be the number of false alarms of normal data, the number of undetected attacks of the targeted type and nontargeted types, respectively. Then, the following relationship holds between these parameters and the regularization parameters:

$$
C_2 = \frac{1}{L_u}\left(N_f C_1 - M_u C_3\right) + \text{constant} \tag{4.36}
$$

Proof: Summation of the Lagrange multipliers α_i, α_j^* and α_k^{**} for observations can be rewritten:

$$\sum_{i=1}^{N}\alpha_i = \sum_{0<\alpha_i<C_1}\alpha_i + N_f C_1 \quad \text{for normal data}$$
$$\sum_{j=1}^{L}\alpha_j^* = \sum_{0<\alpha_j^*<C_2}\alpha_j^* + L_u C_2 \quad \text{for targeted attack type} \qquad (4.37)$$
$$\sum_{k=1}^{M}\alpha_k^{**} = \sum_{0<\alpha_k^{**}<C_3}\alpha_k^{**} + M_u C_3 \quad \text{for nontargeted attack types}$$

The equation (4.20) combined with the equation (4.37) changes:

$$\left(\sum_{0<\alpha_i<C_1}\alpha_i + N_f C_1\right) - \left(\sum_{0<\alpha_j^*<C_2}\alpha_j^* + L_u C_2\right) - \left(\sum_{0<\alpha_k^{**}<C_3}\alpha_k^{**} + M_u C_3\right) = 1 \qquad (4.38)$$

From the equation (4.38), the following relationship holds:

$$C_2 = \frac{1}{L_u}\left(N_f C_1 - M_u C_3\right) + \frac{1}{L_u}\left(\sum_{0<\alpha_i<C_1}\alpha_i - \sum_{0<\alpha_j^*<C_2}\alpha_j^* - \sum_{0<\alpha_k^{**}<C_3}\alpha_k^{**} - 1\right) \qquad (4.39)$$

Assuming that the terms in the second bracket of the equation are constant, the equation is reduced into:

$$C_2 = \frac{1}{L_u}\left(N_f C_1 - M_u C_3\right) + const \qquad \blacksquare$$

It is obvious from the lemma 1 that the number of detections of a targeted attack type increases by raising the value of its regularization parameter and by fixing the regularization parameters for normal data and nontargeted types of attacks. The lemma is a theoretical basis for the proposed differentiated anomaly intrusion detection of the targeted type of attack.

Figure 4.2 shows the effect of the differentiated detection on data set where targeted attack type locates in the opposite place of the nontargeted attack type across the normal data. The dashed boundary of differentiated detection detects more attacks of targeted type but less attacks of nontargeted type than the solid line boundary of the ordinary detection. As seen in figure, differentiated detection moved the boundary away from the targeted attack type, that is, toward the nontargeted attack type and the resulted boundary enclosed more attack data of nontargeted type indicated with small circles.

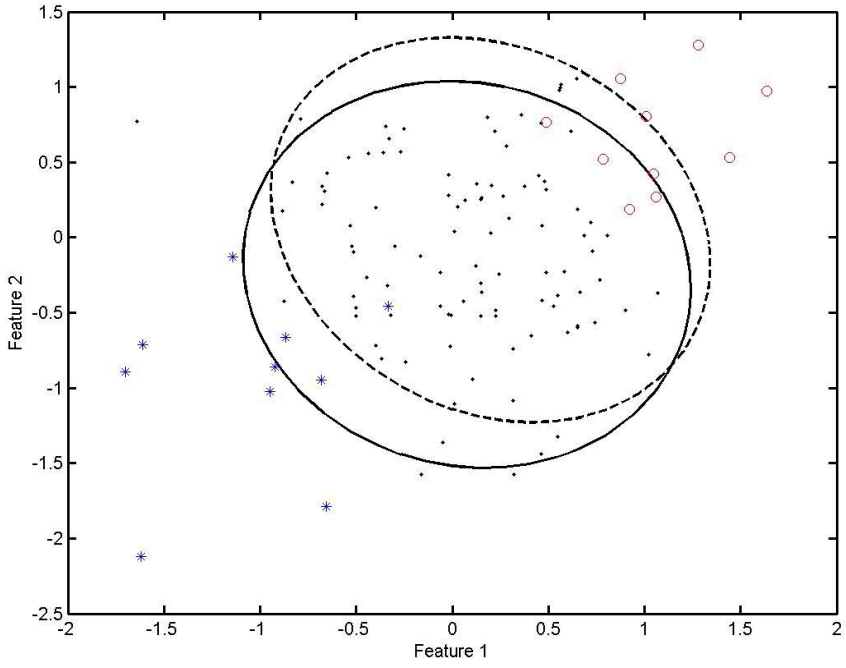


Figure 4.2 Comparison of the ordinary detection boundary with the boundary for the differentiated intrusion detection. Solid line boundary(ordinary detection), dashed line boundary(differentiated detection), dots(normal data), asterisks(targeted attack), tiny circles(nontargeted attacks)

4.4 Differentiated Anomaly Intrusion Detection

4.4.1 Selecting magnitudes of regularization parameter C_2

The magnitude of regularization parameter for targeted attack represents the power of the differentiated detection on the attack. Larger magnitude forces a classifier to detect more intrusions of the attack type. Basic unit of the magnitude is the value of *1/[(1 – desired detection rate for all attacks) × (number of attack data in the training data)]* which is ordinarily set in SVDD training. One unit magnitude of the parameter for targeted attack type means that all attack types are treated with same importance. The magnitude of larger than one unit gives more weights on targeted attack detection than the nontargeted attack type detection. According to our pre-experiments about the maximum magnitude with practical meanings, it was between 200 and 500 at most. After the magnitude reaches the value, there is no change in the differentiated detection results no matter how large the magnitude is over it. Also, the performance of differentiated detection is sensitive on the small magnitudes such as two and three units rather than large value like 100 units. For the differentiated detection to find the best magnitude, we recommend setting its range from two to 500 units and using detailed intervals especially on the smaller values.

4.4.2 Monotonic increase and number of training data

According to the lemma 1, monotonic increase on detections of targeted attacks is expected as a regular result of rising regularization parameter. However, it rarely happens in the real situation with limited number of data which does not distribute evenly. Only

when there are so many data of targeted attack and normal near the boundary that the classifiers from the differentiated detection can be elaborate, the monotonic rising would be possible. It is recommended for finer classifiers of the differentiated detection that as many available data as possible are used for the training.

4.4.3 SVDD parameters

For training its classifiers, SVDD requires parameters to be determined such as desired false alarm rate for normal data, desired detection rate of all attacks, type of kernel, and parameters of the kernel. Usually, desired false alarm rate and detection rate are set to 5% and 90%, respectively. The most commonly used kernel functions are the Gaussian function and the polynomial function. The each parameter for the kernels such as the bandwidth of Gaussian, σ, and the degree of the polynomial, d, is decided to yield the best results by using cross validation on the training data. Selecting a proper kernel type for a data set depends on the nature of the data although Gaussian kernel is better than the polynomial function in SVDD according to Tax and Duin (2004). The kernel showing better result is selected after training the classifier for each kernel function.

4.4.4 Selecting the level of differentiated detection

The differentiated detection of a targeted attack type could produce negative effects for identifying normal data and nontargeted attack types, despite serving best for detection of the targeted attack type. Sometimes, we cannot allow a huge number of false alarms and undergo failures to detect nontargeted types of attacks for the sake of only tiny improvement in detection of a targeted attack. We need to decide what level of

differentiated detection is reasonable while also considering its positive and negative effects. This can be determined by calculating the gain and the loss that will result from increased detection of targeted attacks or from failing to detect them. The gain is the savings from detecting targeted attacks that otherwise would have resulted in losses. The loss can be measured in the costs of handling additional false alarms and undetected non-targeted types of attacks. Let the benefit function of the gain and the loss from regularization parameter magnitude i be $B(i)$. The best level of differentiated detection is i^* that maximizes the value of $B(i)$ over all i.

4.4.5 Steps of differentiated detection

Step 1. Identifying targeted attack type: Identify which attack type is most harmful to the information system and requires differentiated detection.

Step 2. Data preparation: Prepare a data set to be used for the differentiated detection. Based on the required feature formats, data is collected separately for normal activities, the targeted attack type, and nontargeted attack types.

Step 3. Selecting parameters: SVDD parameters, magnitude intervals, and the range of the regularization parameter of the targeted attack need to be given proper values for differentiated detection.

Step 4. Running the model up to convergence: Run the SVDD-based differentiated detection model and collect the results, such as the number of additional detections of targeted attacks and nontargeted attack types, and the increased number of false alarms. If

the results do not converge, raise the maximum magnitude of the regularization parameter, insert more intervals between the existing and new maximum magnitudes, and run the model again. Repeat until the results show convergence.

Step 5. Choosing the best classifier among trained SVDD classifiers: If the benefit function of gain and loss that results from differentiated detection is known, select, based on the results of step 4, the classifier with the maximum benefit function value. Otherwise, choose from among the trained classifiers the classifier showing the largest number of detections of the targeted type of attacks.

4.5 Experiments of Differentiated Intrusion Detection

4.5.1 Experimental setup

The experiment for the proposed differentiated intrusion detection method was conducted by using two data sets: simulated data and the same data as used in the experiment for the new framework. The simulated data was artificially generated from normal and two attack classes in a two-dimensional data space for the experiment. The center of normal class located at (0, 0) and two attack class centers were at (1, 1) and (-1, -1), respectively. Two components of all the samples were independently corrupted by Gaussian noise with standard deviations 0.2 and 0.24.

The performance of the differentiated intrusion detection was demonstrated with the simulated data set in which there were four training data sets and one testing data set as

seen in table 4.2. There were two factors related to the differentiated detection performance: relative magnitude of the regularization parameter for targeted attack type compared to one for nontargeted attack types and the number of samples of targeted attack in the training data. The magnitude of targeted attack's regularization parameter was set to twelve magnitudes ranging two to 500 times bigger than the nontargeted attack's. For the number of targeted attack samples, four different training data groups with 10, 20, 30, and 40 samples of targeted attack, respectively, were considered and 10 data sets were sampled for each group, thereby getting 40 training data sets in total. Then a SVDD classifier was trained with each of 40 data sets combined with one of the twelve regularization parameter's magnitudes. By applying its classifier into the testing data, the result for each data set was measured as the number of detections on targeted attack, the number of false alarms and the number of detections on nontargeted attack. The performance of the differentiated detection for each four training data group was the summation of its ten results. 40 runs of SVDD were required to get desired results from the simulated data. The pre-processed DARPA-MITLL 1998 BSM data in table 3.9 was used to check how well the differentiated intrusion detection works in the real situation.

In this experiment, all the SVDD trainings were performed with outliers since there were attack data available. Polynomial kernel with degree of one was used in training SVDD for the DARPA data set whereas Gaussian kernel was used for the simulated data set because they achieved better performance. In all the experiments, the fraction rejection

Table 4.2 Number of data sets and samples in the simulated data

		Training data				Testing data
		Group #1	Group #2	Group #3	Group #4	
Number of data sets		10	10	10	10	1
Number of samples per data set	Normal	200	200	200	200	2000
	Targeted attack	10	20	30	40	400
	Nontargeted attack	10	20	30	40	400

for SVDD was set to 0.05, where 5% of normal data is expected to lie on or out of the boundary of classifier. The data description toolbox (dd_tools) 1.4.0 of Tax (2005) was modified and used as our SVDD running tool.

4.5.2 Results on simulated data

The proposed differentiated detection was effective on all the training data sets. The detection of targeted attacks was improved by the differentiated detection on the targeted type, showing 54 to 188 more detections compared to the ordinary intrusion detection as seen in table 4.3.

The extent of the improvement became larger along with increasing number of targeted attack samples in the training data. For example, 5.1% improvement was achieved in the training data set number four with 40 targeted attack samples while only 1.5% improvement in detection number came out from the group number one with 10 samples.

Table 4.3 Number of detected samples of targeted attack by the differentiated detection on the targeted type with four training data set groups.

Number of detections on targeted attack type	Training data set group			
	#1	#2	#3	#4
Maximum in the weighed detection	3,665	3,691	3,829	3,874
At ordinary detection*	3,611	3,635	3,653	3,686
Improvement compared to the Ordinary** (percent)	54 (1.5%)	56 (1.5%)	176 (4.8%)	188 (5.1%)

* Ordinary detection was performed with the same weights, magnitude=1, for both of targeted type and nontargeted attack type

** (Maximum number of detections on targeted attack type in the weighed detection) – (Number of detections on targeted attack type in the ordinary detection)

Figure 4.3 depicts the trends of changing detection rates on the targeted attack resulted from magnitude increases for all the four data sets. As the number of targeted attack samples increased in the training data, detection rate improved more smoothly with less deterioration. In case of training with 10 targeted type samples, there were only four time improvements and four time deteriorations on detection rate out of twelve differentiated detections. This contrasts to the results of training with 40 samples that all the differentiated detections showed better detection rates than the ordinary detection and there were only three little deteriorations. Another finding was that detection rates converged earlier with smaller number of attack samples in the training data. As seen in figure 4.3, the convergences appeared at magnitude 20 in data group number one, at 50 in data two, at 100 in data three, and at 200 in data number four. The value of magnitude for the convergence is expected to exist between 100 and 200 based on our experimental results.

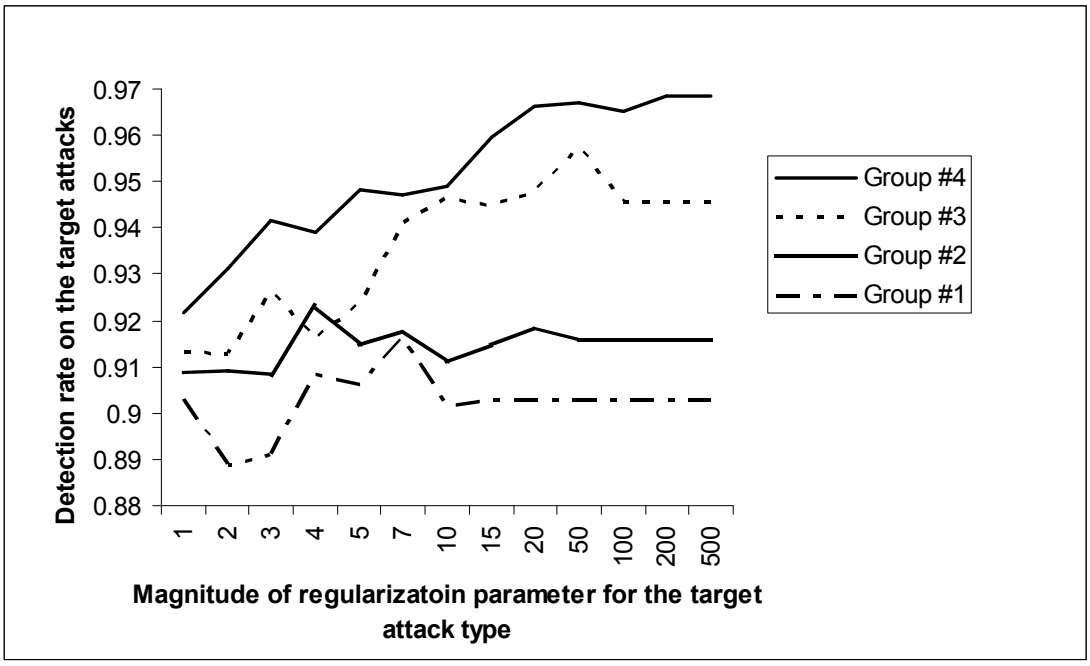


Figure 4.3 Change on detection rates of targeted attack type along with increasing magnitudes of regularization parameter of the targeted type according to differentiated detections on the type with four training data sets.

The differentiated detection with the training data group four showed the most similar results as our expectations based on the lemma. More detection of targeted attacks, more false alarms, and less detection of the nontargeted attacks which locate in the opposite across the normal data were expected as responses to increasing magnitudes in the differentiated detection. Figure 4.4 depicts the results of differentiated detections on the group four data. As the magnitude of the regularization parameter for targeted attack type increased, targeted attack detection increased almost monotonically, nontargeted attacks were gradually less detected, and false alarms increased roughly. It is obvious that those results are consistent with the expectation. More samples of targeted attack type are required to get much finer results of differentiated intrusion detection on the type.

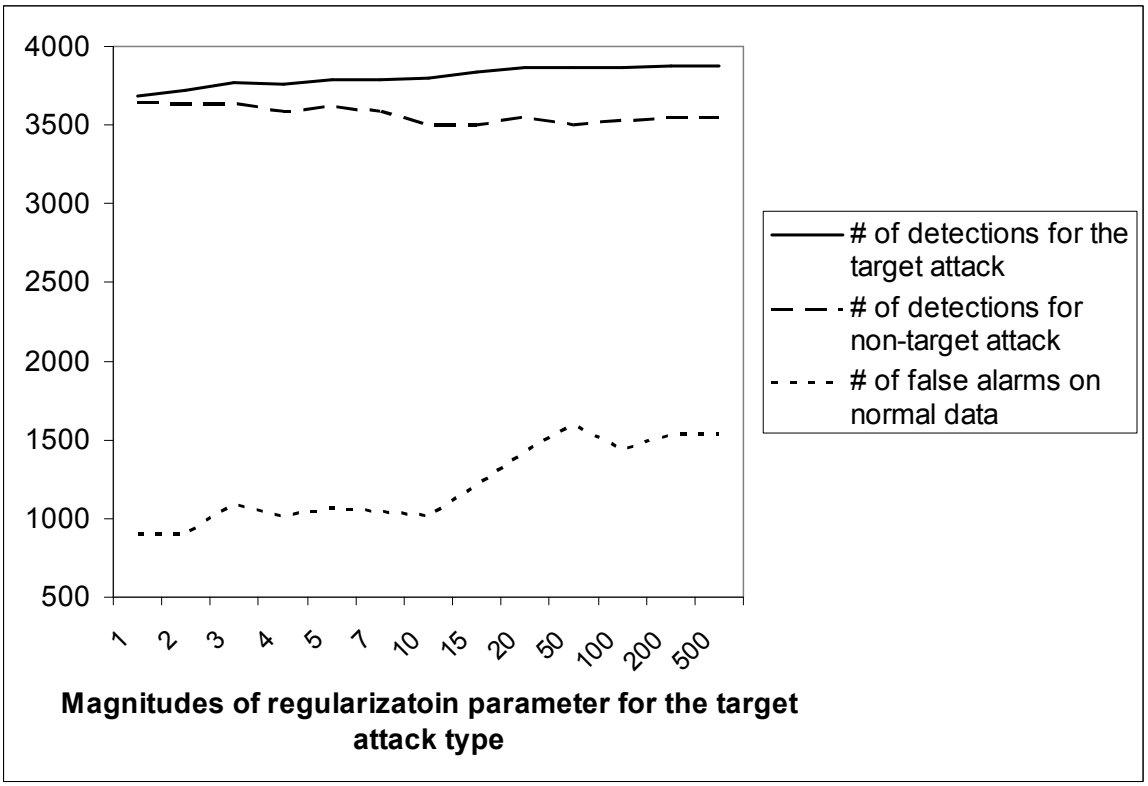


Figure 4.4 Changing trend of number of detections on both targeted attack type and nontargeted attack type and number of false alarms of normal data with increasing magnitudes of regularization parameter of the targeted type resulted from differentiated detections on the targeted type with training data group number four

4.5.3 Results on the DARPA data

The differentiated detection on U2R attack type detected at most six more attacks. U2R attack type was selected to apply the differentiated detection since it was thought the most harmful attack type in our data based on the 1998 DARPA BSM data set. Table 4.4 shows the results for the differentiated detection of U2R attack type. When the magnitude was four, the differentiated detection produced the largest additional detections of U2R attack with 22 more false alarms. Also it detected five additional attacks of the nontargeted types, which means that nontargeted types of attacks locate near U2R and the differentiated detection on U2R is effective on them too. However, there was no further improvement on differentiated detection of U2R after magnitudes reached to four. The convergence of results appeared from the magnitude 50. Effective results of the differentiated detection on U2R came out in only three cases of 2, 4, and 15 magnitudes. From the remaining nine cases of all the twelve magnitudes, we found interesting results, decreased false alarms without any negative impacts on detecting attacks. It is noticeable that these results are better than ordinary intrusion detection because of their less false alarms with same detection capability.

Table 4.4 Results for the differentiated detection on U2R attack type compared to ordinary detection on the type

Regularization parameter magnitudes for the targeted attack type, U2R	Additionally detected number of U2R attacks by the differentiated detection	Number of additional false alarms resulted from the differentiated detection	Additionally detected number of nontargeted attack types by the differentiated detection
2	1	3	0
3	0	-26	0
4	6	22	5
5	0	-2	0
7	0	-26	0
10	0	-31	0
15	1	4	0
20	0	-26	0
50	0	-11	0

Chapter 5 SVDD-based Feature Selection

This chapter provides SVDD-based feature selection methods. Section 5.1 presents motivations of the proposed methods. Introduction to feature selection for anomaly detection is presented in Section 5.2. Section 5.3, 5.4 and 5.5 presents mathematical formulations and algorithms for SVDD-R2-RFE, SVDD-RFE and SVD-Gradient feature selection methods, respectively. Finally, experiment of three methods with simulated data and the DARPA data and the results are presented in Section 5.6.

5.1 Motivation

Feature selection in a classification problem is to select most predictive features for classification results among whole features. The number of features to be considered for a classification can be reduced by using feature selection method. Smaller number of features means less effort to get and process data. Therefore, feature selection contributes cost and time reduction for solving classification problems. When a problem has huge number of features, the trial to get a solution of the problem would fail because of impractically large computation. In that case, only feature selection may make the

problem practical. Also, feature selection can identify irrelevant or redundant features for a problem which does not add any information to the classifier. By removing them from the feature set to be considered, the remaining features have more discriminating power than whole features. Feature selection is invaluable in data dimension reduction and discriminating improvement for classification problems with large number of features.

There are feature selection methods devoted only for a certain type of classification method like SVM-RFE (Support Vector Machine Recursive Feature Elimination) (Youn, 2004) while some algorithms for feature selection such as SBS (Sequential Backward Selection) and SFS (Sequential Forward Selection) are generally applicable in most data mining techniques. SVDD is introduced in 2004 and there is no feature selection method solely dedicated to SVDD. In this paper, SVDD-based feature selection method is developed.

5.2 Introduction to Feature Selection for Anomaly Detection

Feature selection has been deeply studied for the application areas in which datasets with thousands of features exist. The general objectives of feature selection are to improve the performance of the predictors, provide more effective predictors and a better understanding of the dataset (Guyon & Elisseeff, 2003). Although many researched have been performed for classification problems, there are very few studies of feature selection for anomaly detection.

General two components of feature selection are criterion function and subset searching method. Criterion function is to measure the prediction performance of single feature or feature subset. Subset searching method is an algorithm to explore feature subset space to find the best subset of features with maximum value of criterion function. Many researches for feature selection are to find better criterion function and more efficient searching algorithms.

Feature ranking is a feature selection method to evaluate the prediction power of individual feature based on its criterion function. Subset searching algorithm is not required in this method because only individual feature rankings are desired. Since it is simple, scalable, and empirically successful, feature ranking has been widely used in various literatures (Guyon & Elisseeff, 2003). Examples of feature ranking are correlation criteria (Furey et al., 2000; Tusher et al., 2001), single variable classifiers (Forman, 2003), and information theoretic ranking criteria (Bekkerman et al., 2003; Dhillon et al., 2003).

Feature subset selection is a genuine feature selection method which needs both criterion function and subset searching algorithm. There are three categories in feature subset selection: wrappers, filters, and embedded methods. Wrappers proposed by Kohavi and John (1997) uses the prediction performance of a given classifiers to measure the predictive power of feature subsets. SBS and SFS have been used for wrappers. Filters choose feature subset independently of the chosen predictors. Since filters are faster than other two methods by using heuristic algorithms, they can be used as a preprocessing step to reduce dimensionality. Embedded methods carry out feature selection as a part of the

training process and are dependent on given classification method. SVM-RFE is an example of embedded methods.

Those general approaches of feature selection can be applicable to anomaly detection area. Wang et al. (2004) proposed the integration of SVM-based anomaly detection system with feature selection function using specification. The feature selection of Wang et al. (2004) belongs to filters method. However, there is no literature dedicated to the anomaly detection method itself to my best knowledge of it.

5.3 SVDD-R2-RFE Feature Selection Method

5.3.1 Idea

SVDD constructs a boundary that envelops most of normal data. It detects anomalies which locate out of the boundary. The performance of SVDD is dependent on how well the established boundary represents normal data and discriminates anomalies from them. As seen in equation (4.2), the objective function of SVDD is to minimize the size of the boundary that is measured by value of its radius square. Figure 5.1 shows why tighter boundary is desirable then larger one. Larger boundary A can not detect anomalies that are close to normal data while smaller boundary B can. Therefore, a good feature for SVDD is to contribute to making smaller boundary for normal data.

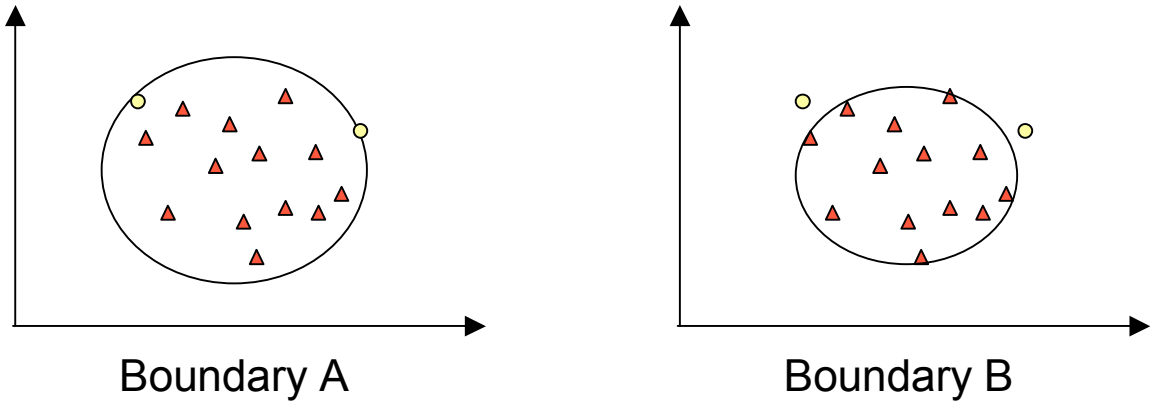


Figure 5.1 Contrast of small boundary to large boundary. Tiny triangles represent normal and small circles anomalies.

The size of boundary is measured by its radius square. Let $J(-k)$ be value of the size of boundary which was trained with $n-1$ features excluding feature k. The worst feature is k^* feature maximizing $J(-k)$ over all ks.

The searching method combined with the proposed criterion function is Recursive Feature Elimination (RFE) which was introduced in the literature. RFE is an iterative procedure: first, train the classifier, second, compute the criterion function for each single feature and finally, remove the worst feature with largest value of the criterion function.

5.3.2 Formulation

Formulations for SVDD-R2-RFE feature selection method are given for two cases of available data: case 1 in which only normal data is available for training and case 2 in which both normal and anomaly data exist in training data set.

5.3.2.1 Case 1: Only normal data

Let us consider a SVDD problem with only normal training data which has N samples. When s is one of support vectors on the boundary, R^2, boundary radius square, is represented as follows from equation (4.3) and (4.6);

$$\begin{aligned} R^2 &= \left\| s - \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \right\|^2 \\ &= (s \cdot s) + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) - 2 \sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot s) \end{aligned} \tag{5.1}$$

where $\mathbf{x}_i$ and α_j are normal data and its Lagrange multiplier. Since there are small differences among R^2 values based different support vectors, average of R^2 over all support vectors is proper as criterion function for the size of boundary. Let $R^2(s_p)$ be boundary radius square based on the support vector s_p. Now, J, criterion function, is calculated as follows:

$$J = \sum_{p \in SV} \frac{R^2(s_p)}{t} \tag{5.2}$$

where SV is a set of support vectors and there are t support vectors. A kernel function $K(\mathbf{x}_i \cdot \mathbf{x}_j)$ can be introduced into the criterion function. By introducing kernel function instead of inner product, the criterion function can be expressed as:

$$J = \frac{1}{t}\sum_{p\in SV}\left(K(s_p \cdot s_p) + \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j K(\mathbf{x}_i \cdot \mathbf{x}_j) - 2\sum_{i=1}^{N}\alpha_i K(\mathbf{x}_i \cdot s_p)\right) \tag{5.3}$$

where SV is a set of support vectors and there are t support vectors. The table 5.1 shows criterion functions for SVDD-R2-RFE feature selection method modified by introducing kernel functions.

Let $J(-k)$ be value of the criterion function for the boundary trained without feature k. The effect to remove feature k in the criterion function is calculated by the equation $DJ(k) = J - J(-k)$. The worst feature is k^* minimizing $DJ(k)$ over all feature k s.

Table 5.1 SVDD-R2-RFE criterion functions with only normal data for kernel functions

Kernel type	SVDD-R2-RFE criterion functions
Linear	$\frac{1}{t}\sum_{p\in SV}\left((s_p \cdot s_p) + \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j(\mathbf{x}_i \cdot \mathbf{x}_j) - 2\sum_{i=1}^{N}\alpha_i(\mathbf{x}_i \cdot s_p)\right)$
Gaussian	$\frac{1}{t}\sum_{p\in SV}\left(1 + \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}\right) - 2\sum_{i=1}^{N}\alpha_i \exp\left(-\frac{\|\mathbf{x}_i - s_p\|^2}{\sigma}\right)\right)$
Polynomial	$\frac{1}{t}\sum_{p\in SV}\left((s_p \cdot s_p)^d + \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j(\mathbf{x}_i \cdot \mathbf{x}_j)^d - 2\sum_{i=1}^{N}\alpha_i(\mathbf{x}_i \cdot s_p)^d\right)$

5.3.2.2 Case 2: Normal and anomaly data

Let us consider a SVDD problem with N samples of normal and M samples of anomalies in training data. R^2 based on s_p being one of support vectors on the boundary is represented as follows from equation (4.11) and (4.22);

$$\begin{aligned} R^2(s_p) &= \left\| s_p - \sum_{i=1}^{N} \alpha_i \mathbf{x}_i + \sum_{k=1}^{M} \alpha_k^* \mathbf{x}_k^* \right\|^2 \\ &= (s_p \cdot s_p) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{k=1}^{M}\sum_{l=1}^{M} \alpha_k^* \alpha_l^* (\mathbf{x}_k^* \cdot \mathbf{x}_l^*) \\ &\quad - 2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot s_p) + 2\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot s_p) - 2\sum_{i=1}^{N}\sum_{k=1}^{M} \alpha_i \alpha_k^* (\mathbf{x}_i \cdot \mathbf{x}_k^*) \end{aligned} \tag{5.4}$$

where $\mathbf{x}_i$, $\mathbf{x}_k^*$, α_i and α_k^* are normal data, anomaly and their Lagrange multipliers. The criterion function combined with a kernel function $K(\mathbf{x}_i \cdot \mathbf{x}_j)$ is as follows:

$$J = \frac{1}{t} \sum_{p \in SV} \begin{pmatrix} K(s_p \cdot s_p) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{k=1}^{M}\sum_{l=1}^{M} \alpha_k^* \alpha_l^* K(\mathbf{x}_k^* \cdot \mathbf{x}_l^*) \\ -2\sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i \cdot s_p) + 2\sum_{k=1}^{M} \alpha_k^* K(\mathbf{x}_k^* \cdot s_p) - 2\sum_{i=1}^{N}\sum_{k=1}^{M} \alpha_i \alpha_k^* K(\mathbf{x}_i \cdot \mathbf{x}_k^*) \end{pmatrix} \tag{5.5}$$

where SV is a set of support vectors and there are t support vectors. The table 5.2 shows criterion functions of SVDD-R2-RFE feature selection method for kernel functions. The worst feature is obtained by the same way explained in Section 5.3.2.1.

Table 5.2 SVDD-R2-RFE criterion functions with anomaly data for kernel functions

Kernel type	SVDD-R2-RFE criterion functions
Linear	$\frac{1}{t}\sum_{p\in SV}\left(\begin{array}{l}\left(s_p\cdot s_p\right)+\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)+\sum_{k=1}^{M}\sum_{l=1}^{M}\alpha_k^*\alpha_l^*\left(\mathbf{x}_k^*\cdot\mathbf{x}_l^*\right)\\ -2\sum_{i=1}^{N}\alpha_i\left(\mathbf{x}_i\cdot s_p\right)+2\sum_{k=1}^{M}\alpha_k^*\left(\mathbf{x}_k^*\cdot s_p\right)-2\sum_{i=1}^{N}\sum_{k=1}^{M}\alpha_i\alpha_k^*\left(\mathbf{x}_i\cdot\mathbf{x}_k^*\right)\end{array}\right)$
Gaussian	$\frac{1}{t}\sum_{p\in SV}\left(\begin{array}{l}1+\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\exp\left(-\frac{\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{\sigma}\right)\\ +\sum_{k=1}^{M}\sum_{l=1}^{M}\alpha_k^*\alpha_l^*\exp\left(-\frac{\left\|\mathbf{x}_k^*-\mathbf{x}_l^*\right\|^2}{\sigma}\right)-2\sum_{i=1}^{N}\alpha_i\exp\left(-\frac{\left\|\mathbf{x}_i-s_p\right\|^2}{\sigma}\right)\\ +2\sum_{k=1}^{M}\alpha_k^*\exp\left(-\frac{\left\|\mathbf{x}_k^*-s_p\right\|^2}{\sigma}\right)-2\sum_{i=1}^{N}\sum_{k=1}^{M}\alpha_i\alpha_k^*\exp\left(-\frac{\left\|\mathbf{x}_i-\mathbf{x}_k^*\right\|^2}{\sigma}\right)\end{array}\right)$
Polynomial	$\frac{1}{t}\sum_{p\in SV}\left(\begin{array}{l}\left(s_p\cdot s_p\right)^d+\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)^d+\sum_{k=1}^{M}\sum_{l=1}^{M}\alpha_k^*\alpha_l^*\left(\mathbf{x}_k^*\cdot\mathbf{x}_l^*\right)^d\\ -2\sum_{i=1}^{N}\alpha_i\left(\mathbf{x}_i\cdot s_p\right)^d+2\sum_{k=1}^{M}\alpha_k^*\left(\mathbf{x}_k^*\cdot s_p\right)^d-2\sum_{i=1}^{N}\sum_{k=1}^{M}\alpha_i\alpha_k^*\left(\mathbf{x}_i\cdot\mathbf{x}_k^*\right)^d\end{array}\right)$

5.3.3 Algorithm of SVDD-R2-RFE feature selection

(a) *Initialize*:

(a.1) Train SVDD with a given training data, $X_{trn} = [\mathbf{x}_1, \cdots \mathbf{x}_k, \cdots \mathbf{x}_m]^T$ under selected kernel function.

(a.2) Initialize subset of surviving features, $\mathbf{s} = [1,2,\cdots n]$, and feature ranking list, $\mathbf{r} = [\ ]$

(b) *Repeat until* $\mathbf{s} = [\ ]$:

(b.1) Construct newly reduced training data

$$X_{reduced} = X_{trn}[:, \mathbf{s}].$$

(b.2) Train SVDD with $X_{reduced}$ to get $\alpha' s$

(b.3) Compute the criterion function for each feature k in table 5.1 and 5.2

$$DJ(k) = J - J(-k)$$

(b.4) Find the feature i such as

$$i = \arg \min_k DJ(k)$$

(b.5) Update feature ranking list

$$\mathbf{r} = [\mathbf{s}(i), \mathbf{r}]$$

(b.6) Eliminate feature i in the subset of surviving features

$$\mathbf{s} = \mathbf{s} - \{\mathbf{s}(i)\}$$

(c) *Output*: feature ranking list, $\mathbf{r}$

5.4 SVDD-RFE Feature Selection Method

5.4.1 Idea

SVDD is to try to find as a small boundary as possible that envelops most of normal data. The solution of the SVDD problem is given as the form of Lagrange multipliers which are a solution to the dual problem of SVDD. SVDD boundary can be obtained from the Lagrange multipliers and their corresponding observations. The dual problem is to find Lagrange multipliers that maximize its objective function value as seen in equation (4.31). Let J and $J(-k)$ be a value of the objective function in the dual problem of SVDD and a recalculated value of the objective function without the feature k. The difference between J and $J(-k)$ becomes larger when feature k is less important and smaller when it is a better feature than others. The proposed idea on criterion function to evaluate the worth of each feature is that the worst feature is a feature with the smallest value of $J(-k)$ among all features. Let $DJ(k)$ be the difference between J and $J(-k)$ such as $DJ(k) = J - J(-k)$. Therefore, the worst feature is $k*$ feature satisfying the following equation:

$$k* = \arg \max_{k} DJ(k) \tag{5.6}$$

Also, Recursive Feature Elimination (RFE) is used as the searching method for the proposed criterion.

5.4.2 Formulation

5.4.2.1 Case 1: Only normal data

The objective function in the dual of a SVDD problem with only normal training data was given in equation (4.8). In order to measure the effect in the value of J resulted from removing a feature in the training data set, we need to train a SVDD boundary for every candidate feature to be excluded. This means that effort to train a boundary for each feature monotonically increases as the number of feature rises, thus forcing this process being impractical. Based on the fact that Lagrange multiplier α corresponds with each observation and does not related with features, we can assume that there are no changes in the values of $\alpha's$ when only one feature is eliminated in the training data set. We can easily compute the effect of removing a feature without retraining boundaries by this assumption. The following $J(-k)$ is the value of the objective function in the dual without feature k:

$$J(-k) = \sum_{i=1}^{N} \alpha(-k)_i \left(\mathbf{x}(-k)_i \cdot \mathbf{x}(-k)_i\right) - \sum_{i=1}^{N}\sum_{p=1}^{N} \alpha(-k)_i \alpha(-k)_p \left(\mathbf{x}(-k)_i \cdot \mathbf{x}(-k)_p\right) \tag{5.7}$$

where $(-k)$ means that the component k has been removed. Now, the effect to remove feature k is calculated by the following criterion function:

$$DJ(k) = J - J(-k) \tag{5.8}$$

By introducing general kernel function, $K(\mathbf{x}_i, \mathbf{x}_j)$, in the objective function, J is changed as follows:

$$J = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i=1}^{N}\sum_{p=1}^{N} \alpha_i \alpha_p K(\mathbf{x}_i \cdot \mathbf{x}_p) \tag{5.9}$$

The objective functions for linear, Gaussian and polynomial kernel are in table 5.3.

5.4.2.2 Case 2: Normal and anomaly data

The objective function in the dual of SVDD problem with training data set including L samples of normal data and M of attack was given from Tax and Duin (2004) as the following:

$$\begin{aligned} J = &\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{j=1}^{L} \alpha_j^* (\mathbf{x}_j^* \cdot \mathbf{x}_j^*) - \sum_{i=1}^{N}\sum_{p=1}^{N} \alpha_i \alpha_p (\mathbf{x}_i \cdot \mathbf{x}_p) \\ &- \sum_{j=1}^{L}\sum_{q=1}^{L} \alpha_j^* \alpha_q^* (\mathbf{x}_j^* \cdot \mathbf{x}_q^*) + 2\sum_{j=1}^{L}\sum_{i=1}^{N} \alpha_j^* \alpha_i (\mathbf{x}_j^* \cdot \mathbf{x}_i) \end{aligned} \tag{5.10}$$

Table 5.3 SVDD-RFE criterion functions with only normal data for kernel functions

Kernel type	SVDD-RFE criterion functions
Linear	$\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i=1}^{N}\sum_{p=1}^{N} \alpha_i \alpha_p (\mathbf{x}_i \cdot \mathbf{x}_p)$
Gaussian	$\sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N}\sum_{p=1}^{N} \alpha_i \alpha_p \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_p\right\|^2}{\sigma}\right)$
Polynomial	$\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i)^d - \sum_{i=1}^{N}\sum_{p=1}^{N} \alpha_i \alpha_p (\mathbf{x}_i \cdot \mathbf{x}_p)^d$

where $\mathbf{x}_i$, $\mathbf{x}_j^*$, α_i and α_j^* are normal data, anomaly and their Lagrange multipliers. In this case, $J(-k)$ is as follows:

$$\begin{aligned} J(-k) = & \sum_{i=1}^{N} \alpha(-k)_i \left(\mathbf{x}(-k)_i \cdot \mathbf{x}(-k)_i\right) - \sum_{j=1}^{L} \alpha(-k)_j^* \left(\mathbf{x}(-k)_j^* \cdot \mathbf{x}(-k)_j^*\right) \\ & - \sum_{i=1}^{N}\sum_{p=1}^{N} \alpha(-k)_i \alpha(-k)_p \left(\mathbf{x}(-k)_i \cdot \mathbf{x}(-k)_p\right) \\ & - \sum_{j=1}^{L}\sum_{q=1}^{L} \alpha(-k)_j^* \alpha(-k)_q^* \left(\mathbf{x}(-k)_j^* \cdot \mathbf{x}(-k)_q^*\right) \\ & + 2\sum_{j=1}^{L}\sum_{i=1}^{N} \alpha(-k)_j^* \alpha(-k)_i \left(\mathbf{x}(-k)_j^* \cdot \mathbf{x}(-k)_i\right) \end{aligned} \tag{5.11}$$

The effect to remove feature k is calculated by the same way in Section 5.4.2.1. By introducing general kernel function, $K(\mathbf{x}_i, \mathbf{x}_j)$ J is changed as follows:

$$\begin{aligned} J = & \sum_{i=1}^{N} \alpha_i K\left(\mathbf{x}_i \cdot \mathbf{x}_i\right) - \sum_{j=1}^{L} \alpha_j^* K\left(\mathbf{x}_j^* \cdot \mathbf{x}_j^*\right) - \sum_{i=1}^{N}\sum_{p=1}^{N} \alpha_i \alpha_p K\left(\mathbf{x}_i \cdot \mathbf{x}_p\right) \\ & - \sum_{j=1}^{L}\sum_{q=1}^{L} \alpha_j^* \alpha_q^* K\left(\mathbf{x}_j^* \cdot \mathbf{x}_q^*\right) + 2\sum_{j=1}^{L}\sum_{i=1}^{N} \alpha_j^* \alpha_i K\left(\mathbf{x}_j^* \cdot \mathbf{x}_i\right) \end{aligned} \tag{5.12}$$

Table 5.4 shows objective functions for linear, Gaussian and polynomial kernels.

Table 5.4 SVDD-RFE criterion functions with anomaly data for kernel functions

Kernel type	SVDD-RFE criterion functions
Linear	$\sum_{i=1}^{N}\alpha_i(\mathbf{x}_i\cdot\mathbf{x}_i)-\sum_{j=1}^{L}\alpha_j^*(\mathbf{x}_j^*\cdot\mathbf{x}_j^*)-\sum_{i=1}^{N}\sum_{p=1}^{N}\alpha_i\alpha_p(\mathbf{x}_i\cdot\mathbf{x}_p)$ $-\sum_{j=1}^{L}\sum_{q=1}^{L}\alpha_j^*\alpha_q^*(\mathbf{x}_j^*\cdot\mathbf{x}_q^*)+2\sum_{j=1}^{L}\sum_{i=1}^{N}\alpha_j^*\alpha_i(\mathbf{x}_j^*\cdot\mathbf{x}_i)$
Gaussian	$\sum_{i=1}^{N}\alpha_i-\sum_{j=1}^{L}\alpha_j^*-\sum_{i=1}^{N}\sum_{p=1}^{N}\alpha_i\alpha_p\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_p\|^2}{\sigma}\right)$ $-\sum_{j=1}^{L}\sum_{q=1}^{L}\alpha_j^*\alpha_q^*\exp\left(-\frac{\|\mathbf{x}_j^*-\mathbf{x}_q^*\|^2}{\sigma}\right)+2\sum_{j=1}^{L}\sum_{i=1}^{N}\alpha_j^*\alpha_i\exp\left(-\frac{\|\mathbf{x}_j^*-\mathbf{x}_i\|^2}{\sigma}\right)$
Polynomial	$\sum_{i=1}^{N}\alpha_i(\mathbf{x}_i\cdot\mathbf{x}_i)^d-\sum_{j=1}^{L}\alpha_j^*(\mathbf{x}_j^*\cdot\mathbf{x}_j^*)^d-\sum_{i=1}^{N}\sum_{p=1}^{N}\alpha_i\alpha_p(\mathbf{x}_i\cdot\mathbf{x}_p)^d$ $-\sum_{j=1}^{L}\sum_{q=1}^{L}\alpha_j^*\alpha_q^*(\mathbf{x}_j^*\cdot\mathbf{x}_q^*)^d+2\sum_{j=1}^{L}\sum_{i=1}^{N}\alpha_j^*\alpha_i(\mathbf{x}_j^*\cdot\mathbf{x}_i)^d$

5.4.3 Algorithm of SVDD-RFE feature selection

(a) *Initialize*:

(a.1) Train SVDD with a given training data, $X_{trn} = [\mathbf{x}_1, \cdots \mathbf{x}_k, \cdots \mathbf{x}_m]^T$ under selected kernel function.

(a.2) Initialize subset of surviving features, $\mathbf{s} = [1,2,\cdots n]$, and feature ranking list, $\mathbf{r} = [\ ]$

(b) *Repeat until* $\mathbf{s} = [\ ]$:

(b.1) Construct newly reduced training data

$$X_{reduced} = X_{trn}[:, \mathbf{s}].$$

(b.2) Train SVDD with $X_{reduced}$ to get $\alpha' s$

(b.3) Compute the criterion function for each feature k

$$DJ(k) = J - J(-k)$$

(b.4) Find the feature i such as

$$i = \arg \max_k DJ(k)$$

(b.5) Update feature ranking list

$$\mathbf{r} = [\mathbf{s}(i), \mathbf{r}]$$

(b.6) Eliminate feature i in the subset of surviving features

$$\mathbf{s} = \mathbf{s} - \{\mathbf{s}(i)\}$$

(c) *Output*: feature ranking list, $\mathbf{r}$

5.5 SVDD-Gradient Feature Selection Method

5.5.1 Ideas

The SVDD decision function to decide whether a test sample z is normal or outlier can be rewritten from equation (4.10) as:

$$f(z) = I(\|z - \mathbf{\mu}\|^2 \le R^2) = \begin{cases} 1 & when \quad \|z - \mathbf{\mu}\|^2 \le R^2 \\ 0 & otherwise \end{cases} \tag{5.13}$$

Figure 5.2 shows the value of decision function (5.13) in the case of two features. The value of decision function is one when a test sample is inside or on the SVDD boundary and zero when a sample is outside the boundary.

The gradient of a scalar function is a vector which points the direction of the greatest rate of increase of the function, and whose magnitude is the greatest rate of change according to Wikipedia encyclopedia (2006). Figure 5.3 shows the gradient applied into the SVDD decision function with two features. The small arrows represent the gradient for only the points near-outside and on the boundary while the other points have zero magnitude of gradient. It is obvious in figure 5.4 that feature B is the better feature compared to feature A since feature B has smaller region than feature A. As we divide all gradients into feature A axis and B axis components and sum absolute values of the components for each axis, feature B has very larger value than feature A. We can draw a clue for feature selection that a feature is important to the SVDD classification if its sum of all the

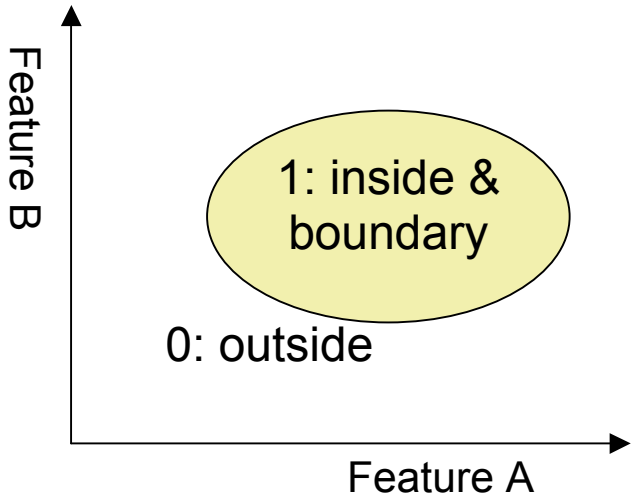


Figure 5.2 Value distribution of SVDD decision function in two-feature case

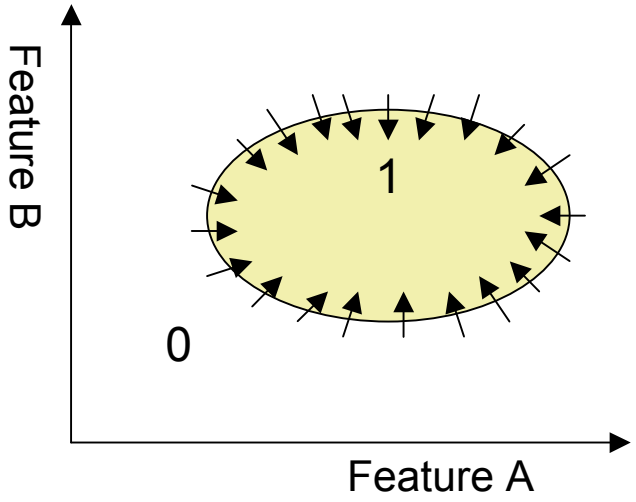


Figure 5.3 Gradient field of decision function in two-feature case

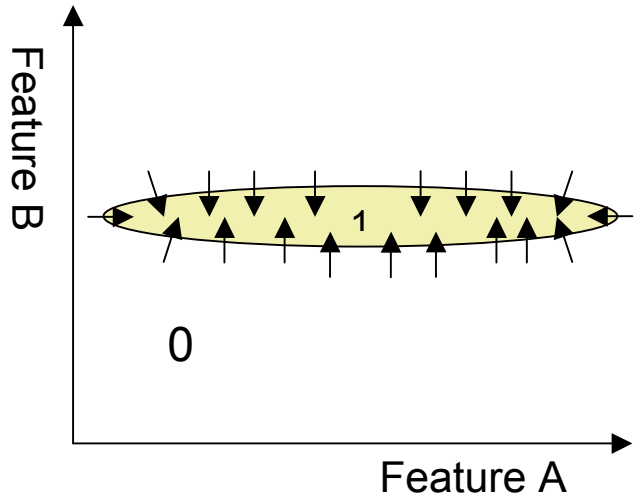


Figure 5.4 Gradient field of decision function at ellipse shape boundary

absolute values of gradient components is bigger. In SVDD the boundary is based on the support vectors which are objects on the boundary. It is reasonable that gradient is calculated for support vectors. Our feature selection is performed by computing axis component of gradient on each support vector, summing all absolute values of its components for each feature, and sorting them in descending order. The order is preference ranking for all features.

5.5.2 Formulation

5.5.2.1 Case 1: Only normal data

The equation of SVDD decision boundary for a test sample z can be written from equation (5.13) as:

$$g(z) = R^2 - \|z - \boldsymbol{\mu}\|^2 \tag{5.14}$$

The equation of SVDD (5.14) is as follows by inserting the equation (4.6) for center $\boldsymbol{\mu}$ and (5.1) for radius:

$$\begin{aligned} g(z) &= \left[(s \cdot s) + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) - 2 \sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot s) \right] \\ &\quad - \left[(z \cdot z) + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) - 2 \sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot z) \right] \\ &= (s \cdot s) - (z \cdot z) - 2 \sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot s) + 2 \sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot z) \end{aligned} \tag{5.15}$$

By introducing kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, equation (5.15) is as follows:

$$g(z) = K(s \cdot s) - K(z \cdot z) - 2 \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i \cdot s) + 2 \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i \cdot z) \tag{5.16}$$

Calculation of gradient for kernel functions was known by Hermes and Buhmann (2000). For the linear kernel, the gradient of equation (5.16) with respect to z is:

$$\nabla g(z) = \frac{\partial g(z)}{\partial z} = -2z + 2 \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \tag{5.17}$$

Using the Gaussian function as a kernel, equation (5.16) becomes:

$$g(z) = -2\sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\mathbf{x}_i - s\|^2}{\sigma}\right) + 2\sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\mathbf{x}_i - z\|^2}{\sigma}\right) \tag{5.18}$$

The gradient of equation (5.18) with respect to z is:

$$\begin{aligned} \nabla g(z) &= \frac{\partial}{\partial z}\left[2\sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\mathbf{x}_i - z\|^2}{\sigma}\right)\right] \\ &= \frac{4}{\sigma}\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i - z)\exp\left(-\frac{\|\mathbf{x}_i - z\|^2}{\sigma}\right) \end{aligned} \tag{5.19}$$

For polynomial kernel with degree d, the SVDD boundary equation is:

$$g(z) = (s \cdot s)^d - (z \cdot z)^d - 2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot s)^d + 2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot z)^d \tag{5.20}$$

The gradient of equation (5.20) with respect to z is:

$$\begin{aligned}\nabla g(z) &= \frac{\partial}{\partial z}\left[-(z\cdot z)^d + 2\sum_{i=1}^{N}\alpha_i(\mathbf{x}_i\cdot z)^d\right] \\ &= -2d(z\cdot z)^{d-1} + 2d\sum_{i=1}^{N}\alpha_i(\mathbf{x}_i\cdot z)^{d-1}\end{aligned} \tag{5.21}$$

Let $\nabla g(SV_i)$ be the gradient computed at the *i*th support vector among l support vectors. Since the gradient is a vector with n dimension which is the number of features in the training data, $\nabla g(SV_i)$ can be represented with its components as in the following equation:

$$\nabla g(SV_i) = g_{i1}\mathbf{e}_1 + g_{i2}\mathbf{e}_2 + \cdots g_{ij}\mathbf{e}_j + \cdots g_{in}\mathbf{e}_n \tag{5.22}$$

Criterion function for *j*th feature is the summation of absolute value of *j*th component over gradients of all support vectors. The following equation is for the criterion function for *j*th feature.

$$J_j = \sum_{i\in SV}\left|g_{ij}\right| \tag{5.23}$$

where SV is a set of support vectors.

5.5.2.2 Case 2: Normal and anomaly data

The equation for $\boldsymbol{\mu}$ can be represented in equation (5.24) when normal data $\mathbf{x}_i$ and attack data $\mathbf{x}_k^*$ are available.

$$\boldsymbol{\mu} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i - \sum_{k=1}^{M} \alpha_k^* \mathbf{x}_k^* \tag{5.24}$$

The equation of SVDD decision boundary for a test sample z can be written as follows by inserting the equation (5.24) for center $\boldsymbol{\mu}$ and (5.4) for radius:

$$\begin{aligned} g(z) = & \begin{bmatrix} (s \cdot s) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{k=1}^{M}\sum_{l=1}^{M} \alpha_k^* \alpha_l^* (\mathbf{x}_k^* \cdot \mathbf{x}_l^*) \\ -2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot s) + 2\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot s) - 2\sum_{i=1}^{N}\sum_{k=1}^{M} \alpha_i \alpha_k^* (\mathbf{x}_i \cdot \mathbf{x}_k^*) \end{bmatrix} \\ & - \begin{bmatrix} (z \cdot z) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{k=1}^{M}\sum_{l=1}^{M} \alpha_k^* \alpha_l^* (\mathbf{x}_k^* \cdot \mathbf{x}_l^*) \\ -2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot z) + 2\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot z) - 2\sum_{i=1}^{N}\sum_{k=1}^{M} \alpha_i \alpha_k^* (\mathbf{x}_i \cdot \mathbf{x}_k^*) \end{bmatrix} \\ & = (s \cdot s) - (z \cdot z) - 2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot s) + 2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot z) \\ & \quad + 2\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot s) - 2\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot z) \end{aligned} \tag{5.25}$$

Equation (5.25) can be rewritten as follows by introducing kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$:

$$\begin{aligned} g(z) = & K(s \cdot s) - K(z \cdot z) - 2\sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i \cdot s) + 2\sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i \cdot z) \\ & + 2\sum_{k=1}^{M} \alpha_k^* K(\mathbf{x}_k^* \cdot s) - 2\sum_{k=1}^{M} \alpha_k^* K(\mathbf{x}_k^* \cdot z) \end{aligned} \tag{5.26}$$

The gradient of equation (5.26) with respect to z for linear kernel is:

$$\nabla g(z) = \frac{\partial g(z)}{\partial z} = -2z + 2\sum_{i=1}^{N} \alpha_i \mathbf{x}_i - 2\sum_{k=1}^{M} \alpha_k^* \mathbf{x}_k^* \tag{5.27}$$

For Gaussian kernel, equation (5.26) becomes:

$$\begin{aligned} g(z) = \exp\left(-\frac{\|s-s\|^2}{\sigma}\right) - \exp\left(-\frac{\|z-z\|^2}{\sigma}\right) - 2\sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\mathbf{x}_i - s\|^2}{\sigma}\right) \\ + 2\sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\mathbf{x}_i - z\|^2}{\sigma}\right) + 2\sum_{k=1}^{M} \alpha_k^* \exp\left(-\frac{\|\mathbf{x}_k^* - s\|^2}{\sigma}\right) \\ - 2\sum_{k=1}^{M} \alpha_k^* \exp\left(-\frac{\|\mathbf{x}_k^* - z\|^2}{\sigma}\right) \end{aligned} \tag{5.28}$$

The gradient of equation (5.28) with respect to z is:

$$\begin{aligned} \nabla g(z) &= \frac{\partial}{\partial z}\left[2\sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\mathbf{x}_i - z\|^2}{\sigma}\right) - 2\sum_{k=1}^{M} \alpha_k^* \exp\left(-\frac{\|\mathbf{x}_k^* - z\|^2}{\sigma}\right)\right] \\ &= \frac{4}{\sigma}\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i - z)\exp\left(-\frac{\|\mathbf{x}_i - z\|^2}{\sigma}\right) - \frac{4}{\sigma}\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* - z)\exp\left(-\frac{\|\mathbf{x}_k^* - z\|^2}{\sigma}\right) \end{aligned} \tag{5.29}$$

The SVDD boundary equation for polynomial kernel with degree d , is:

$$g(z) = (s \cdot s)^d - (z \cdot z)^d - 2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot s)^d + 2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot z)^d + 2\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot s)^d - 2\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot z)^d \tag{5.30}$$

The gradient of equation (5.30) with respect to z is:

$$\nabla g(z) = \frac{\partial}{\partial z}\left[-(z \cdot z)^d + 2\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot z)^d - 2\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot z)^d \right] = -2d(z \cdot z)^{d-1} + 2d\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot z)^{d-1} - 2d\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot z)^{d-1} \tag{5.31}$$

The criterion function for *j*th feature is the same in equation (5.23)

5.5.3 Algorithm of SVDD-Gradient feature selection

(a) Initialize feature ranking list

$\mathbf{r} = [\ ]$.

(b) Train SVDD with a given training data under selected kernel function.

(c) Compute the gradient for each support vectors. Refer to the table 5.5 for gradients of kernel functions.

Table 5.5 Gradients of kernel functions

Kernel type	Available data	Gradient $\nabla g(SV_i)$
Linear	Only normal	$-2SV_i + 2\sum_{i=1}^{N} \alpha_i \mathbf{x}_i$
	Normal and anomaly	$-2SV_i + 2\sum_{i=1}^{N} \alpha_i \mathbf{x}_i - 2\sum_{k=1}^{M} \alpha_k^* \mathbf{x}_k^*$
Gaussian	Only normal	$\frac{4}{\sigma}\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i - SV_i) \exp\left(-\frac{\|\mathbf{x}_i - SV_i\|^2}{\sigma}\right)$
	Normal and anomaly	$\frac{4}{\sigma}\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i - SV_i) \exp\left(-\frac{\|\mathbf{x}_i - SV_i\|^2}{\sigma}\right)$ $-\frac{4}{\sigma}\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* - SV_i) \exp\left(-\frac{\|\mathbf{x}_k^* - SV_i\|^2}{\sigma}\right)$
Polynomial	Only normal	$-2dSV_i^T (SV_i \cdot SV_i)^{d-1} + 2d\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot SV_i)^{d-1}$
	Normal and anomaly	$-2dSV_i^T (SV_i \cdot SV_i)^{d-1} + 2d\sum_{i=1}^{N} \alpha_i (\mathbf{x}_i \cdot SV_i)^{d-1}$ $-2d\sum_{k=1}^{M} \alpha_k^* (\mathbf{x}_k^* \cdot SV_i)^{d-1}$

(d) Calculate the criterion function for *j*th feature

$$J_j = \sum_{i=1}^{l} \left| g_{ij} \right|$$

(e) Sort $\mathbf{J} = \begin{bmatrix} J_1 & \cdots & J_j & \cdots & J_n \\ 1 & \cdots & j & \cdots & n \end{bmatrix}$ in descending order of the first row

(f) Output: feature ranking list

$$\mathbf{r} = \left[sorted\_\mathbf{J}(2,:) \right]$$

5.6 Experiments

5.6.1 Experimental setup

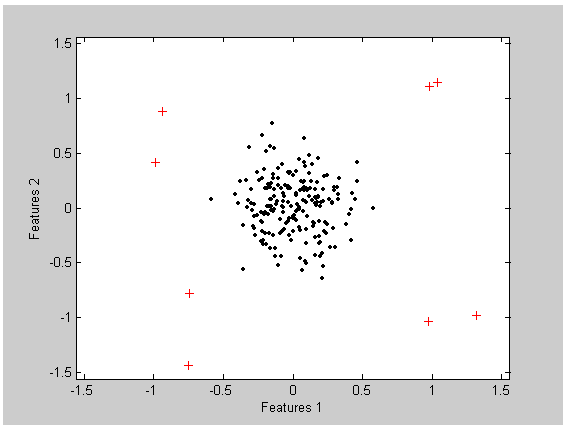
Two datasets were used for the experiment of the proposed SVDD-based feature selection methods. One is based on the DARPA dataset which was used in the previous chapters. We normalized the dataset to reduce variance effect due to range difference among its features. The other dataset was artificially generated to make it clear which feature is better or worse. It is called simulated dataset. We made up normal and anomaly samples in a 20-dimensional data space for the experiment. The center of the normal samples was located at $(0,\cdots,0)^T$, and anomaly samples were based on one of four centers being located at $(1,\cdots,1)^T$, $(-1,\cdots,-1)^T$, $(1,-1,\cdots,1,-1)^T$ and $(-1,1,\cdots,-1,1)^T$. Each feature of all the samples was independently corrupted by Gaussian noise with zero mean and standard deviations dependent on the feature index i as $0.2 \times 1.2^{i-1}$ (Hermes & Buhmann, 2000). Table 5.6 shows the Gaussian noise's mean and sigma charged to the 20 features.

Table 5.6 Gaussian noises for features

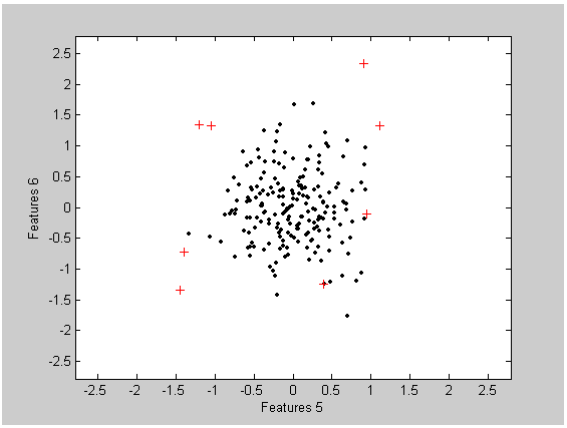
Feature number	Noise	Feature number	Noise
1	$N(0, 0.2^2)$	11	$N(0, 1.24^2)$
2	$N(0, 0.24^2)$	12	$N(0, 1.49^2)$
3	$N(0, 0.29^2)$	13	$N(0, 1.78^2)$
4	$N(0, 0.35^2)$	14	$N(0, 2.14^2)$
5	$N(0, 0.41^2)$	15	$N(0, 2.57^2)$
6	$N(0, 0.5^2)$	16	$N(0, 3.08^2)$
7	$N(0, 0.6^2)$	17	$N(0, 3.7^2)$
8	$N(0, 0.72^2)$	18	$N(0, 4.44^2)$
9	$N(0, 0.86^2)$	19	$N(0, 5.32^2)$
10	$N(0, 1.03^2)$	20	$N(0, 6.39^2)$

By increasing noises along with feature index, the first feature is the most favorable and the last becomes the most confusing feature to distinguish anomalies from normal data. Figure 5.5 shows the trend for normal and anomaly data to approach closely as the feature indices increase. Normal data and anomalies are clearly away from each other in the space with feature 1 and 2 as seen in figure 5.5(a). However, anomalies locate in the middle of normal data in the figure 5.5 (d), thereby being difficult to distinguish them from normal.

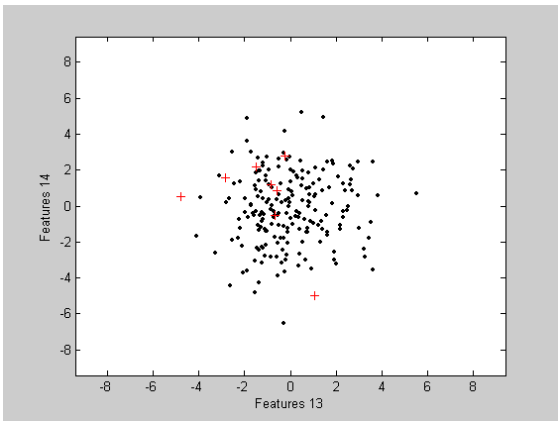
The performances of the proposed three SVDD feature selection methods were compared with the performance of SVM-RFE feature selection method that is regarded as one of the most effective feature selection method in classification problems. The comparison was performed in two cases dependent on available data: case 1 is when only normal data is available in training data and case 2 is the situation that normal and anomaly data are



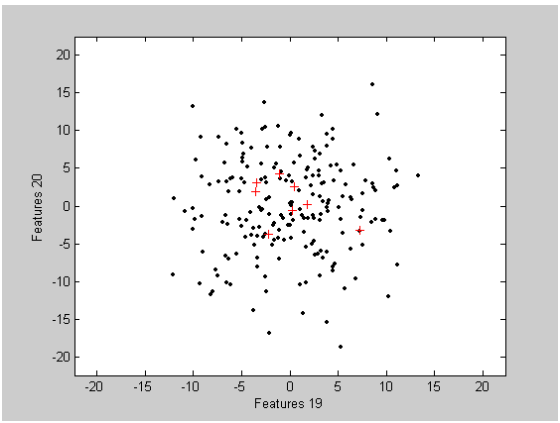
(a) Feature 1 & 2



(b) Feature 5 & 6



(c) Feature 13 & 14



(d) Feature 19 & 20

Figure 5.5 Two-dimensional pictures to show distribution of normal data represented with dot and anomaly with plus sign in various feature combinations.

available. In the case 1, only proposed SVDD-based feature selection methods were evaluated with the simulated data since SVM-RFE feature selection method does not work. The DARPA data was not used in the case 1. Since what feature is better or worse is clear in the simulated dataset, the performance of the method in the case 1 was tested by checking whether SVDD-RFE results agree with the designed feature ranking in the dataset. To avoid bias due to randomness of simulated data, 100 data sets were used and the average performances were used. In the case 2, three SVDD-based methods and the SVM method were compared for the simulated data set in two ways. The first way is similar with one in the case 1 that inspects correctness of the selected feature order based on the 100 data sets. The second way is to measure the performance of selected features by applying detection method to the feature. For the DARPA data in case 2, only SVDD-RFE method was compared with the SVM-RFE method because the DARPA data is heavy and takes long time to run feature selection methods. The performance of feature selection methods in the case 2 was measured by false alarm rate and detection rate. In addition, Gaussian kernel function was used in all the experiment since the kernel is recognized as the best in most of classification cases. Table 5.7 shows the summary of this experimental setup.

5.6.2 Results on Case 1

Table 5.8 shows results to compare performances of the proposed SVDD-based feature selection methods and SVM-RFE feature selection method with simulated data in the case 1. The best performance achieved by both of the SVDD-R2-RFE and the SVDD-

Table 5.7 Summary of experimental setup and performance measure

<table>
<tr><td colspan="2">Available data situation</td><td>Case 1</td><td colspan="3">Case 2</td></tr>
<tr><td colspan="2">Data set</td><td>Simulated data</td><td colspan="2">Simulated data</td><td>DARPA data</td></tr>
<tr><td rowspan="2">Comparison</td><td>Proposed methods</td><td rowspan="2">Inspecting feature order</td><td rowspan="2">Inspecting feature order</td><td>False alarm rate & detection rate by applying SVDD to feature order output</td><td>False alarm rate & detection rate by applying SVDD to feature order output</td></tr>
<tr><td>SVM-RFE</td><td>False alarm rate & detection rate by applying SVDD and SVM to feature order output</td><td>False alarm rate & detection rate by applying SVDD and SVM to feature order output</td></tr>
</table>

RFE feature selection methods showing 100% correctness that right order of 20 features was identified perfectly without any wrong selection. The second best performance was carried out by the SVDD-Gradient individual ranking method that identified correctly best 15 out of 20 features. However, it was impossible for the SVM-RFE method to perform feature selection for the data set with only normal data. Only the proposed SVDD-based feature selection methods are able to perform feature selection for the data set like case 1 where anomaly data is not available in training data set.

5.6.3 Results on Case 2

Table 5.9 shows feature orders selected by the proposed SVDD-based feature selection methods and SVM-RFE with simulated data in the case 2. The best performance achieved by both of the SVDD-R2-RFE and the SVDD-RFE feature selection methods showing

Table 5.8 Feature order selected by the proposed SVDD-based feature selection methods and SVM-RFE with simulated data in the case 1

Best feature order	SVDD-based feature selection method			SVM-RFE
	R2-RFE	SVDD-RFE	Gradient	
1	Feature 1	Feature 1	Feature 1	N/A
2	Feature 2	Feature 2	Feature 2	N/A
3	Feature 3	Feature 3	Feature 3	N/A
4	Feature 4	Feature 4	Feature 4	N/A
5	Feature 5	Feature 5	Feature 5	N/A
6	Feature 6	Feature 6	Feature 6	N/A
7	Feature 7	Feature 7	Feature 7	N/A
8	Feature 8	Feature 8	Feature 8	N/A
9	Feature 9	Feature 9	Feature 9	N/A
10	Feature 10	Feature 10	Feature 10	N/A
11	Feature 11	Feature 11	Feature 11	N/A
12	Feature 12	Feature 12	Feature 12	N/A
13	Feature 13	Feature 13	Feature 13	N/A
14	Feature 14	Feature 14	Feature 14	N/A
15	Feature 15	Feature 15	Feature 15	N/A
16	Feature 16	Feature 16	Feature 17	N/A
17	Feature 17	Feature 17	Feature 18	N/A
18	Feature 18	Feature 18	Feature 20	N/A
19	Feature 19	Feature 19	Feature 19	N/A
20	Feature 20	Feature 20	Feature 16	N/A

100% correctness that right order of 20 features was identified perfectly without any wrong selection. The second best performance was carried out by the SVDD-Gradient individual ranking method that identified correctly 16 out of 20 features. The SVM-RFE method showed the worst performance to feature selection for the simulated data set, in which identified correctly only 10 out of 20 features.

Table 5.10 shows performance results in comparison of the proposed SVDD-based feature selection methods and SVM-RFE with simulated data in the case 2. The best subsets of features with best performance in terms of false alarm rate and detection rate are 4-feature set with 100% detection rate and 7.1% false alarm rate from the SVDD-Gradient, 2-feature set with 100% and 6.9% from the SVDD-R2-RFE, 2-feature set with 100% and 6.9% from the SVDD-RFE, one-feature set with 87% and 0.2% from the SVM-RFE with SVM detection, and 4-feature set with 100% and 7% from the SVM-RFE with SVDD detection. The best performance came from the 2-feature subsets of both the SVDD-R2-RFE and the SVDD-RFE. The second was the performance of 4-feature subset obtained by both of the SVDD-Gradient and the SVM-RFE with SVDD detection. Considering the entire performances for all the best feature subset, both of the SVDD-R2-RFE and he SVDD-RFE are slightly better to anomaly feature selection than the SVDD-Gradient and the SVM-RFE with SVDD detection. The SVM-RFE with SVM detection showed the worst performance in all selected feature subsets. Therefore, the proposed SVDD-based feature selection methods are better than or equal performance with the SVM-RFE feature selection method.

Table 5.9 Feature order selected by the proposed SVDD-based feature selection methods and SVM-RFE with simulated data in the case 2

Best feature order	SVDD-based feature selection method			SVM-RFE
	R2-RFE	SVDD-RFE	Gradient	
1	Feature 1	Feature 1	Feature 1	Feature 2
2	Feature 2	Feature 2	Feature 2	Feature 1
3	Feature 3	Feature 3	Feature 3	Feature 5
4	Feature 4	Feature 4	Feature 4	Feature 3
5	Feature 5	Feature 5	Feature 5	Feature 4
6	Feature 6	Feature 6	Feature 6	Feature 7
7	Feature 7	Feature 7	Feature 8	Feature 10
8	Feature 8	Feature 8	Feature 7	Feature 8
9	Feature 9	Feature 9	Feature 9	Feature 9
10	Feature 10	Feature 10	Feature 10	Feature 6
11	Feature 11	Feature 11	Feature 11	Feature 11
12	Feature 12	Feature 12	Feature 13	Feature 12
13	Feature 13	Feature 13	Feature 12	Feature 13
14	Feature 14	Feature 14	Feature 14	Feature 14
15	Feature 15	Feature 15	Feature 15	Feature 15
16	Feature 16	Feature 16	Feature 16	Feature 16
17	Feature 17	Feature 17	Feature 17	Feature 17
18	Feature 18	Feature 18	Feature 18	Feature 18
19	Feature 19	Feature 19	Feature 19	Feature 19
20	Feature 20	Feature 20	Feature 20	Feature 20

Table 5.10 Comparison results of the proposed SVDD-based feature selection methods and SVM-RFE with simulated data in the case 2

Feature selection method	Criteria function	SVDD-Gradient		SVDD-R2		SVDD-Dual objective		SVM-Dual objective			
	Searching method	Individual ranking		RFE		RFE		RFE			
Detection method		SVDD		SVDD		SVDD		SVM		SVDD	
Performance measure		FA*	DR**	FA*	DR**	FA*	DR**	FA*	DR**	FA*	DR**
All features		99.6%	100%	99.1%	100%	99.1%	100%	0.3%	0%	99.1%	100%
Number of best features	19	97.8%	100%	93.9%	100%	89.2%	100%	0.2%	0%	89.2%	100%
	18	87.2%	100%	72.4%	98%	72.4%	98%	0.3%	1%	72.4%	98%
	17	81.3%	100%	55.4%	97%	55.4%	97%	0.2%	6%	58.7%	98%
	16	75.9%	100%	36.7%	94%	36.7%	94%	0.1%	7%	36.7%	94%
	15	57.0%	99%	24.8%	89%	24.8%	89%	0.0%	7%	33.0%	97%
	14	41.6%	99%	23.5%	96%	18.6%	92%	0.0%	6%	23.5%	96%
	13	32.1%	99%	17.6%	95%	12.4%	91%	0.0%	11%	14.5%	92%
	12	32.4%	100%	12.4%	96%	7.9%	95%	0.0%	15%	7.9%	95%
	11	36.7%	100%	8.5%	96%	8.5%	96%	0.0%	19%	10.2%	96%
	10	9.3%	99%	9.1%	98%	9.1%	98%	0.0%	26%	9.1%	98%
	9	7.9%	100%	7.6%	100%	9.3%	100%	0.0%	23%	9.4%	100%
	8	11.7%	100%	12.1%	100%	11.6%	100%	0.0%	34%	12.1%	100%
	7	10.5%	100%	10.5%	100%	10.5%	100%	0.0%	33%	11.8%	100%
	6	7.7%	100%	9.2%	100%	10.2%	100%	0.0%	34%	7.6%	100%
	5	11.0%	100%	7.6%	100%	10.8%	100%	0.0%	51%	10.9%	100%
	4	**7.1%**	**100%**	9.5%	100%	9.5%	100%	0.0%	52%	**7.0%**	**100%**
	3	7.9%	100%	10.3%	100%	10.3%	100%	0.0%	79%	7.8%	100%
	2	7.3%	99%	**6.9%**	**100%**	**6.9%**	**100%**	0.0%	58%	7.3%	99%
	1	10.5%	100%	10.5%	100%	10.5%	100%	**0.2%**	**87%**	10.5%	100%

* FA: false alarm rate for normal data
** DR: detection rate for anomaly data

Table 5.11 shows the comparison results of the proposed SVDD-RFE feature selection method and SVM-RFE with the DARPA data in the case 2. The best subsets of features are 6-feature set with 100% detection rate and 1.6% false alarm rate from the SVDD-RFE, 14-feature set with 90.1% and 0.04% from the SVM-RFE with SVM detection, and 99-feature set with 100% and 2.2% from the SVM-RFE with SVDD detection. The 6-feature subset of the SVDD-RFE is the best than the other two methods' in terms of number of feature and performance results. The performances of three methods with all the features are 100% detection rate and 5.7% false alarm rate from the SVDD-RFE and the SVM-RFE with SVDD detection, and 90.7% and 1.1% from the SVM-RFE with SVM detection. Both of the SVDD-RFE and the SVM-RFE with SVDD detection achieved same performance because they use same detection technique, SVDD, for the same data set. They performed better with all the features than the SVM-RFE with SVM detection method. As comparing the performances of three methods along with various feature subsets, only the SVDD-RFE method achieved reasonable performance until 10 feature subsets. For example, the method showed 95.6% detection rate with 1% false alarm rate from single feature subset and 100% with 4.7% from 10 feature subset. However, the SVM-RFE with SVM detection and the SVM-RFE with SVDD detection methods showed 0% and 9.3% detection rate, respectively, in both single and 10 feature subsets. After 40 feature subset, the SVDD-RFE method showed as good performance as the other two methods'. It is clear that the SVDD-RFE method performed much better than SVM-RFE with SVM detection and the SVM-RFE with SVDD detection methods.

Table 5.11 Comparison results of the proposed SVDD-RFE feature selection method and SVM-RFE with the DARPA data in the case 2

Feature selection method		SVDD-RFE		SVM-RFE			
Detection method		SVDD		SVM		SVDD	
Performance measure		FA*	DR**	FA*	DR**	FA*	DR**
All features		5.7%	100%	1.1%	90.7%	5.7%	100%
Best performance (Number of features)		1.6% (6)	100% (6)	0.04% (14)	90.1% (14)	2.2% (99)	100% (99)
Performance by best features	200	5.4%	100%	0.7%	90.7%	5.3%	100%
	160	5.3%	100%	0.6%	90.7%	4.5%	100%
	120	5.0%	100%	0.5%	90.7%	4.0%	100%
	80	5.4%	100%	0.4%	90.7%	10.1%	100%
	40	3.2%	100%	0.1%	90.1%	2.8%	9.9%
	10	4.7%	100%	0%	0%	2.7%	9.3%
	1	1.0%	95.6%	0%	0%	0.1%	9.3%

* FA: false alarm rate for normal data
** DR: detection rate for anomaly data

Chapter 6 Conclusion and Future Research

This chapter provides conclusion of this dissertation in Section 6.1 and presents future research area in Section 6.2.

6.1 Conclusion

The proposed differentiated anomaly intrusion detection method was effective according to the experimental results. The differentiated detection was motivated by the fact that there exists more critical type of intrusions against information systems. The system administrator needs to focus on detecting as precisely intrusions of the type as possible even compromising with false alarms and detection of non-target attack types. With the simulated data experiment, our differentiated detection method demonstrated that it had enough potential to fit well with those practical needs. It was noticeable that using more training samples of target attack type can provide more detailed performance of the differentiated intrusion detection. Another experiment with the preprocessed DARPA BSM data confirmed our method's usefulness in the real situation. Since the concept of differentiated anomaly detection can be applicable into other anomaly detection areas,

this method would be beneficial to broader application areas beyond intrusion detection field.

The new framework for host-based feature extraction showed promising results from the experiment with features which were extracted from DARPA_MITLL 98 BSM data set by the framework. This new framework was studied to widen feature searching space and explore further searching directions for better features. Based on new viewpoints about user activities, the framework brought new feature categories such as length, intensity, and event type. According to the experiment with SVDD classifiers, event type category was the most effective single category among three, and each category of length and intensity was not practical. This result supports why most existing researches have used event type features. However, another significant finding was that combination of length and intensity features could be powerful features. This suggests importance of two feature categories and requires further investigation of features combined with the two categories. In addition, all features combined from three categories showed better performance than existing features. Therefore, the proposed new framework is worthy enough to be regarded as an efficient approach for host-based feature development.

In this dissertation, SVDD-based feature selection methods such as SVDD-R2-RFE, SVDD-RFE and SVDD-Gradient have been presented to provide feature selection tools for anomaly detection field. The proposed feature selection methods were compared with well-known SVM-RFE feature selection method using simulated data and the DARPA data set. The results showed that the proposed methods performed much better than SVM-RFE for both datasets. Only the proposed methods were able to perform feature

selection for training data without anomalies whereas the SVM-RFE was not able to do. With the DARPA data, the proposed SVDD-RFE method showed better performance than the SVM-RFE. In comparison of the proposed methods, SVDD-R2-RFE and SVDD-RFE were better in feature selection than SVDD-Gradient.

6.2 Future Research

Future research is to examine the strengths and weaknesses of the proposed differentiated detection and SVDD-based feature selection methods by applying them to other applications in anomaly detection area. The possible application fields are product quality inspection, nuclear power plant control management, and medical examination in which there are huge normal outputs and very few anomalies.

Another future work is to explore the effectiveness of the differentiated anomaly detection combined with SVDD-based feature selection method. The idea is to make the differentiated detection more powerful by using the feature selection method in finding more predictive features to distinguish a target attack type from non-target attacks.

LIST OF REFERENCES

List of References

[1] Anderson, D., Frivold, T. and Valdes, A. (1995). Next-generation intrusion detection expert system (NIDES): A summary. *Technical Report SRI-CSL-97-07*. SRI International, Menio Park, CA.

[2] Bekkerman, R., El-Yaniv, R., Tishby, N. and Winter, Y. (2003). Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research* **3**: 1183-1208.

[3] Chen, W., Hsu, S., and Shen, H. (2005). Application of SVM and ANN for intrusion detection. *Computers & Operations Research* **32**: 2617-2634.

[4] Cortes, C. and Vapnik, V. N. (1995). Support vector networks. *Machine Learning* **20**: 273-297.

[5] Depren, O., Topallar, M., Anarim, E. and Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications* **29(4)**: 713-722.

[6] Dhillon, I., Mallela, S. and Kumar, R. (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research* **3**: 1265-1287.

[7] Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research* **3**: 1289-1306.

[8] Forrest, S,. Hofmeyr, S.A., Somayaji, A. and Longstaff, T.A. (1996). A sense of self for UNIX processes. *Proceedings of the 1996 IEEE Symposium on Security and Privacy*.

[9] Forrest, S., Hofmeyr, S.A. and Somayaji, A. (1997). Computer immunology. *Communications of the ACM* **40(10)**: 88-96.

[10] Furey, T., Cristianini, N., Duffy, Bednarski, N., Schummer, D.M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16**: 906-914.

[11] GeNUA company web site. URL: www.genua.de/dateien/gd-installation-en.jpg

[12] Gordon, L.A., Loeb, M.P., Lucyshyn, W. and Richardson, R. (2007). 2006 CSI/FBI computer crime and security survey. *Computer Security Institute*.

[13] Guyon, I. And Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research* **3**: 1157-1182.

[14] Hermes, L. and Buhmann, J.M. (2000). Feature selection for support vector machines. *Proceedings of ICPR 2000* **2**: 716-719.

[15] Jiang, S., Song, X., Wang, H., Han, J. and Li, Q. (2006). A clustering-based method for unsupervised intrusion detections. *Pattern Recognition Letters* **27**: 802-810.

[16] Jou, Y., Gong, F., Sargor, C., Wu, X., Wu, S., Chang, H. and Wang, F. (2000). Design and implementation of a scalable intrusion detection system for the protection of network infrastructure. *Proceedings of the DARPA Information Survivability Conference and Exposition. IEEE Computer Society*. CA: 69-83.

[17] Kendall, K. (1999). A database of computer attacks for the evaluation of intrusion detection systems. *MS thesis*, Massachusetts Institute of Technology, Cambridge, Massachusetts.

[18] Ko, C., Fink, G. and Levitt, K. (1997). Execution monitoring of security-critical programs in distributed systems: A specification-based approach. *Proceedings of the 1997 IEEE Symposium on Security and Privacy*: 175-187.

[19] Kohavi, R. and John, G. (1997). Wrappers for feature selection. *Artificial Intelligence* **97(1-2)**: 273-324.

[20] Lee, W. and Stolfo, S.J. (1998). Data mining approaches for intrusion detection. *Proceedings of the 7th USENIX Security Symposium*. Texas.

[21] Lee, W., Stolfo S.J. and Chan, P.K. (1997). Learning patterns from Unix process execution traces for intrusion detection. *Proceedings of AAAI97 Workshop on AI Methods in Fraud and Risk Management*.

[22] Levin, I. (2000). KDD-99 conference reports: KDD-99 classifier learning contest LLSoft's results overview. *ACM SIGKDD Explorations Newsletter* **1(2)**: 67-75.

[23] Li, X. and Ye, N. (2002). Grid- and dummy-cluster-based learning of normal and intrusive clusters for computer intrusion detection. *Quality & Reliability Engineering International* **18**: 231-242.

[24] Li, X. and Ye, N. (2006). A supervised clustering and classification algorithm for mining data with mixed variables. *IEEE Transactions on Systems, Man, and Cybernetics* **36(2)**: 396-406.

[25] Li, Z., Das, A. and Zhou, J. (2005). Theoretical Basis for Intrusion Detection. *Proceedings of the 2005 IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop*: 184-192.

[26] Lippmann, R., Fried, D., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Webster, S., Wyschogrod, D., Cunningham, R. and Zissman, M. (2000). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. *Proceedings of the DARPA Information Survivability Conference and Exposition*. IEEE Computer Society. Los Alamitos, CA: 12-26.

[27] Liu, Y., Chen, K., Liao, X. and Zhang, W. (2004). A genetic clustering method for intrusion detection. *Pattern Recognition* **37**: 927-942.

[28] MIT Lincoln Labs, *DARPA intrusion detection evaluation 1998 data sets*. URL: www.ll.mit.edu/IST/ideval/data/data_index.html.

[29] MIT Lincoln Labs, *MIT Lincoln laboratory offline component of DARPA 1998 intrusion detection evaluation*. URL:www.ll.mit.edu/IST/ideval/docs/1998/introduction/index.htm.

[30] Oh, S. H. and Lee, W. S. (2003). An anomaly intrusion detection method by clustering normal user behavior. *Computers & Security* **22(7)**: 596-612.

[31] Park, J., Kang, D., Kim, J., Kwok, J. T. and Tsang, I. W. (2005). Pattern De-Noising Based on Support Vector Data Description. *Proceedings of International Joint Conference on Neural Network*, Canada.

[32] Schölkopf, B., Smola, A. J. and Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* **10**: 1299-1319.

[33] Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H. and Zhou, S. (2002). Specification-based anomaly detection: a new approach for detecting network intrusions. *Proceedings of ACM Conference on Computer and Communication Security*: 265-274.

[34] Tao, X., Liu, F., and Zhou, T. (2004). A novel approach to intrusion detection based on support vector data description. *Proceedings of 30th Annual Conference of IEEE Industrial Electronics Society* **3**: 2016-2021.

[35] Tax, D. M. J. and Duin, R. P. W. (2004). Support vector data description. *Machine Learning* **54**: 45-66.

[36] Tax, D. M. J. (2005). *Data description toolbox dd_tools 1.4.0*. URL:www-ict.ewi.tudelft.nl/~davidt/dd_manual.pdf.

[37] Tohmatsu, D. T. (2003), 2003 Global Security Survey

[38] Tusher, V.G., Tibshirani, R. and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences* **98**: 5116-5121.

[39] Wang, Y. (2005). A multinomial logistic regression modeling approach for anomaly intrusion detection. *Computers & Security* **24(8)**: 662-674.

[40] Wang, Y., Miner, A., Wong, J. and Uppuluri, P. (2004). Improving feature selection in anomaly intrusion detection using specifications. *Workshop On Datamining, Security & Application*. India.

[41] Wikipedia, (2006). URL:en.wikipedia.org/wiki/Gradient

[42] Wu, N. and Zhang, J. (2006). Factor-analysis based anomaly detection and clustering. *Decision Support Systems* **42(1)**: 375-389.

[43] Yang, M., Zhang, H., Fu, J. and Yan, F. (2004). A framework for adaptive anomaly detection based on support vector data description. *In Lecture Notes in Computer Science* **3222**: 443-450.

[44] Ye, N. and Chen, Q. (2001). An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality & Reliability Engineering International* **17**: 105-112.

[45] Ye, N., Ehiabor T. and Zhang, Y. (2002). First-order versus high-order stochastic models for computer intrusion detection. *Quality & Reliability Engineering International* **18**: 243-250.

[46] Ye, N., Vilbert S. and Chen, Q. (2003). Computer intrusion detection through ewma for autocorrelated and uncorrelated data. *IEEE Transactions on Reliability* **52(1)**: 75-82.

[47] Youn, E.S. (2004). Feature selection and discriminant analysis in data mining. *Ph.D. dissertation*, University of Florida.

[48] ZDNet Security News Article, January 2004

[49] Zhang, Z. and Shen, H. (2005). Application of online-training SVMs for real-time intrusion detection with different considerations. *Computer Communications* **28**: 1428-1442.

VITA

Inho Kang was born in Samcheonpo, Korea on August 28, 1965. He went to Samcheonpo Elementary School and Jae-il Middle School in the city. He moved to Jin-ju to enter Jin-ju High School. After graduating the high school, he went to the Seoul National University. In 1988, he received a B.S. degree in Industrial Engineering and continued his Master's program focusing on Operations Research in the university. In 1990, he received a M.A. and began to work for the Korea Institute for Defense Analyses (KIDA) that is a think-tank for the Ministry of National Defense in Korea. As a researcher in the institute, he participated in many military research projects and was responsible of several projects. He won five prizes for research excellence and recognition of achievement from the institute. In 2003, he started a Ph.D. program in Industrial Engineering at the University of Tennessee under the support of KIDA.

Inho comes back to his position at KIDA in Seoul, Korea and continues to research policies for national defense science and technology. He can be reach at kih@kida.re.kr.