ANOMALY DETECTION THROUGH STATISTICS-BASED MACHINE LEARNING FOR COMPUTER NETWORKS

by

Xuejun Zhu

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2006

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation

prepared by Xuejun Zhu

entitled Anomaly Detection Through Statistics-based Machine Learning for Computer Networks

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy

____Date: 3/31/2006
Judy Jin

____ Date: 3/31/2006
Ronald Askin

____ Date: 3/31/2006
Ferenc Szidarovszky

____ Date: 3/31/2006
Daniel Zeng

____ Date: 3/31/2006
Salim Hariri

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

____ Date: 3/31/2006
Dissertation Director: Judy Jin

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirement for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: ______________________________
Xuejun Zhu

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Jionghua Jin. Without her guidance and kindly support, I might not have the opportunity to explore the field of reliability and quality engineering. Also thank Dr. Zhisheng Zhang to discuss many issues together. I would like to thank members of my committee, Dr. Askin, Dr. Szidarovszky, Dr. Zeng, and Dr. Hariri. I appreciate their help in both my course work and research projects.

I would like to thank my friends, former and current graduate students in the Department of Systems & Industrial Engineering: Dr. Huairui Guo, Dr. Li Li, Mr. Zhihong Zhou and Mr. Seunghyun Baek for their advice, support and friendship.

Special thanks to my wife, Hongyan Chen. It was her understanding and support that made this journey possible.

DEDICATION

This work is dedicated to my family, my mother Chunrong Wang, my father Jinpeng Zhu, my wife Hongyan Chen, and my elder brother Yongjun Zhu. Their love, support, understanding and encouragement have been the major motivation for me in working towards the Ph. D. degree.

TABLE OF CONTENTS

TABLE OF CONTENTS - Continued

TABLE OF CONTENTS - Continued

TABLE OF CONTENTS - Continued

LIST OF FIGURES

LIST OF FIGURES - Continued

LIST OF FIGURES - Continued

LIST OF TABLES

ABSTRACT

The intrusion detection in computer networks is a complex research problem, which requires the understanding of computer networks and the mechanism of intrusions, the configuration of sensors and the collected data, the selection of the relevant attributes, and the monitor algorithms for online detection. It is critical to develop general methods for data dimension reduction, effective monitoring algorithms for intrusion detection, and means for their performance improvement. This dissertation is motivated by the timely need to develop statistics-based machine learning methods for effective detection of computer network anomalies.

Three fundamental research issues related to data dimension reduction, control charts design and performance improvement have been addressed accordingly. The major research activities and corresponding contributions are summarized as follows:

(1) Filter and Wrapper models are integrated to extract a small number of the informative attributes for computer network intrusion detection. A two-phase analyses method is proposed for the integration of Filter and Wrapper models. The proposed method has successfully reduced the original 41 attributes to 12 informative attributes while increasing the accuracy of the model. The comparison of the results in each phase shows the effectiveness of the proposed method.

(2) Supervised kernel based control charts for anomaly intrusion detection. We propose to construct control charts in a feature space. The first contribution is the use of multi-objective Genetic Algorithm in the parameter pre-selection for SVM based control

charts. The second contribution is the performance evaluation of supervised kernel based control charts.

(3) Unsupervised kernel based control charts for anomaly intrusion detection. Two types of unsupervised kernel based control charts are investigated: Kernel PCA control charts and Support Vector Clustering based control charts. The applications of SVC based control charts on computer networks audit data are also discussed to demonstrate the effectiveness of the proposed method.

Although the developed methodologies in this dissertation are demonstrated in the computer network intrusion detection applications, the methodologies are also expected to be applied to other complex system monitoring, where the database consists of a large dimensional data with non-Gaussian distribution.

CHAPTER 1 INTRODUCTION

1.1 MOTIVATION

The security of computer networks plays a strategic role in modern computer systems. Computer network vulnerability is a growing problem. It is very important to build systems for the purpose of intrusion detection. As long as we can detect suspicious connection, we can take actions to prevent its further propagation in the networks. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to bypass the security mechanisms of a computer or network ("compromise the confidentiality, integrity, availability of information resources"). Intrusion Detection System (IDS) is a combination of software and hardware that attempts to perform intrusion detection and raise alarm when possible intrusions are detected.

Computer network intrusion usually includes a series of activities. Existing work on intrusion detection has primarily used system activities data to detect intrusions. Other kinds of data (e.g., system state and performance data) in computer and network systems may also be useful for intrusion detection.

The goal of intrusion detection is to detect intrusive activities while they are taking place on computer and network systems. System activities usually are monitored by collecting data of system activities and analyzing the data to detect intrusive activities. Once an intrusion is detected, intrusion reaction is then triggered to assess the damage of the intrusion and to take actions for system recovery and further intrusion prevention.

1.2 RESEARCH FRAMEWORK IN THIS DISSERTATION

This dissertation is focused on applying attribute selection, nonlinear feature extraction methods and control chart techniques for computer network intrusion detection. Anomaly detection methods through statistics-based machine learning will be proposed for computer networks. Two problems are studied in this dissertation. The first one is to reduce the number of attributes for data dimension reduction, i.e., selecting only informative attributes to reduce monitoring attribute dimension for effective intrusion detection. The other is to design control charts for the computer networks anomaly detection. In the case of having both in-control training samples and out-of-control training samples present, we will develop supervised control charts; otherwise, when only in-control training samples are available, we will develop unsupervised control charts.

1.3 CONTRIBUTION OF THIS DISSERTATION

In order to deal with the first problem, we propose a two-phase attribute selection algorithm to reduce data dimension for effective intrusion detection. In phase I, a filter model is used to reduce dimensionality and to keep the correlated attributes only. This is followed by Phase II where a wrapper model is employed to exploit the most important attributes. The proposed algorithm is applied to the intrusion detection problem that has five classes of network connection states. We use correlation based filter model in Phase I and GA based attribute selection model in Phase II. Multiclass support vector machine (SVM) is used as the learning algorithm embedded in GA based attribute selection model; minimal output coding (MOC) is applied to improve computing efficiency.

The second problem requires more detailed research on the improvement of currently used multivariate control charts. Most of the prevailing multivariate control charts are constructed on pre-known distribution data. However, in the application of network intrusion detection, many attributes are in non-Gaussian distributions. A new type of control chart designed in feature space is proposed in this dissertation for those problems that are not suitable for designing control charts in the original data space. We develop two types of non-parametric control charts based on kernel methods to solve the problem of non-Gaussian distributed data based on different initial situations.

In the case of having both in-control samples and out-of-control samples, we extend support vector machine to construct supervised control charts that are able to deal with both Gaussian data and non-Gaussian distribution data. We also propose a multi-objective genetic algorithm to evaluate the pre-selected parameters in radial basis function (RBF) based SVM to obtain the optimal tradeoff between type I and type II errors in the design of control charts.

In case only in-control samples exist, we develop two types of unsupervised multivariate control charts. The first one is kernel principal components analysis (KPCA) based control chart that is able to conduct nonlinear transform of the original data to a feature space and then construct control chart on the first several orthogonal principal components in the feature space that contribute most of the variance of the data. The second one is support vector clustering (SVC) based control chart, which is used to find the minimal hypersphere to enclose most of the data in a feature space.

We also propose methods for probabilistic output of SVM based control charts to facilitate implementation. The method is to use logistic curves to directly transform the evaluation function values of kernel methods to probabilistic values for decision-making of anomaly detection.

By combining the advantages of the kernel-based method and the simplicity of control charts, a generic kernel based multivariate control chart is proposed. This framework is tested by computer network intrusion detection application in Chapter 4 and Chapter 5 of this dissertation.

1.4 OUTLINE OF THE DISSERTATION

Three fundamental research issues related to attribute selection and supervised or unsupervised nonlinear kernel based control chart construction will be addressed in this dissertation. The outline of the dissertation is provided in Figure 1.1.

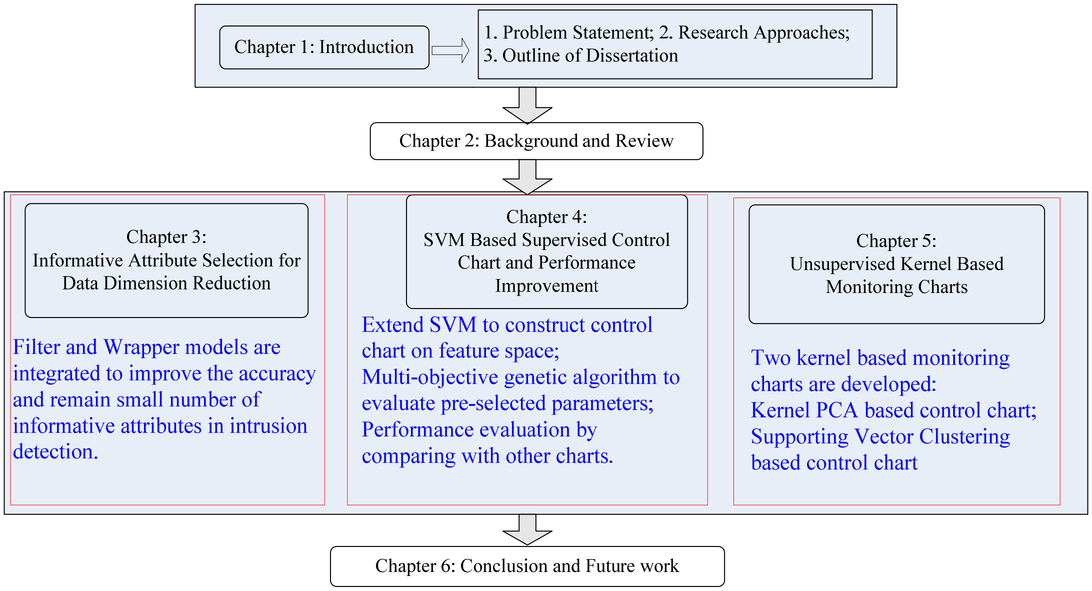

Chapter 1: Introduction
1. Problem Statement; 2. Research Approaches; 3. Outline of Dissertation
Chapter 2: Background and Review
Chapter 3: Informative Attribute Selection for Data Dimension Reduction
Filter and Wrapper models are integrated to improve the accuracy and remain small number of informative attributes in intrusion detection.
Chapter 4: SVM Based Supervised Control Chart and Performance Improvement
Extend SVM to construct control chart on feature space; Multi-objective genetic algorithm to evaluate pre-selected parameters; Performance evaluation by comparing with other charts.
Chapter 5: Unsupervised Kernel Based Monitoring Charts
Two kernel based monitoring charts are developed: Kernel PCA based control chart; Supporting Vector Clustering based control chart
Chapter 6: Conclusion and Future work

Figure 1.1 Flow chart of dissertation

Chapter 1 introduces the problems and outlines the proposed research approaches. The corresponding state of the art and the need of the proposed research approaches are reviewed in Chapter 2.

The development of the proposed methodologies is discussed in Chapters 3, 4 and 5. Chapter 3 proposes a method to make effective attribute selection in the data-rich environment and applies this method to computer network intrusion detection problem. Chapter 4 discusses the design of non-linear supervised kernel based multivariate control charts and compares its performance with other control charts and methods in the context of intrusion detection applications. Chapter 5 is focused on two unsupervised kernel based multivariate control charts: kernel based PCA chart and unsupervised support vector clustering based control chart. The performance of the support vector clustering based control chart on intrusion data is also studied. Finally, conclusions and discussions of possible future research are presented in Chapter 6.

CHAPTER 2 REVIEW ON FEATURE EXTRACTION METHODS AND COMPUTER NETWORK INTRUSION DETECTION TECHNIQUES

In this chapter, the state of art of feature extraction methods is reviewed. The topics relate to feature extraction methods are introduced, i.e., linear and nonlinear feature extraction methods, attribute selection methods and novelty detection based on feature extraction. This chapter also reviews the background of computer network intrusion detection and the currently used techniques for intrusion detection in details. In conclusion of this chapter, we give the critical research issues for anomaly detection in this dissertation.

This chapter is organized as follows: First in Section 2.1, the feature extraction methods are reviewed by its categories. In Section 2.2, intrusion detection techniques are reviewed. Then in Section 2.3, the problems to be solved in this dissertation are summarized.

2.1 REVIEW ON FEATURE EXTRACTION METHODS

Feature extraction methods have been widely used for human facial recognition, hand-written recognition, image processing and other fields that need to extract most important information from original data for identification, detection, etc. The extracted feature can be part of the original data, or transform of the original data. Feature extraction is accomplished by constructing a mapping from the measurement space to another space, either through a linear or nonlinear mapping.

In this section a brief review of feature extraction methods will be given. This review includes linear feature extraction methods, nonlinear feature extraction methods, feature selection and anomaly detection based on feature extraction.

2.1.1 Linear Feature Extraction Methods

Linear feature extraction is mainly based on multivariate statistical methods. Most of the multivariate statistical analysis methods such as PCA, Fisher discriminate analysis (FDA), factor analysis are examples of linear feature extraction methods. Because of its simplicity, linear feature extraction methods are widely used in many applications.

Jain et al [1] reviewed on pattern recognition, and pointed out that feature extraction and selection are most important issues in pattern recognition. In this review, most of the linear feature extraction methods are summarized and compared in detail.

2.1.2 Nonlinear Feature Extraction Methods

Linear feature extraction only transforms data linearly (rotation, linear projection), but does not change the shape of the data. In some cases, when linear transforms could not discover the key feature of data, nonlinear feature extraction methods are used instead. Methods falling in nonlinear feature extraction methods include principle curves, artificial neural networks, radial basis functions, etc.

Koontz et al. [2] proposed a scalar distance function using one-dimensional function approximation for pattern recognition. The multivariate mapping is obtained by the distance function. Fukunaga et al. [3] converted the problem of optimal feature extraction to an intuitive function of the posterior probabilities. Park et al. [4] presented a nonlinear feature extraction method to reduce dimension through an implicitly mapping, and then extract orthonormal basis of centroids which can maximally separate classes.

Mao et al. [5] proposed nonlinear networks include a network for nonlinear projection, a nonlinear discriminant analysis (NDA) network, and a network for nonlinear projection (NP-SOM) based on Kohonen's self-organizing map for feature extraction and data projection. They are all adaptive learning algorithms and powerful for high dimensional data. Kocsor et al. [6] discovered the application of nonlinear feature extraction methods such as kernel principal component analysis (KPCA), kernel independent component analysis (KICA), kernel linear discriminant analysis (KLDA), and kernel springy discriminant analysis (KSDA) to the classification of phonemes in a phonological awareness drilling software package.

2.1.3 Attribute Selection Methods

Attribute selection (also called feature selection or feature subset selection in some papers) is one category of feature extraction methods. Attribute selection is to select the most significant attributes directly from the existing attributes without any transform.

In multivariate regression, the number of independent variables is reduced by attribute selection and the most significant variables are obtained through statistical hypothesis testing. The selection methods include forward selection, backward selection and stepwise selection. Recent years attribute selection is widely applied to information technology. When huge dataset is available and an optimal subset is desirable, attribute selection method can obtain most important attributes representing the information of original dataset.

Jain et al. [1] reviewed the existing attribute selection methods such as exclusive search, brand-and-bound search, sequential forward search, sequential backward search and sequential forward / backward floating search. Kudo et al. [7] compared several attribute selection methods for large-scale attribute selection. 1-NN classifier is used by leave-one-out correct-classification rate for the evaluation of the methods. The authors found that sequential floating search methods are suitable for small and medium-scale problems and genetic algorithms are suitable for large-scale problems.

Sebban et al. [8] exploited the geometrical information contained in the minimum spanning tree (MST) built on the learning set and use statistical test of relative certainty gain as the criteria for goodness of selection. By forward selection algorithm, the authors developed a hybrid model for attribute selection.

2.1.4 Novelty Detection Based on Feature Extraction

Novelty detection is the identification of new or unknown data / signal / pattern that a learning system is not aware of during training, or the detection of novel or abnormal events or patterns. Novelty detection has been a popular research topic and addressed a wide range of applications in signal processing, statistical process control, fault detection, sensor networks, hand written digit recognition, health care, epidemiology, information security, computer intrusion detection, homeland security and bioinformatics etc.

Traditional statistical approaches are applied to novelty detection [9]. Statistical approaches process data by estimating the distribution of data, i.e., constructing the probability density function. Two main approaches are parametric and non-parametric methods. Parametric approaches assume that data is independent and in identical statistical distribution, thus the problem of novelty detection is converted to parameter estimation. When normal distribution is used, mean and variance are the main parameters to be estimated. If there is more than one Gaussian model, Gaussian mixture model (GMM) is used and expectation-maximization (EM) algorithm is used to estimate model parameters. Typical non-parametric approaches are KNN (K-nearest neighborhood), rule-based methods, and string matching approaches. Neural network based approaches can be applied to novelty detection [10]. If the data is not suitable to be fitted with a statistical distribution, neural network based approaches are good alternatives. Many modern methods fall into neural network based approaches, such as multi-layer perceptions, RBF networks, Hopfield networks and self-organizing maps (SOM).

If the novelty is difficult to be detected on original data or original data has a large number of attributes, neither statistical nor neural networks could perform very well for the detection. In that case, feature extraction methods can reduce the data dimension or detect the novelty more easily. The approaches to be reviewed in this section include wavelet-based approaches [11, 12], Multivariate statistics based approaches, nonlinear feature extraction approaches and spatio-temporal hotspot analysis approaches.

2.1.4.1 Multivariate statistics based feature extraction for novelty detection

PCA is a traditional way to extract principal components that contribute for most data variance, therefore it can be used to reduce the dimension of data. After the small numbers of features (principal components) are extracted, the traditional multivariate statistical methods can be constructed on the reduced features to detect novelty.

T^2 control chart [13] is a typical multivariate statistical way to detect out-of-control samples. The key point of T^2 control chart is to construct a feature – distance for multi-normal distribution.

Lowry et al. [14] gives a review on multivariate control chart. Actually most of the multivariate control charts use the idea of feature extraction to reduce dimension and construct features that are easy to distinguish in-control samples from out-of-control samples, and then construct statistics to check if a new coming sample is in-control or out-of-control.

2.1.4.2 Nonlinear feature extraction for novelty detection

Multivariate statistics based feature extraction methods apply linear transforms directly on original data to construct features for novelty detection. Sometimes the data is not in known distribution, traditional multivariate statistical methods fail to extract informative features. Recently kernel methods obtain wide attention and many applications are developed [15-17]. The key idea of kernel methods is to transform the original data from input space to feature space through nonlinear kernel transform. In feature space the transformed data can be linearly separated, while in original space (input space) it can not. This approach is often useful for non-Gaussian distribution data. In this dissertation several kernel based feature extraction methods will be extended for control charts, which are used for anomaly detection.

2.1.4.3 Spatio-temporal hotspot analysis for novelty detection

The application of spatio-temporal clustering mainly uses the model of spatial-correlated and /or time-correlated data to detect pattern changes in space and /or time.

There are two typical ways for spatio-temporal hotspot analysis: retrospective model and prospective model. The example of retrospective model is in [18] which provides a framework for spatial clustering. Lawson [19] and Clark et al. [20] discussed statistical spatial analysis of small area health data. The typical example of prospective model is prospective Space-Time Scan Statistic [21] that is designed to detect the geographical disease outbreaks irrespective of its location and size.

The spatio-temporal modeling is a dynamic problem, and the key is how to model the spatial and temporal correlations. The model relates to the type of output (either continuous or categorical/discrete). Models for continuous longitudinal data form have been well-developed [22, 23]. Nowadays, categorical (nominal, ordinal and binary) and discrete outcomes are also very prominent in statistical practice.

Two fairly different views can be adopted. The first one, supported by large-sample results, states that normal distribution theory should be applied as much as possible, even to non-normal data such as ordinal scores and counts. A different view is that each type of outcome should be analyzed to exploit the nature of the data, giving categorical data, counts, etc., by using the proper methods for analysis.

2.2 REVIEW ON COMPUTER NETWORK INTRUSION DETECTION

2.2.1 Types of Computer Attacks

Kendall [24] describes taxonomy of attacks, grouping them into four major categories: DoS, R2L, U2R and Probe. Lazarevic et al. [25] added one more category: Trojan horses/worms. So the major types of computer attacks are listed as following:

- DoS (Denial of Service) attacks
    - DoS attacks attempt to shut down a network, computer, or process, or otherwise deny the use of resources or services to the authorized users
    - Distributed DoS attacks
- Probe (probing, scanning) attacks
    - Attacker uses network services to collect information about a host (e.g. list of valid IP addresses, what services it offers, what is the operating system)
- Compromises - attackers use known vulnerabilities such as buffer overflows and weak security to gain privileged access to hosts
    - R2L (Remote to Login) attacks - attacker who has the ability to send packets to a machine over a network (but does not have an account on that machine), gains access (either as a user or as a root) to the machine and does harmful operations
    - U2R (User to Root) attacks - attacker who has access to a local account on a computer system is able to elevate his or her privileges by exploiting a bug in the operating system or a program that is installed on the system

- Trojan horses / worms – attacks that are aggressively replicating on other hosts (worms – self-replicating; Trojan horses are downloaded by users)

2.2.2 Test Data Available For the Network Intrusion Detection

Ideally an IDS (Intrusion Detection System) should be evaluated on a real network and tested with real attacks. Unfortunately it is difficult to replicate those tests on which other researchers can evaluate their methods. In order to repeat those tests, the network traffic would have to be captured and reused. This raises privacy concerns, because real traffic can contain sensitive information such as email messages and passwords. The DARPA/Lincoln Laboratory IDS evaluation (IDEVAL) data sets Lippmann et al. [26] [27] overcome this difficulty. This project had two goals. The first goal was to test a wide variety of systems (host or network, signature or anomaly, four different operating systems) on a wide range of attacks. The second goal was to provide off-line data to encourage development of new systems and algorithms by publishing a standard benchmark so that researchers could compare systems and replicate results. Evaluations were conducted in 1998 and 1999. The 1999 evaluation improved on the 1998 evaluation by simplifying the scoring procedure, providing attack-free data to train anomaly detection systems, adding many new attacks and one new target (Windows NT) to the three 1998 UNIX based targets. Figure 2.1 is the topology of 1999 simulation network.

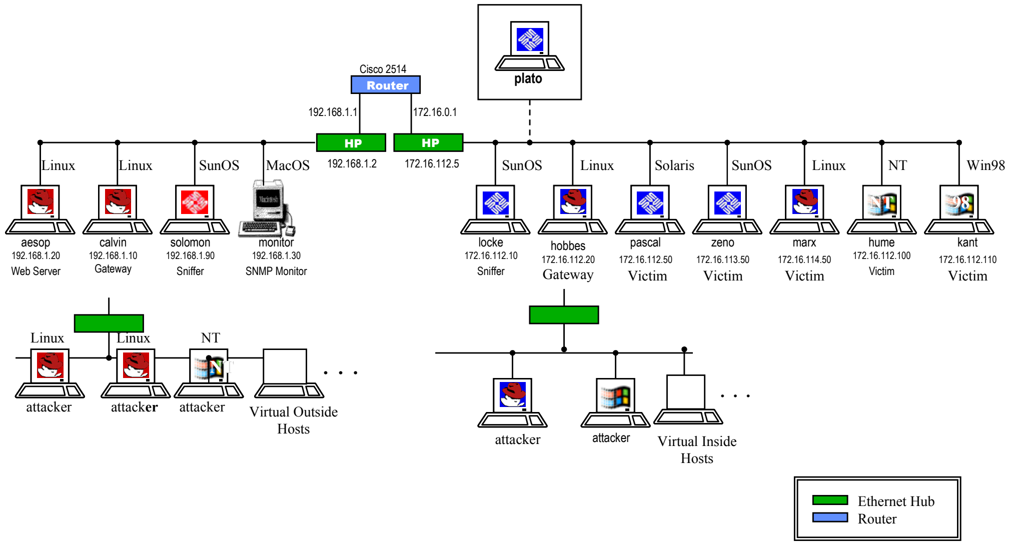


Figure 2.1 Simulation network 99 topology

Three groups of measurements called features are constructed [26, 27]:

- Content-based features within a connection. This group includes number of packets, acknowledgments, data bytes from source to destination) and intrinsic characteristics of data packets
- Time-based traffic features included number of connections or different services from the same source or to the same destination considering recent time interval (e.g.a few seconds) is useful for detecting scanning activities.
- Connection based features included number of connections from same source or to same destination or with the same service considering in last N connections. It is useful for detecting SLOW scanning activities

2.2.3 Evaluation of IDS Systems

In an intrusion detection system, there are only two outputs of the detection for a specific connection (record), true or false. And there are only two true connection outputs, also true or false. In Table 2.1 there are definitions of the combination of actual connection label and predicted connection label.

Table 2.1 Evaluation metrics of IDS systems

Standard metrics		Predicted connection label	
		Normal	Abnormal (Intrusions/Attacks)
Actual connection label	Normal	True Negative (TN)	False Alarm (FP) α error
	Abnormal (Intrusions/Attacks)	False Negative (FN) β error	Correctly detected intrusions i.e., Detection rate (TP) = 1- β

The standard measurements for evaluating IDSs are:

- Detection rate - ratio between the number of correctly detected attacks and the total number of attacks. Detection rate (TP) = 1- β.
- False alarm (false positive) rate - ratio between the number of normal connections that are incorrectly misclassified as attacks (False Alarms in Table) and the total number of normal connections. Also called α error.
- Trade-off between detection rate and false alarm rate.
- Performance (Processing speed + propagation + reaction).
- Fault tolerance (resistant to attacks, recovery, resist subversion).

2.2.4 Intrusion Detection Techniques Review

In this section most often used intrusion detection techniques will be reviewed. As shown in Figure 2.2, in this part the IDS taxonomy includes three main entries, information source, analysis strategy, and time aspects. Actually there are other taxonomies such as architecture (single centralized and distributed & heterogeneous), activeness (active reaction and passive reaction) and continuality (continuous analysis and periodic analysis) [25]. Because those taxonomies are not closely related to this dissertation, we have excluded them in this review.

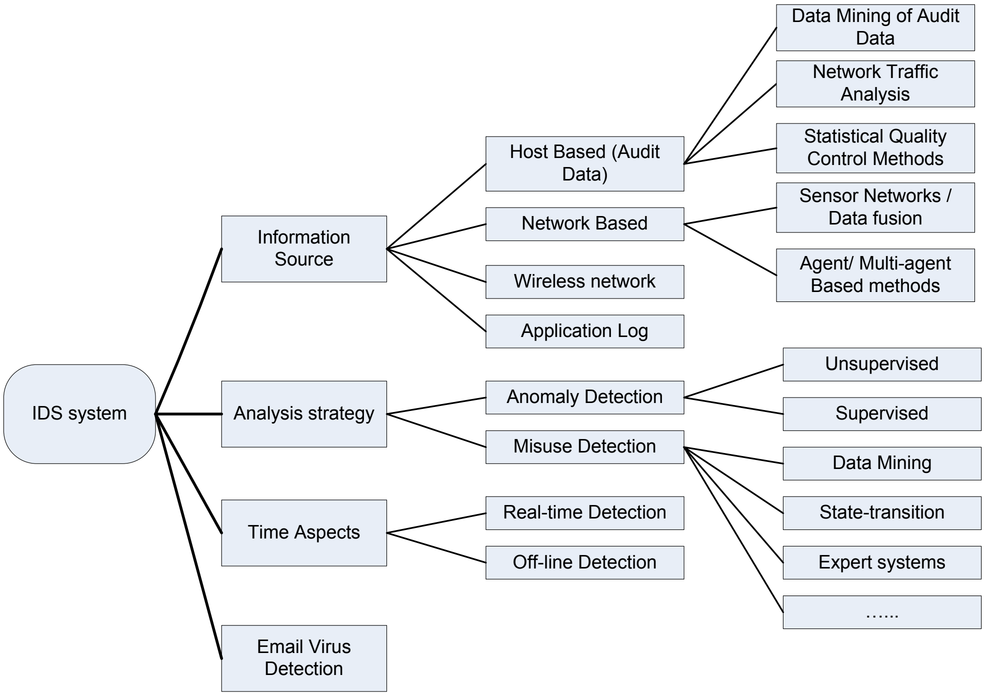


Figure 2.2 Main IDS Taxonomy

2.2.4.1 IDS according to information source

2.2.4.1.1 Host-based (Audit data) intrusion detection

(1) Data mining of audit data

A variety of data can be collected from a host machine to capture activities. Typical examples are computer audit data, system log data, and application log data [28]. Auditable events are those actions that may have security implications. The following are examples of auditable events: actions involved in authentication/ identification/ authorization; addition and deletion of objects in a user's address space; actions of adding or deleting user accounts by system administrators; use of printer, network interface card, and other I/O (input/output) devices. In short, the following information can be obtained from computer audit data: access to files, users, and processes.

Computer audit/log data can be useful in detecting intrusive activities in the gaining-access, maintaining-access, launching-further-attack, and covering-track phases, for example, gaining root user privileges, creating a user account, installing a DoS attack program, and modifying audit/log files.

(2) Network traffic analysis

Network activities can be captured by network traffic data. Because network activities involve mainly the transmission of data, network activity data are a collection of data packets being transmitted over network links. Data packets are the traffic on a network, therefore network activity data are also called network traffic data. A data packet consists of the following two parts [29, 30]: data payload and header. A data packet travels over network links in binary form. Special software programs (called sniffers) can be used to capture and interpret the binary information in a data packet. Tcpdump is a commonly used sniffer.

Network traffic data can be used in detecting intrusive. For example, access to online sources (including whois databases, DNS servers, and Web sites) for reconnaissance (attacker investigates a target computer and network system using publicly available information), and access to network services (including HTTP, FTP, and SMTP applications) for scanning active hosts and open ports for network services such as HTTP. Those intrusive activities are captured in network traffic data and can be identified using information in the header and data payload of a data packet.

(3) Statistical quality control methods

Nong Ye et al. [28] proposed to use time series modeling and single variable quality control technique to monitor network traffic data. In their study, they applied, tested, and compared two EWMA techniques to detect anomalous changes in event intensity for intrusion detection: EWMA for autocorrelated data and EWMA for uncorrelated data. Different parameter settings and their effects on performance of these EWMA techniques are also investigated to provide guidelines for practical use of these techniques. The problem of using this method is how to fit a suitable λ value for different time period. Obviously, it is better to use an adaptive EWMA model.

Nong Ye et al. [31] also investigated a multivariate quality control technique to detect intrusions by building a long-term profile of normal activities in information systems (norm profile) and using the norm profile to detect anomalies. The multivariate quality control technique is based on Hotelling's T^2 test that detects both counter-relationship anomalies and mean-shift anomalies. The performance of the Hotelling's T^2 test is examined on two sets of computer audit data: a small data set and a large multiday data set. Both data sets contain sessions of normal and intrusive activities. For the small data set, the Hotelling's T^2 test signals all the intrusion sessions and produces no false alarms for the normal sessions. For the large data set, the Hotelling's T^2 test signals 92 percent of the intrusion sessions while producing no false alarms for the normal sessions.

Markov chain was also used by several researchers [32, 33]. The application of a Markov model helps answer the question about whether the ordering property of activity data provides additional advantage to intrusion detection, or whether we can detect intrusions from only the frequency property of activity data without the ordering property. First-order and high-order Markov models can produce comparable intrusion detection performance. An intrusive event sequence is expected to receive a low probability of support from the Markov chain model of the norm profile. So, if a transition with low probability happens, it has a high probability that the event is caused by an intrusion.

Nong Ye, et al. [34] studied and compared several probabilistic techniques that were used in intrusion detection. Those include Hotelling’s T^2 test, chi-square multivariate test, and Markov chain modeling. These methods are applied to the same training set and the same testing set of computer audit data for investigating the frequency property and the ordering property of computer audit data. Their study shows that the frequency property of multiple audit event types in a sequence of events is necessary for intrusion detection. A single audit event at a given time is not sufficient for intrusion detection. They also found that the ordering property of multiple audit events provides additional advantage to the frequency property for intrusion detection.

2.2.4.1.2 Network based intrusion detection

(1) Sensor networks / data fusion

Base [35] stated the importance of data fusion for sensor networks on intrusion detection. He also mentioned that the next-generation cyberspace intrusion detection systems would require the fusion of data from myriad heterogeneous distributed network sensors to effectively create cyberspace situational awareness. He summarized the framework to use multi-sensor data fusion to combine data and information from numerous heterogeneous distributed agents (and managers) into a coherent process, which can be used to evaluate the security of cyberspace: from data to information and finally to knowledge.

Technical details on the development of multisensor data fusion and its applications can be found in [36].

(2) Agent /Multi-agent based methods

In recent years, agent /multi-agent is widely investigated and used in software engineering. Because agent has the advantage of autonomous, collaborate, flexibility, self-learning, etc., agent /multi-agent catches attention in computer network intrusion detection area.

Spafford and Zamboni [37] proposed an early prototype of multi-agent intrusion detection system and issued future research questions. Followed by him, there are many investigations on this topic. Hegazy et al. [38] proposed a multi-agent system framework for intrusion detection that has four main modules: the sniffing module, the analysis module, the decision module and the reporting module. In this framework, sniffing agents collect network traffic information first and send it to analysis module. At this stage, different analysis agent is used for different attacks, like ping sweep, DoS, security code, etc. Afterwards decision agents wok on the information and alerting agents in reporting module generate alert for the suspecting results.

Harmer et al. [39] proposed a self-adaptive distributed agent-based defense immune system based on biological strategies is developed within a hierarchical layered architecture. This intrusion detection function is based on string matching against a library of signatures. In nature this method is still in misuse detection. The communication of the agents makes this system effective. Gorodetski and Kotenko [40, 41] designed a similar system with more features such as an offline detection, learning, testing and modification.

(3) Wireless network intrusion detection

Akyidiz et al. [42] gave a review on wireless sensor networks. Samfat and mlova [43] proposed a multilevel intrusion detection architecture for GSM network. It includes: Level 1- Velocity and Clone Verification; Level 2- Componentwise Verification; Level 3- Intrusion Detection per User. The basic idea of detecting an intruder relies on the system's ability to learn the normal behavior of the subscriber by creating a user profile. In the case of GSM, the signature of the user is defined by three profiles: a mobility profile, an activity profile, and a speech profile. Each profile will help in raising different intrusion alarms that a rule-based system will analyze in order to give the final decision.

Mishra et al. [44] presented the reason of big vulnerability because of the characteristics of wireless ad hoc networks. The wireless links between nodes are highly susceptible to link attacks, which include passive eavesdropping, active interfering, leakage of secret information, data tampering, impersonation, message replay, message distortion, and denial of service. The authors also compared different proposed architectures against ideal characteristics for IDSs in mobile ad hoc network (MANETs).

2.2.4.2 IDS according to analysis strategy

In this category, there are mainly two widely investigated methods, named anomaly detection [45, 46] and misuse detection (also called signature detection, signature recognition in some literatures).

(1) Anomaly detection

Anomaly detection is based on profiles that represent normal behavior of users, hosts, or networks, and detecting attacks as significant deviations from this profile. The major benefit is that anomaly detection is potentially able to recognize unforeseen attacks. The major limitation is possible high false alarm rate, since detected deviations do not necessarily represent actual attacks. The major approaches for anomaly detection are statistical methods, expert systems, clustering, neural networks and outlier detection schemes.

Anomaly detection techniques capture both known intrusions and unknown intrusions if the intrusions demonstrate a significant deviation from a norm profile. Existing anomaly detection techniques differ mainly in the representation of a norm profile and the inference of intrusions using the norm profile.

Many studies, such as Ye, Denning [28, 34, 47], use statistical distributions to model the frequency feature and the intensity feature of normal activities for a norm profile and employ statistical tests to determine whether observed activities deviate significantly from the norm profile. An advantage of statistical-based anomaly detection techniques is their capability of explicitly representing and handling variations and noises in normal activities.

Strings-based anomaly detection techniques must rely on a large, costly repository of short sequences of normal events to capture variations of normal activities.

Both artificial neural networks and stochastic models (e.g., Markov chain model and hidden Markov model) have been used to model the order feature of normal activities (e.g., event transitions or event sequences) for a norm profile and to detect intrusions based on the deviation of the observed events from the expected event or based on the probabilistic support of the norm profile [34, 46, 48]. In the category of anomaly detection, there are two techniques, named supervised anomaly detection and unsupervised anomaly detection. Supervised anomaly detection is based on the available data containing both normal and abnormal connection records.

If we do not know which connection is normal and which is abnormal, we need to use unsupervised anomaly detection, also called outlier detection. Most widely used technique is clustering. Model based clustering and distance based clustering are typical ways for unsupervised anomaly detection.

(2) Misuse detection

Misuse detection compares activities in a computer and network system with signatures of known intrusions, and signal intrusions when there is a match. For a subject (user, file, privileged program, host, network, etc.) of interest, anomaly detection techniques establish a profile of the subject's long-term normal behavior (norm profile), compare the observed behavior of the subject with its norm profile, and signal intrusions when the subject's observed behavior deviates significantly from its norm profile.

Misuse detection techniques utilize intrusion signatures, profiles of intrusion characteristics, and consider the presence of an intrusion signature as evidence of an intrusion. Anomaly detection techniques use only data of normal activities in a computer system for training and building a norm profile. Signature recognition techniques rely on data of both normal and intrusive activities for learning intrusion signatures, either manually or automatically, through data mining.

Most commercial intrusion detection systems are based on misuse detection techniques [49]. Intrusion signatures have been characterized as strings (e.g., command names), event frequency distributions, event sequences, activity graphs, and intrusion scenarios with event sequences, event preconditions, and target compromised states. Intrusion signatures have been represented using finite state machines, association rules [50] and decision trees [34] to store and recognize intrusion signatures. Intrusion signatures are either manually encoded or automatically learned through data mining. However, signature recognition techniques have a limitation in that they cannot detect novel intrusions which have unknown signatures.

Misuse detection is based on extensive knowledge of patterns associated with known attacks provided by human experts. Existing approaches include pattern (signature) matching, expert systems, state transition analysis, and data mining. Major limitations of misuse detection are:

- Unable to detect novel & unanticipated attacks
- Signature database has to be revised for each new type of discovered attack

(3) Data mining methods

Denning [47] proposed a real-time intrusion detection expert system for intrusion detection. It includes profiles for representing the behavior of subjects with respect to objects in terms of metrics and statistical models, and rules for acquiring knowledge about this behavior from audit records and for detecting anomalous behavior.

Lee et al. [50, 52, 53] proposed a framework of data mining on intrusion detection. The key ideas are to use data mining techniques to discover consistent and useful patterns of system features that describe program and user behavior, and use the set of relevant system features to compute (inductively learned) classifiers that can recognize anomalies and known intrusions.

(4) State-transition methods

Ilgun et al. [54] presented state-transition method for the first time. State transition analysis models penetrations as a series of state changes that lead from an initial secure state to a target compromised state. In this paper state transition diagrams the graphical representation of penetrations identify precisely the requirements for and the compromise of a penetration and present only the critical events that must occur for the successful completion of the penetration. State transition diagrams are written to correspond to the states of an actual computer system and these diagrams form the basis of a rule-based expert system for detecting penetrations called the State Transition Analysis Tool (STAT).

(5) Expert system methods

The rule-based methods, such as fuzzy logic [55] and rough set [56, 57] are subsets of expert system methods. Expect system methods can be incorporated into data mining methods.

2.2.4.3 IDS according to time aspects

(1) Real-time detection methods

Luo and Bridges [58] gave a real-time implementation of intrusion detection. Because of the large amount of data stream in computer networks, the real-time detection methods are still in investigation. Mainly used real-time detection methods are misused detection methods because the string match is relatively simple. Real-time anomaly detection is still a research issue.

(2) Off-line detection methods

Up to now, most intrusion detection methods reviewed in this dissertation are off-line detection methods. Off-line detection is used to understanding the attackers' behavior.

2.2.4.4 Some other techniques

Data mining applies machine learning and statistical techniques to automatically discover and detect misuse patterns, as well as anomalous activities in general. When applied to network-based activities and user account observations for detection of errant or misuse behavior, these methods are referred to as behavior-based misuse detection [59].

Email virus is another computer intrusion. In [60] behavior profiles of user email accounts are built to detect viral propagations. Totally 3 modeling techniques of behavior profiles are used: user cliques, Hellinger distance and daily cumulative distribution of emails. To achieve high detection rates with remarkably good FP rates, the combinations of these three models are investigated.

2.3 PROBLEMS NEED TO BE SOLVED

There are many challenges in the research field of computer network intrusion detection, such as:

- Large data size and high dimensionality. Millions of network connections are common for commercial network sites. Hundreds of dimensions are possible because of the availability of hardware/software to collect large amount of properties of transaction data.
- Skewed distribution. Interesting events are very rare; unkown or complex distributions for many attributes in computer networks data
- Difficulty for online detection. Currently misuse detection is mainly used for online detection, but it lacks the ability to detection new intrusion types.

This dissertation is focused on integrating statistical and machine learning on transformation and extraction methods for computer network intrusion detection. Two problems are raised in this dissertation, one is how to reduce the dimension (number of attributes) for effective data processing, i.e., selecting only informative attributes to reduce data dimension for effective intrusion detection. The other is to design control charts for the anomaly intrusion detection, in which the data can only be separable with a nonlinear boundary.

CHAPTER 3 INFORMATIVE ATTRIBUTE SELECTION FOR DATA DIMENSION REDUCTION FOR EFFECTIVE INTRUSION DETECTION

This chapter presents a two-phase attribute selection method for data-rich environment and applies this method to computer intrusion detection data. Filter and Wrapper models are integrated to improve the accuracy and extract a small number of informative attributes for computer network intrusion data. In the proposed hybrid attribute selection method, Filter model based on dependency is firstly applied In Phase I to reduce the dimension of large dataset while keeping the most significant attributes. This is followed by Phase II, where a wrapper model is employed to exploit the most important attributes without redundancy. The performance of each step in Phase I and II is examined to illustrate the effectiveness of the joint model.

This chapter is organized as follows. In Section 3.1, the detailed methods of feature selection methods are discussed. In Section 3.2, a new hybrid attribute selection method combining filter and wrapper models is proposed and discussed in details. Section 3.3 demonstrates how the proposed method is applied to the intrusion detection. Discussion is given in Section 3.4.

3.1 INTRODUCTION

In recent years, computer networks security has been a growing problem. More and more computers have undergone vulnerability problems by cyber attacks, such as network intrusion, resulting in huge loss. Therefore, computer network intrusion detection is becoming an increasing important research topic since the last decade. To implement intrusion detection in a computer network environment, tools such as *TCPdump* have been used to trace the connections to the server. The collected records can be used to detect whether these connections are normal or abnormal. In the popularly used database developed by MIT Lincoln Labs, the network connections are mainly classified into five states, which are corresponding to the normal state representing legal connections and other four attack states named DoS, U2R, R2L and Probe [24]. Table 3.1 lists the detailed definitions of these states with some typical examples of attack actions. In this database, 41 attributes are used in the network operational tracing database, which are classified as content-based attributes, time-based traffic attributes, and connection based attributes. Based on the network operation structure, intrusion detection methods are further divided into host based, network based, wireless network, application log and sensor alerts [25]. In this dissertation, we mainly deal with the host based monitoring for misuse detection, in which the intrusion detection methodology will be developed based on the public KDD CUP 1999 data, the description of the attributes can be found from Appendix A.

Table 3.1 Types of network states (including normal and attacks)

Label	Name	Definition	Examples
1	Normal	Legitimate connection	
2	DoS	Denial of service	ping-of-death, teardrop, smurf, syn flood, etc
3	R2L	Remote-to-local. Unauthorized access from a remote machine	guessing password
4	U2R	User-to-root. Unauthorized access to local superuser privileges by a local unprivileged user	various of buffer overflow attacks
5	Probe	Surveillance and probing	port-scan, ping-sweep, etc

There are two main different approaches for intrusion detection according to analysis strategy: misuse detection and anomaly detection [61]. Misuse detection is based on users' signatures, which are characterized by the operation rules and procedures, to detect illegal connections. Anomaly detection relies on statistical monitoring of network operational data. In this approach, the normal behavior baselines are first built based on historical legal operation data. The anomalies are suspected when monitored operational data has a significant deviation from the normal baseline. In this dissertation, we will focus on the investigation of statistical anomaly detection methods.

It is known that the performance of statistical detection methods is usually severely degraded for a large dimension of data. Therefore, the data dimension reduction is always considered as a critical step for the detection method development. In this chapter, we will first study how to select effective monitoring attribute subset for the intrusion detection purpose. The resultant attribute subset will be used for the detection method development, which will be discussed in Chapter 4 and Chapter 5.

3.1.1 Discussion on Attribute Selection

The purpose of the attribute subset selection is to reduce the number of attributes used to characterize a dataset under a given data analysis objective. Attribute selection in machine learning has shown its impressive performance gains by reducing a large dimensionality through removing many irrelevant attributes [62], thus leading to enhanced analysis results. In such research, the problems are usually exposed as search problems, in which many heuristic search algorithms have been developed in order to solve those problems more efficiently. In general, a search algorithm needs to address following four basic issues:

- Starting point in the search space;
- Organization of the search;
- Evaluation strategy of the selected subset;
- Stopping criterion for halting the search.

The selection of starting point determines the initial search space and provides distinction between forward selection (starting with no attribute and then adding new attributes sequentially) and backward selection (starting with the whole attribute set and then shrinking it until the desired subset is reached).

The organization of the search determines the search strategy in the space of size $2d$ where d is the number of attributes. One popularly used non-exhaustive optimal search strategy is the branch and bound algorithm [63], and its optimality is guaranteed if the evaluation function is monotonic. When the monotonic condition is not satisfied, heuristic search methods are often used, such as simple deterministic heuristic algorithms of sequential forward search (SFS) and sequential backward search (SBS), and more sophisticated strategy of floating search and best-first search [64]. Results from [65] suggested that those simple greedy hill-climbing approaches may get trapped on local peaks caused by interdependencies among attributes.

On the other hand, non-deterministic approaches using a random search have been recently investigated for the purpose of avoiding local optimum, such as Genetic Algorithms (GA) [66], evolutionary computation [67], and Las Vegas Algorithms [68], etc. By applying these non-deterministic search algorithms, different analysis results may be generated from different runs.

An evaluation function is used to measure the effectiveness of a selected attribute subset under the given objective of its maximization or minimization. Depending on whether such a measure can directly carry out this objective, two different types of evaluation approaches are used [64, 69]: wrapper approaches and filter approaches.

In the wrapper approach, a good attribute subset is determined by using classification performance itself as the evaluation function, which directly relates to the objective of minimizing the classification errors. Therefore, the wrapper approach can guarantee the final learning accuracy. However, a wrapper approach needs to design a classifier in every step of searching, thus requiring extremely high computation.

The filter approach assesses the attribute only based on its intrinsic data properties, i.e., whether it is potentially relevant to the classification learning algorithm. The name is due to the fact that the attribute selection is done by filtering out the irrelevant attributes before applying the learning algorithm. Almuallim, et al [70] designed filters by checking data consistency, i.e. the association between the combination of every value for a attribute subset and the class label. Koller [71] eliminates the redundant attributes that are already included in the selected attributes. The other popularly used filter method is based on predefined relevancy score [72], such as distance measure (Euclidean distance measure), information (entropy, information gain), dependency (correlation coefficient) and consistency (minimal-attributes bias), etc. Because a filter approach is independent of the learning algorithm and has the simplicity of the measures, it shows significant advantage on the computation time by comparing with the wrapper method. However, it cannot guarantee the final learning performance by using the selected attributes.

In conclusion, the filtering approach is flexible that can be integrated with any learning algorithms, while the wrapper approach is strictly dependent on the specific learning algorithms. The filter approach requires less computation complexity and is suitable to efficiently handle a large dataset, but it cannot guarantee the final accuracy of the learning algorithm followed. The wrapper approach integrates learning algorithms into evaluation functions, thus requiring extensive computation before it can determine the optimal subset of attributes. It is usually suitable to handling a small dataset in terms of both data dimension and sample size with the guaranteed learning performance.

3.1.2 Contribution of This Chapter

In order to fully utilize the advantages of both filter and wrapper approaches, a hybrid approach is proposed in this dissertation for a large dataset analysis through two-phases attribute selection analysis. In Phase I, a filter model is used to filter out those attributes irrelevant to network states, in which the correlation between attributes and labeled classes is used as an evaluation function. This pre-filtering can be efficiently used for the whole dataset analysis. The elimination of those irrelevant attributes in this step can efficiently reduce the attribute dimension, which is essential to perform a wrapper approach in Phase II. In Phase II, a further attribute selection is exploited from the remaining attribute subset after Phase I filtering. The classification learning performance is used for final decision of the optimal attribute subset. Multiclass SVM is used as the learning algorithm embedded in the GA (Genetic Algorithm) searching algorithm. Different from traditional two-class SVM, a minimal output coding (MOC) is used to achieve a higher computing efficiency for a multiclass SVM. The comparison of the results in each phase shows the goodness of the method presented.

3.2 PROPOSED HYBRID ATTRIBUTE SELECTION METHOD FOR MULTI-CLASS CLASSIFICATION

3.2.1 General Analysis Framework

The proposed hybrid approach for attribute selection is a generic approach which provides a feasible way to handle a large dataset analysis (In the exemplary database, the dataset consists of 41 attributes and 311029 records). The two-phase analysis framework is illustrated in Figure 3.1.

In this framework, Step 1 is considered as Phase I analysis, in which the correlation analysis is used as a data filtering approach to remove the irrelevant attributes. Steps 2~6 are considered as Phase II analysis, in which the supervised SVM classification combined with GA is used for attribute selection through the wrapper approach. The whole dataset is divided into training dataset and test dataset, in which the former one is used for Step 1~6 analysis to determine an optimal attribute subset, and the later one is used for evaluation of the selected attribute subset. The details of these analyses in each step are described as follows:

Step 1: Attribute dimension reduction through the filter approach. The correlation analysis between the selected attribute and the labeled classes corresponding to the network operational states is used as an evaluation function, in which insignificant correlated attributes are removed.

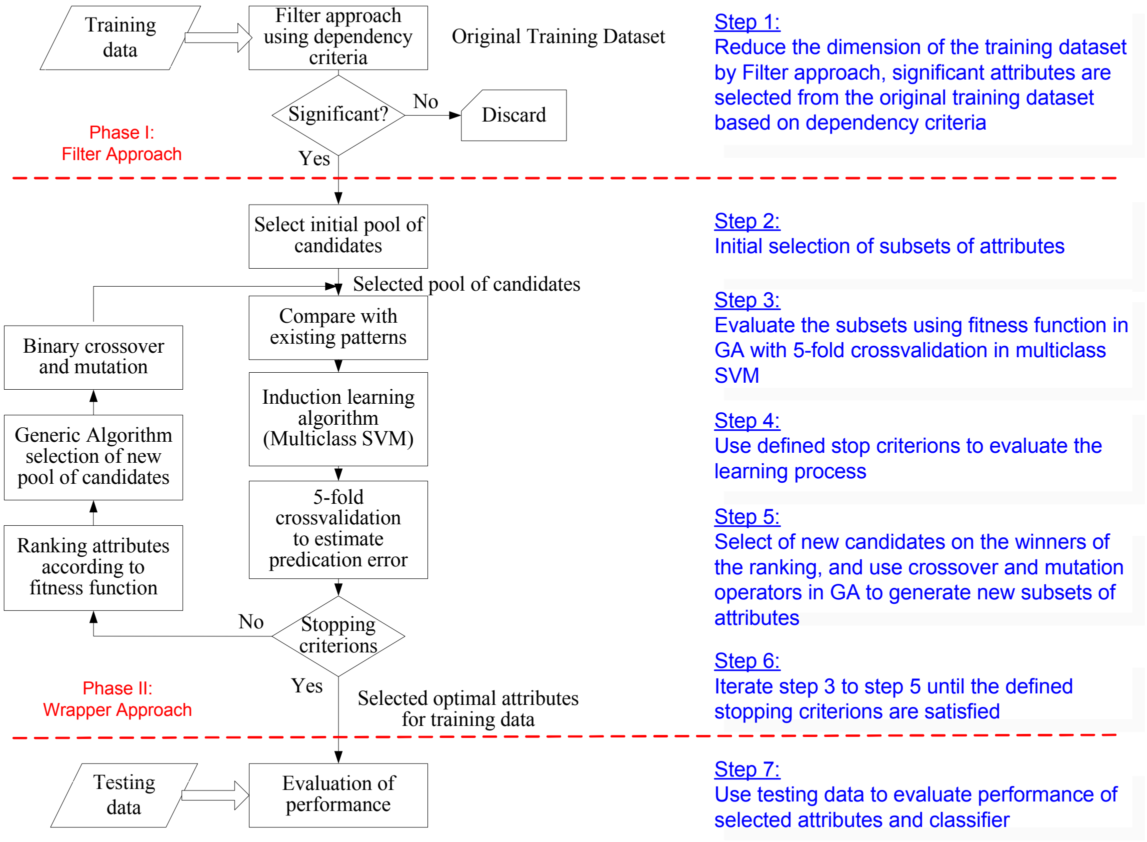

Training data
Filter approach using dependency criteria
Original Training Dataset
Step 1:
Reduce the dimension of the training dataset by Filter approach, significant attributes are selected from the original training dataset based on dependency criteria
Significant?
No
Discard
Yes
Phase I:
Filter Approach
Select initial pool of candidates
Step 2:
Initial selection of subsets of attributes
Selected pool of candidates
Compare with existing patterns
Binary crossover and mutation
Step 3:
Evaluate the subsets using fitness function in GA with 5-fold crossvalidation in multiclass SVM
Induction learning algorithm (Multiclass SVM)
Generic Algorithm selection of new pool of candidates
Step 4:
Use defined stop criterions to evaluate the learning process
5-fold crossvalidation to estimate predication error
Ranking attributes according to fitness function
Step 5:
Select of new candidates on the winners of the ranking, and use crossover and mutation operators in GA to generate new subsets of attributes
No
Stopping criterions
Yes
Phase II:
Wrapper Approach
Selected optimal attributes for training data
Step 6:
Iterate step 3 to step 5 until the defined stopping criterions are satisfied
Testing data
Evaluation of performance
Step 7:
Use testing data to evaluate performance of selected attributes and classifier

Figure 3.1 Framework of combining Filter and Wrapper models for attribute selection

Step 2: The remaining significant attributes after Step 1 are considered as an initial pool of candidates for the wrapper analysis.

Step 3: The reduced dimensional training candidates is sent to GA based wrapper attribute selection model, where GA starts through the configuration like population size, maximum generations, probability of crossover and probability of mutation, etc. In this step the main task is to generate initial selection of the attribute subset. In GA based attribute selection, all the bits in the initial chromosomes are set to 0, which means at the beginning of attribute subset selection, there is no attribute to be selected in the subset. In this sense, it is similar to a forwarding selection approach. The classification accuracy is used as the performance evaluation of the selected attribute subset. Therefore, the fitness function in GA is set as the accuracy of the classifier, which is evaluated by the 5-fold cross-validation estimate of the classification errors in multi-class SVM. The optimal attribute subset is determined, which has the highest classification accuracy and the minimal number of attributes. This is an iterate search approach, in which each subset is compared with the best subset in the historical runs. The historical best subset will be replaced by the current subset either having a higher accuracy or the same accuracy but with less number of attributes included in the subset.

Step 4: After each generation of a subset is finished in fitness function evaluation, the stop criterion will be checked. In this step, two stopping criteria are used. One criteria is the generation limit set by the largest generation of GA. The other criteria is that if there is no change on the optimal attribute subset for more than 10 generations, it will stop and break the iterate loop.

Step 5: Generate a new candidate subset based on the winners of the ranked attributes. The chromosome with the higher classification accuracy will have higher probability to be selected, so the winners of the ranking will be reproduced more pieces than others. Two operators are used and a new subset of attributes is created, in which each selected chromosome will be operated according to the probability of the operators.

Step 6: The loop from Step 3 to Step 5 will be iterated until either of two stop criterions is satisfied.

Step 7: The best attribute subset is obtained based on the training dataset through Steps 1~6 analyses. This step is used to further evaluate the performance of the selected optimal attribute subset and the multi-class SVM classifier based on the testing dataset.

It is worthwhile to suggest that in the implementation of the proposed GA based wrapper, the program should keep track of the history of all existed chromosomes. The reason is that the new chromosome may sometime be created same as one historic pool of chromosomes. In this case, its performance can be directly recalled from the historical record instead of training and evaluating it again. This trick is especially helpful to save computing time when a large dataset is analyzed.

3.2.2 Proposed Filter Approach for Numeric and Discrete Data Analysis

3.2.2.1 Data types

The data types used in the network operational tracing database are very diverse, which are generally classified as two major categories as shown in Figure 3.2: quantitative measurements (also called numerical variables) and qualitative measurements (also called categorical variables).

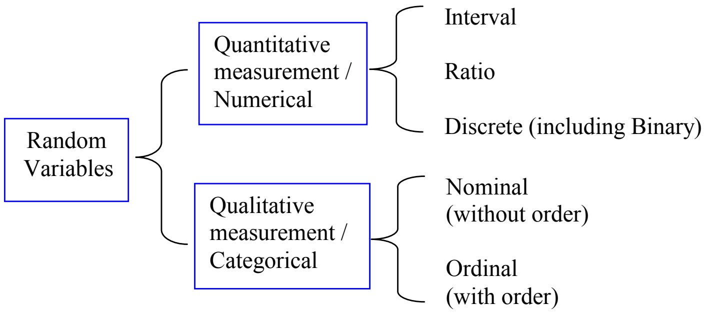

Random Variables
Quantitative measurement / Numerical
Interval
Ratio
Discrete (including Binary)
Qualitative measurement / Categorical
Nominal (without order)
Ordinal (with order)

Figure 3.2 Data type taxonomy

In the first category of these quantitative measurements, three types of data, i.e., interval, ratio, and discrete variables are further classified. In the second category of qualitative measurements, the data are further classified as nominal and ordinal variables.

In the following subsection, different correlation analysis will be used as the filtering model for different types of data.

3.2.2.2 Filter analysis

Hall, et al [73] pointed out that a good attribute subset should contain attributes highly correlated with the class labels, yet uncorrelated with attributes themselves. Based on this principal, two steps of correlation analysis are used in filtering analysis. The first step of the correlation analysis is to remove those irrelevant attributes, which are not correlated with the class labels. The second step of the correlation analysis is to remove those redundant attributes, which are highly correlated with other selected attributes.

In the first step of removing irrelevant attributes, the correlation coefficient between each attribute and the class labels is calculated. If the resultant correlation coefficient is smaller than a predefined threshold value under a given α level (α=0.05 is most often used), this attribute will be considered as the irrelevant attribute to be removed. In this step, the computation time is O(p). p equals to the number of attributes.

If only filter model is used, all correlation coefficients between each pair of attributes need to be calculated and represented by a correlation matrix in the second step of removing redundant attributes. If a correlation coefficient is larger than the predefined threshold, it means one of these two corresponding attributes is redundant that needs to be removed. The computation time in this step is $O(p^2)$. Also the correlation computed by this matrix may contain multicollinearity. In this chapter, because we propose two-phase approach, we will leave the second step of removing redundant attributes to Phase II.

For the numerical data type, Pearson's correlation is calculated as follows:

$$\rho_{xy} = \frac{\Sigma xy - \Sigma x \Sigma y / n}{\sqrt{(\Sigma x^2 - \frac{(\Sigma x)^2}{n})(\Sigma y^2 - \frac{(\Sigma y)^2}{n})}} = \frac{SS(xy)}{\sqrt{SS(x)SS(y)}} \tag{3.1}$$

where x and y are the samples of attributes and n is the number of samples. The following hypothesis test is used to check whether the correlation between x and y is significant.

H_0: $\rho = 0$

H_1: $\rho \neq 0$

The corresponding test statistic is defined as:

$$t_0 = \rho \sqrt{\frac{n-2}{1-\rho^2}} \sim t_{n-2} \tag{3.2}$$

which follows a t-distribution with n-2 degrees of freedom. The significant correlated coefficients are identified if the resultant p-value of this test is smaller than the given α value, i.e., the null hypothesis of ρ=0 is rejected.

In the case that one attribute y is numerical and the other one x is categorical, a weighted Pearson's correlation is used:

$$\rho_{xy} = \sum_{i=1}^{k} p(x = x_i)\rho_{x_{bi}y} \tag{3.3}$$

Here, variable x is assumed to have k different categorical values and each x_i is a binary data, in which x_i=1 if the ith category of x occurs, and x_i=0 for all other cases. $p(x = x_i)$ is the prior probability that x takes value x_i, which is used as the weight for the weighted Pearson's correlation analysis.

Similarly, when both attributes involved are categorical, the weighted correlations are calculated for all possible combinations as follows:

$$\rho_{xy} = \sum_{i=1}^{k}\sum_{j=1}^{l} p(x = x_i, y = y_j)\rho_{x_{bi}y_{bj}} \tag{3.4}$$

3.2.3 Proposed Wrapper Approach Based on Supervised Classification

3.2.3.1 Wrapper search strategy

After the filtering analysis in Phase I, only a small number of attributes are remained for the wrapper model in Phase II. In the wrapper model, a genetic algorithm (GA) is used as the search strategy because GA is a stochastic searching technique which can avoid local optima. In this approach, binary representation of chromosome is used to indicate the selection state of attributes, in which each bit of the chromosome has only two values with 0 standing for "not selected" and 1 standing for "selected".

Normalized geometric ranking P_i is defined for individual attribute as:

$$P(\text{Selecting the } i\text{th individual}) = \tfrac{q}{1-(1-q)^N}(1-q)^{r-1} \tag{3.5}$$

where

q = the probability of selecting the best individual

r = the rank of the individual, where 1 is the best

N = the population size

Simple crossover that is based on a random number r generated by a uniform distribution from 1 to m, is used to create two new individuals according to the following equations

$$x' = \begin{cases} x_i, & \text{if } i < r \\ y_i, & \text{otherwise} \end{cases} \tag{3.6}$$

$$y' = \begin{cases} y_i, & \text{if } i < r \\ x_i, & \text{otherwise} \end{cases} \tag{3.7}$$

Binary mutation flips each bit in every individual with probability p_m as

$$x_i' = \begin{cases} 1 - x_i, & \text{if } U(0,1) < p_m \\ x_i, & \text{otherwise} \end{cases} \tag{3.8}$$

Based on these two operators of crossover and mutation, the chromosome can be changed stochastically into another combination of attributes. In fact, this approach is mainly relied on some heuristics that ensure the better attributes are generated through such stochastic combinations.

The classifier is embedded into the wrapper for the task of classification for the attribute performance evaluation, in which support vector machines (SVM) is used as the classifier [25, 74]. SVM is mainly used as a binary classifier that deals with 2-classes classification problem. In this chapter, we extend SVM for the multi-class classification problem by using the coding-decoding scheme. The coding-decoding scheme is integrated with SVM to encode and decode a multi-class classification task into multiple binary classifiers.

For solving multi-class classification problems, we reformulate the multi-class problem (n_c classes) into a set of n_y binary classification problems. For each class, C_i is a unique codeword $[y_i^{(1)}; y_i^{(2)}; ..., y_i^{(n_y)}] \in \{-1, +1\}^{n_y}$ for i = 1, 2, …, n_C. There are several coding techniques available:

(1) Minimum Output Coding (MOC)

The minimal number of bits n_b is used to encode the n_c classes as:

$$n_b = \lceil \log_2 n_c \rceil \tag{3.9}$$

(2) Error Correcting Output Code (ECOC).

This coding scheme is derived from information theory and data communication applications. It uses redundant bits. Normally, the bounds of the number of binary classifiers n_b for n_c classes are

$$n_b \le 15\lceil \log_2 n_c \rceil \tag{3.10}$$

However, it is not guaranteed to have a valid n_b-representation of n_c classes for all combinations. This method is computationally extensive because it bases on backtracking.

(3) One versus All Coding (OneVsAll)

Each binary classifier k = 1, 2, …, n_c is trained to discriminate between class k and the union of the others.

(4) One Versus One Coding (OneVsOne)

Each of the n_b binary classifiers is used to discriminate between a specific pair of n_c classes

$$n_b = \frac{n_c(n_c - 1)}{2} \tag{3.11}$$

The proposed attribute selection method mainly deals with a large dataset. Therefore, computation time is an important factor in the coding method selection. As shown in **Table 3.1**, different coding schemes have different codeword and different length of coding. Among them, Minimal output coding (MOC) has the minimal code length, which will be used in this chapter to save computing cost. It should be clarified that the coding schemes are not dependent on what the original classes are as long as different numbers codes them.

Table 3.1 Different Encoding Schemes with Illustration

Classes	MOC	ECOC	OneVsAll	OneVsOne
1	-1 -1 -1	-1 1 -1 1	1 -1 -1 -1 -1	1 1 1 1 0 0 0 0 0 0
2	-1 -1 1	-1 1 1 -1	-1 1 -1 -1 -1	-1 0 0 0 1 1 1 0 0 0
3	-1 1 -1	1 1 1 1	-1 -1 1 -1 -1	0 –1 0 0 –1 0 0 1 1 0
4	-1 1 1	-1 -1 1 1	-1 -1 -1 1 -1	0 0 –1 0 0 –1 0 –1 0 1
5	1 -1 -1	1 1 -1 -1	-1 -1 -1 -1 1	0 0 0 –1 0 0 –1 0 –1 -1

In decoding schemes, there are two different approaches either using Hamming distance or Bayesian distance measure. Hamming distance equals to the number of the corresponding different bits between the binary result of output and the codeword. The Bayesian distance uses a matrix of probability for the binary classifiers to estimate the posterior probability in order to set a class for an input. We suppose not knowing the relationship between each class, and because Hamming distance has a good performance on error correction, Hamming decoding will be used in our analysis.

3.2.3.2 Model assessment for wrapper

It is important to have a good assessment on the classifier performance in the wrapper model. It is known that cross-validation is an effective way for estimation of classification errors. The basic ideas of cross-validation is to split the data into *K* roughly equal-sized parts, this is called *K*-fold cross validation. Every time, the other *K*-1 parts of the data are used to build the classifier. Afterwards, the *k*th part of the data is served as the test dataset to calculate the prediction error of the fitted classifier in the classification of the *k*th part of data. Suppose *N* is the total sample size, and *K* is the number of folders for cross-validation, the final prediction error is the mean of those prediction errors obtained from all *k*-fold analyses (*k* = 1, 2, …, *K*).

$$CV = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i)) \tag{3.12}$$

where $k:\{1,2,...,N\} \mapsto \{1,...,K\}$ is an indexing function that indicates the partition to which observation *i* is allocated by the randomization. $\hat{f}^{-k(i)}(x_i)$ is the fitted function of the data without the *k*th part, no matter whether this function is explicit or implicit. The typical choices of *K* are 5 or 10. The higher the *K* value is, the higher the variance is. The lower the *K* value is, the higher the bias is. So, the choice of *K* should be a good compromise of bias and variance. In this dissertation, *K*=5 is used to have faster computation.

3.3 APPLICATION IN INTRUSION DETECTION

3.3.1 Review of Datasets

The labeled KDD CUP 1999 dataset is used to illustrate the implementation and effectiveness of the proposed method. This dataset is modified from DARPA 1998 data from MIT Lincoln Laboratory [75]. The attribute selection method is developed for the purpose of classifying 5-classes network operation states, which are named as Normal and four different attacks of DOS, U2R, R2L and Probing.

There are three groups of attributes in the dataset collected from TCPdump tracing data.

- Content-based attributes: Attributes within a connection, such as the number of packets, acknowledgments, data bytes from *src* (source) to *dest* (destination), and intrinsic characteristics of data packets, etc.

- Time-based traffic attributes: This category of attributes includes the number of connections or services from the same source or to the same destination during a given time interval. This kind of attributes is sensitive to detect scanning activities.
- Connection based attributes: This category of attributes includes the number of connections from the same source or to the same destination or with the same service considering in last *N* connections. They are useful for detecting slow scanning activities.

3.3.2 Parameters Used in the Analysis

The dataset used in this chapter contains 311,029 records of connections. Each connection has 41 attributes. We randomly select the certain percentage of records of each class as the training dataset, and the rest of records are used as the testing data. The detail descriptions of the sample size used in the training and testing are given in Table 3.2. Note this dataset is specially designed to simulate computer network intrusion data, so most of the connections are attacks.

Table 3.2 Sample size of the training and testing dataset

	Overall entry	Individual				
		Normal	DoS	U2R	R2L	Probe
Total data size	311029	60593	4166	231455	70	14745
Training data size	1302	243	167	695	49	148
Percentage of training	0.42%	0.40%	4.01%	0.30%	70.00%	1.00%
Testing data size	309727	60350	3999	230760	21	14597

In the implementation of GA, the major parameters are used as follows:

- Population size: 20

- Number of generation limit: 50
- Probability of selection of the highest ranked individual: 0.6
- Probability of crossover q: 0.6
- Probability of mutation p_{m}: 0.05
- Generation limit used in the stop criteria if there is no change for the best individual: 10
- Fitness function: overall accuracy of the classifier with the selected attributes,

$$\mathrm{Accuracy}_{\mathrm{Training}} = \frac{1}{5}\sum_{i=1}^{5} \frac{\text{Number of connections correctly identified without } i\text{th folder}}{\text{Total number of connections}}$$

- Selection algorithm: normal geometric selection
- Crossover algorithm: simple crossover
- Mutation algorithm: binary mutation
- Classification accuracy at the test stage

$$\mathrm{Accuracy}_{\mathrm{Test}} = \frac{\text{Number of connections correctly identified}}{\text{Total number of connections}}$$

After 50 generations running of wrapper model, there is neither new population generated with the higher classification accuracy nor the decreased number of attributes with the same accuracy. Figure 3.3 shows the accuracy evolution for the attribute selection based on GA and multiclass SVM classifier.

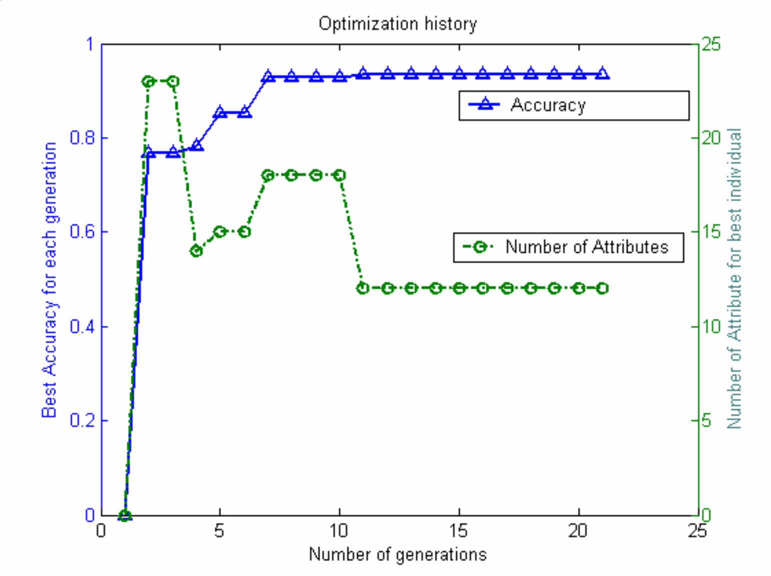


Figure 3.3 Accuracy evolution in the attribute selection

3.3.3 Analysis Results and Conclusions

There are total 41 attributes in the original dataset, where 13 are content-based attributes, 19 are time-based traffic attributes and 9 are connection-based attributes as show in Table 3.3. After Phase I attribute selection by a correlation-based filter model, there are 28 attributes are selected. After Phase II attribute selection by the wrapper model, there are only 12 attributes selected, in which 3 of them are connection-based attributes, 2 are content-based attribute, and 7 are time-based traffic attributes, which are summarized in Table 3.4. The specific features that are selected in each phase analysis are also marked as "x" in Table 3.3.

Table 3.3 Details of selected attributes in each phase

Category	Attribute	Type	Phase I	Phase II
	duration	Numeric		
	src_bytes	Numeric		
	dst_bytes	Numeric	X	X

Connection based attributes	land {1,0}	Numeric		
	wrong_fragment	Numeric		
	urgent	Numeric	X	X
	protocol_type {tcp,udp,icmp}	discrete	X	
	service {private,smtp,auth,ftp,domain	discrete	X	X
	flag {SF,SO,REJ,RSTR,S1}	discrete	X	
Content based attributes	hot	Numeric	X	
	num_failed_logins	Numeric	X	
	logged_in {0,1}	Binary	X	
	num_compromised	Numeric	X	
	root_shell	Numeric	X	X
	su_attempted	Numeric		
	num_root	Numeric	X	
	num_file_creations	Numeric	X	
	num_shells	Numeric	X	X
	num_access_files	Numeric	X	
	num_outbound_cmds	Numeric		
	is_host_login {1,0}	Binary	X	
	is_guest_login {1,0}	Binary		
Traffic based attributes	count	Numeric	X	
	srv_count	Numeric	X	X
	serror_rate	Numeric		
	srv_serror_rate	Numeric		
	rerror_rate	Numeric	X	
	srv_rerror_rate	Numeric	X	X
	same_srv_rate	Numeric	X	X
	diff_srv_rate	Numeric	X	X
	srv_diff_host_rate	Numeric	X	
	dst_host_count	Numeric	X	X
	dst_host_srv_count	Numeric		
	dst_host_same_srv_rate	Numeric		
	dst_host_diff_srv_rate	Numeric	X	X
	dst_host_same_src_port_rate	Numeric	X	X
	dst_host_srv_diff_host_rate	Numeric	X	
	dst_host_serror_rate	Numeric		
	dst_host_srv_serror_rate	Numeric		
	dst_host_rerror_rate	Numeric	X	
	dst_host_srv_rerror_rate	Numeric	X	

Table 3.4 Number of attributes in each stage

		Original dataset	After Phase I (Filter)	After Phase II (Wrapper)
Overall		41	28	12
Individual	Content-based attributes	13	10	2
	Time-based traffic attributes	19	13	7
	Connection-based attributes	9	5	3

Table 3.5 Comparison of classification and prediction accuracy

		Training dataset			Testing dataset	
		Use all 41 attributes	Use 28 attributes (Phase I)	Use 12 attributes (Phase II)	12 attributes	41 attributes
Overall accuracy		92.70%	92.74%	93.39%	92.08%	87.00%
Individual	Normal	87.65%	86.83%	84.77%	76.57%	68.01%
	DoS	85.03%	85.63%	96.41%	98.37%	99.52%
	U2R	95.11%	95.11%	95.54%	95.84%	94.25%
	R2L	87.76%	83.67%	77.55%	38.10%	14.29%
	Probe	100%	100%	99.32%	95.05%	47.66%

Table 3.5 shows the comparison of classification accuracy in each phase of attribute selection based on the training dataset and the prediction accuracy based on the testing dataset. It can be seen that when the training dataset is used, the classification accuracy using the final selected 12 attributes has a slight better overall performance (93.39% in column 4 and row 2) than using all 41 features (92.70% in column 2 and row 2). However, when the testing dataset is used, the prediction accuracy using the final selected attributes has a significant better overall performance (92.08% in column 5 and row 2) than using all 41 features (87.00% in column 6 and row 2).

3.4 DISCUSSION

It is worthwhile to point out that the dataset used in our analysis has a significant unbalance samples in each class of the dataset as shown in Figure 3.4 and Table 3.2. This is one of the major reasons that cause the poor performance in the identification of R2L class (only 70 samples available). In fact, the different percentages of the total samples are purposely used for each class in the selection of the training dataset, which is tried to reduce the unbalance of the training samples in our analysis.

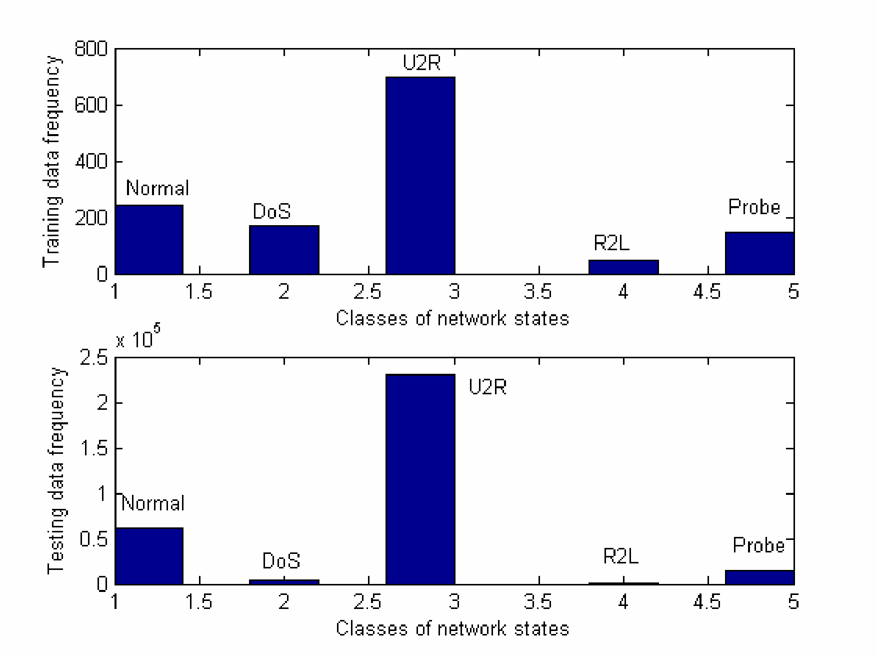
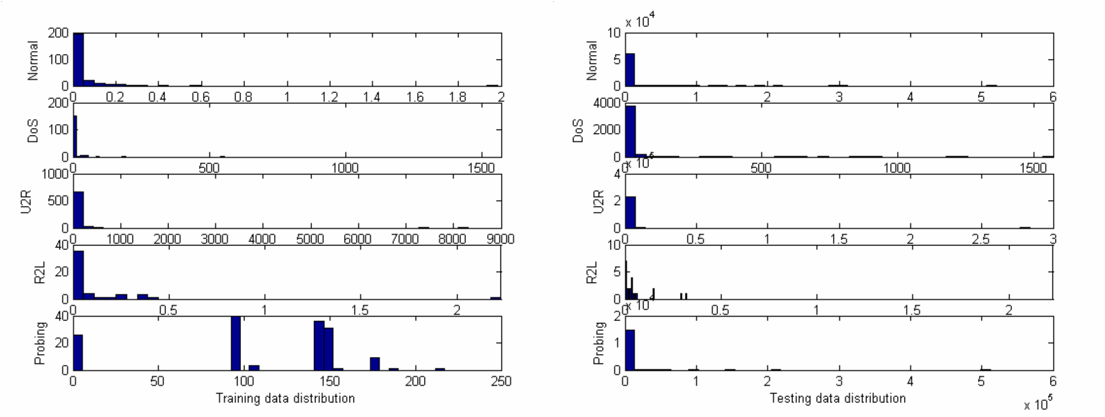


Figure 3.4 Frequencies of training data and testing data

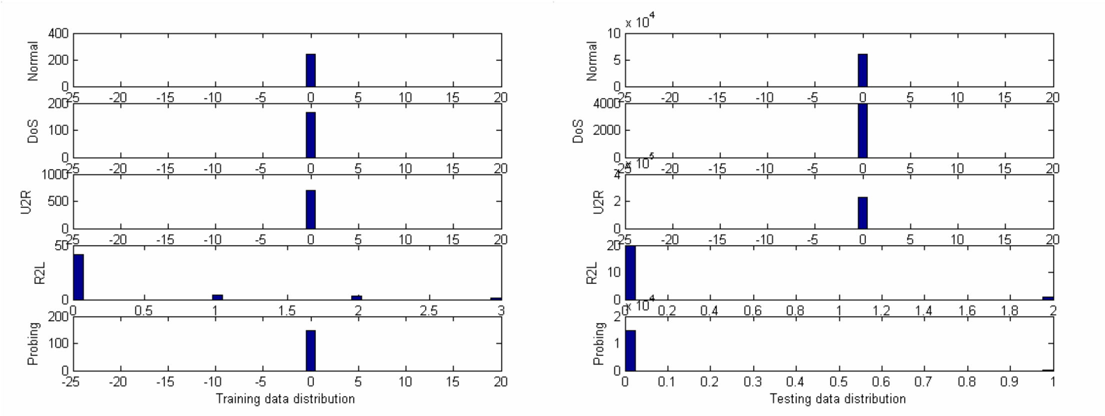
Another major reason causes some poor classification performance is due to the data heterogeneity between the testing dataset and the training dataset. Figure 3.5 shows the histogram of 12 selected attributes in the training and testing dataset (the left plot represents the training dataset, the right plot shows testing dataset) under each of 5 classes of network operation states.

From Figure 3.5 we can find some examples of different histograms between training and testing data, for example, R2L in (a), (b), (e) and (i), Normal connection in

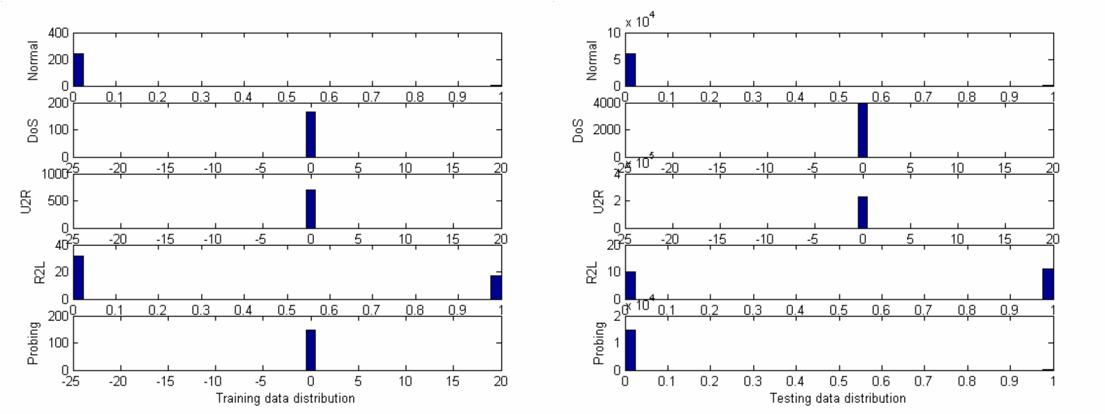
(a) and (f). Those differences account for the difference in detection performance for normal connection and R2L. Also we can find that the data in most attributes is difficult to find distribution, so it is difficult to investigate the computer network intrusion detection data using statistical ways.



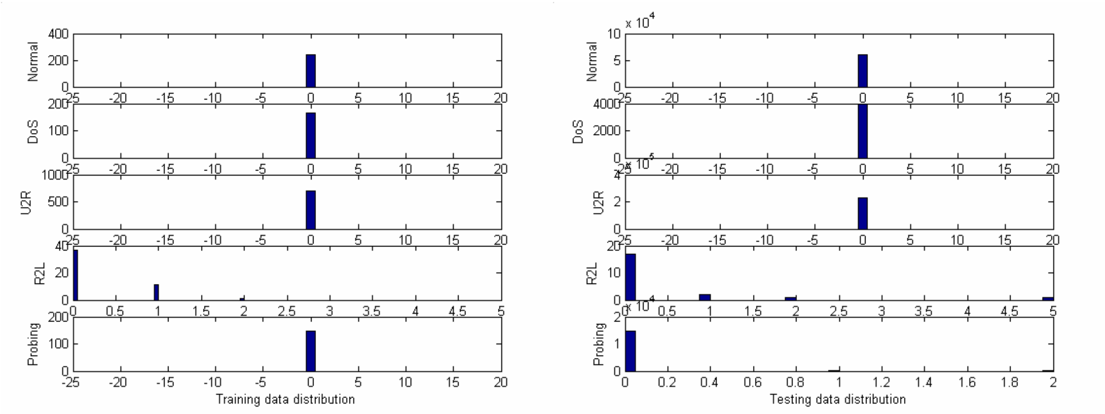
(a) Attribute 1: dst_bytes



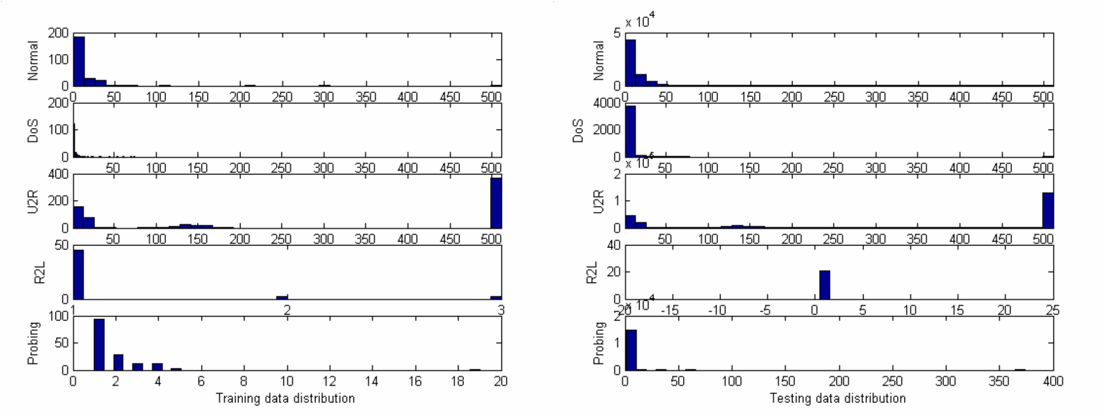
(b) Attribute 2: urgent



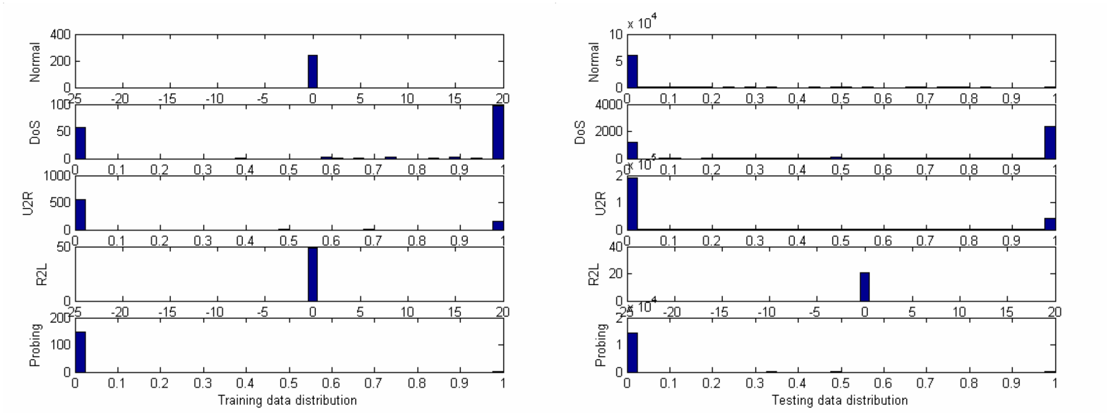
(c) Attribute 3: root_shell



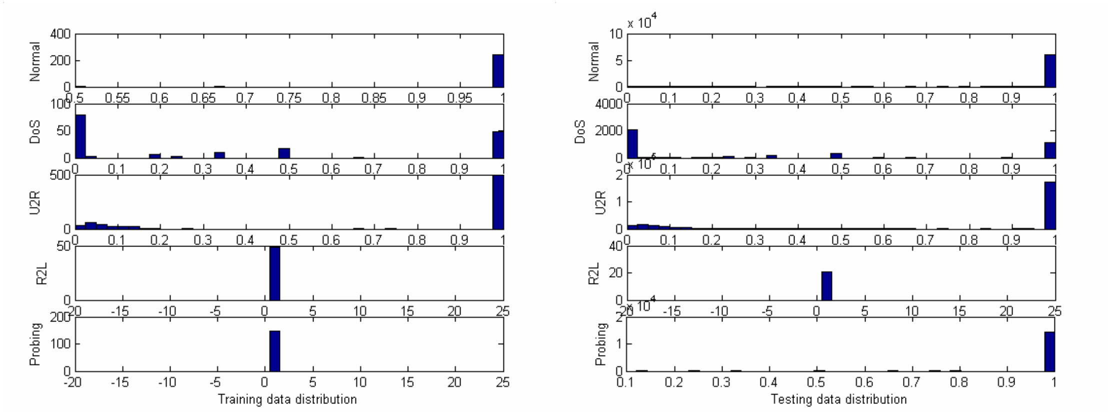
(d) Attribute 4: num_shells



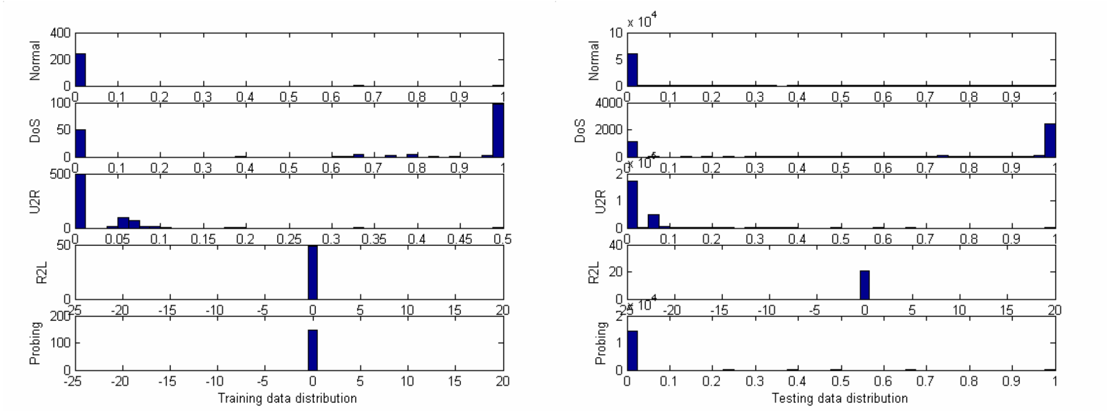
(e) Attribute 5: srv_count



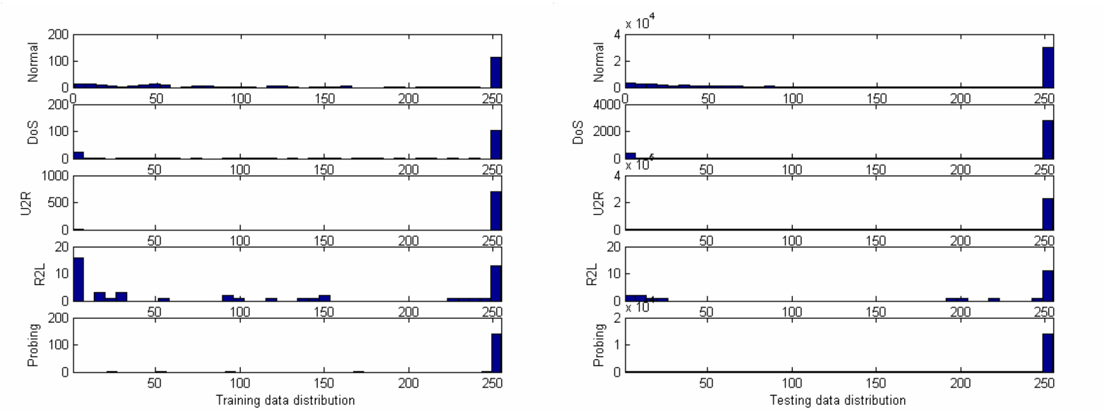
(f) Attribute 6: srv_rerror_rate



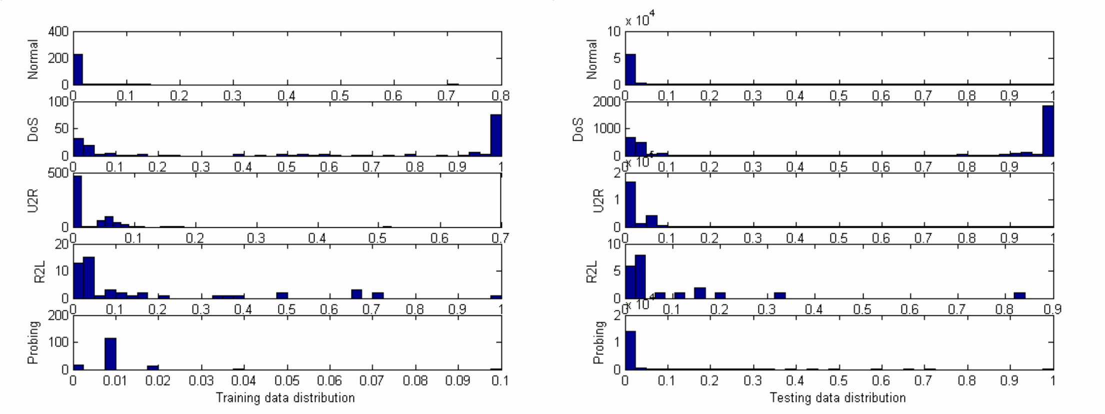
(g) Attribute 7: same_srv_rate



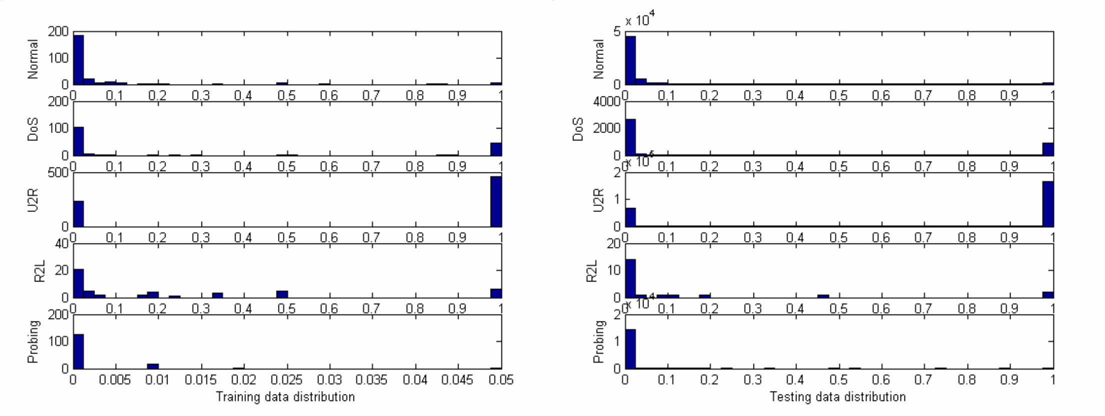
(h) Attribute 8: diff_srv_rate



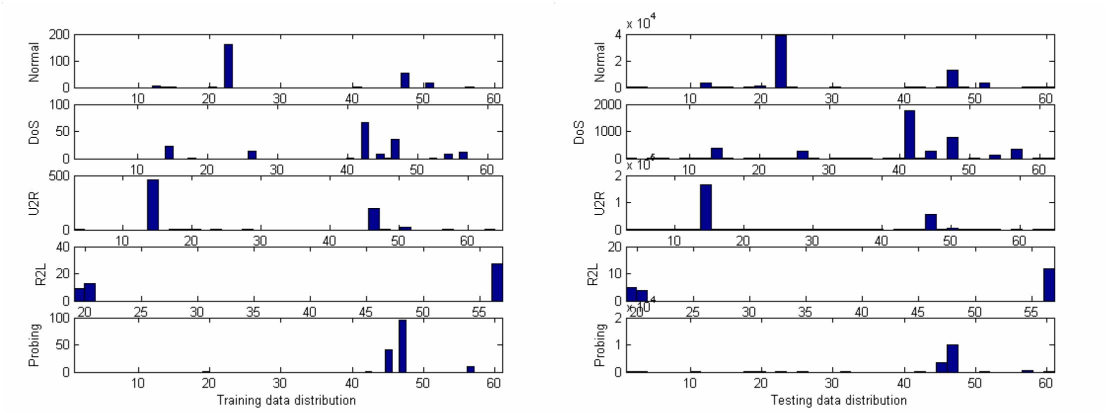
(i) Attribute 9: dst_host_count



(j) Attribute 10: dst_host_diff_srv_rate



(k) Attribute 11: dst_host_same_src_port_rate



(l) Attribute 12: service

Figure 3.5 Histogram of selected attributes in training data vs. testing data

CHAPTER 4 SVM BASED CONTROL CHART AND CASE STUDY FOR ANOMALY DETECTION

Traditional multivariate control charts, e.g. T^2 charts, have a limitation that the data has to be multivariate normal distribution. In this chapter, a systematic procedure is proposed for monitoring control chart design, which is an extension of existing multivariate control charts without the normal distribution assumption. For this purpose, the support vector machine (SVM) is used to extract monitoring features for non-Gaussian distribution data. In the framework on supervised kernel-based multivariate control chart, there are three important contributions in this chapter. The first contribution is the extension of SVM methods to construct multivariate control charts. The second contribution is the use of multi-objective Genetic Algorithm in parameter pre-selection for SVM based control chart. With the fine-tuning of pre-selected soft margin constant C and RBF kernel parameter σ we can obtain the best combination of Pareto optimal α and β errors. The third contribution is the performance evaluation of supervised kernel-based control chart by different scenarios.

This chapter is organized as follows: In Section 4.1, multivariate control charts are reviewed. Section 4.2 gives the framework of the proposed supervised multivariate control chart based on kernel methods. The performance improvement for the optimal selection of pre-selected parameters using multi-objective genetic algorithm is discussed in Section 4.3. The proposed control chart is applied to intrusion detection data in Section 4.4. Section 4.5 compares the performance of proposed control chart with other methods.

4.1 INTRODUCTION

It is very important to develop effective monitoring methods for online detection of network intrusions. Traditional multivariate control charts [13], such as Chi-square chart and Hotelling T^2 chart, have been well studied for monitoring of normally distributed data. Recently, some other multivariate control charts like multivariate EWMA chart [13] are not very sensitive to the normal distribution assumption. However, they are not very efficient to deal with high dimensional multivariate data, especially non-Gaussian distribution data.

4.1.1 Review of Multivariate Control Charts

Figure 4.1 shows a typical multivariate data scatter plot with 2 variables of x_1 and x_2 and dimension p=2. If we monitor each variable individually, there should be 2 univariate control limits as shown in the control rectangle range of Fig. 4.1. However, if these two variables are highly correlated, univariate charts will not be very effective for process monitoring because the correlation relationships between the variables are not well considered in the control limits. For example, if a process is in-control, the probability of p means in control is $(1 - \alpha)^p$ for the independent univariate case. But the joint probability of type I error may be much larger then $(1 - \alpha)^p$ if there is positive correlation between variables [76]. Therefore, a multivariate control chart is needed when the correlation among individual variables is significant.

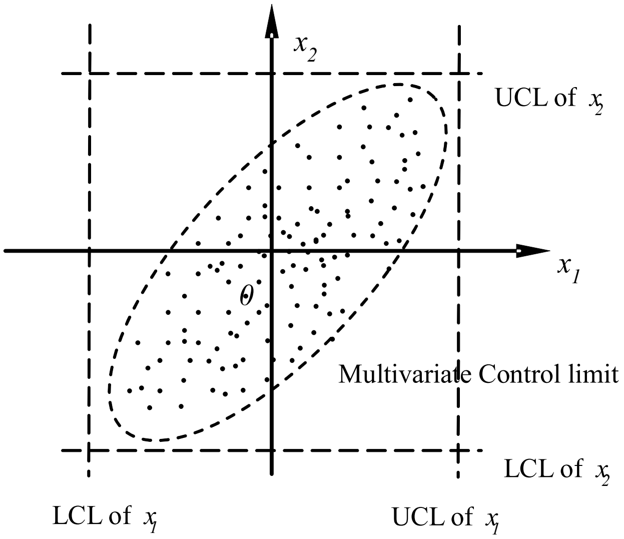


Figure 4.1 Multivariate quality control

For multivariate normally distributed data, one often uses Chi square control chart and Hotelling T^2 control chart for large mean shift., and uses MEWMA and MCUSUM charts for small mean shift. The main drawbacks for these control charts are: (1) they may not be very effective for extreme large dimensional data because the variance-covariance matrix may be poorly estimated for the large the dimensional data; (2) they are designed based on the normal distribution assumption. If the assumption is not satisfied, the misuse of those control charts will lead to misleading results.

4.1.2 Contribution of This Chapter

In recent years, non-parametric (distribution free) control charts have caught increasing attentions [77-79]. But most of such research is still limited in the univariate cases or the rank-based methods. This chapter is to propose a new method for developing non-parametric multivariate statistical control charts based on kernel methods, which can deal with non-Gaussian or unknown distribution data frequently encountered in real world problems.

There are two scenarios of designing a multivariate control chart for non-Gaussian distribution data. The first scenario is that we have the sample data under both in-control state and out-of-control states of all possible anomaly states. In this case, we normally suppose that we only care about these known anomaly states. Therefore, we can use supervised kernel methods to build the hyperplane to separate the in-control samples from faulty or anomaly samples, which is considered as the control limits with the maximum separation margin. For this purpose, Support Vector Machine (SVM) is used as the supervised classifier that is discussed in Chapter 4. The other scenario is that there are only in-control training samples but without the out-of-control training samples. In this case, the unsupervised kernel methods are used to build the control limits, which will be discussed in Chapter 5.

4.2 SUPERVISED MULTIVARIATE CONTROL CHART BASED ON KERNEL METHODS

When there are enough in-control samples and out-of-control samples, the design of control charts can be taken as the problem to obtain the boundary to separate the two categories of samples. In this chapter, a new SVM-based multivariate control chart will be developed, in which a multi-objective genetic algorithm will be integrated for selecting the adjustable parameters in the SVM.

4.2.1 Classification Based on SVM

Assume input-output training data pairs follow an independently identical distribution (i.i.d.) with an unknown probability function of $P(\mathrm{x}, y)$

$$\begin{aligned} &(x_1, x_2, ..., x_n) \in \mathbf{R}^p \\ &(y_1, y_2, ..., , y_n) \in \{-1, +1\} \end{aligned} \tag{4.1}$$

where p is the dimension of input data, n is number of samples. We need to estimate a function $f : \mathbf{R}^p \to \{-1, +1\}$. In the output set, Y=+1 represents in-control data, and –1 represents out-of-control ones.

Since two classes {-1,+1} are used to represent in-control and out-of-control states, the problem of multivariate control chart design becomes a 2-class classification problem. For constructing a linear separation boundary in the feature space, a feature transform mapping the input data space into a higher dimensional feature space using $\varphi(\cdot)$ needs to be defined first.

Given a training set of n data points as in (4.1), the support vector method approach is used to construct a classifier as:

$$y(x) = sign\left[w^T \varphi(x) + b \right] \tag{4.2}$$

where w are positive real constant vector and b is a real constant.

When the data of the two classes are separable, it yields

$$\begin{aligned} &w^T \varphi(x_j) + b \geq 1, \quad \text{if} \;\; y_j = +1 \\ &w^T \varphi(x_j) + b \leq -1, \quad \text{if} \;\; y_j = -1 \end{aligned} \tag{4.3}$$

These two sets of inequalities can be combined as

$$y_j [w^T \varphi(x_j) + b] \geq 1, \qquad j = 1, ..., n \tag{4.4}$$

This is called a hard margin formulation, which assumes these two classes can be perfectly separated by the optimal hyperplane without overlap.

$$y(x) = w^T \varphi(x) + b = 0 \tag{4.5}$$

In this case, the optimal hyperplane is constructed by solving the following quadratic programming:

$$\begin{aligned} &\min_{w \in \boldsymbol{H}} \quad \frac{1}{2} w^T w \\ &s.t. \quad y_j [w^T \varphi(x_j) + b] \geq 1, \quad j = 1,...,n \end{aligned} \tag{4.6}$$

In practice, the perfect separating hyperplane may not exist when there are some overlaps between two classes due to noise. In this case, the slack variables $\xi_j \geq 0$ are introduced (*j*=1,…, *n*) in order to relax the perfect separation constraints. This is called a soft margin optimization problem, which is defined in the primal weight space as

$$\begin{aligned} &\min_{w,\xi_k} \quad \frac{1}{2} w^T w + C \sum_{j=1}^{n} \xi_j \\ &s.t. \quad y_j [w^T \varphi(x_j) + b] \geq 1 - \xi_j, \quad j = 1,...,n \\ &\qquad \xi_j \geq 0, \quad j = 1,...,n \end{aligned} \tag{4.7}$$

C is the scalar to balance two objectives: one is to maximize margin, the other is to minimize the violation caused by overlap of data points. This constrained quadratic programming problem is solved by introducing Lagrange multipliers $\alpha_j \geq 0$, $\nu_j \geq 0$ (*j*=1,…, *n*)

$$L_1(w, b, \xi_j, \alpha_j, \nu_j) = l_1(w, \xi_j) - \sum\nolimits_{j=1}^{n} \alpha_j \{ y_j [w^T \varphi(x_j) + b] \text{-} 1 + \xi_j \} - \sum\nolimits_{j=1}^{n} \nu_j \xi_j \tag{4.8}$$

where $l_1(w, \xi_j) = \frac{1}{2} w^T w + C \sum_{j=1}^{n} \xi_j$

The saddle point of the Lagrangian gives the solution by computing

$$\min_{\alpha_j,v_j} \max_{w,b,\xi_j} L_1(w,b,\xi_j,\alpha_j,v_j) \tag{4.9}$$

By partial derivation it yields

$$\begin{aligned} &\tfrac{\partial L_1}{\partial w} = 0 \rightarrow w = \sum\nolimits_{j=1}^{n} \alpha_j y_j \varphi(x_j) \\ &\tfrac{\partial L_1}{\partial b} = 0 \rightarrow \sum\nolimits_{j=1}^{n} \alpha_j y_j = 0 \\ &\tfrac{\partial L_1}{\partial \xi_j} = 0 \rightarrow 0 \le \alpha_j \le C,\ j = 1,\dots,n \end{aligned} \tag{4.10}$$

which leads to the Dual problem by replacing (4.8) by using (4.10)

$$\begin{aligned} &\max_{\alpha_j} J_D(\alpha_j, \varphi(x_j)) = -\frac{1}{2}\sum_{j,l=1}^{n} y_j y_l \varphi(x_j)^T \varphi(x_l)\alpha_j \alpha_l + \sum_{j=1}^{n} \alpha_j \\ &s.t.\quad \sum_{j=1}^{n} \alpha_j y_j = 0, \quad 0 \le \alpha_j \le C,\ j = 1,\dots,n \end{aligned} \tag{4.11}$$

Compared with the hard margin formulation in the linear separable case, (4.11) has additional box constraints $0 \le \alpha_j \le C$.

Based on (4.11), the kernel is defined as $k(x,x_j) = \varphi(x)^T \varphi(x_j)$ (4.12)

The kernel $k(x,x_j)$ is any symmetric continuous function satisfying Mercer's condition [80], which states that any positive definite kernel can be expressed as a dot product in a high-dimensional space, and more specifically, if a kernel is positive semidefinite, there exists a function $\varphi(x)$ whose range is an inner product space of all possible high dimensions, such that $k(x,x_j) = \varphi(x)^T \varphi(x_j)$. By considering (4.10), (4.2) can be rewritten as

$$y(x) = sign\left[\sum_{j=1}^{n} \alpha_j y_j k(x,x_j) + b\right] \tag{4.13}$$

Normally we have the following kernels to pick up

Linear kernel: $k(x,x_j)=x_j^T x$

RBF kernel: $k(x,x_j)=\exp\{-\left\|x-x_j\right\|^2/(2\sigma^2)\}$

By using kernel trick [15]: any positive semi-definite kernel can be expressed as a dot product in a high-dimensional space, the classifier (4.13) can be obtained by solving

$$\begin{aligned}&\max_{\alpha_j} J_D(\alpha_j, k(x_j,x_l)) = -\frac{1}{2}\sum_{j,l=1}^{n} y_j y_l k(x_j,x_l)\alpha_j\alpha_l + \sum_{j=1}^{n}\alpha_j \\ &s.t. \quad \sum_{j=1}^{n}\alpha_j y_j = 0, \quad 0 \le \alpha_j \le C,\ j=1,...,n\end{aligned} \tag{4.14}$$

The solution to this problem is unique and global for linear SVM case as well as positive definite kernel [80]. For a positive semidefinite kernel the solution is global but not necessarily unique. In the solution of (4.14), only some α_j are not equal to zero, those data are on the boundary of the whole data called support vectors, the corresponding non-zero α_j are called support vectors. The solution for *b* is

$$b=\frac{1}{n_{NBSV}}\sum_{i=1}^{n_{NBSV}}\{y_i-\sum_{j=1}^{n_{SV}}\alpha_j y_j k(x_i,x_j)\} \tag{4.15}$$

where x_l is an example which is non-bound support vector (i.e. $0<\alpha_j<C$), n_{SV} is the number of support vectors, n_{NBSV} is the number of non-bound support vector. It can be found that only non-bound support vector (i.e. $0<\alpha_j<C$) plays the role of $w^T\varphi(x_j)+b=1$ or -1 (margin with $\xi_j=0$), other support vectors with $\alpha_j=C$ are overlap samples with $\xi_j>0$ to be compensated by soft margin..

4.2.2 Multivariate Control Chart Using SVM

In this section we design a control chart based on the classifier hyperplane constructed in binary SVM. After finding a unique globally optimal hyperplane by using a positive definite kernel function, the hyper-parameters in the model (4.14) are all fixed. When there is a new observation coming, we use the following equation to obtain the decision state for new observations.

$$y(x) = \sum_{j=1}^{n} \alpha_j y_j k(x, x_j) + b \tag{4.16}$$

The control limit of this control chart is zero, a data falls above zero is in-control, and below zero is out-of-control. Because decision function value $y(x)$ represents the distance of the sample from hyperplane in feature space, we can use this distance to evaluate how close this new observation is toward the control limit. The parameters in the decision function are obtained as following:

Lagrange multiplier α_j, i.e. the variable in the dual problem (4.14) is obtained by solving this dual problem from the training labeled samples. Kernel $k(x, x_j)$ is pre-selected from linear, Polynomial, or RBF kernels. Bias b is obtained by (4.15).

4.3 OPTIMAL SELECTION OF KERNEL PARAMETERS USING GENETIC ALGORITHMS FOR SVM-BASED CONTROL CHART DESIGN

Optimal selection of kernel function is a critical issue in SVM classification problems, which has been extensively studied [81]. This section aims to study how to adjust kernel parameters for the best performance of the control chart under a given kernel function.

From previous section, it can be found that the decision boundary of kernel methods are determined by solving a quadratic programming (QP), but inside the formulation of QP, we need to select optional parameters such as soft margin constant C and RBF kernel parameter σ. In this section we first summarize the existing methods for this purpose and give some comments on them, and then propose a new method which uses multi-objective Pareto genetic algorithm to obtain a set of optimal combination for the two objectives we want to minimize: α and β errors.

4.3.1 Review of Existing Methods

(1) Trial and error with cross-validation

Despite its simplicity, the trial and error method is the most widely used method. Normally one can pick up several values of a parameter by trial and error and compare the effect of them then fix one as the final choice.

Cross-validation is a typically re-sampling method that is widely used in machine learning algorithms. Cross validation is more preferred than residuals method as a model evaluation method because it can predict the performance of the model on new data and can overcome the problem of outliers. The basic idea of cross-validation is to remove part of the data before the training of model, and use the removed data as validation data to evaluate the performance of the model. There are mainly three methods included in cross-validation method, named holdout method, K-fold cross validation and Leave-one-out cross validation. Among them K-fold cross validation is the most widely used one.

Normally K equals to 5 or 10. Details for this method can be found in [82]. Because of the advantage of obtaining stable solution, cross-validation is often connected with search methods.

(2) Grid search with cross-validation

The basic idea of grid search is to divide the search space (e.g. two parameters to be decided) into uniformly distributed grids, and evaluate the performance on each point, finally pick up the combination with best performance. This method is quite computationally extensive. Definitely, if the grid resolution is large enough, we can find the optimal value of the parameters after all the grids result are obtained. But the cost of the computational is usually unaffordable to reach this end.

4.3.2 Multi-objective Optimization For Parameters Fine-tuning

All the methods above have the common limitation: there is no heuristic inside the algorithm and so they are both blind search methods. The number of possible values for the parameters must be small enough otherwise the searching time would be very long. Even though, the optimal parameters by those searches may not be the true optimal values because it depends on how close the best results based on the selected grids to the true optimal results. Another problem of the existing methods is that those methods are not suitable for the control chart design application, because we have 2 objectives to consider: we want to minimize: α and β errors at the same time.

As we all know, α and β errors are calculated according to the corresponding class the sample belonging to:

$$\begin{aligned}\alpha &= \Pr(x \text{ is measured as out of control}|x \text{ is actually in control})\\ &= \frac{\text{number of sample(measurement is out|true value is in)}}{\text{In control sample size}}\end{aligned} \tag{4.17}$$

$$\begin{aligned}\beta &= \Pr(x \text{ is measured as in control}|x \text{ is actually out of control})\\ &= \frac{\text{number of sample(measurement is in|true value is out)}}{\text{Out of control sample size}}\end{aligned} \tag{4.18}$$

While the total error for classification is

$$\begin{aligned}\text{Total error rate} &= \Pr(x \text{ is measured as different value as } x \text{ actually is})\\ &= \frac{\text{number of sample(measurement is different from true value)}}{\text{Total sample size}}\end{aligned} \tag{4.19}$$

Take a close look at the three formula (4.17) to (4.19), we can find that $\alpha+\beta>\text{Total error rate}$. The reason is that α and β errors are constructed on their corresponding classes either in control or out of control, but total error rate is constructed on total sample size.

We also provide with the explanation of the fact that $\alpha+\beta>\text{Total error rate}$. Let U$_1$ = {all in-control sample}, U$_2$ = {all out of control sample}, it is easy to know that $U_1 \cup U_2 = \Omega$ and $U_1 \cap U_2 = \Phi$. So the probability of detection error is

$$\begin{aligned}\Pr(Error) &= \Pr(Error\,|\,U_1 \cup Error\,|\,U_2)\\ &= \Pr(Error\,|\,U_1)\Pr(U_1) + \Pr(Error\,|\,U_2)\Pr(U_2)\\ &= \alpha\Pr(U_1) + \beta\Pr(U_2)\end{aligned} \tag{4.20}$$

where $\Pr(U_1) = \dfrac{\text{Number of In Control samples}}{\text{Total sample size}}$

and $\Pr(U_2) = \dfrac{\text{Number of Out of Control samples}}{\text{Total sample size}}$

Because $0 \le \Pr(U_1) \le 1$ and $0 \le \Pr(U_2) \le 1$, it is easy to see $\alpha+\beta>\text{Total error rate}$.

In this section a multi-objective optimization method for fine-tuning parameters is presented. The basic idea is to take use of multi-objective genetic algorithm by its stochastic searching and guided searching power ability and dealing with multi-objectives at the same time to deal with fine-tuning problem of SVM parameter optimization.

The typical problem for multi-objective optimization is as following

$$\begin{array}{rll} \text{Minimize/Maximize} & f_m(x), & m = 1,2,...,M; \\ \text{subject to} & g_j(x) \geq 0, & j = 1,2,...,J; \\ & h_k(x) = 0, & k = 1,2,...,K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1,2,...,n; \end{array} \tag{4.21}$$

where $f_m(x)$: objective functions. Assume there are M objective functions to be mimimized or maximized at the same time.

$g_j(x)$: The constraints with values larger than or equal to zero.

$h_k(x)$: The constraints with value equal to zero.

x_i: decision variables..

There are two goals in a multi-objective optimization (MOO): to find a set as close as possible to the Pareto-optimal front, and to find a set of solutions as diverse as possible. In the field of multi-objective optimization it is usually assumed that optimization takes places before decision making. So the goal is to find or approximate the Pareto-optimal set.

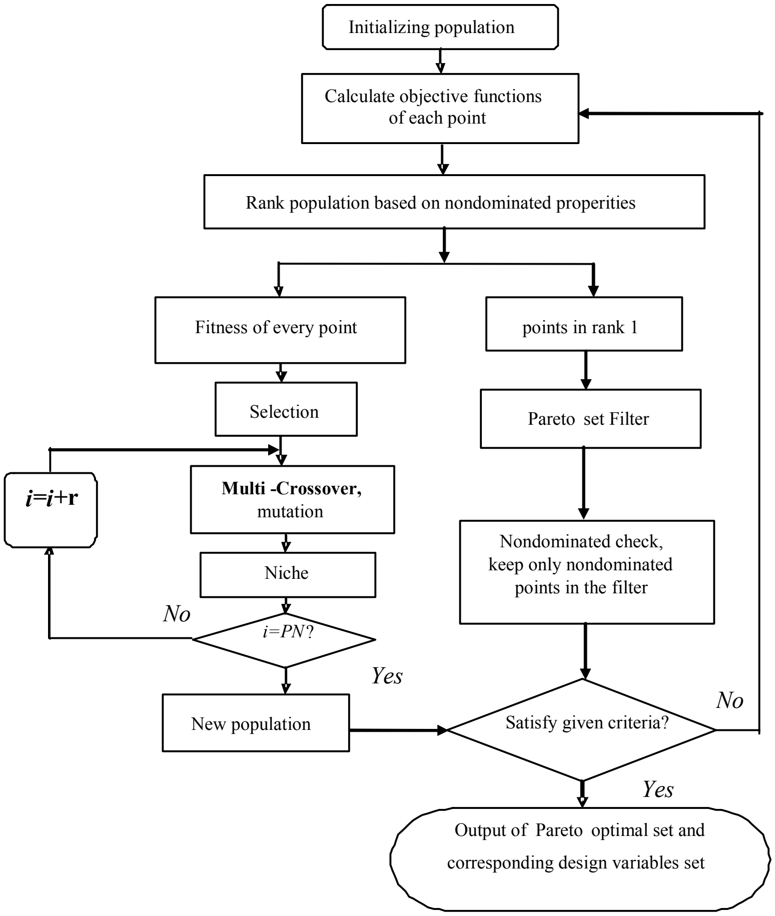

Initializing population
Calculate objective functions of each point
Rank population based on nondominated properities
Fitness of every point
points in rank 1
Selection
Pareto set Filter
i=i+r
Multi -Crossover, mutation
Niche
Nondominated check, keep only nondominated points in the filter
No
i=PN?
Yes
New population
Satisfy given criteria?
No
Yes
Output of Pareto optimal set and corresponding design variables set

Figure 4.2 Framework of multi-objective Pareto Genetic Algorithm

Figure 4.2 is the flowchart of the algorithm presented in this chapter. In original method two individuals are chose for crossover and mutation. PN/2 times (PN is the size of population) operations for crossover, mutation and ranking are needed in every evolution, which makes traditional multi-objective genetic algorithm very time-consuming. We implement a multiple chromosomes crossover to speed up while remain the convergence of the algorithm, the process needs only PN/*r* times operations, where *r* is the number of individuals which take part in crossover every time. In theory calculating time will be decreased to *r*/2 of original time when using this way. Generally, the more the number of individuals which take part in crossover, the faster the multi-objective genetic algorithm calculates. Details can be found in [83]. Also Appendix B has details of each step in the flowchart as in Figure 4.2.

In this section we use the two pre-defined parameters in SVM: C value and σ as design variables, and use α and β errors as the two objectives in the multi-objective optimization. The settings used in this section are:

Kernel: RBF

C value: 0.1 to 100

σ: 0.01 to 1

Population size: 20

Pareto optimal set size: 30

Maximum number of iteration: 50

Crossover probability q: 0.2

Mutation probability p_m: 0.2

After 50 iterations of the revolution, we obtain the optimal Pareto objective values of α vs. $1-\beta$ errors as shown in Figure 4.3 and corresponding optimal Pareto set of design variables shown in Figure 4.4.

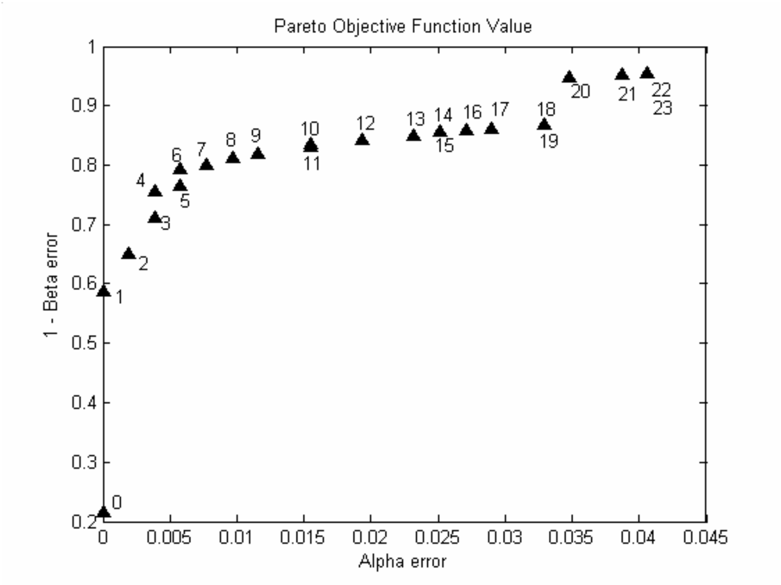


Figure 4.3 Optimal Pareto objective values

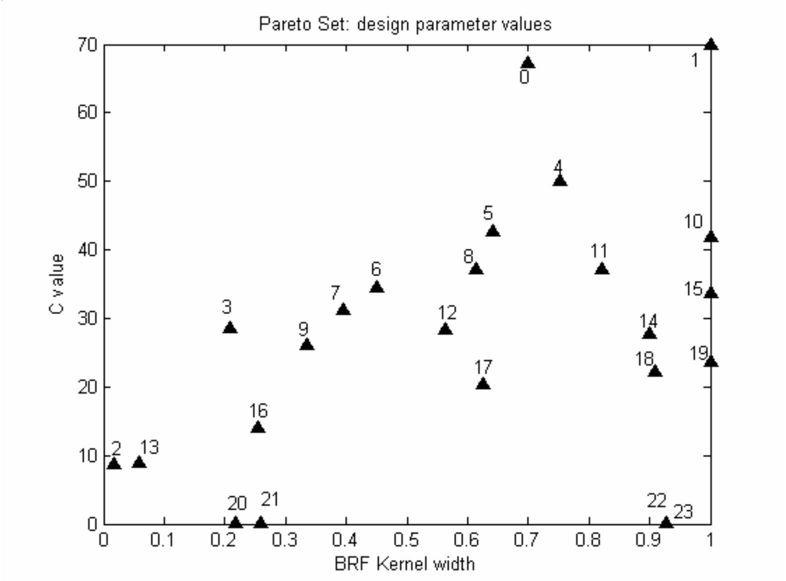


Figure 4.4 Optimal Pareto set

According to Figure 4.5, first we can find the two objectives: α and β errors are exclusive to decrease, i.e., if we want to decrease α error, we have to pay the price of increasing β error. Also we can find that neither the C value nor kernel width σ can cause α or β error to change monotonously. The reason of this fact is that SVM control chart is data dependent. Different combinations of C value and kernel width σ may produce different seperating hyperplanes and different support vectors, but they may have same α and β errors. Another finding from Figure 4.5 is that the optimal Pareto frontier is not a smoothly changed curve. This fact tells that it is not easy to find theoretic Pareto frontier for this type of multi-objective optimization problem. It is necessary to use simulation to estimate the optimal Pareto frontier so that we can pick up the most satisfied combination for our needs. So when we use SVM control chart, we need to use multi-objective Genetic Algorithm to fine tune the pre-defined parameters: C value and kernel width σ to reach the highest performance of SVM control chart.

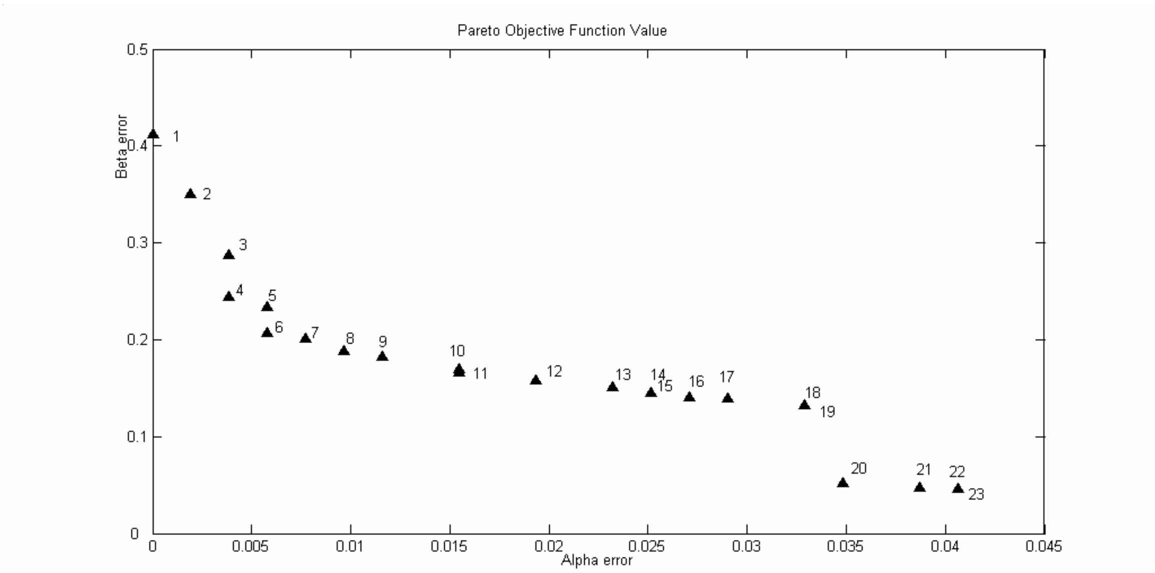


Figure 4.5 Relationship between α , β errors and C, σ values

We can also find the relationship between α , β errors and C, σ values in Table 4.1. It is clear from the table that Pareto α , β errors are conflict objectives, also C and σ

values have no intrinsic relations. It means we could not obtain the optimal Pareto by simply increase either C or σ value, but need to use multi-objective genetic algorithm (MOGA) to find the Pareto optimal set.

Table 4.1 The table for relationship between α , β errors and C, σ values

Index	Kernel width σ	C value	α error	β error
1	1	69.986	0	0.41201
2	0.017434	8.6273	0.001934	0.3499
3	0.20846	28.625	0.003869	0.28778
4	0.75315	50.032	0.003869	0.24431
5	0.64129	42.707	0.005803	0.23395
6	0.4495	34.523	0.005803	0.20704
7	0.39529	31.238	0.007737	0.20083
8	0.6146	37.089	0.009671	0.18841
9	0.33462	26.186	0.011605	0.18219
10	1	41.853	0.015474	0.16977
11	0.82081	37.089	0.015474	0.16563
12	0.56298	28.361	0.019342	0.15735
13	0.058476	8.8233	0.023211	0.15114
14	0.90072	27.794	0.025145	0.14493
15	1	33.75	0.025145	0.14493
16	0.25509	14.071	0.027079	0.14079
17	0.62495	20.307	0.029014	0.13872
18	0.9081	22.204	0.032882	0.13251
19	1	23.61	0.032882	0.13251
20	0.21677	0.1	0.034816	0.05176
21	0.25913	0.1	0.038685	0.047619
22	0.92783	0.1	0.040619	0.045549
23	0.92688	0.1	0.040619	0.045549

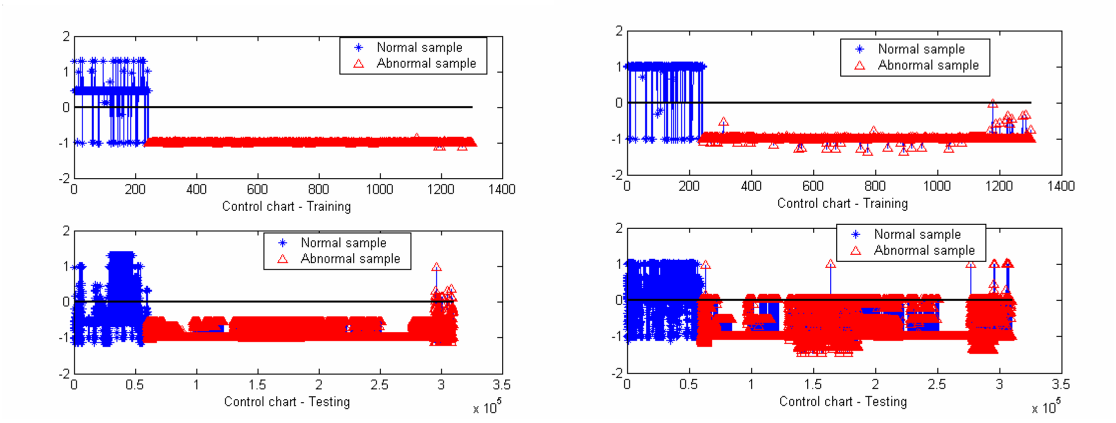
4.4 APPLICATION ON ANOMALY DETECTION

In this section the SVM based control chart is used for computer network anomaly detection. Similar to Chapter 3, we use KDD1999 labeled data for SVM based control chart. We simply put normal connection as normal data and all the others attacks as abnormal data.

One result coming from the tables and figures is that the 12 attributes has better performance than the original 41 attributes not only on the accuracy, α error and β error, but also on the fact that 12-attribute data has much less computing time on both training and testing. In the real-world intrusion detection application, short computing time is highly desired.

Table 4.2 Normal vs. all intrusion

Normal vs. All Intrusion performance		41 attributes	12 attributes
Training	Training accuracy	98.31%	98.08%
	α error	9.0535%	10.288%
	β error	0	0
	Model SV number	884	641
	Margin	0.0584	0.051
	CPU time (Sec.)	42.0460	84.1400
Testing	Testing accuracy	81.997%	94.982%
	α error	92.328%	12.717%
	β error	0.01%	3.1543%
	CPU time (Sec.)	279.3910	102.4530



(a) Control chart on 41 attributes (b) Control chart on 12 attributes

Figure 4.6 Normal vs. all intrusions

Table 4.2 shows the performance of SVM based control chart on normal connection versus all the abnormal connections (intrusions). With selected 12 attributes, the testing accuracy increases a lot (from about 82% to 95% and significant decrease in α error) while the testing time is reduced from 279 seconds to 102 seconds.

We also use part of the intrusion data to illustrate the use of SVM control chart. 1000 samples are selected from original dataset, among them 700 is from normal connection, 100 each for DoS, U2R and Probe.

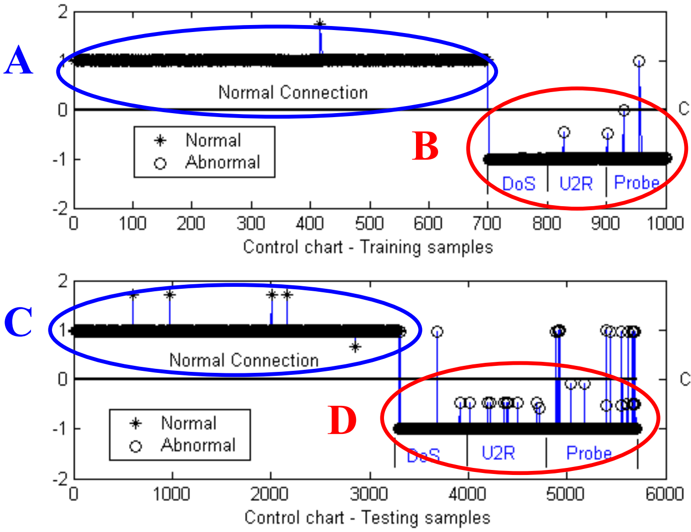


Figure 4.7 Control chart for selected samples on intrusion detection data

If we have a detailed look at the training and testing data on the kernel distance, we can find that the distribution of each part in Figure 4.7: A, C represent normal samples in training and testing data, B, D represent abnormal samples in training and testing data individually.

Figure 4.8 shows the kernel distance and probabilistic fitting for data in area A and B. Based on the fact that the kernel distance is not an exact way to explain the result that one new sample is in-control or out-of-control, we propose a probabilistic way to explain the result: a logistic curve fitting is used to estimate the probability of the sample to be in the state (in-control or out-of-control), then we can compare with the two probabilities to see which one is more suitable to describe the state of that sample.

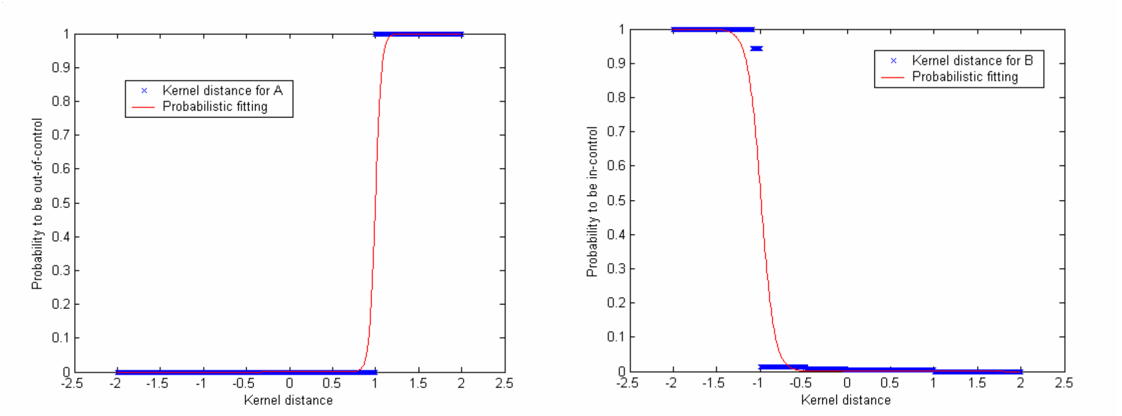


Figure 4.8 kernel distance and probabilistic fitting for data in area A and B

In detail, a hypothesis

H_0: x_0 is in control

H_1: x_0 is out of control

is used for the new sample x_0. Given a new sample x_0, according to the probabilistic curve in Figure 4.9, we can have the probabilities:

$P_0(x_0 \in H_0 \mid x_0) =$ logistic function value of probability to be in-control

$P_1(x_0 \in H_1 \mid x_0) =$ logistic function value of probability to be out-of-control

Then we compare $P_0(x_0 \in H_0 \mid x_0)$ and $P_1(x_0 \in H_1 \mid x_0)$. If $P_1(x_0 \in H_1 \mid x_0) \geq P_1(x_0 \in H_1 \mid x_0)$, then we make decision that x_0 is in control, otherwise x_0 is out of control.

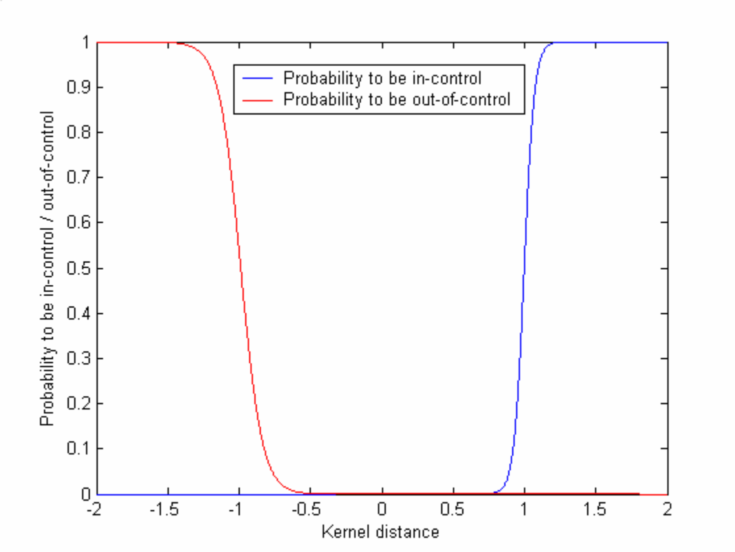


Figure 4.9 Probabilistic curve for the final state of a new sample

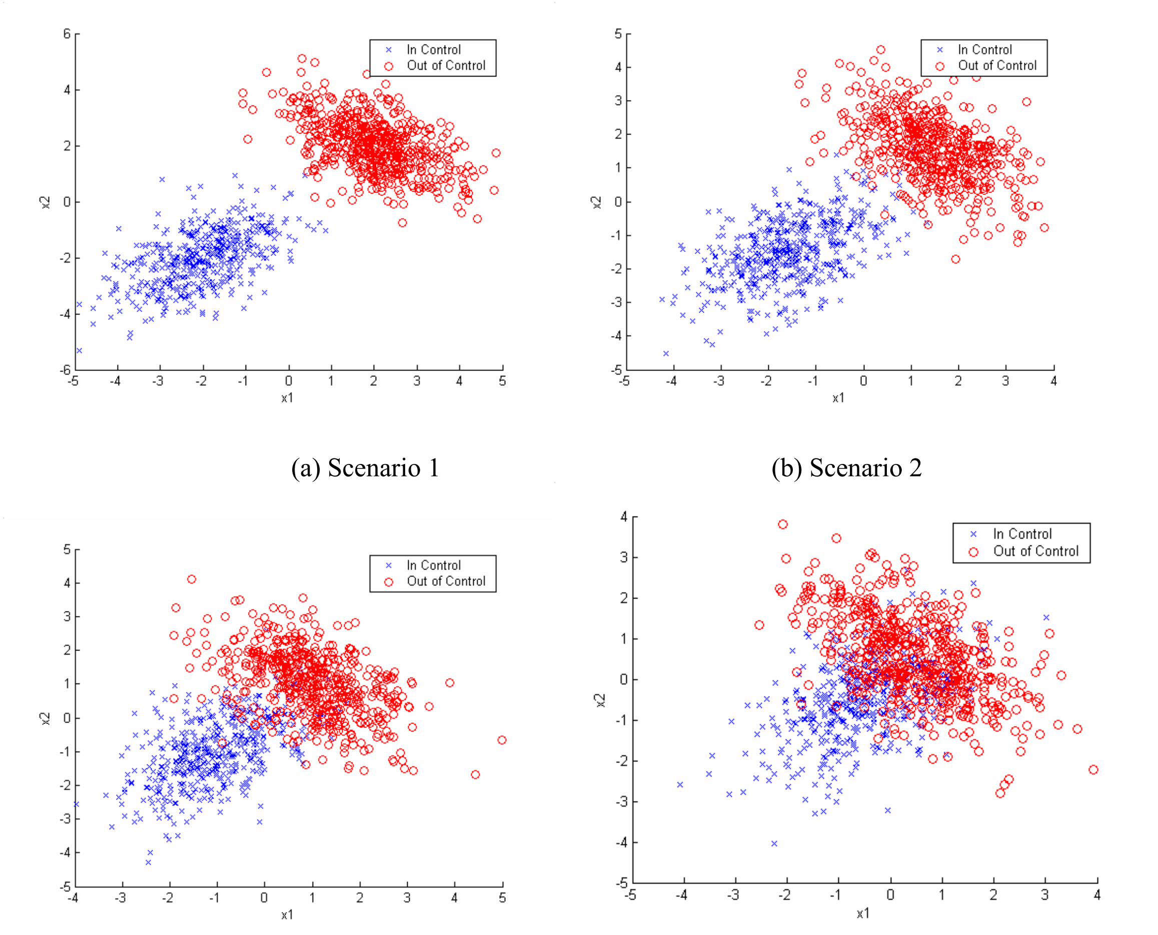
4.5 COMPARISON OF SVM CONTROL CHART WITH OTHER METHODS

4.5.1 Comparison on Known Distribution Data

We still use the two-dimensional data sets shown in Figure 4.10 with 4 scenarios. One of the two clusters is in-control data, the other is out-of-control data. Both the two clusters are in normal distribution. Details can be found in Table 4.3.

Table 4.3 The 4 scenarios of data (sample size: 1000, 500 for in-control and 500 for out-of-control data)

	In-control data		Out-of-control data	
	mean	Covariance matrix	mean	Covariance matrix
Scenario 1	(-2, -2)	$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	(2, 2)	$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$
Scenario 2	(-1.5, -1.5)	$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	(1.5, 1.5)	$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$
Scenario 3	(-1, -1)	$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	(1, 1)	$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$
Scenario 4	(-0.5, -0.5)	$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	(0.5, 0.5)	$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$



(a) Scenario 1

(b) Scenario 2

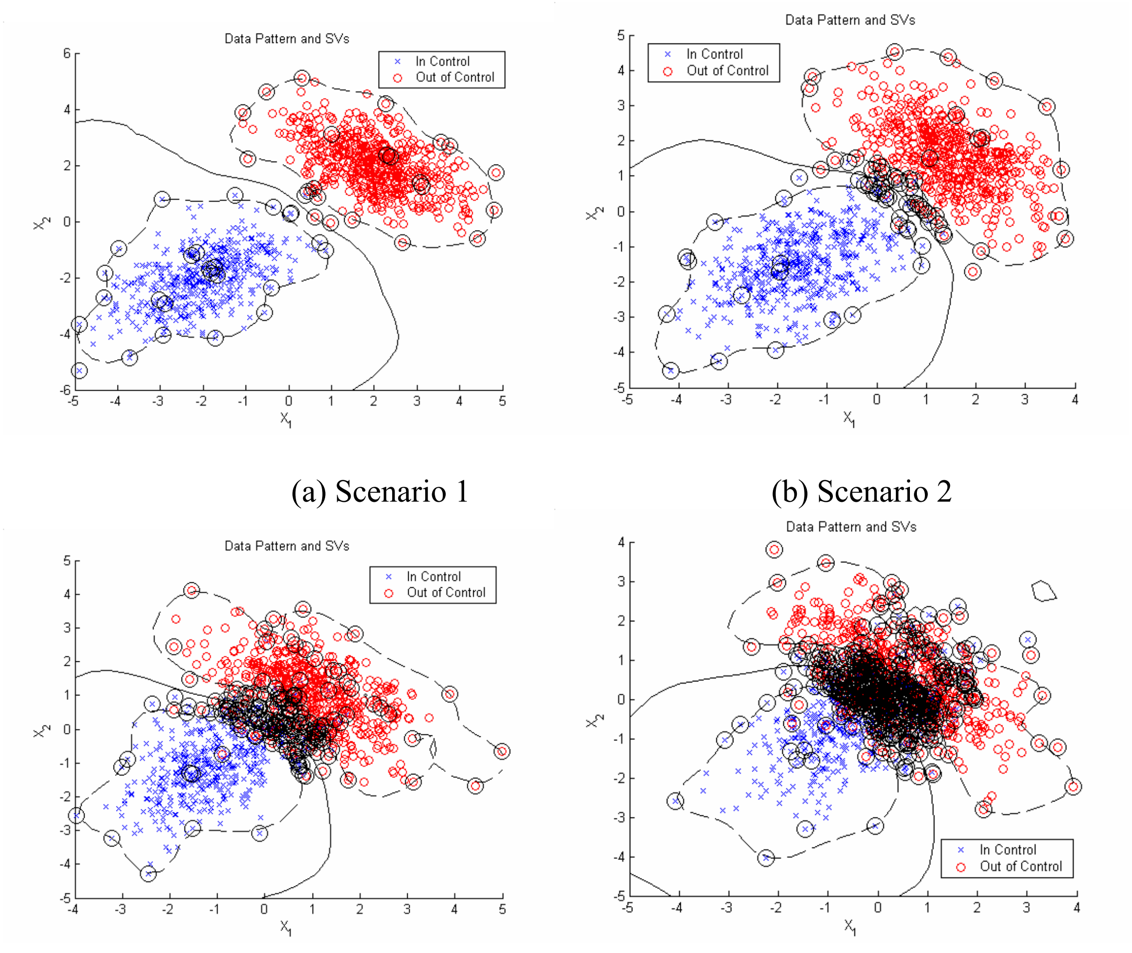
(c) Scenario 3

(d) Scenario 4

Figure 4.10 The 4 scenarios for performance comparison

4.5.1.1 SVM based control chart on normal distribution data

SVM based control chart does not require data to be in normal distribution, it does not mean that that performance of SVM based control chart is not as good as traditional control chart based on the known distribution of data. In this section we will demonstrate SVM based control chart used for normal distribution data and compare with T^2 chart.



(a) Scenario 1 (b) Scenario 2

(c) Scenario 3 (d) Scenario 4

Figure 4.11 SVM based separating lines for in-control and out-of-control data

Figure 4.11 shows the separating lines as the solid lines for the 4 scenarios. The dotted lines represent +1 or -1 of the decision function value. They are used to show the margin. SVM based control chart use both in-control data and out-of-control data to construct the optimal separating line (hyperplane) to obtain largest margin as well as minimal penalty of the overlap. The performance of SVM based control chart on the four scenarios is shown in Table 4.4. Note that in this section RBF kernel is used (parameters: C=$1, \sigma = 1$).

Table 4.4 Performance of SVM based control chart on the four scenarios

	α error	β error	Total accuracy
Scenario 1	0	0.0021	0.999
Scenario 2	0.0082	0.0136	0.989
Scenario 3	0.0542	0.0725	0.937
Scenario 4	0.1134	0.2792	0.81

In Table 4.4, α error is corresponding to Phase 1 in T^2 chart using in-control data, and β error is corresponding to Phase 2 in T^2 chart using out-of-control data.

4.5.1.2 T^2 chart on normal distribution data

Given data following a multivariate normal distribution, the sample mean vector is

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.22}$$

Also, the sample covariance matrix is

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})' \tag{4.23}$$

Then T^2 statistic is [13]

$$T^2 = n(\mathbf{x} - \bar{\mathbf{x}})'\mathbf{S}^{-1}(\mathbf{x} - \bar{\mathbf{x}}) \tag{4.24}$$

The design of T^2 chart is divided into 2 phases. Phase 1 obtains an in-control set of observations so that control limits can be established for phase 2, and phase 2 is used for monitoring future process data.

Phase 1 control limits are [13]:

$$\begin{aligned} UCL &= \frac{(m-1)^2}{m}\beta_{\alpha,p/2,(m-p-1)/2} \\ LCL &= 0 \end{aligned} \tag{4.25}$$

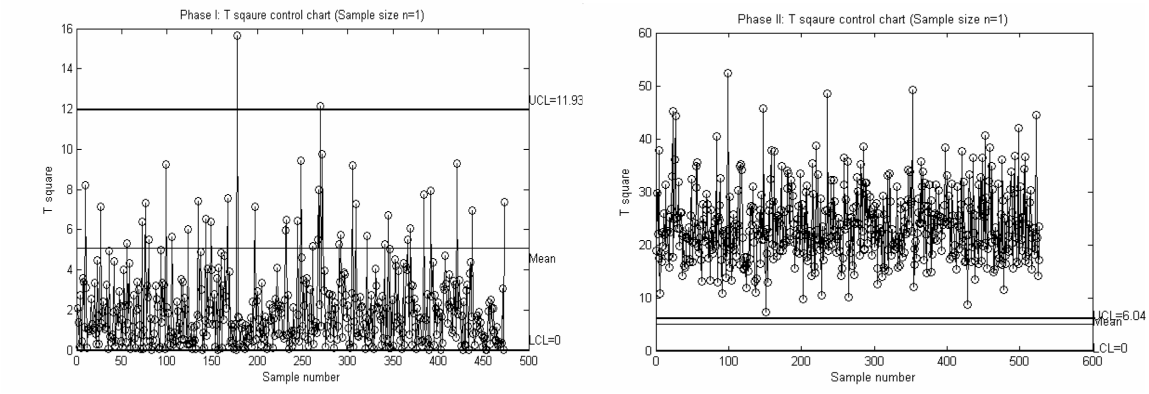
where m is the number of preliminary samples, p is the number of quality characteristis, $\beta_{\alpha,p/2,(m-p-1)/2}$ is the upper α percentage point of a beta distribution with parameters $p/2$ and $(m-p-1)/2$. Here we set α =0.05.

Phase 2 control limits are:

$$\begin{aligned} UCL &= \frac{p(m+1)(m-1)}{m^2-mp} F_{\alpha,p,m-p} \\ LCL &= 0 \end{aligned} \tag{4.26}$$

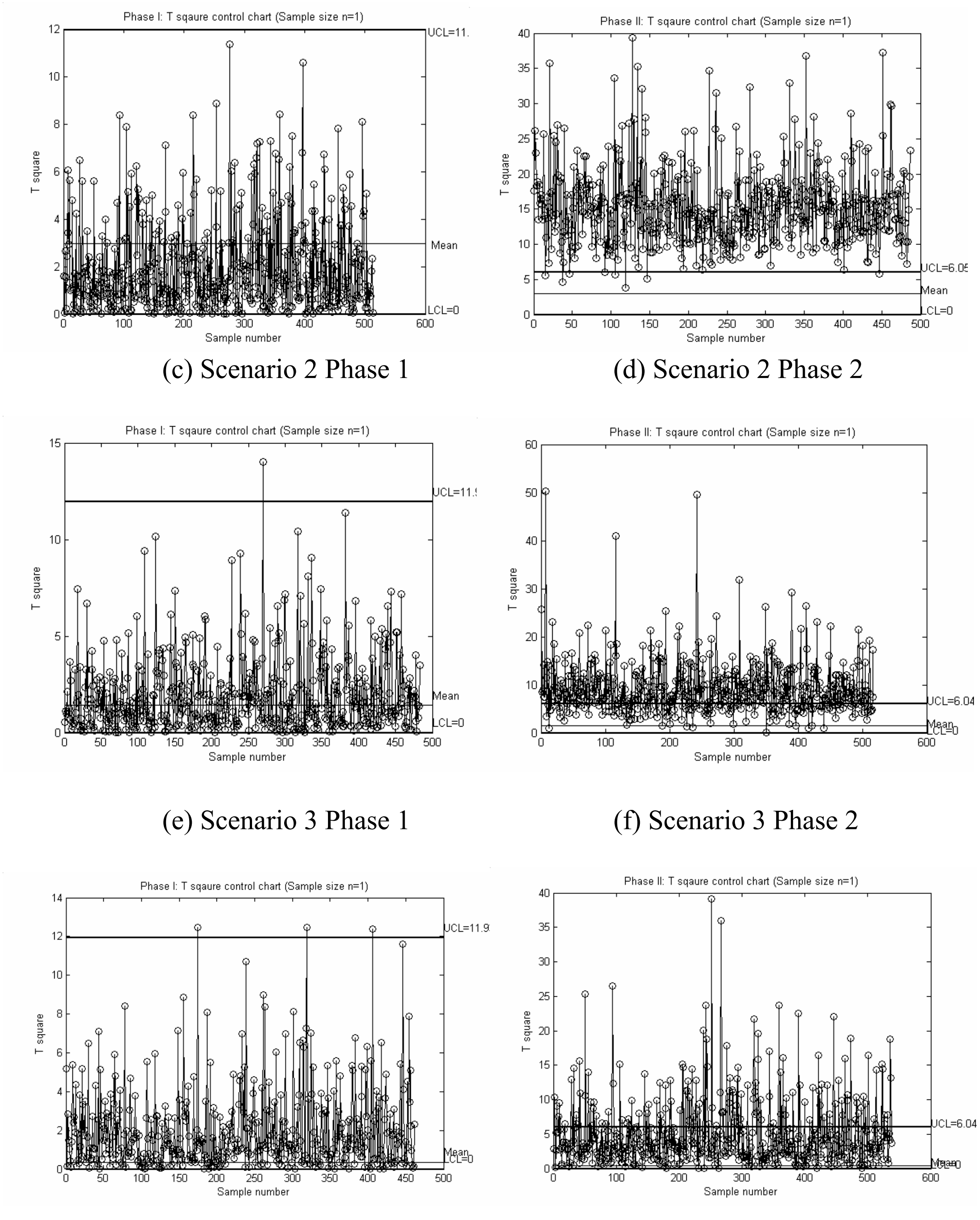
The T^2 chart can deal with only one cluster of data. It uses in-control samples to construct the control chart elements: the sample mean, sample variance-covariance, and then obtain the control limits for in-control data, then using the result to construct Phase 2 control limits.

Using T^2 chart only on in-control samples, and then using the constructed control limits for out-of-control data, we can obtain the performance of T^2 chart on these scenarios.



(a) Scenario 1 Phase 1

(b) Scenario 1 Phase 2



(c) Scenario 2 Phase 1 (d) Scenario 2 Phase 2

(e) Scenario 3 Phase 1 (f) Scenario 3 Phase 2

(g) Scenario 4 Phase 1 (h) Scenario 4 Phase 2

Figure 4.12 T^2 chart on the four scenarios

The performance of T^2 chart on the 4 scenarios is shown in Table 4.5.

Table 4.5 Performance of T^2 chart on the 4 scenarios

	Phase 1 Upper Limit	Phase 2 Upper Limit	α error	β error	Total accuracy
Scenario 1	11.9323	6.0553	0.00423	0	0.998
Scenario 2	11.9362	6.0535	0	0.01643	0.992
Scenario 3	11.9333	6.0498	0.00207	0.28820	0.850
Scenario 4	11.9311	6.0475	0.00649	0.72862	0.605

Note that α error of all the scenarios are different from $\alpha = 0.05$ because all the scenarios are numerical simulations, which have the α errors specific obtained from those scenarios. According to the results for scenario 1 shown in Table 4.5, T^2 chart can obtain good performance when the in-control samples are in normal distribution. However, T^2 chart does not use the information of out-of-control data, when there is big overlap between in-control and out-of-control samples, the β error will be significant. For example, in scenario 4, Phase 1 has α error only 6.49%, but the β error in Phase 2 is 72.86%. So T^2 chart is only good to apply to the data that there is clear difference between in-control and out-of-control clusters.

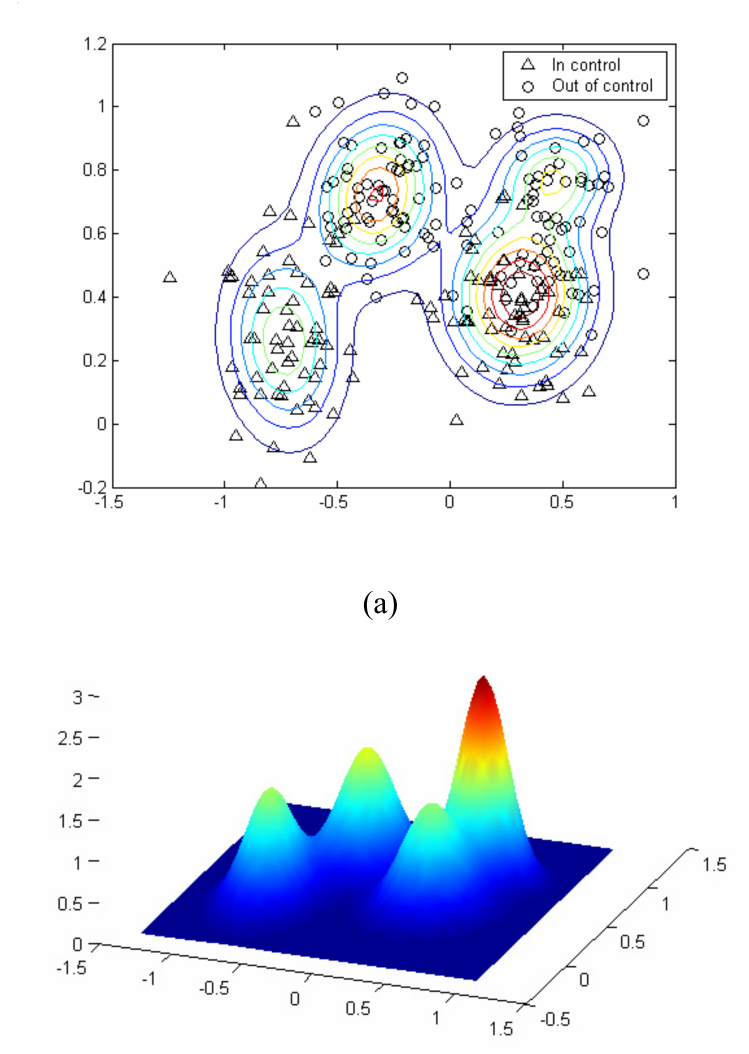
Compare Table 4.4 with Table 4.5, it can be found that although SVM based control chart has higher α error when the overlap is increasing between in-control and out-of-control data, β error is much more decreased than T^2 chart. So is the total error. When the training data available includes both in-control and out-of-control samples and there may be overlap between in-control and out-of-control samples, SVM based control chart is a good choice for multivariate control chart.

If the in-control data is in several clusters instead of a single normal distribution, T^2 chart will be no use because it can consider only one multivariate normal cluster to be the normal condition. To get some further discussion on this problem, in section 4 we will consider using Gaussian mixture model (GMM) method to estimate the parameters of known Gaussian models, and then use general way to construct control chart. We will also discuss the performance comparison of this method and SVM based control chart.

4.5.2 Comparison on Mixture Model Data

Among real-world problems, many of them have the following characteristics: there are several discrete clusters that the performance of the products is marked satisfied, and also the fault pattern is combined with several other discrete clusters. Those clusters are either multivariate normal distributed or with non-Gaussian distributions. Traditionally when encountered with this kind of problem, people first try to figure out the distributions of the data, mainly multivariate normal distribution is used to test the significance of fitness, and then if the test is significant, T^2 chart is used. If the data is not in normal distribution, approximation or transform will be used first. If the samples are in multiple clustering, Gaussian mixture model needs to be used to estimate the mean and variance (covariance) of each cluster, then for each cluster T^2 chart is used. This is a technically sound solution, but not easy to use. In this section, we will demonstrate that SVM based control chart is also capable to solve the multiple cluster problems, and can have better performance than traditional Gaussian mixture model method.

In this section, a two-dimensional data performance comparison is given as shown in Figure 4.13.



(a)

(b)

Figure 4.13 EM estimation of 2-d data

The data is generated according to the following parameters in Table 4.6:

Table 4.6 Data used for comparison of EM and SVM methods (Sample size: 400)

Cluster	Mean	Var-Covariance	Prior	Status
1	[0.3136; 0.4052]	[0.0372,0.0009; 0.0009,0.0260]	0.25	In Control
2	[-0.7382; 0.2603]	[0.0255,-0.0028; -0.0028,0.0346]	0.25	In Control
3	[0.4664; 0.7815]	[0.0255,0.00085;0.0008,0.0104]	0.25	Out of Control
4	[-0.3251; 0.7241]	[0.0295,0.0040;0.0040, 0.0241]	0.25	Out of Control

In the comparison, we use 2-d data to test EM (expectation maximization) and SVM for 50 times each. EM algorithm for GMM (Gaussian Mixture Model) use the information that the data is composed of 4 Gaussian distribution, so the task is to estimate the mean, variance-covariance matrix of each Gaussian and the weight of each Gaussian. The EM algorithm is an iterative procedure that monotonically increases log-likelihood of the current estimate until it reaches a local optimum. We use K-means clustering to separate the original dataset into 4 clusters and K-NN (K-nearest neighborhood) method to provide labels of each point. After 50 times calculation for each method, we obtain results that are in Table 4.7.

Table 4.7 Comparison of average performance of EM and SVM on 2-d simulation data

	EM average	EM best performance	SVM chart	SVM chart best performance*
Total_accuracy	0.514	0.8680	0.856	0.884
α error	0.505	0.1200	0.152	0.128
β error	0.464	0.1440	0.136	0.104

Note(*): SVM chart best performance is obtained by multi-objective Genetic Algorithm mentioned in section 4.3. At the best performance the parameters are: C=1.5685, σ=0.40019. SVM normal performance is obtained by C=1 and σ=1. Both use RBF basis.

The results show that the average performance of EM method is far below SVM control chart, although the best performance of EM is a little higher than normal SVM obtained by C=1 and σ=1. Also with multi-objective Genetic Algorithm to optimize performance, we can obtain the best performance with α error 12.8% and β error 10.4%. Because SVM control chart does not require detailed information of data, and the optimization method is a convex optimization, so SVM control chart will generate a global optimal solution which is the reason why all 50 times simulation can generate the unique solution. The EM method depends on the cluster accuracy related to the true models, and it will fall to a local optimum. That is the reason why each of the 50 runs of EM method will generate different result. Figure 4.14 gives an example of wrong initial clustering cause bad performance of EM method with total accuracy only 44%.

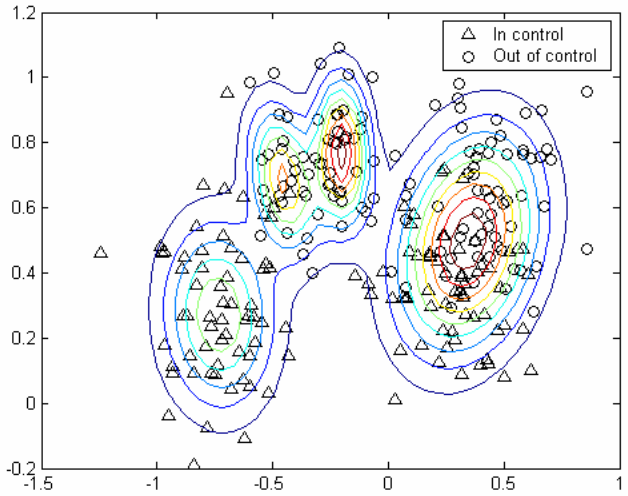


Figure 4.14 Wrong initial clustering will cause bad performance of EM method

From the viewpoint of average performance, SVM is much better than EM algorithm, and the best performance is almost same for both methods.

4.5.3 Comparison on Non-Gaussian Distribution Data

In this section we illustrate SVM based control chart for non-Gaussian distributed data and compare with T^2 chart and EM based method. Also we want to illustrate that T^2 chart will give misleading results for non-Gaussian data.

Figure 4.15 is an non-Gaussian distributed dataset. As shown in the figure, the data does not satisfy with any existing distribution. Using the method derived in section 4.2, we can find the optimal separating hyperplane as the solid line shown in Figure 4.16, and the two dashed lines are two margins with evaluation function values +1/-1. We use $C = 1$ and $\sigma = 1$ in this example.

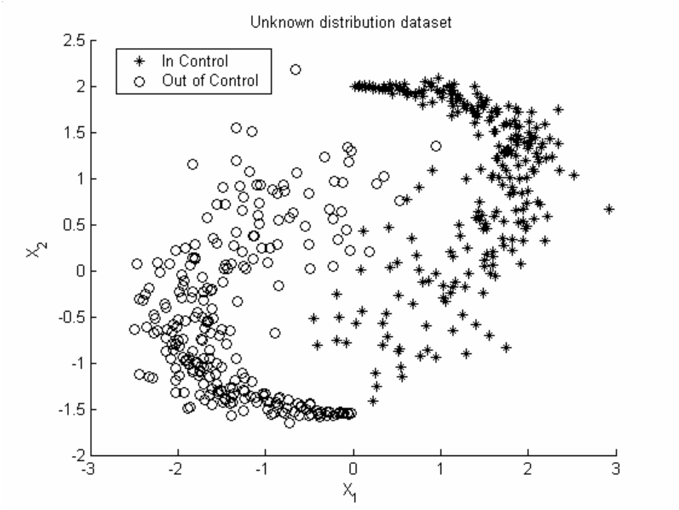


Figure 4.15 Non-Gaussian distribution dataset

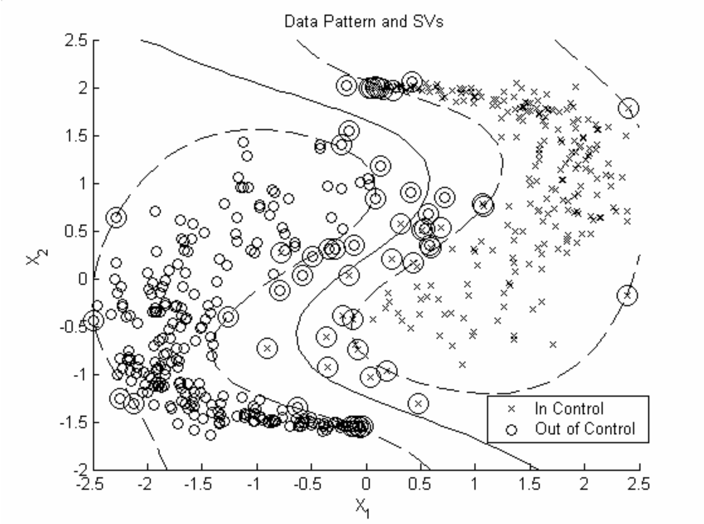


Figure 4.16 SVM separating plane for in-control and out-of-control samples

Figure 4.17 is the control chart constructed by SVM based method. The samples above the control limit are detected as in-control samples, and below control limit is detected as out-of-control samples. Because SVM based control chart build a nonlinear separating hyperplane so a nonlinear control limit is set in the feature space instead of input space. By this setting we obtain overall 98.6% total accuracy.

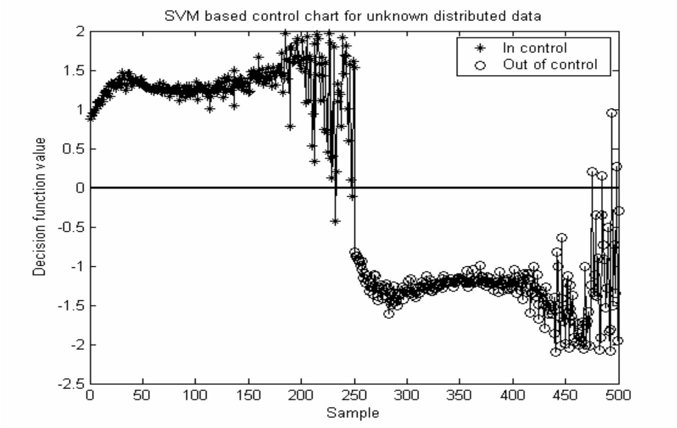


Figure 4.17 SVM based control chart for non-Gaussian distributed data

If we assume the data is in normal distribution by mistake and use T^2 chart, the derived control limits are UCL=11.89 and LCL=0. Phase 1 plot is shown in Figure 4.18.

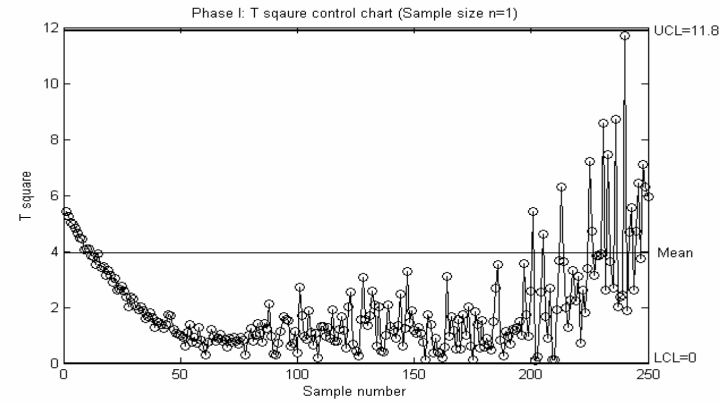


Figure 4.18 T^2 chart (Phase 1) for non-Gaussian distribution in-control training samples

After the control limits are set by Phase 1, T^2 chart is applied to forthcoming samples. Here we use out-of-control samples as new samples to test T^2 chart. Because we suppose the samples are satisfied with normal distribution in T^2 chart, the testing samples have big detecting errors as shown in Figure 4.19. The total accuracy for T^2 chart in this case is only 90% and the β error reaches 8% for this specific case.

From the truth that the data is not in normal distribution, we can tell that the results are very misleading. The distribution of data needs to be tested before we apply T^2 chart, otherwise the results could not be explained.

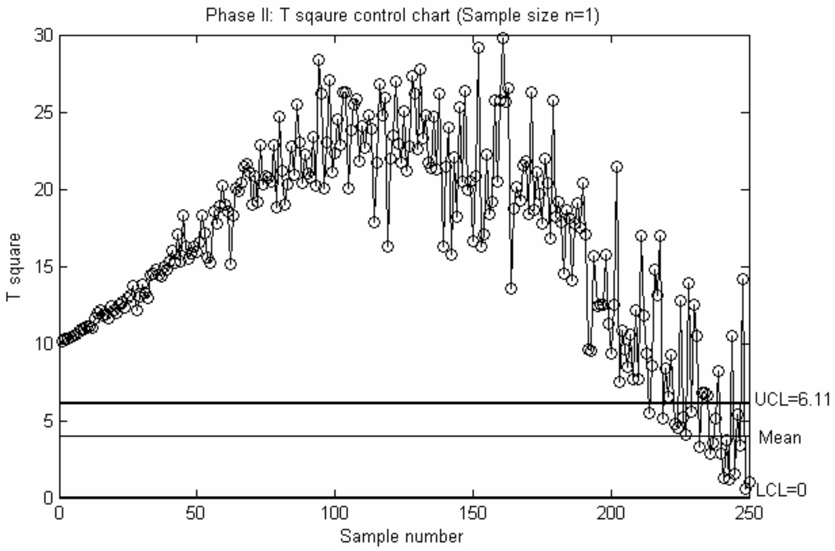


Figure 4.19 T^2 chart (Phase 2) for non-Gaussian distribution out-of-control testing samples

Then we use EM method to estimate the distribution of in-control and out-of-control samples. Expectation Maximization (EM) algorithm is used to estimate the probability density of a set of given data by obtaining maximum likelihood estimates of parameters in probabilistic models. EM executes with a loop with an expectation (E) step, which computes the expected value of the hidden variable, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters given the data and setting the latent variables to their expectation. Normally finite Gaussian mixture model is used in order to model the probability density of the data.

By using EM method, we still suppose the data are in normal distribution. EM method considers all of the existing samples, both in-control and out-of-control ones. Figure 4.20 shows the contours of the estimation for distributions of in-control and out-of-control samples. From Figure 4.20 we find that EM method catches part of the characteristics of the samples such as mean and variance-covariance, but because the samples themselves are not in normal distribution, there are still somewhat large detecting errors. The total accuracy of EM method for this case is 91.6%, a little higher than T^2 chart.

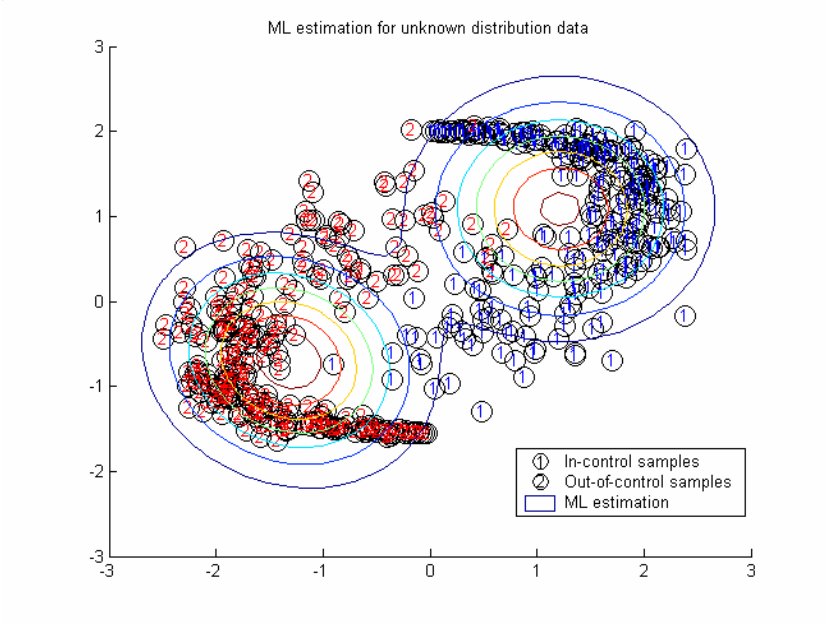


Figure 4.20 EM estimation for non-Gaussian distribution samples

To sum up, the comparison of the performance for SVM chart, T^2 chart and EM method is shown in Table 4.8. It is easy to be found that for non-Gaussian distributed data, SVM chart shows higher total accuracy than traditional T^2 chart and EM method. So SVM chart is much more suitable for the samples with non-Gaussian distribution.

Table 4.8 Comparison between SVM chart and T^2 chart on non-Gaussian distribution data

	α error (Phase 1)	β error (Phase 2)	Total accuracy
SVM chart	0.016	0.012	0.986
T^2 chart (mis-using)	0	0.2	0.90
EM method	0.092	0.016	0.916

4.6 CONCLUSION

With the wide use of computer networks and monitoring techniques such as sensor networks, huge amount of data is available for online monitoring. Although multivariate process monitoring methods have been investigated for a long time, the current multivariate control charts still have a lot of limitations. In this chapter we present SVM based control chart and compare the performance with the current multivariate control chart. SVM based control chart does not require the distribution of the samples, so it has highly applicable in real world problems.

We have the following conclusions for the chapter:

(1) T^2 chart has similar performance with SVM based control chart when the data is Gaussian distribution. However, when the data is in non-Gaussian distribution, T^2 chart gives misleading results. Because T^2 chart does not consider out-of-control data during the construction of control chart (phase 1), it has large β error when the in-control data and out-of-control data have large overlap. SVM based control chart can resolve this problem by constructing an

optimization problem to maximize the margin between in-control data clusters and out-of-control clusters, at the same time minimize the penalty caused by the overlap. So if the in-control data and out-of-control data has large overlap, SVM based control chart is preferred.

(2) When the in-control and out-of-control has more than one clusters, traditional multivariate control chart does not work. SVM based control chart can still resolve this problem by construct the hyperplane between in-control and out-of-control data no matter how many clusters the data has.

(3) We can optimize the performance of SVM based control chart by finding the optimal pair of pre-defined parameters C and σ by multi-objective GA. Multi-objective GA can find the optimal Pareto set of the combinations of C and σ to obtain the Pareto frontier of α and β error. This method provides us intuitive way to understand the mechanism of SVM based control chart for the specific data.

(4) SVM based control chart is good for real-time application. Except for the relatively long training for the model parameters, the calculation of kernel distance of new samples is very fast. In the application of intrusion detection, training algorithm can run off-line whenever enough samples is obtained from online data. We can improve the training time by using higher performance computer, but the monitoring and detection only need computers with normal performance. We have also shown in this chapter that SVM based control chart

can be applied to computer networks anomaly detection with very good performance.

CHAPTER 5 UNSUPERVISED KERNEL BASED MONITORING CHARTS

This chapter addresses the development of control charts when only in-control samples are available and the data is not linearly separable in the input space. Two unsupervised kernel-based multivariate monitoring control charts are proposed. The first method is kernel principal component analysis based control charts. This control chart is constructed in the feature space based on the first several orthogonal principal components that contribute most of the variance in the data. The second method is support vector clustering based control charts, which are constructed by formulizing an optimization problem to obtain the center and radius of a minimal hypersphere to enclose most of the in-control data in the feature space.

This chapter is organized as follows: Section 5.1 gives a brief introduction on the motivation of the research in this chapter. Section 5.2 proposes the kernel principal component analysis (KPCA) based monitoring charts. Section 5.3 develops the unsupervised support vector clustering (SVC) based monitoring charts. The application of SVC based monitoring chart to the computer network intrusion data is discussed in Section 5.4. Conclusion is drawn in Section 5.5.

5.1 INTRODUCTION

Unsupervised learning methods are critically important when there are no sufficient labeled training samples of attacks especially for new or unknown attacks. However, it is generally much easier to get the sufficient samples under the normal operational condition, which are often used to train the normal operational boundary. For this purpose, linear models have been widely investigated, such as PCA, discriminate analysis and linear clustering. These methods assume that in-control data and out-of-control data are linearly separable, which in reality may not be satisfied in many cases, like the computer network audit data used in this dissertation. In this chapter we propose two types of unsupervised kernel based control charts: kernel principal component analysis (KPCA) based control charts and support vector clustering (SVC) based control charts with their applications to anomaly detection in computer networks.

5.2 KERNEL PCA BASED CONTROL CHARTS

5.2.1 Introduction of Kernel PCA

PCA as a linear transform is popularly used for feature extraction and data dimension reduction. In this approach, the first several principal components reflecting the majority of the data variance are selected to construct control charts. Those principal components are corresponding to those larger eigenvalues and the associated eigenvectors as the projection directions. PCA intends to use a smaller dimension of linearly transformed features to reconstruct the original large dimension of data while keeping the most structural variance of original data. Analogically, when in the need of nonlinear transform for the data with the complex non-Gaussian distributions, kernel-PCA (KPCA) uses the nonlinear kernel transform to map the original data having the nonlinear relationship in the original data space into a feature space that can be separated by a linear classifier.

Again, let $\varphi(x)$ be a nonlinear mapping to some feature space *F*. In feature space *F*, $\varphi(x_j)$ is centered as $\sum_{j=1}^{n}\varphi(x_j)=0$ (*n* is the dimension of $\varphi(x_j)$), the covariance matrix is

$$\bar{C}=\frac{1}{n}\sum_{j=1}^{n}\varphi(x_j)\varphi(x_j)^T \tag{5.1}$$

Then the principal components are obtained by finding eigenvalue $\lambda>0$, and eigenvector $\mathbf{V}\neq 0$ that

$$\lambda\mathbf{V}=\bar{C}\mathbf{V}=\frac{1}{n}\sum_{j=1}^{n}(\varphi(x_j)\cdot\mathbf{V})\varphi(x_j) \tag{5.2}$$

Suppose $\mathbf{V}=\sum_{j=1}^{n}\beta_j\varphi(x_j)$ is the linear combination of elements $\varphi(x_j)$ with coefficients β_j, it can be rewritten as $\lambda(\varphi(x_j)\cdot\mathbf{V})=(\varphi(x_j)\cdot\bar{C}\mathbf{V})$. So the eigenvalue problem is now

$$n\lambda\boldsymbol{\beta}=\mathbf{K}\boldsymbol{\beta}\quad(\boldsymbol{\beta}=(\beta_1,\ldots,\beta_n)^T) \tag{5.3}$$

where **K** is kernel matrix defined as $\mathrm{K}_{ij}=\varphi(x_i)^T\varphi(x_j)$. Formula (5.3) is to solve eigenvalue in feature space *F*. The solutions are (λ_j,β_j). Eigenvector are normalized in feature space, i.e. $\mathbf{V}^T\mathbf{V}=1$. This can derive to

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j\varphi(x_i)^T\varphi(x_j)=1=\boldsymbol{\beta}^T(\mathbf{K}\boldsymbol{\beta})=\lambda_j\boldsymbol{\beta}^T\boldsymbol{\beta} \tag{5.4}$$

When we have a new observation *x*, we can extract features of *x* by projecting the mapped pattern $\varphi(x)$ onto $\mathbf{V}$ (in feature space)

$$(\mathbf{V}\cdot\varphi(x)) = \sum_{j=1}^{n} \beta_j (\varphi(x_j)^T \varphi(x)) = \sum_{j=1}^{n} \beta_j K(x_j, x) \tag{5.5}$$

Here we can find that the only difference between KPCA and PCA is that PCA solve eigenvalue problem in input space while KPCA solve it in feature space. So the choice of PCA or KPCA only depends on the linearity of the input space. If the input space is nonlinear, we need to use kernel matrix to transform into a linear feature space, then use PCA in the feature space.

5.2.2 KPCA Based Control Chart

Normally, if the in-control samples and out-of-control samples can be linearly separated, PCA is enough to handle this situation. But if those two types of samples can not be linearly separated, we need to get the help of kernel transform to transfer the input space into a higher dimensional feature space where the two types of samples can be separated by a linear hyperplane, then apply PCA in the feature space. This is the basic idea behind KPCA control charts.

In KPCA based control charts, we take the advantage of the kernel transform to first map the nonlinear input space data into a linear feature space, and then use linear PCA to get the directions (eigenvectors) that accounts for most variance. We can use the first several principal components to compose the control charts which account for the majority of the variance (e.g. over 90% of the variance).

5.2.3 Simulation

Figure 5.1 illustrates the difference between linear PCA and kernel PCA. Linear PCA can only rotate the current coordinates to make the largest variance happens on the new x_1 axis, i.e. the direction of the first principal component, but can not change the shape of data. So if the data is not linear separable, as shown in Figure 5.1, Linear PCA cannot detect outliers in the new samples. The data is generated by samples along a nonlinear trend line with specific variance. Out-of-control samples are generated by adding more than 3 σ variances along the trend line.

Kernel PCA is superior to linear PCA in the fact that it provides the possibility to change the shape of the data in feature space so that the new data is easy to be used for outlier detection. In Figure 5.1 the grids (contour lines) represents the directions of first and second principal components. Here the polynomial kernel is used with the parameter of d = 0.03596 by trail and error.

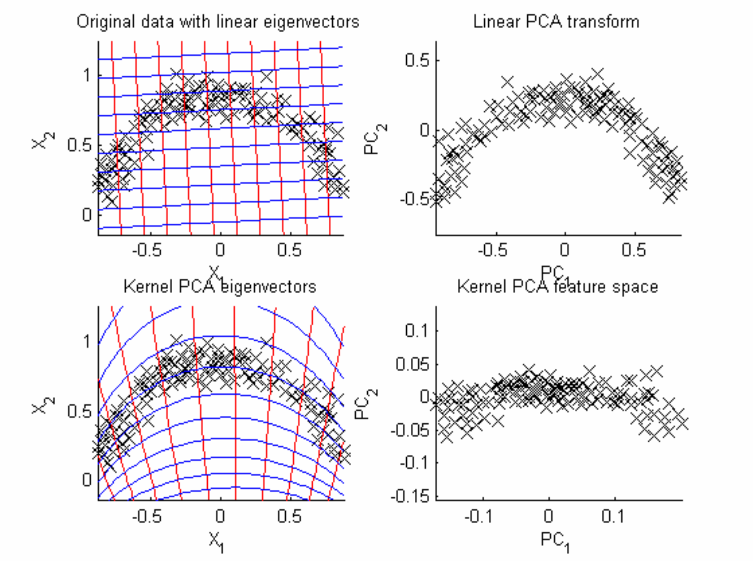


Figure 5.1 Comparison between linear PCA and kernel PCA

Because principal components in feature space are orthogonal to each other, there is no correlation between any pair of principal components, i.e. they are independent to each other. In this case, the first and second eigenvalues have accounted for over 95% of the total variance, so the control chart is constructed in the feature space based on those two principal components.

The testing data set is shown in the input space and feature space in Figure 5.2. Figure 5.3 shows control charts constructed in the feature space on the first and second kernel principal components. Combining those two *X*-bar control charts together we can tell which sample is out of control.

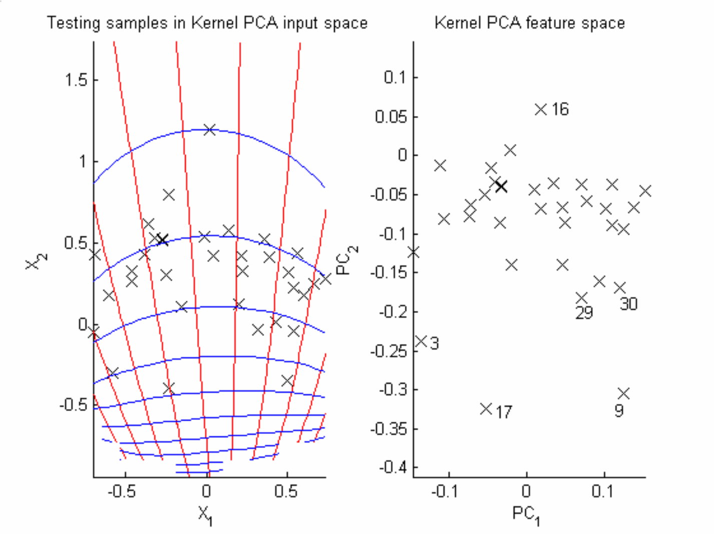


Figure 5.2 Testing data set in input space and feature space

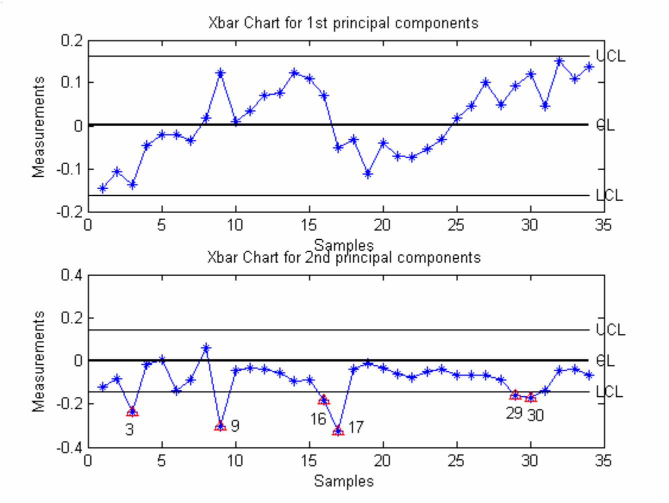


Figure 5.3 *X*-bar control charts on first and second kernel principal components in feature space

5.3 UNSUPERVISED SUPPORT VECTOR CLUSTERING (SVC) BASED CONTROL CHART

5.3.1 Introduction to SVC

Support vector clustering is also called one-class classification [84, 85]. There are two types of algorithms for one-class classification [81]: one is to construct a hyperplane in the feature space which makes a specified fraction of the training samples above the hyperplane, and the distance of this hyperplane to the origin is maximized [86]; the other method simply uses a hypersphere with soft margin in the feature space to enclose data. The hypersphere is characterized by a center $\mathbf{a}$ and a radius R (R>0). This method is to design a hypersphere with the minimal volume that can contain all the data in the case of no outliers or contain most of the data in the case of having some outliers. In this chapter we use the idea of the second method (hypersphere).

Actually the idea of hypersphere is used in [80] to obtain a minimum separating dimension. The difference is that in [80] the smallest ball contains all the training data no matter whether they are in the same class or not, but in the support vector clustering, there is only one class enclosed by this hypersphere and the radius R is decided to separate the normal data from outliers.

The clustering boundary is defined in feature space by

$$\left\|\varphi(x_j)-\mathbf{a}\right\|^2 \le R^2 \quad (\forall j) \tag{5.6}$$

for data without outliers. In case of outliers exist, slack variables $\xi_j \ge 0$ is added.

$$\left\|\varphi(x_j)-\mathbf{a}\right\|^2 \le R^2+\xi_j \quad (\forall j) \tag{5.7}$$

Without loss of generality, we construct an optimization problem

$$\min f(R,\mathbf{a}) = R^2 + C\sum_j \xi_j \tag{5.8}$$

Using Lagrange multipliers $\alpha_j \ge 0$ and $\gamma_j \ge 0$ we get

$$L(R,\mathbf{a},\alpha_j,\gamma_j,\xi_j) = R^2 + C\sum_j \xi_j - \sum_j \alpha_j (R^2+\xi_j-\left\|\varphi(x_j)-\mathbf{a}\right\|^2) - \sum_j \gamma_j \xi_j \tag{5.9}$$

Set partial derivative to zero for extreme point yields:

$$\begin{aligned} &\tfrac{\partial L}{\partial R} = 0 \rightarrow \sum_j \alpha_j = 1 \\ &\tfrac{\partial L}{\partial \mathbf{a}} = 0 \rightarrow \mathbf{a} = \frac{\sum_j \alpha_j \varphi(x_j)}{\sum_j \alpha_j} = \sum_j \alpha_j \varphi(x_j) \\ &\tfrac{\partial L}{\partial \xi_j} = 0 \rightarrow C-\alpha_j-\gamma_j = 0 \end{aligned} \tag{5.10}$$

and Karush-Kuhn-Tucker (KKT) condition yields

$$\xi_j \gamma_j = 0 \tag{5.11}$$

$$\alpha_j(R^2 + \xi_j - \left\|\varphi(x_j) - \mathbf{a}\right\|^2) = 0 \tag{5.12}$$

We can define $0 \le \alpha_j \le C$ for the (5.12) above and put (5.10) back to (5.9)

$$\begin{aligned} &Max: L = \sum\nolimits_j \alpha_j K(x_j, x_j) - \sum\nolimits_{i,j} \alpha_i \alpha_j K(x_i, x_j) \\ &s.t. \quad \sum_j \alpha_j = 1 \text{ and } 0 \le \alpha_j \le C \end{aligned} \tag{5.13}$$

By solving (5.13) we can obtain minimum radius R for the enclosing sphere. In feature space only the data associated with $0 < \alpha_j < C$ are needed to define the boundary. Those data are called support vectors (SVs) and lie on the surface of the feature space sphere. The two controlling variables are C (soft margin constant, used to balance radius of the minimal sphere and number of outliers) and σ (RBF kernel parameter, used to determine the scale of data probing), detailed discussion can be found in [84]. C is the weight to balance the two objective functions: min $\sum_j \xi_j$ and min R^2. If let C to be infinity, a sphere is obtained to just enclose all data, then there will be no sample to be excluded. So C can be used to determine the number of samples to be excluded, decreasing C can lead the increase of the number of samples to be excluded from the control limit. That is the why we can use C value to control the boundary for excluding some outliers in the training data.

5.3.2 Extend SVC Method to Construct Multivariate Control Charts

The SVC based multivariate control chart is based on the distance R' from the center $\mathbf{a}$ in the feature space to the mapping of new observation $\varphi(x)$ in the feature space.

$$R^{'} = \sqrt{(\varphi(x) - \mathbf{a})^{T} (\varphi(x) - \mathbf{a})} \tag{5.14}$$

By substituting $\mathbf{a}$ using (5.10)

$$R^{'} = \sqrt{\varphi(x)^{T} \varphi(x) - 2\sum_{i} \alpha_{i} k(\varphi(x), \varphi(x_{i})) + \sum_{i,l} \alpha_{i} \alpha_{l} k(x_{i}, x_{l})} \tag{5.15}$$

The control limit is R obtained from the optimization problem (5.13). If $R' > R$, we conclude that this new observation is out-of-control sample. The choice on the value of C and σ can adapt the method to different problems.

Overall, similar to T^2 chart with subgroup size 1, the SVC multivariate control chart for individual observation also needs two phases to implement. In Phase 1, the in-control samples in the original space are transformed to the feature space using a selected kernel, then a minimal ball (m dimensional, m is the dimension of feature space instead of original variable dimension p) with unknown center and radius is designed to envelop all the in-control samples in the feature space. The optimized radius from the center in the feature space is defined to be the control limit. The advantage of the kernel transform in Phase 1 is that no matter the in-control samples in the original space is in Gaussian distribution or not, they will have an envelop to be enclosed by a m dimensional ball. In Phase 2 we construct the control chart in the feature space using the already fixed kernel transform and then use the optimal radius as the control limit. If the new sample has radius less than or equal to the control limit, we consider this new sample in control; otherwise we consider the new sample out of control.

Since the way of constructing SVC multivariate control chart is similar to T^2 chart with subgroup size 1, we will compare the performance of SVC control chart to T^2 chart with both Gaussian distribution data and non-Gaussian distribution data in the next sections.

5.3.3 Simulation

We first use a data set with the same shape of Figure 5.1 to test SVC based control chart. In this section we use the RBF kernel to transform the data from the input space data into the feature space, and then search a hypersphere to enclose the data if the data are all in-control samples. The parameters used in this case is: $\sigma = 0.25$. If there is an outlier in training samples, we either delete it first or adjust controlling parameters to isolate the outliers from in-control samples.

In Figure 5.4 the line connecting support vectors shows the hypersphere mapped back to the input space, and the circles are support vectors. Other lines are contour lines. It may be a high dimension of data in the feature space and the hypersphere is a hyberball. The contours in Figure 5.4 show the hyperspheres with the equal distance from the center in the feature space.

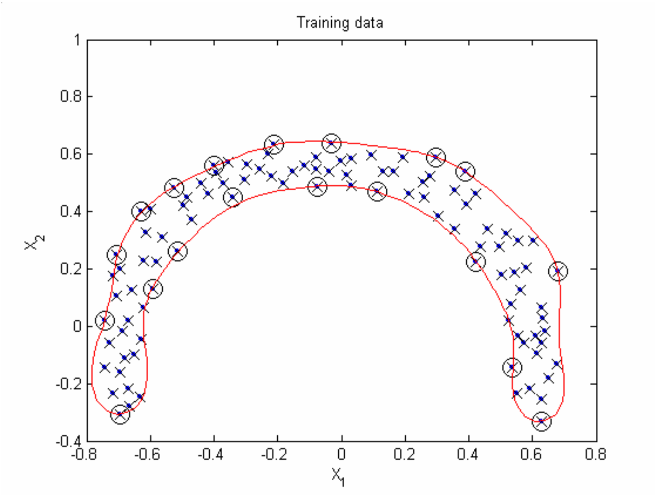


Figure 5.4 Training data for SVC based control chart and decision boundary

Figure 5.5 shows the testing data used for this case. As mentioned earlier, the out-of-control samples are generated by applying more than 3σ variance to the trend-line. The in control and out of control samples are marked with different symbols by the discrimination of SVC control limit. In the plot, the samples marked with labels are with true status as out of control.

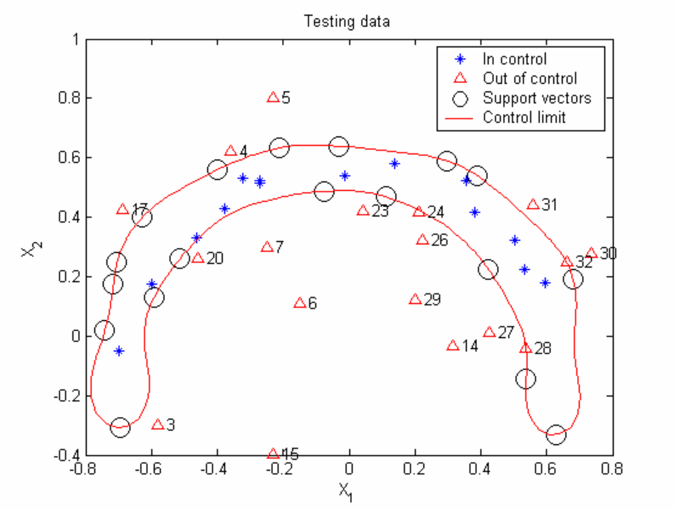


Figure 5.5 Testing data set for SVC based control chart

Figure 5.6 is the control chart based on SVC method. The upper plot is for training data. Note that the support vectors lie on the decision boundary, and the kernel distance of the support vectors are all equal. The in-control samples having different distances from the kernel center are all below the control limit. The lower part of Figure 5.6 is testing data plots in the control chart. Based on the kernel distance, the status of new sample can be determined. Compared with the true status of all the samples, we find that all the out-of-control samples are detected in this example. At the same time, there is no false alarm for those samples with true status as in-control.

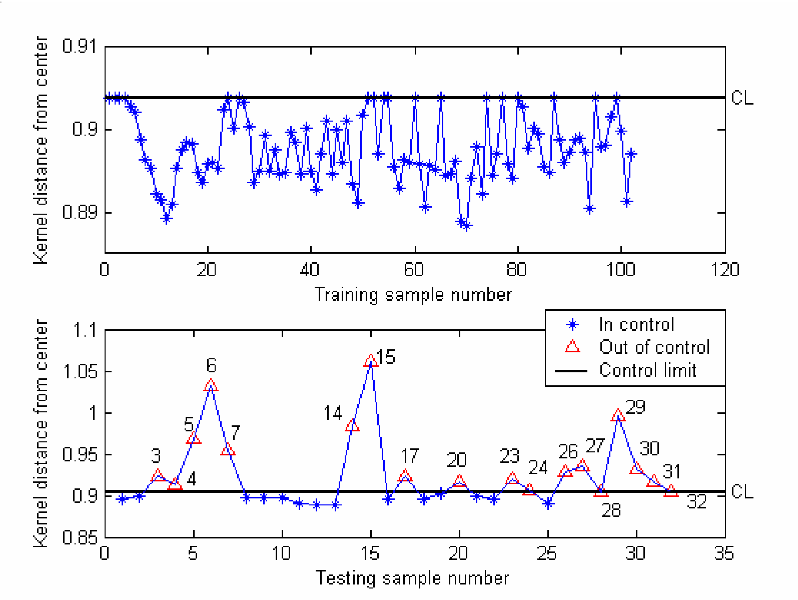


Figure 5.6 SVC based control chart $\sigma = 0.25$

Now consider the case that there is an outlier in (0,0), as shown in Figure 5.7. If the SVC was trained with original parameters, i.e. $\sigma = 0.25$ with no data excluded, we would obtain a model taking this outlier as a good sample thus resulting a misleading result. To exclude this outlier, we adjust parameter C to obtain a soft margin in the training stage. In Figure 5.8, we can find the outlier has been excluded from the clustering, but the decision boundary is not a single continuous enclosure area as in Figure 5.5. Figure 5.9 is the testing data plot using the trained model. Figure 5.10 is the control chart based on SVC with *C*=0.8 and $\sigma = 0.25$. It can be found that for the purpose of excluding outliers, some in-control training sample points are also excluded from the clustering (detected as false alarm) for the compromise.

So if there are outliers in the training data, the better way is to clean the data before we use SVC to get control limit. Otherwise we need to adjust the parameter *C* very carefully to exclude the outliers from the model.

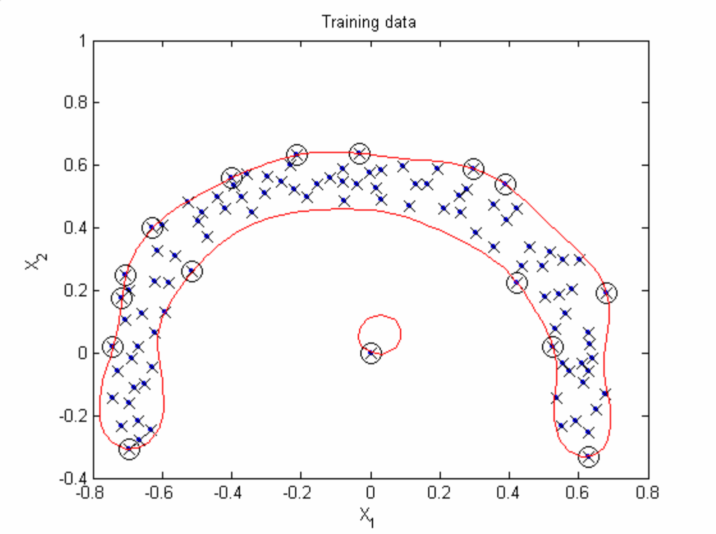


Figure 5.7 Control limits boundary including all samples with outlier at (0,0)

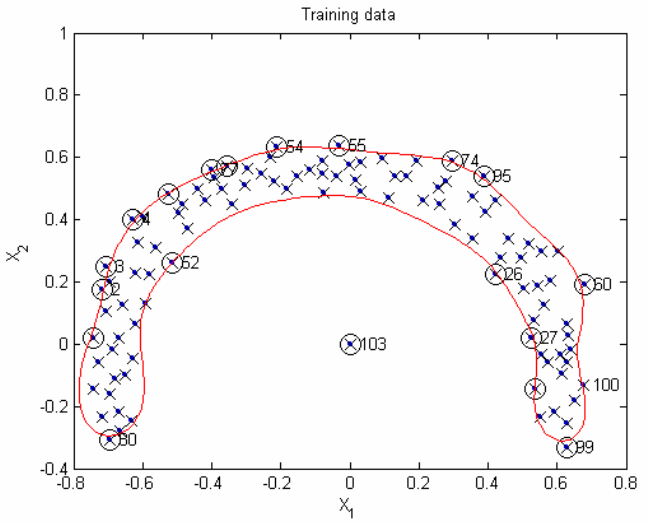


Figure 5.8 Control limits boundary to exclude extreme points including outlier at (0,0)

(*C*=0.8 and $\sigma = 0.25$)

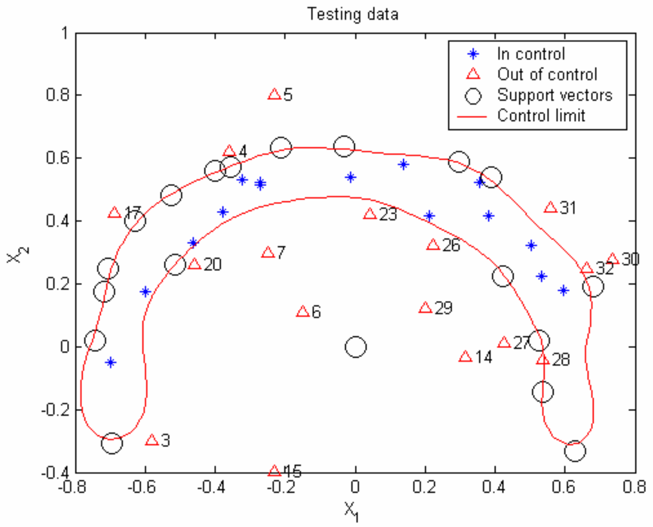


Figure 5.9 Detection performance for testing data (*C*=15 and $\sigma = 0.25$)

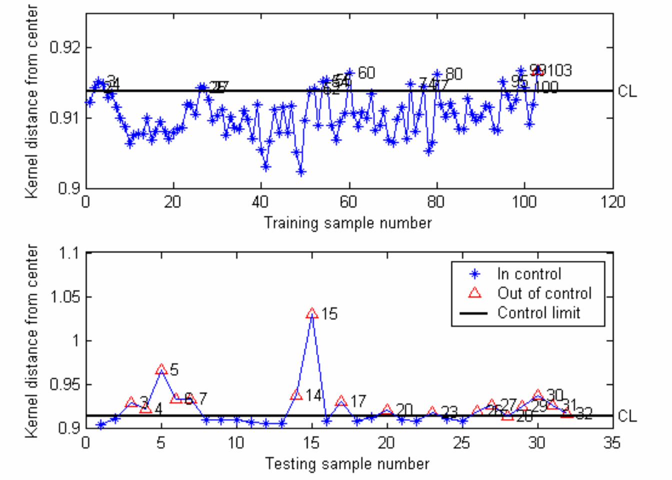
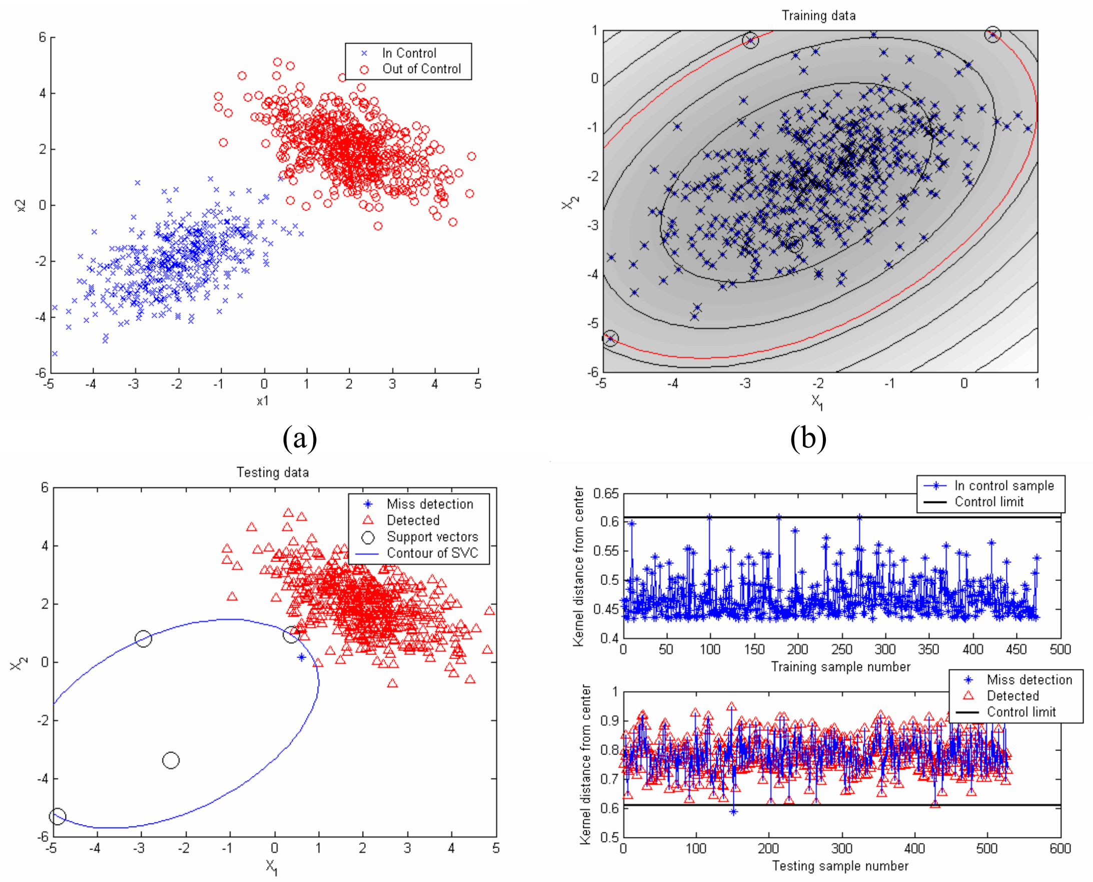


Figure 5.10 Control chart plots using SVC with C=15 and $\sigma = 0.25$

5.3.4 Comparison Between SVC Based Control Chart and Other Control Charts

5.3.4.1 Gaussian distribution data

Similar to Section 4.3.2, we will compare SVC based control charts with T^2 charts using the data shown in Figure 4.9. The following Figures (Figure 5.11 to Figure 5.13) show the results of each scenario of SVC control chart. In each Figure, (a) shows the data (both in control and out of control samples), (b) shows the SVC control limit mapped back to original space, also shows the contours of the same radius. (c) shows the control limit and the out-of-control samples to demonstrate how many of them are detected and miss-detected. (d) shows the control chart for Phase 1 and Phase 2. In this section, we use all the in-control samples as training samples, and all the out-of-control samples as testing samples.



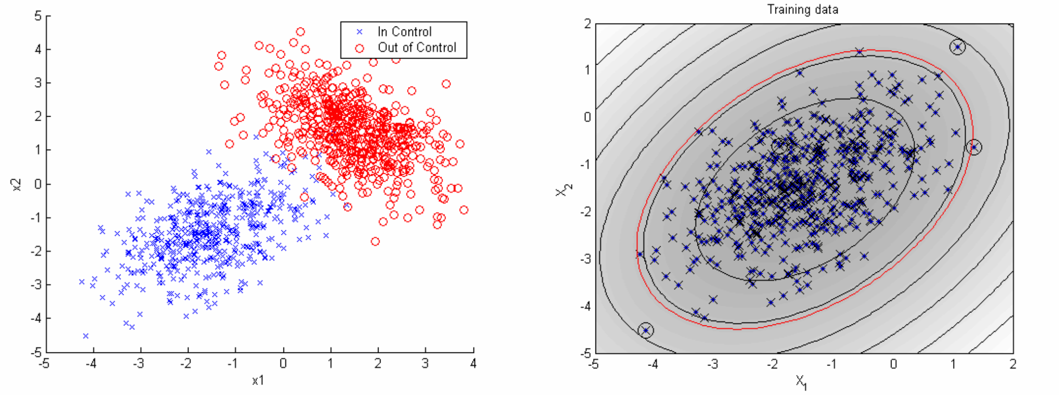
(a)

(b)

(c)

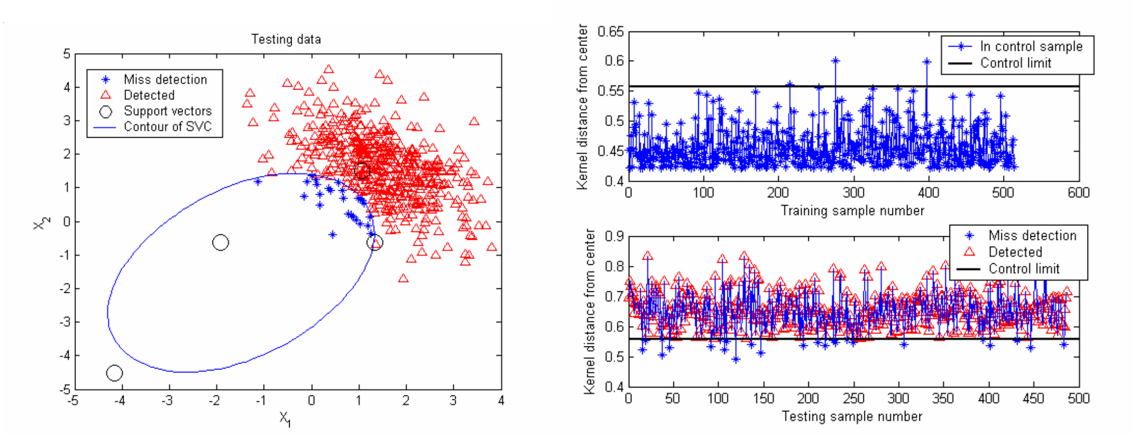
(d)

Figure 5.11 SVC control chart for scenario 1 ($\sigma = 5, C = \infty$)



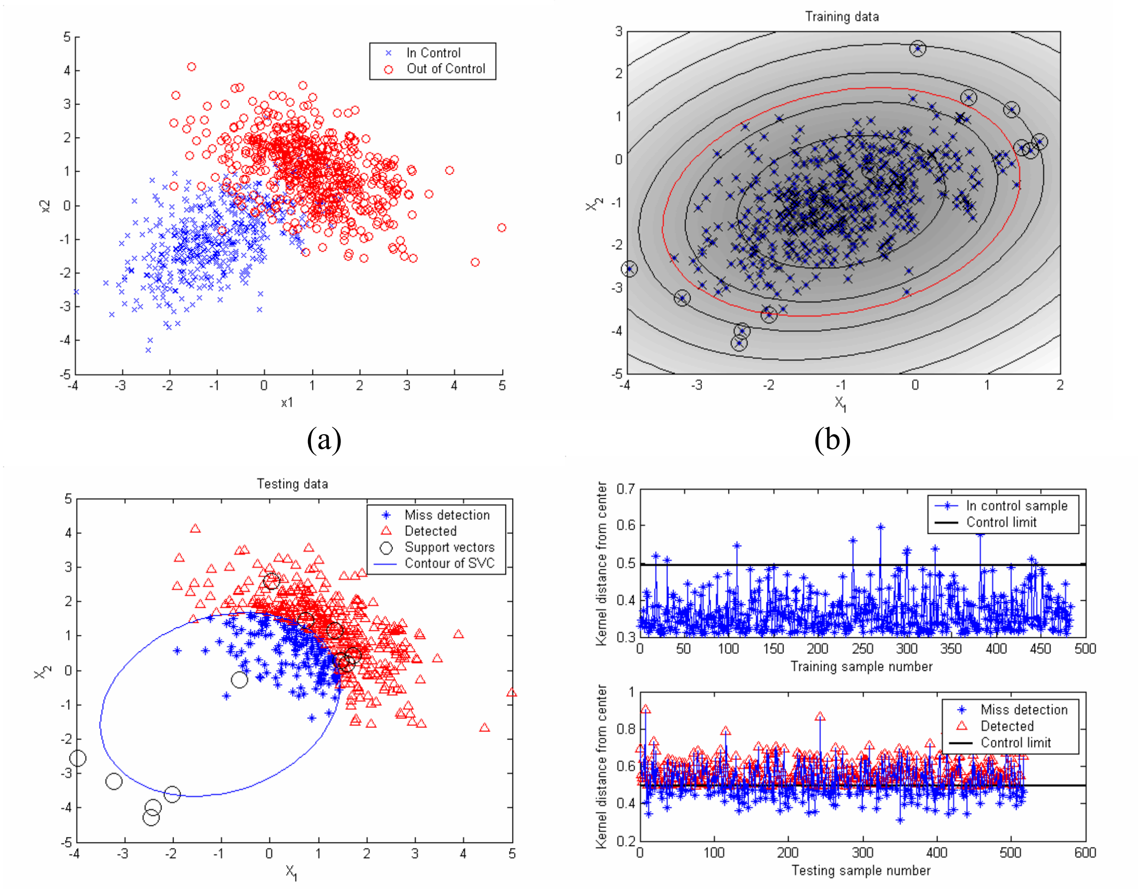
(a)

(b)



(c) (d)

Figure 5.12 SVC control chart for scenario 2 ($\sigma = 5, C = 5$)



(a) (b)

(c) (d)

Figure 5.13 SVC control chart for scenario 3 ($\sigma = 5, C = 1$)

The comparison of performance between SVC control chart and T^2 chart is shown in Table 5.1. We can find that in Scenario 1 to 3 T^2 chart has slightly better performance in total accuracy.

Table 5.1 Performance of SVC control chart and T^2 chart on the 3 scenarios

	SVC control chart			T^2 chart		
	α error	β error	Total accuracy	α error (Phase 1)	β error (Phase 2)	Total accuracy
Scenario 1	0.0042283	0.0018975	0.997	0.00423	0	0.998
Scenario 2	0.005848	0.053388	0.971	0	0.01643	0.992
Scenario 3	0.022774	0.31721	0.825	0.00207	0.28820	0.850

5.3.4.2 Comparison on non-Gaussian distribution data:

σ =1, C=2.5 are used to construct a SVC control chart for non-Gaussian distribution data shown in Figure 4.9. In Phase 1, the control limit is built based only on in-control training data as shown in Figure 5.14. Figure 5.15 shows the control limit after optimization, and Figure 5.16 shows the control charts for both Phase 1 and Phase 2 in this case.

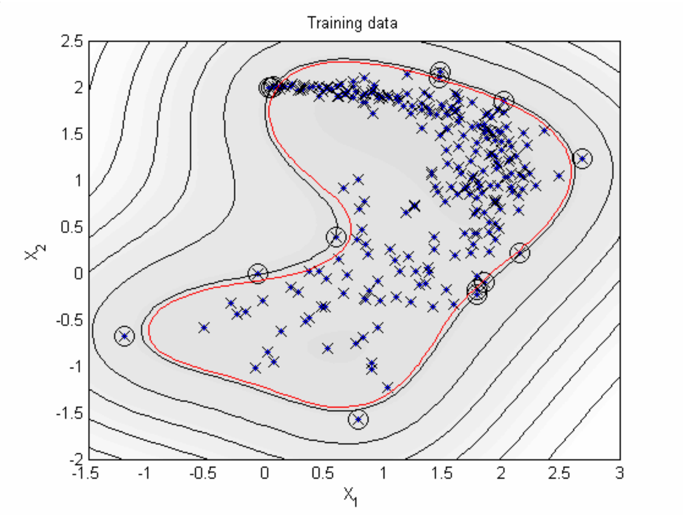


Figure 5.14 Phase 1 control limit (σ =1, C=2.5)

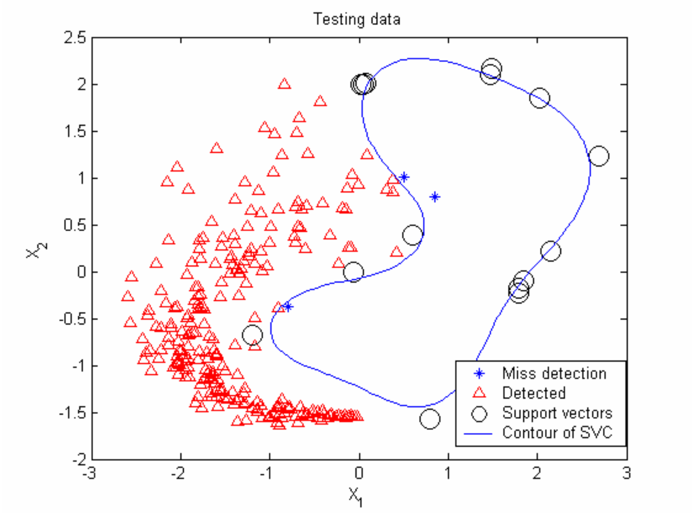


Figure 5.15 Detection performance for out-of-control data (σ =1, C=2.5)

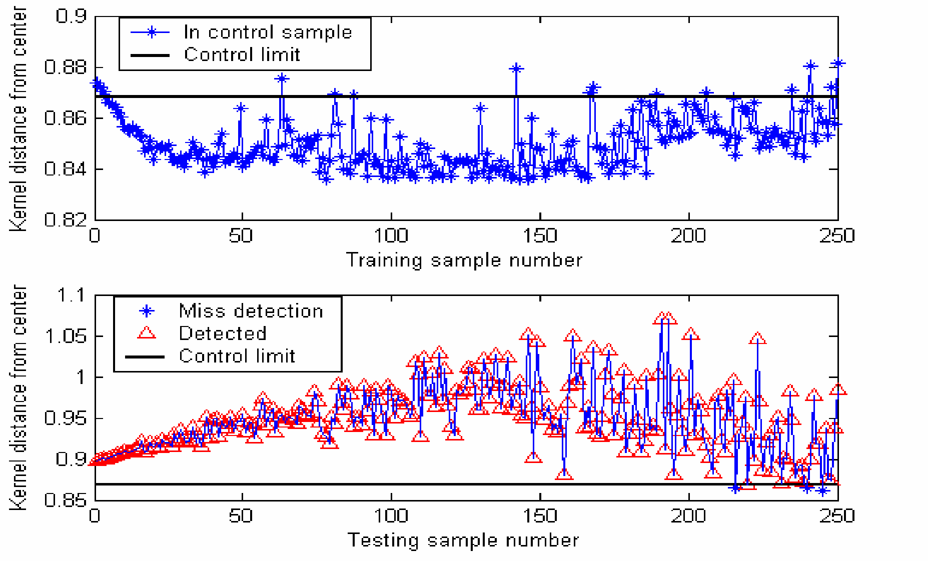


Figure 5.16 SVC Control chart for non-Gaussian distribution data

The performance comparison between SVC control chart and other charts can be found in Table 5.2. It is easy to find that T^2 charts performed much worse than all other charts, because it is miss-used without satisfying the Gaussian distribution assumption. EM method is used better in this case, however it needs to specify the correct number of Gaussians classes. SVC chart and SVM chart do not need the knowledge of data distribution. The reason SVM chart performs a little better than SVC chart is that SVM tries to minimize the overlap of in-control data and out-of-control data when there is, but SVC only takes in-control data as reference. So if there is both in-control and out-of-control samples to construct control chart, we will prefer SVM based control charts.

Table 5.2 Comparison of SVC chart with other charts (refer to: Table 4.4)

	α error (Phase 1)	β error (Phase 2)	Total accuracy
SVC chart	0.064	0.012	0.962
SVM chart	0.016	0.012	0.986
T^2 chart	0	0.2	0.90
EM method	0.092	0.016	0.916

5.4 APPLICATION TO COMPUTER NETWORK ANOMALY DETECTION

In this section, we apply the SVC based control chart for anomaly detection in computer networks. In Phase 1 we use the normal connection data to construct a SVC model on which the SVC based control chart is built, and in Phase 2 we will test the performance of this control chart under 4 classes of intrusions.

Same as Chapter 3 and Chapter 4, we use KDD1999 labeled data for constructing the SVC based control chart, and the training and test data size can be found in Table 3.3. We use the normal connection as normal data used in Phase 1 and all other data as abnormal data in Phase 2. Because the sample size in training dataset for normal connection is too small (just 243 samples), we put some test samples into training data, so the actual training sample size is 1752. We use only the 12 selected attributes in both training and testing data.

In Phase 1, we use the 1752 samples (each sample contains 12 attributes) to construct minimal hypersphere to enclose all data in the feature space, the parameter of the model is set as σ=1. After that, we use the rest of the training sample to test if all the training data is inside of the hypershere, i.e. the kernel distance within the radius of the hypersphere. 58841 samples are used to verify this model, and we find the false alarm rate as 3.798%.

After the Phase 1 control limit is decided, we use the control chart to test on different connection data (DoS, U2R, R2L and Probe). Figure 5.17 to Figure 5.20 show the performance on each intrusion type individually. It can be found that for DoS and U2R connection, the SVC control chart can have 100% detection rate. It has about 45% missing detection (β error) for Probe connection. It means the Probe has very similar signatures with normal connection, so it is hard to detect by SVC chart. It is better for the detection of Probe to collect enough samples and use supervised SVM control chart. Also it is worthwhile to investigate the pattern for the miss-detection. It looks like those miss-detected samples are in several clusters, so we can study the conditions that the data are collected for further conclusion of those miss-detections. All the results are put inside Table 5.3.

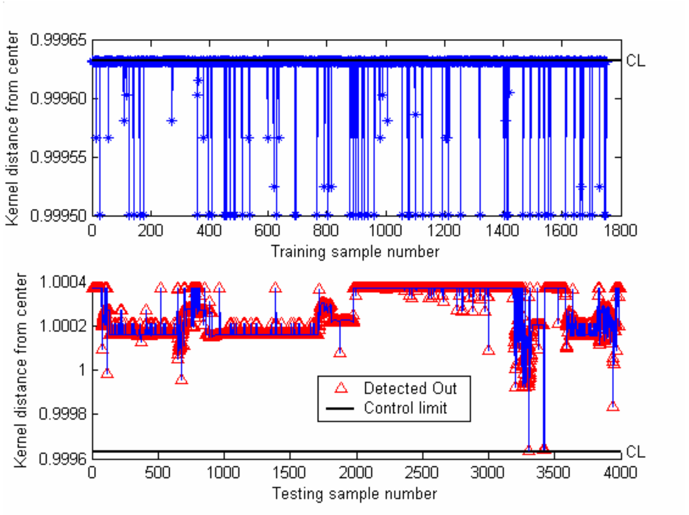


Figure 5.17 SVC control chart for DoS detection

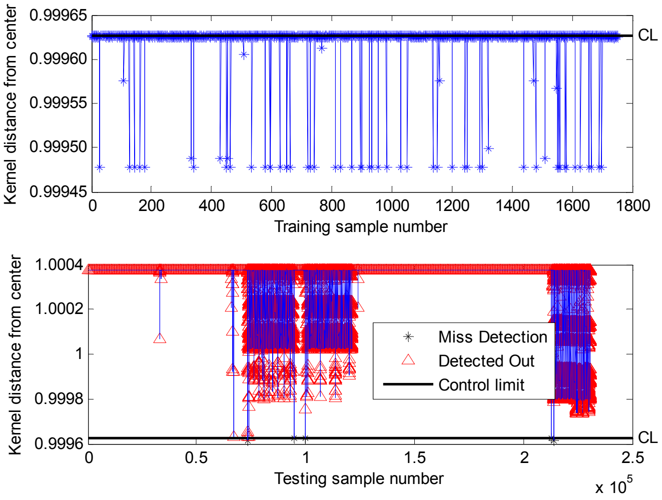


Figure 5.18 SVC control chart for U2R detection

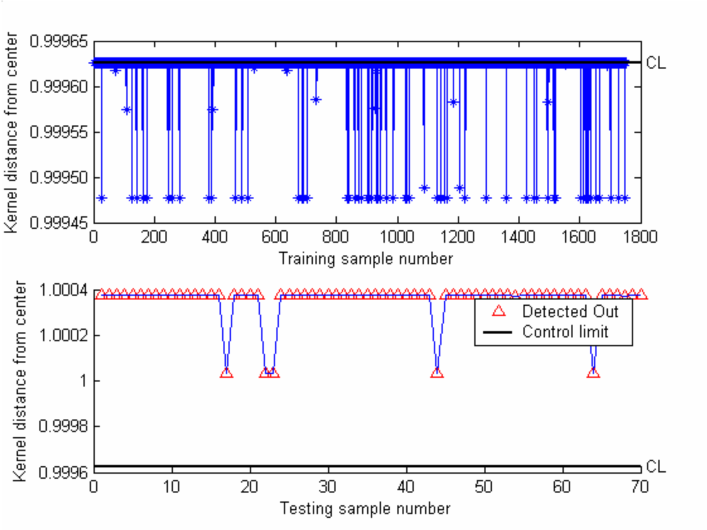


Figure 5.19 SVC control chart for R2L detection

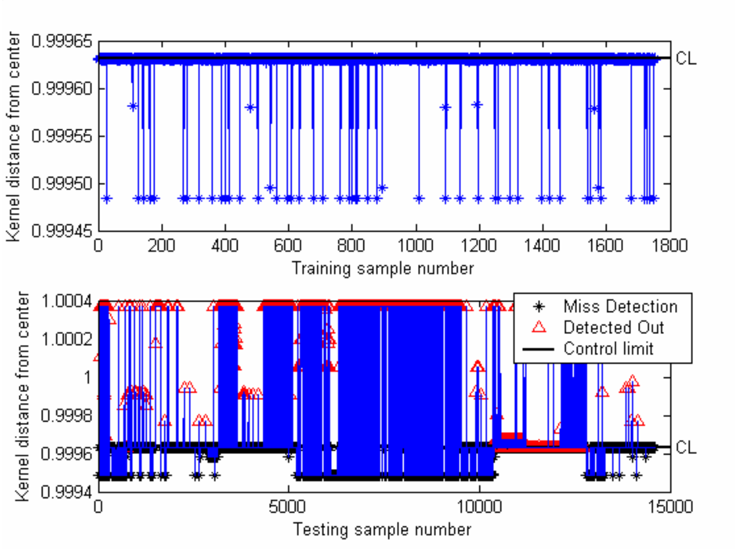


Figure 5.20 SVC control chart for Probe detection

Table 5.3 Performance of Phase 2 control chart

	Phase 2			
	DoS	U2R	R2L	Probe
β error (Phase 2)	0	0.002%	0	55.65%
Training sample size	1752	1752	1752	1752
Testing sample size	3999	230760	70	14597

5.5 CONCLUSION

In this chapter we formulate two types of unsupervised kernel based control chart, one is KPCA control chart, which is a technique to find PCA in feature space instead of original space in order to deal with the situation that data itself can not be linearly separated by the eigenvector. The other is SVC control chart, which is to transform the originally non-Gaussian distributed data into a feature space, then find a closed hypersphere around the data by an optimization problem. After Phase 1 training, the center and radius can be used for Phase 2 testing data. The unsupervised kernel based control charts are used when the out-of-control samples are absent. For the purpose of performance evaluation, we use several cases to compare the performance of SVC control chart with T^2 chart, SVM chart and EM methods.

The following conclusions can be drawn from this chapter:

- SVC control chart is suitable for the situation that there is only in-control data available. SVC control chart construct a closed sphere in the feature space in Phase 1, then use the minimal radius from the center as the control limit for the testing data in Phase 2.
- By fine-tuning of parameters σ and C we can control the boundary of the enclosing hypersphere, and also SVC control chart has the capability to exclude outliers from the training data.
- The comparison of SVC control chart with SVM control chart, T^2 chart and EM method shows that SVC control chart has good performance on both Gaussian data and non-Gaussian distribution data.

- The application on computer networks anomaly detection shows that SVC control chart has good performance. SVC control chart does not work well on Probe data, but it can still be found that the miss-detection samples and the patterns can be further analyzed.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The intrusion detection in computer networks is a complex research problem, which requires the understanding of computer networks and the mechanism of intrusions, the configuration of sensors and the collected data, the selection of the relevant attributes, and the monitor algorithms for online detection.

This dissertation mainly focuses on developing statistics-based machine learning methods for anomaly detection of computer network intrusions. The critical problems of how to reduce data dimension and improve monitoring performance are addressed in detail. More specifically, several research issues have been investigated and the respective contributions are summarized as follows:

(1) *Filter and Wrapper models are integrated to extract a small number of the informative attributes for computer network intrusion detection.* A two-phase analyses method is proposed for the integration of Filter and Wrapper models. In Phase I, Filter model based on data correlation is firstly applied to reduce the dimension of attributes by removing the irrelevant attributes. This is followed by Phase II, where a wrapper model is employed to further extract most important attributes relevant to the intrusion classes without redundancy. The performance of each step in Phase I and II is examined to illustrate the effectiveness of the joint model. We apply correlation based filter model in Phase I and GA based attribute selection model in Phase II. Multiclass SVM is employed

as the learning algorithm embedded in GA based attribute selection model and minimal output coding (MOC) is applied for high computing efficiency.

One significant aspect of this dissertation describes how to monitor a large amount of computer network audit data. The attribute selection algorithm we proposed addresses one of the key problems on data dimension reduction. The proposed method has successfully reduced the original 41 attributes to 12 informative attributes while increasing the accuracy of the model. The comparison of the results in each phase shows the effectiveness of the proposed method.

(2) *Supervised kernel based control charts for anomaly intrusion detection.* Different from conventional methods that construct monitoring control charts in the original multivariate data space, we propose to construct control charts in a feature space. For instance, multivariate control charts based on the normal distribution (e.g. T^2 chart) would result in misleading solutions if data distributions are not following normal distribution. Kernel-based control charts, which are constructed based on non-parametric methods, map the data from an original space to a feature space, then use a hyperplane to separate in-control and out-of-control samples with a maximal margin. When there is an overlap between the in-control and out-of-control samples, a more complex objective function is formed to make the trade-off between maximizing the margin and minimizing the violation caused by the overlap of those two categories. Kernel-based control charts have the advantage of not requiring the specific distribution of data, thus have the great potential to be applied widely.

There are two important contributions in this part. The first contribution is the use of multi-objective Genetic Algorithm in the parameter pre-selection for SVM based control charts. With the fine-tuning of pre-selected soft margin constant C and RBF kernel parameter σ, we can obtain an optimal combination of Pareto optimal α and β errors. The second contribution is the performance evaluation of supervised kernel based control charts. We design several scenarios using different types of data (i.e. Gaussian data, mixutre data and non-Gaussian distribution data) to evaluate kernel-based control chart and to compare its performance against the T^2 chart and the EM method.

(3) *Unsupervised kernel based control charts for anomaly intrusion detection.* When only in-control data (e.g. normal connection data in intrusion detection) is available, unsupervised kernel based control charts are proposed. Two types of unsupervised kernel based control charts are investigated: Kernel PCA control charts and Support Vector Clustering based control charts. The applications of SVC based control charts on computer networks audit data are also discussed to demonstrate the effectiveness of the proposed method.

Although the developed methodologies in this dissertation are demonstrated in the computer network intrusion detection applications, the methodologies are also expected to be applied to other complex system monitoring, where the database consists of a large dimensional data with non-Gaussian distribution.

6.2 FUTURE WORK

Developing effective monitoring algorithms for computer network intrusion detection is a new and challenging research area. There are many remaining research issues, both in theory and in practice, that need to be further investigated in the future. For example:

(1) Algorithms incorporating the detection of network traffic pattern change and adaptive model parameter estimation (e.g. using EWMA method or time series method).

(2) Agent based intrusion detection and control. Newly developed multi-agent systems could be applied to collect real-time operation information of networks and take proactive actions cooperatively to reduce the impact of attacks.

(3) Network based intrusion detection. Consideration of the correlation between different nodes in the detection model. Because of the propagation of computer network attacks, it is important to monitor the whole network instead of single host machine to obtain propagation patterns of attacks. Spatio-temporal analysis is potential method to incorporate both spatial and temporary data change into one model patterns.

(4) Research on combining statistics-based anomaly detection and knowledge-based misuse detection rules to create highly efficient and accurate algorithms for real world problems.

APPENDIX A COMPLETE LISTING OF THE SET OF FEATURES DEFINED FOR THE CONNECTION RECORDS [50]

Table 1: Basic features of individual TCP connections

feature name	*description*	*type*
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of ``wrong'' fragments	continuous
urgent	number of urgent packets	continuous

Table 2: Content features within a connection suggested by domain knowledge

feature name	*description*	*type*
hot	number of ``hot'' indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of ``compromised'' conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if ``su root'' command attempted; 0 otherwise	discrete
num_root	number of ``root'' accesses	continuous

num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the ``hot" list; 0 otherwise	discrete
is_guest_login	1 if the login is a ``guest"login; 0 otherwise	discrete

Table 3: Traffic features computed using a two-second time window

feature name	*description*	*type*
count	number of connections to the same host as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-host connections.	
serror_rate	% of connections that have ``SYN" errors	continuous
rerror_rate	% of connections that have ``REJ" errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-service connections.	
srv_serror_rate	% of connections that have ``SYN" errors	continuous
srv_rerror_rate	% of connections that have ``REJ" errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

APPENDIX B MULTI-OBJECTIVE GENETIC ALGORITHM FORMULA AND FLOWCHART

(1) Selection

Ranking method is used for selection. This probabilistic selection method is performed based upon the individual's fitness ranking such that the better individuals have an increased chance of being selected.

Define P_i is the probability of *i*th individual to be selected

$$P_i = q'(1-q)^{r-1}$$

where q - probability of selection on best individual;

r - ranking of individual, the best individual has ranking 1, then 2, 3, etc;

q' - $q' = \dfrac{q}{1-(1-q)^P}$

P - sample size of the chromosome.

Crossover for real number (floating point number) can be as following:

- Simple Crossover

$$x_i' = \begin{cases} x_i & if\ i < r \\ y_i & otherwize \end{cases}$$

$$y_i' = \begin{cases} y_i & if\ i < r \\ x_i & otherwize \end{cases}$$

where r is a uniform distributed random number within (0, 1)

- Arithmetic Crossover

$$\overline{X}' = r\overline{X} + (1-r)\overline{Y}$$

$$\overline{Y}' = (1-r)\overline{X} + r\overline{Y}$$

(2) Mutation

- uniform mutation

$$x_i' = \begin{cases} U(a_i, b_i) & \mathit{if}\ i = j \\ x_i & \mathit{otherwise} \end{cases}$$

- non-uniform mutation

$$x_i' = \begin{cases} x_i + (b_i - x_i) f(G) & \mathit{if}\ r_1 < 0.5 \\ x_i + (b_i - x_i) f(G) & \mathit{if}\ r_1 \geq 0.5 \\ x_i & \mathit{otherwise} \end{cases}$$

where

$$f(G) = (r_2(1 - \frac{G}{G_{\max}}))^b$$

r_1, r_2 - uniform random number in (0,1);

G - current generation；

$G_{\max}$ - maximum generation；

b - predefined parameter.

- boundary mutation

$$x_i' = \begin{cases} a_i & \mathit{if}\ i = j, r < 0.5 \\ b_i & f\ i = j, r \geq 0.5 \\ x_i & \mathit{otherwise} \end{cases}$$

(3) Ranking

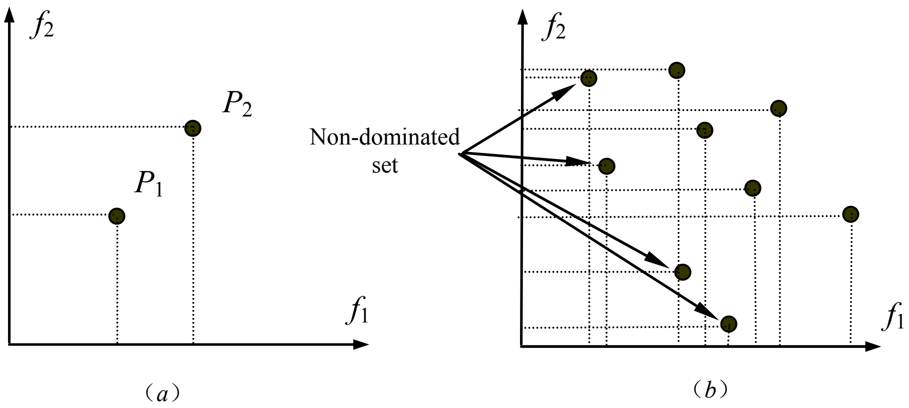


Figure 1 Nondominated Points for Minimize Problem

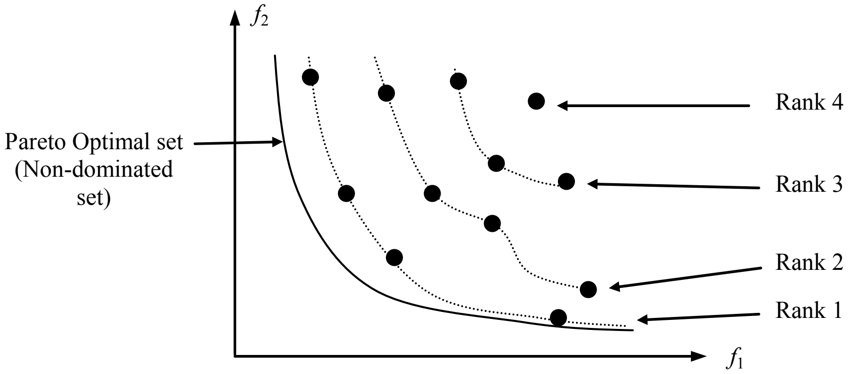


Figure 2. Population Rank (Two Objectives)

Where the ranking is finished, the fitness value of each individual is obtained by

$$F_i = (N_r - i + 1)/SS$$

$$SS = \sum_{i=1}^{N_r}(N_r - i + 1)P_{Si} \Big/ M$$

where M – population size

N_r – total ranking number of the population

P_{Si} - number in population with rank i

F_i - fitness value of members with rank i

(4) Filter

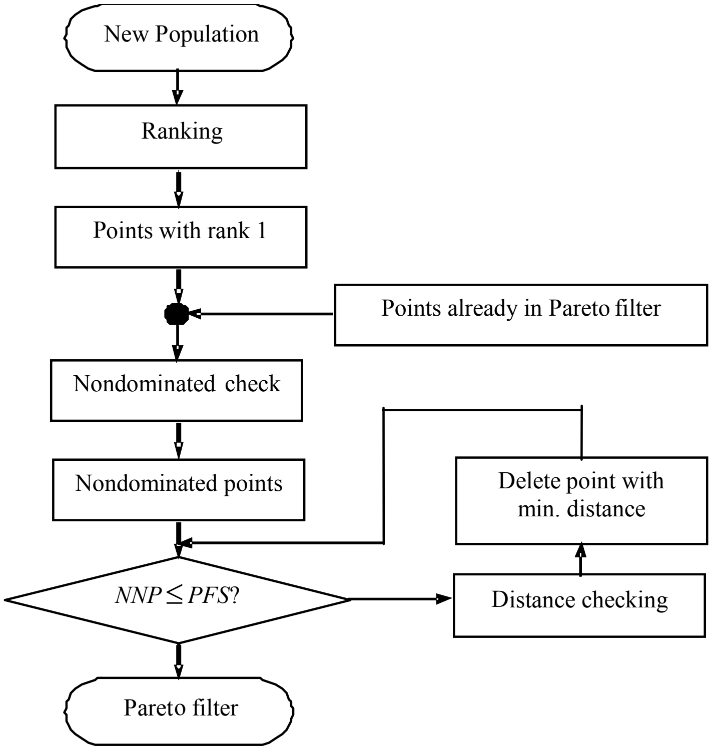


Figure 3. Pareto Set Filter Operation

Where NNP is the number of non-dominated points after the nondominated check, PFS is the sample size of Perato set filter.

(5) Niche

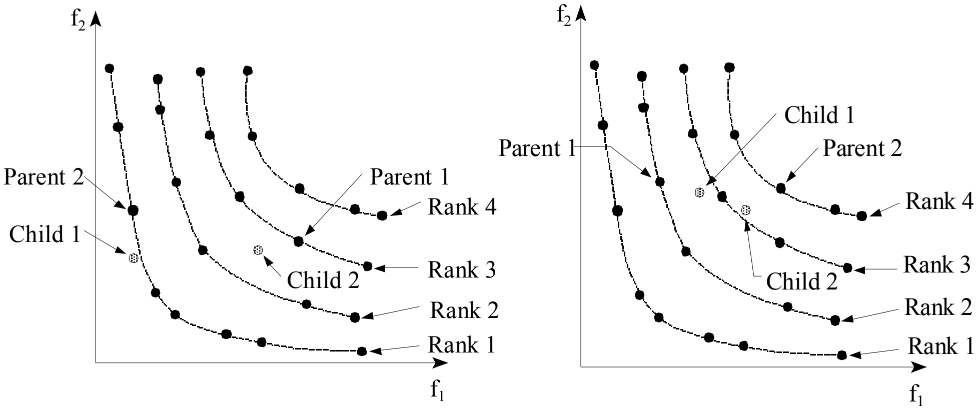


Figure 4. Illustration of Niche Technology

Parent 1 + Parent 2 → Child 1 and Child 2

```
Parent.Rank = min(Parent1.Rank, Parent2.Rank)
Child.Rank = min(Child1.Rank, Child2.Rank)
Test = Child.Rank ≤  Parent.Rank ?
New.Child 1 = if (test) Child 1 else Parent 1
New.Child 2 = if (test) Child 2 else Parent 2
```

(6) Multiple individuals crossover

The core operator in genetic algorithm is crossover. Because genetic algorithm using float representation is found to be superior to binary genetic algorithm in terms of efficiency and quality of solution, this dissertation uses float representation in multiple individual crossover. The way presented in this dissertation to crossover is that *b* parental individual take part in crossover every crossover and get *b* new individuals. Three crossover operators are developed in reference [1], namely simple crossover, arithmetic crossover and heuristic crossover. In this dissertation only arithmetic crossover is developed to deal with multiple crossover because the effect of arithmetic crossover is the best among the three ones. It is named multiple individual arithmetic crossover.

The principle of arithmetic crossover is that the two new individuals are linear combination of two parental ones, i.e., it produces a uniform distribution from 0 to 1 from which a random number r is selected, and then two parental individuals x_i and y_i are operated as follows:

$$\overline{X}'=r\overline{X}+(1-r)\overline{Y}$$

$$\overline{Y}'=(1-r)\overline{X}+r\overline{Y}$$

Multiple individual arithmetic crossover can be derived from above. Corresponding to the *b* parental individuals $\overline{X}_i \, (1 \le i \le b)$, *b* random numbers are selected from a uniform distribution ranged from 0 to 1, and then we get b new individuals $\overline{X}_i'$ as follows:

$$\overline{X}_i' = \sum_{j=1}^{b} r_j \overline{X}_i \Big/ b$$

The possible number of random sequence of number r_1, r_2, …, r_{b} can be infinitive, so new individuals can traverse the whole design space. From it we can see that the algorithm presented in this dissertation can increase the diversity of solutions, and then increase the calculating efficiency.

(7) Floatchart for the whole Pareto Multi-objective genetic algorithm

Please refer to Figure 4.2.

APPENDIX C SVM PARAMETERS OBTAINED BY SIMULATION (1-D AND 2-D)

(1) 1-D problem

Linear kernel: $k(x, x_j) = x_j^T x$

$$y(x) = \sum_{j=1}^{n} \alpha_j y_j k(x, x_j) + b$$

$$= \sum_{j=1}^{nsv} \alpha_j y_j x_j{}^T x + b$$

$$b = \frac{1}{n_{SV}} \sum_{i=1}^{n_{SV}} \{y_i - \sum_{j=1}^{n_{SV}} \alpha_j y_j k(x_i, x_j)\}$$

where x_l is an example which is non-bound support vector (i.e. $0 < \alpha_j < C$), n_{SV} is the number of support vectors.

Margin = $\frac{1}{\sqrt{w^T w}}$

(2) 2-D problem

Kernel function for two dimensions:

RBF kernel: $k(x, x_j) = \exp\{-\left\|x - x_j\right\|^2 / (2\sigma^2)\}$

$x = [x_1, x_2, \ldots, x_\mathrm{n}]^\mathrm{T}$ & $x_\mathrm{i} = [x_{i1}\ x_{i2}]$

kernel matrix K [n x n]

K(i,j) = k(x(:,i),x(:,j)) for all i=1..n, j=1..n

$$y(x) = \sum_{j=1}^{n} \alpha_j k(x, x_j) + b$$

$$= \sum_{j=1}^{nsv} \alpha_j x_j{}^T x + b$$

$$b = \frac{1}{n_{SV}} \sum_{j=1}^{n_{SV}} y_j (1 - \frac{\tau_j}{2C}) - \alpha_j k(x_j, x_j) \qquad \text{(L2 soft margin)}$$

where x_l is an example which is non-bound support vector (i.e. $0 < \alpha_j < C$),

nsv is the number of support vectors.

$$\tau_j = \begin{cases} \alpha_j & \text{if } y_j = 1 \\ -\alpha_j & \text{if } y_j = -1 \end{cases}$$

$$\xi_j = \frac{\tau_j}{2C}$$

APPENDIX D THEORETICAL SOLUTION FOR α AND β ERROR FOR 1-D CONTROL CHART WITH BOTH IN-CONTROL AND OUT-OF-CONTROL DATA

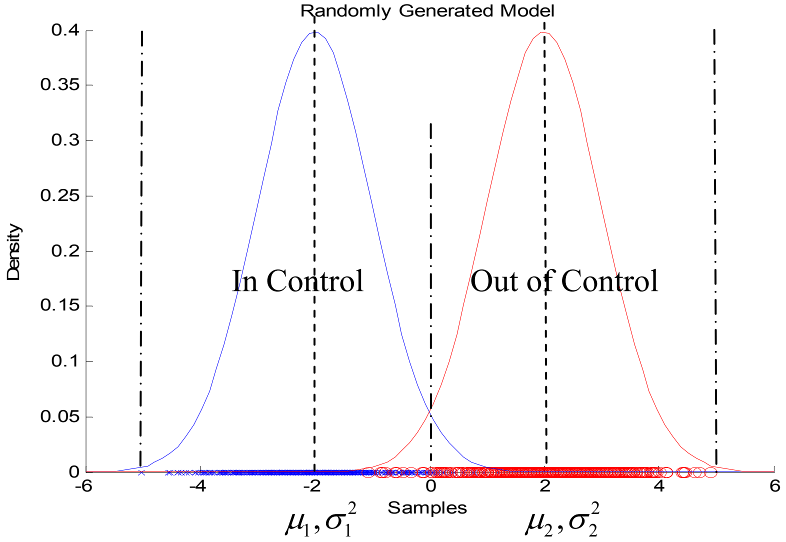


Figure 1. Two classes of Gaussian distribution

$$
\begin{aligned}
\alpha &= \Pr(x \text{ is out of control}|x \text{ is in control}) \\
&= \Pr(x \geq UCL \mid \mu = \mu_1, \sigma^2 = \sigma_1^2) \\
&= 1 - \Phi(\frac{UCL - \mu_1}{\sigma_1 / \sqrt{n}}) \\
&= 1 - \Phi(\frac{\mu_2 + L\sigma_2 - \mu_1}{\sigma_1 / \sqrt{n}}) \\
&= 1 - \Phi(\frac{\mu_2 - \mu_1 + L\sigma_2}{\sigma_1 / \sqrt{n}})
\end{aligned}
$$

$$
\begin{aligned}
\beta &= \Pr(x \text{ is in control}|x \text{ is out of control}) \\
&= \Pr(x \le UCL \mid \mu = \mu_2, \sigma^2 = \sigma_2^2) \\
&= \Phi(\frac{UCL - \mu_2}{\sigma_2 \big/ \sqrt{n}}) \\
&= \Phi(\frac{\mu_1 + L\sigma_1 - \mu_2}{\sigma_2 \big/ \sqrt{n}}) \\
&= \Phi(\frac{\mu_1 - \mu_2 + L\sigma_1}{\sigma_2 \big/ \sqrt{n}})
\end{aligned}
$$

where $x_i \sim N(\mu_i, \sigma_i^2 / n_i)$, $UCL_i = \mu_i + L\sigma / \sqrt{n}$, $LCL_i = \mu_i - L\sigma / \sqrt{n}$, i = 1, 2. n is sample size. If we only consider the mean shift or variance change, it will go to traditional framework of type I or type II errors. We keep the μ_1, σ_1^2, μ_2, σ_2^2 individually to remain the potential that both mean and variance will change together.

REFERENCES

1. Jain, A.K., R.P.W. Duin, and J. Mao, *Statistical pattern recognition: a review.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. 22(1): p. 4-37.
2. Koontz, W.L.G. and K. Fukunaga, *A nonlinear feature extraction algorithm using distance transformation.* IEEE Transactions on Computing, 1972. 21(1): p. 56-63.
3. Fukunaga, K. and R. Short, *Nonlinear feature extraction with a general criterion function.* IEEE Transactions on Information Theory, 1978. 24(5): p. 600-607.
4. Park, C.H. and H. Park, *Nonlinear feature extraction based on centroids and kernel functions.* Pattern Recognition, 2003. 37(4): p. 801-810.
5. Mao, J. and A.K. Jain, *Artificial neural networks for feature extraction and multivariatedata projection.* IEEE Transactions on Neural Networks, 1995. 6(2): p. 296-317.
6. Kocsor, A. and L. Toth, *Kernel-based feature extraction with a speech technology application.* IEEE Transactions on Signal Processing, 2004. 52(8): p. 2250-2263.
7. Kudo, M. and J. Sklansky, *Comparison of algorithms that select features for pattern classifiers.* Pattern Recognition, 2000. 33: p. 25-41.
8. Sebban, M. and R. Nock, *A hybrid filter/wrapper aproach of feature selection using information theory.* Pattern Recognition, 2002. 35: p. 835-846.
9. Markou, M. and S. Singh, *Novelty detection: a review - part 1: statistical approaches.* Signal Processing, 2003a. 83: p. 2481-1497.
10. Markou, M. and S. Singh, *Novelty detection: a review - part 2: neural network based approaches.* Signal Processing, 2003b. 83: p. 2499-2521.
11. Jin, J. and J. Shi, *Feature-preserving data compression of stamping tonnage information using wavelets.* Technometrics, 1999. 41(4): p. 327-339.

12. Jin, J. and J. Shi, *Automatic feature extraction of waveform signals for in-process diagnostic performance improvement.* Journal of Intelligent Manufacturing, 2001. 12(3): p. 257-268.

13. Montgomery, D.C., *Introduction to Statistical Quality Control*, ed. t. ed. 2000, New York, NY.: Wiley.

14. Lowry, C.A. and D.C. Montgomery, *A review of multivariate control charts.* IIE Transactions, 1995. 27: p. 800-810.

15. Vapnik, V.N., *Statistical learning theory*. 1998, New York ; Chichester [England]: Wiley. xxiv, 736 p.

16. Vapnik, V.N., *The nature of statistical learning theory*. 2nd ed. 2000, New York: Springer. xix, 314 p.

17. Scholkopf, B., et al., *Input space vs. feature space in kernel-based methods.* IEEE Transactions on Neural Networks, 1999. 10(5): p. 1000-1017.

18. Lawson, A.B. and D. Denison, *Spatial cluster modelling*. 2002, Boca Raton, Fla. ; London: Chapman & Hall/CRC.

19. Lawson, A.B., *Issues in the Spatio-Temporal Analysis of Public Health Surveillance Data*, in *Monitoring the Health of Populations:Statistical Methods for Public Health Surveillance*, R. Brookmeyer and D. Stroup, Editors. 2003, Oxford University Press.

20. Clark, A.B. and A.B. Lawson, *Spatio-temporal clustering of small area health data*, in *Spatial Cluster Modelling*, A.B. Lawson and D. Denison, Editors. 2002, Chapman & Hall: London.

21. Kulldorff, M., *Prospective time periodic geographical disease surveillance using a scan statistic.* Journal of the Royal Statistical Society Series A, 2001. 164(1): p. 61-72.

22. Verbeke, G. and G. Molenberghs, *Linear mixed models in practice : a SAS-oriented approach*. 1997, New York: Springer. xiii, 306 p.

23. Verbeke, G. and G. Molenberghs, *Linear mixed models for longitudinal data*. 2000, New York: Springer. xxii, 568 p.

24. Kendall, K., *A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*, in *C.S.* 1998, Massachusetts Institute of Technology: Boston.

25. Lazarevic, A., J. Srivastava, and V. Kumar. *Data Mining for Intrusion Detection*. in *Tutorial at the Pacific-Asia Conference on Knowledge Discovery in Databases*. 2003. Seoul.

26. Lippmann, R. and J.W. Haines. *Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation, in Recent Advances in Intrusion Detection*. in *Proc. Third International Workshop RAID*. 2000a.

27. Lippmann, R., et al., *The 1999 DARPA Off-Line Intrusion Detection Evaluation.* Computer Networks, 2000b. 34(4): p. 579-595.

28. Ye, N., *Modeling and analysis of cyber-security data*. 2003, London: Springer-Verlag.

29. Stevens, W.R., *TCP/IP illustrated (Vol. 1)*. 1994, Boston: Addison-Wesley.

30. Northcutt, S., et al., *Intrusion signatures and analysis*. 2001, Indianaplolis, IN: New Riders.

31. Ye, N., et al., *Multivariate statistical analysis of audit trails for host-based intrusion detection.* IEEE Transactions on Computers, 2002. 51: p. 810-820.

32. Ye, N., X. Li, and S.M. Emra. *Decision tree for signature recognition and state classification*. in *Information Assurance and Security Workshop of IEEE Systems, Man, and Cybermetics*. 2000. West Point, USA.

33. Scott, S.L., *Detecting network intrusion using a markov modulated nonhomogeneous poisson process*, in *http://www-rcf.usc.edu/~sls/fraud.ps*. 2002.

34. Ye, N., et al., *Probablistic techniques for intrusion detection based on computer audit data.* IEEE Transactions on Systems, Man, and Cybernetics, 2001. 31: p. 266-274.

35. Base, T., *Intrusion detection systems and multisensor data fusion.* Communications of the ACM, 2000. 43(4): p. 99-106.

36. David, L.H. and L. James, *Handbook of Multisensor Data Fusion*. 2001: CRC Press.

37. Spafford, E.H. and D. Zamboni, *Intrusion detection using autonomous agents.* Computer Networks, 2000. 34(4): p. 553-570.

38. Hegazy, I.M., et al., *A multi-agent based system for intrusion detection.* IEEE Potentials, 2003. 22(4): p. 28-31.

39. Harmer, P.K., et al., *An artificial immune system architecture for computer security applications.* IEEE Transactions on Evolutionary Computation, 2002. 6(3): p. 252-280.

40. Gorodetski, v. and I. Kotenko. *The multi-agent systems for computer network security assurance: frameworks and case studies*. in *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02)*. 2002.

41. Gorodetski, V., I. Kotenko, and O. Karsaev, *Multi-agent technologies for computer network security: attack simulation, intrusion detection and intrusion detection learning.* International journal of Computer Systems Science & Engineering, 2003. 18(4): p. 191-200.

42. Akyildiz, I.F., et al., *Wireless sensor networks: a survey.* Computer Networks, 2002. 38: p. 393-422.

43. Samfat, D. and R. Molva, *IDAMN: an intrusion detection architecture for mobile networks.* IEEE Journal on Selected Areas in Communications, 1997. 15(7): p. 1373-1380.

44. Mishra, A., K. Nadkarni, and A. Patcha, *Intrusion detection in wireless ad hoc networks.* IEEE Wireless Communications, 2004. 11(1): p. 48-60.

45. Debar, H., M. Dacier, and A. Wepsi, *A Revised Taxonomy for Intrusion-Detection Systems*, in *IBM Research Report*. 1999.

46. Debar, H., M. Dacier, and A. Wespi, *Towards a taxonomy of intrusion-detection systems.* Computer Networks, 1999a. 31: p. 805-822.

47. Denning, D., *An intrusion-detection model.* IEEE Transactions on Software Engineering, 1987. 13(2).

48. Debar, H., M. Dacier, and A. Wepsi, *A Revised Taxonomy for Intrusion-Detection Systems*, in *IBM Research Report*. 1999b.

49. Escamilla, T., *Intrusion detection: Network security beyond the firewall*. 1998, New York: Wiley.

50. Lee, W., *A data mining framework for constructing features and models for intrusion detection systems*, in *CS*. 1999, Columbia University.

51. Caswell, B., et al., *Snort 2.0 Intrusion Detection*. 2003, Rockland, MA: Syngress Publishing.

52. Lee, W. and S.J. Stolfo, *A framework for constructing features and models for intrusion detection systems.* ACM Transactions on Information and System Security, 2000. 3(4).

53. Lee, W. and D. Xiang. *Information-theoretic measures for anomaly Detection*. in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*. 2001.

54. Ilgun, K., R.A. Kemmerer, and P.A. Porras, *State Transition Analysis: A Rule-Based Intrusion Detection Approach.* IEEE Transactions on Software Engineering, 1995. 21(3).

55. Luo, J. and S.M. Bridges, *Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection.* International Journal of Intelligent Systems, 2000. 15(8): p. 687-703.

56. Fayyad, U., G. Piatetsky-Shapiro, and P. Smyth, *The KDD process for extracting useful knowledge from volumes of data.* Communications of the ACM, 1996. 39(11): p. 27-34.

57. Cai, Z., et al., *A rough set theory based method for anomaly intrusion detection in computer network systems.* Expert Systems, 2003. 20(5): p. 251-260.

58. Paxson, V., *Bro: A System for Detecting Network Intruders in Real-Time.* Computer Networks, 1999. 31: p. 23-24.

59. Stolfo, S.J., et al. *Behavior Profiling of Email*. in *1st NSF/NIJ Symposium on Intelligence & Security Informatics(ISI 2003)*. 2003. Tucson,Arizona,USA.

60. Stolfo, S.J., et al., *Detecting Viral Propagations Using Email Behavior Profiles*, in *CU Tech Report*. 2003.

61. Axelsson, S., *Intrusion Detection Systems: A Survey and Taxonomy*. 2000, Dept. of Computer Engineering, Chalmers University of Technology, Sweden.

62. NG, A.Y. *On feature selection: learning with exponentially many irrelevant features as training examples*. in *Proc. 15th Intl. Conf. on Machine Learning*. 1998.

63. Narendra, P. and K. Fukunaga, *A branch and bound algorithm for feature subset selection.* -922IEEE transactions on Computers, 1977. 26(9): p. 917.

64. Kohavi, R. and G. John, *Wrappers for feature subset selection.* Artificial Intelligence, 1997. 97: p. 273-324.

65. Vafaie, H. and K. DeJong. *Robust feature selection algorithms*. in *Proc. 5th Intl. Conf. on Tools with Artificial Intelligence*. 1993. Rockville, MD.

66. Yang, J. and V. Honaver, *Feature Subset Selection Using A Genetic Algorithm.* IEEE Intelligent Systems, 1998.

67. Etxeberria, R., et al., *Feature subset selection by bayesian network-based optimization.* Artificial Intelligence, 2000. 123: p. 157-184.

68. Skalak, D.B. *Prototype and feature selection by sampling and random mutation hill-climbing algorithms*. in *Proc. 11th Intl. Conf. on Machine Learning*. 1993. New Brunwick, NJ.

69. Das, S. *Filters, wrappers and a boosting-based hybrid for feature slection*. in *Proc. 18th Intl. Conf. on Machine Learning*. 2001.

70. Almuallim, H. and T.G. Dietterich. *Efficient algorithms for identifying relevant features*. in *Proceedings of the ninth Canadian Conference on Artificial Intelligence*. 1992: Morgan Kaufmann.

71. Koller, D. and M. Sahami. *Towards optimal feature selection*. in *Machine Learning: Proceedings of the thirteenth International Conference*. 1996: Morgan Kaufmann.

72. Holmes, G. and C.G. Nevil-Manning. *Feature selection via the discovery of simple classification rules*. in *Proceedings of the International symposium on Intelligent Data Analysis*. 1995.

73. Hall, M., *Correlation based feature selection for machine learning*, in *Dept. of Computer Science*. 1999, University of Waikato.

74. Mukkamala, S. and A.H. Sung. *A comparative study of techniques for intrusion detection*. in *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03)*. 2003.

75. Haines, J.W., et al., *1999 DARPA Intrusion Detection Evaluation: Design and Procedures*, in *technical report TR-1062*. 2001, MIT Lincoln Laboratory.

76. Alt, F.B., *Multivariate quality control*, in *The Encyclopedia of Statistical Sciences*. 1984. p. 110-122.

77. Chakraborti, S., P. van de Laan, and S.T. Bakir, *Nonparametric control charts: an overview and some results.* Journal of Quality Technology, 2001. 33(3): p. 304-315.

78. Chakraborti, S., P. van der Laan, and M.A. van de Wiel, *A class of distribution-free control charts.* Applied Statistics, 2004. 53(3): p. 443-462.

79. Vermaat, M.B., et al., *A comparison of schewhart individuals control charts based on normal, non-parametric, and extreme-value theory.* Quality and Reliability Engineering International, 2003. 2003(19).

80. Vapnik, V., *Statisitcal Learning Theory*. 1998: Wiley.

81. Muller, K.-R., et al., *An introduction to kernel-based learning algorithms.* IEEE Transactions on Neural Networks, 2001. 12(2): p. 181-201.

82. Hastie, T., R. Tibshirani, and J.H. Friedman, *The elements of statistical learning : data mining, inference, and prediction*. 2001, New York: Springer. xvi, 533 p.

83. Zhu, X., *Pareto multiobjective genetic algorithm with multiple-chromosomes crossover.* Acta Electronica Sinica, 2001. 29(1): p. 106-109.

84. Ben-Hur, A., et al., *Support vector clustering.* Journal of Machine Learning Research, 2001. 2: p. 125-137.

85. Tax, D.M.J. and R.P.W. Duin, *Support vector data description.* Machine Learning, 2004. 54: p. 45-66.

86. Scholkopf, B., et al., *Estimating the Support of a High-Dimensional Distribution.* Neural Computation, 2001. 13(7): p. 1443-1471.