جامعة الشرق الأوسط
MEU MIDDLE EAST UNIVERSITY

Comparative Study between (SVM) and (KNN) Classifiers after adding (PCA) to Improve of Intrusion Detection System

دراسة مقارنه بين مصنفات آلة دعم المتجهات (SVM) والجار الاقرب (KNN) بعد إضافة تحليل
المكونات الرئيسية(PCA) لتحسين نظام كشف التسلل

Preparation

Nafea Ali Majeed Alhammadi

Student Number: (401220073)

Supervisor

Dr.Sadeq AlHamouz

A Thesis Submitted in Partial Fulfillment of the Requirements for Master Degree in Computer Information Systems

Department of Computer Information Systems

Faculty of Information Technology

Middle East University

Amman, Jordan

January 2016

بسم الله الرحمن الرحيم

(اللَّهُ لَا إِلَٰهَ إِلَّا هُوَ الْحَيُّ الْقَيُّومُ ۚ لَا تَأْخُذُهُ سِنَةٌ وَلَا نَوْمٌ ۚ لَهُ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ ۗ مَنْ
ذَا الَّذِي يَشْفَعُ عِنْدَهُ إِلَّا بِإِذْنِهِ ۚ يَعْلَمُ مَا بَيْنَ أَيْدِيهِمْ وَمَا خَلْفَهُمْ ۖ وَلَا يُحِيطُونَ بِشَيْءٍ مِنْ عِلْمِهِ إِلَّا
بِمَا شَاءَ ۚ وَسِعَ كُرْسِيُّهُ السَّمَاوَاتِ وَالْأَرْضَ ۖ وَلَا يَئُودُهُ حِفْظُهُمَا ۚ وَهُوَ الْعَلِيُّ الْعَظِيمُ)

صدق الله العظيم

سورة البقرة / اية (255)

Authorization Statement

I, Nafea Ali Majeed Al-Hammadi, authorize Middle East University to supply copies of my thesis to libraries, establishments or individuals upon their request, according to the university regulations.

Signature: ....

Date: 3 / 1 / 2016

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled " **Comparative Study between (SVM) and (KNN) Classifiers after adding (PCA) to Improve of Intrusion Detection System**" was successfully defended and approved in 3 / 1 /2016.

Examination Committee MembersSignature

Dr. SadeqAlHamouz (Supervisor & Member)
Associate Professor, Department of Computer Science
Head Department of Computer Information Systems
Middle East University (MEU)

Dr. Ahmed Abu Shariha (Chairman)
Associate Professor, Department of Computer Science
Middle East University (MEU)

Dr. Mohammed Al-Alkasasbeh (External Member)
Associate Professor.
Head Department of Computer Information Systems
Mutah University

Acknowledgment

I would like to thank Dr.Sadeq AlHamouzfor his supervision and cooperation all through this thesis and all my doctors who helped me to enhance my knowledge and skills

My great thanks for my family and friends

Dedication

I like to get this opportunity to donate this project for my Parents and friends for their precious support in my life.

May God bless them

List of Contents

List of Figures

List of Tables

List of Abbreviations

CC	Control Chart
CPU	Central Processing Unit
CR	Classification Rate
DARPA	Defense Advanced Research Projects Agency
DoS	Denial of service
DR	Detection Rate
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GA	Genetic Algorithm
HIDS	Host based Intrusion Detection System
IDS	Intrusion Detection System
KDD	Knowledge Discovery and Data Mining
KNN	K-Nearest Neighbor
LCC	Lower Control Chart
MAC	Media Access Control
MDS	Multi-Dimensional Scaling
MSE	Mean Square Error
NIDS	Network based Intrusion Detection System
NSL	National Security Letter
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
R2L	Remote to Local
RBF	Radial Basis Function
RST	Rough Set Theory
SRM	Structure Risk Minimization
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TCM	Transductive Confidence Machines
TN	True Negative
TP	True Positive
U2R	User to Root
UCC	Upper Control Chart

Student Name

Nafea Ali Majeed Alhammadi

Supervisor

Dr.Sadeq AlHamouz

Abstract

Intrusion Detection Systems (IDSs) are efficient applications that monitor activities of specific network or system to detect any abnormal activity and then send alarms for a defined management station. However, the current IDSs generate a high number of false alarms; False Positives (FP) and False Negatives (FN), which decreases the accuracy of distinguishing attacks from normal activities. Thus, this thesis introduced the implementation of a binary classifier based IDS. The used classifiers within the system were Principal Component Analysis-Support Vector Machine(PCA-SVM) and Principal Component Analysis-K-Nearest Neighbor(PCA-KNN). The performance of the system with using these classifiers was compared using the National Security Letter-Knowledge Discovery and Data Mining(NSL-KDD) dataset to determine the optimal classifier in terms of detection rate and the number of generated false alarms. This was performed based on dividing the dataset into training and testing sets, where the Control Chart was then applied on the training set to improve the results, where it filtered the data to remove the out-bound data and keep the data in the range from Mean-3sigma to Mean+3sigma.

Six evaluation metrics; FP, FN, True Positive (TP), True Negative (TN), Detection Rate (DR) and Classification Rate (CR)were computed for both classifiers for three sets of features; F1: [4,5,10,11,23,24,29,31,33,38,41], F2: [4,5,10,11,23,24,29,31,33] and F3: [4,5,10,11,23,24,29] with and without applying a control chart. The obtained results demonstrated that the PCA-KNN based IDS with control chart offered the best detection rate with minimum number of generated false alarms for sets F2 and F3, while the PCA-SVM based IDS with control chart offered the best detection rate with minimum number of generated false alarms for F1. The average achieved detection rate for the PCA-KNN based IDSwas 98.17% with control chart and 88.7738% without control chart. On the other hand, the average achieved detection rate for the PCA-SVM based IDS was 97.62% with control chart and 96.63587% without control chart. Based on these outcomes, the application of control chart enhancedthe detection rate and decreased the number of false alarms for both classifiers.In addition, the PCA-KNNwas the best classifier to be applied on the IDS with minimum number of false alarms and highest security and detection rate.

Keyword: Support Vector Machine (SVM), K-Nearest Neighbor (PCA), optimal classifier.

دراسة مقارنه بين مصنفات آلة دعم المتجهات (SVM) والجار الاقرب (KNN) بعد إضافة تحليل المكونات الرئيسية(PCA) لتحسين نظام كشف التسلل

إعداد

نافع علي مجيد الحمادي

اشراف

الدكتور صادق الحموز

الملخص

أنظمة كشف التسلل (IDSs)هي تطبيقات فعالة لرصد أنشطة شبكة، أو نظام محدد للكشف عن أي نشاط غير طبيعي، ومن ثم ارسال تنبيهات لمحطة إدارة معرفة. ومع ذلك، فإن النظم الحالية تولد عدد كبير من الانذارات الكاذبة: ايجابيات كاذبة (FP) وسلبيات كاذبة (FN)، مما يقلل من دقة تمييز الهجمات من الأنشطة العادية. لذلك هذه الرسالة قدمت تنفيذ نظام كشف تسلل معتمد على مصنف ثنائي. المصنفات التي تم استخدامها ضمن النظام هي مصنف آلة دعم المتجهات (SVM) والجار الاقرب (KNN) بعد إضافة تحليل المكونات الرئيسية (PCA). تم مقارنة اداء النظام بعد اضافة كل مصنف باستخدام قاعدة البيانات NSL-KDD لتحديد المصنف الأمثل من حيث معدل اكتشاف الهجمات، وعدد الإنذارات الكاذبة التي تم إنشاؤها. اعتمد هذا على تقسيم قاعدة البيانات الى مجموعتين: التدريب والاختبار حيث تم تطبيق مخطط السيطرة على المجموعة التدريبية لتحسين النتائج من خلال تصفية البيانات لإزالة البيانات خارج نطاق معين والحفاظ على البيانات داخل النطاق من Mean-3sigma الى Mean+3sigma.

ستة مقاييس تقييم وهي FP, FN, صحيح إيجابي (TP) صحيح سلبي (TN)، معدل الكشف (DR) ومعدل الصواب (CR) تم حسابها لكل من المصنفين لثلاث مجموعات من الميزات[F1: [4,5,10,11,23,24,29,31,33,38,41 و F2: [4,5,10,11,23,24,29,31,33] و F3: [4,5,10,11,23,24,29] مع وبدون تطبيق مخطط السيطرة. أوضحت النتائج أن النظام المعتمد علىPCA-KNNمع تطبيق مخطط السيطرة وفر أفضل معدل اكتشاف مع الحد الأدنى لعدد الإنذارات الكاذبة للمجموعات الثانية والثالثة، في حين أن النظام المعتمد على PCA-SVM مع تطبيق مخطط السيطرة وفر أفضل معدل اكتشاف مع الحد الأدنى لعدد الإنذارات الكاذبة للمجموعة الاولى. متوسط معدل الاكتشاف للنظام المعتمد على PCA-KNN مع تطبيق مخطط السيطرة كان98.17% وبدون تطبيق مخطط السيطرة كان88.7738%. من ناحية أخرى، فإن

متوسط معدل الاكتشاف للنظام المعتمد على PCA-SVM مع تطبيق مخطط السيطرة كان97.62% وبدون تطبيق مخطط السيطرة كان96.63587%. بناء على هذه النتائج، تطبيق مخطط السيطرة عزز معدل اكتشاف وقلل من عدد الإنذارات الكاذبة لكل من المصنفين. بالإضافة إلى ذلك،PCA-KNN هو أفضل مصنف ليتم تطبيقه على النظام مع الحد الأدنى لعدد الإنذارات الكاذبة وأعلى مستوى من الأمان ومعدل اكتشاف.

الكلمات المفتاحية: مصنفات آلة دعم المتجهات (SVM)، الجار الاقرب (KNN)، تحليل المكونات الرئيسية(PCA)، تحسين نظام كشف التسلل.

Chapter One

Introduction

Chapter One: Introduction

1.1. Background

Computers and communication are considered as essential parts of human life. The world is becoming more and more interconnected with both the internet and networking techniques. Thus, the amount of the available commercial, personal, government and military data that increases the importance of network security due to vulnerability of important data. Practically, various security tools, such as firewalls, anti-viruses and policies have been proposed to reduce threats, reach statutory compliance and address the information security problems. IDSs are software applications that used to monitor the activities of networks or systems, detect unauthorized records, activities and events, such as attacks and then respond automatically to these activities. On the other hand, these systems do not completely guarantee the security issue in networks and have some restrictions.

The rapid improvements and enhancements in internet based technologies and techniques, various application domains in both computers and communication have emerged and considered as main parts of human life. The accessibility of cheap broad band, mobile technologies, and internet connectivity raised the number of connected computers to the internet. Nowadays, the world is becoming more and more interconnected with both the internet and networking techniques. Thus, the amount of the available commercial, personal, government and military data on the networking infrastructures is being increased daily where thus in turn increases the importance of network security due to vulnerability of important data and intangible intellectual property to various types of attacks and threats. (Dacier & Alessandri, 1999; Alessandri, 2004)

Practically, various security tools, such as firewalls, anti-viruses and policies have been proposed to reduce threats, reach statutory compliance and address the information security problems. However, the prevention of attacks using these tools is a real challenge due to the presence of unknown bugs and weaknesses in systems and applications and complicated, unexpected interactions among software components and network protocols which are frequently used by attackers. Therefore, Intrusion Detection Systems (IDSs) have been proposed as advanced security tools instead of the traditional ones (Dacier & Alessandri, 1999; Porras et al., 2000).

Intrusion Detection represents the approach of controlling and monitoring the performed processes in a certain network or computer system, which differ from usual system activities to detect them. IDSs are applications software that used to monitor the activities of networks or systems, detect unauthorized records, activities and events, such as attacks and then respond automatically to these activities. Those systems have no impact on the utilization of defensive mechanisms in computer systems, while they represent the final defensive mean in the security of those systems. Intrusion detection is an essential issue in network security field. The main two intrusion detection methods are the misuse and anomaly detections. IDSs gather and inspect data to monitor and detect intrusions in that computer systems and networks (Ghali, 2009).

On the other hand, these systems do not completely guarantee the security issue in networks and have some restrictions, such as the complexity in describing the normal performance metrics and high range of false alarms that causes trust lack in the systems. But, when certain advanced methods are used in these systems, they can efficiently enhance the network security. (Dacier & Alessandri, 1999)

1.2. Problem Statement

IDSs are advanced security tools that can be used to detect various types of attacks, such as Denial of service (DoS), User to Root (U2R), Remote to Local (R2L) and probe attacks in networks. The main problem of these systems is their low accuracy. The current IDSs are not precise enough to offer reliable detection, where this problem resulted in a high number of generated false alarms: False Positives (FP) and False Negatives (FN). This large number of false alarms makes the process of filtering out false attacks without missing real ones a real challenge. Furthermore, it makes security administrators unable to respond correctly for risks.

Both the Support Vector Machine (SVM) and K Nearest Neighbor (KNN) are some of the classifiers methods that can be used to detect FP, FN and accuracy for NSL-KDD CUP 99 dataset. The SVM is supervised learning classifier, which depends on creating a hyper-plane using support vectors to separate normal from abnormal data, while the KNN is a machine learning technique that can be utilized to discover new added data for training set. In this work, both methods will be applied to the developed IDS to determine the most efficient one.

1.3. Aims and Objectives

This study aims to compare the performance of NSL-KDD dataset in the system using two classifiers; PCA-SVM and PCA-KNN to determine the optimal classifier. The following objectives must be met in order to achieve the proposed purposes of this work:

- Review the main concepts, definitions, and terms of Intrusion Detection Systems (IDSs)
- Review the main types of Intrusion Detection Systems (IDSs) and compare them with other defense methods
- Explore the main types of attacks that threat computer systems and networks
- Develop advanced IDS using the MATLAB software program
- Add Principal Component Analysis (PCA) method to the developed system
- Apply both PCA-KNN and PCA-SVM to the system using the same dataset; NSL-KDD CUP 99
- Apply the control chart on data to distinguish normal records from attack ones
- Measure the False Positive (FP), False Negative (FN), True Positive (TP), True Negative (TN), and Detection Rate (DR) of each method to find the best one that has the lowest FN and FP and highest DR and CR in the detection of attacks
- Determine the most optimal method
- Explore the main problems and limitations of the current work

1.4. Research Importance

The current Intrusion Detection Systems (IDSs) cannot detect all types of attacks as well as they generate false alarms. The research offers a modified intrusion detection system based on the most effective statistical method to assist in the detection of various types of attacks. The comparison between PCA-KNN and PCA-SVM will be applied.

1.5. Thesis Outlines

This thesis is divided into four chapters as follows:

- Chapter two: Literature review

It includes a review of some of the related works concerning the developed Intrusion Detection Systems (IDSs) using various machine learning techniques with a comparison among them

- Chapter three: System analysis and design

It includes the analysis of the developmentof efficient IDS with applying both PCA-KNN and PCA-SVM in details

- Chapter four: Results discussion and evaluation

It demonstrates the main results of the application and evaluation it based on applying it on a dataset to determine the most efficient method

- Chapter five: Conclusion and recommendations

It offers a summary concerning the conducted work and its outcomes, problems and improvements and demonstrates the main works that can be performed in the future to enhance this work

Chapter Two

Literature Review

Chapter Two: Literature Review

2.1 Background

This section offers a background concerning the IDSs, the main types of attack trends and threats that intrude computer networks and systems, the main available defense methods, the attacks types and the classification of IDSs.

2.1.1 Overview of IDSs

Various researches explored the implementation of IDSs that provide details and information concerning the features of those systems, which are in turn appropriate and applicable in the detection of various types of attacks. The implementation of those systems is based on the experiences that resulted from both the development and utilization of IDS and the analysis of various kinds of threats. (Dacier & Alessandri, 1999)

The main IDS characteristics are the information that utilized in the analysis, the verification and interpretation levels of protocols and the utilized approaches in finding activities, which can signify attacks. IDSs are mainly range from simple to complicated systems based on their properties. Two simple parameters can be used to represent IDS characteristics. The first one represents the general characteristics of the system, such as the ability to determine conventional expression similarity on data, but this parameter cannot define the target of that characteristic or determine its type. The second parameter can define the target of the system characteristic to decide the validity of the system characteristics. (Alessandri, 2004)

The IDS scope which is an iterative method that consists of three main high level scopes is explored, these scopes are: Networking, user and host. Both networking and host are divided into several low level scopes, like application layer and process. User scope is the human that uses the IDS. (Alessandri, 2004)

2.1.2 Threats and attack trends of Networks and Systems

Threats of computer systems and networks can be persons, objects or events that can cause damages in those systems and networks. They can be classified into accidental threats, such as errors in computations and malicious threats, that as intended changes in data. On the other hand, network security threats can be classified into two types; internal and external threats. The internal ones occur by persons who have known access to networks or systems, where this access is can be an account or physical access. Conversely, the external threats occur by persons who have no known accesses to network or systems, where those threats can be resulted from internet or access servers. (Xu & Shelton, 2008; Dewaele&Fukuda, 2007)

The main types of attack trends are vulnerability, phishing activity, and fraud activity and malicious code trends. Vulnerability trends represent the network weakness, which permit attacks to collaborate its integrity and accessibility. Phishing activity trends represent the ability of attacks to get personal data of users who can be persons, groups or organizations, where those trends mainly require fatalities to provide their main qualifications. Fraud activity trends represent the illegal utilization of certain data, which are relative to specific persons, by attacks. Malicious code trends represent a set of software threats that attack systems and networks. (Lakhina & Crovella, 2005; Ye& Emran, 2002)

2.1.3 Network Defense

Network defense represents the process of monitoring, defending, exploring, discovering and responding to illegal activities in computer systems and networks. The main defense systems are the firewall, encryption, authentication, IDSs and physical security. Firewall ranges from personnel array firewall systems, which are mainly used to protect huge computer networks and distinguish among networks based on using specific rules to determine the legal connections. The encryption is mainly utilized in hiding data using secret techniques to be then decrypted via known secret keys only. Thus, attacks will not be able to get those data. The authentication allows transmitting messages among users and network access routers via protocols to prevent attacks from reaching those messages, where users are defined by Media Access Control (MAC) addresses to accesses those messages. IDSs can discover several types of attacks based on monitoring computer systems and networks (Tandon & Chan, 2005; Hofmeyr, Forrest & Somayaji, 1998). The physical security helps in the evaluation of various risks to allow performing right actions. (Faria, 2006)

2.1.4 Types of Attacks

The main types of attacks are: (Gogoi, Bhattacharyya, Borah & Kalita, 2013)

- Denial of service (DoS) attacker uses obtainable or unobtainable memory sources in order to control requirements or to ignore rights of users from service using such as SYN flood, neptune,back, smurf, land and teardrop.
- User to Root (U2R) attacker uses an account of a system user in order to realize root access to the required system as the user privilege (e.g. buffer overflow)
- Remote to Local (R2L) attacker sends several packets to the system without having an account on this system (e.g. password guessing).
- Probe attacker finds out information or recognized threats. Attackers can easily make an attack with the use of this information (e.g. ping sweep , port scan)

Table 2.1 elucidating different types of sub attacks that belong to the main attacks above along with their popular name (Kezih&

Taibi, 2013).

Attack name	Attack type	Attack name	Attack type
Back	DOS	Per1	U2R
Buffer_ overflow	U2R	Phf	U2L
Ftp_ write	R2L	Pod	DOS
Guess_ passwd	R2L	Portsweep	Prob
Imap	R2L	Rootkit	U2R
Ipsweep	Prob	Satan	Prob2
Land	DOS	Smurf	DOS
Loadmodule	U2R	Spy	R2L

Multihop	R2L	Warezclient	R2L
Neptune	DOS	Warezmaster	R2L
Nmap	Prob		

Table 2.1 types of sub attacks occur within a network, source (Kezih&Taibi, 2013).

2.1.5 Classification of IDSs

IDSs can be classified based on the used intrusion detection method or protected system. IDSs that based on the used intrusion detection method can be categorized into anomaly detection, pattern matching and protocol analysis systems, while IDSs that based on the protected systems can be categorized into hit based, network based and hybrid systems.

2.1.5.1. Intrusion detection method based IDSs

IDSs that based on the used intrusion detection method can be categorized into anomaly detection, pattern matching and protocol analysis systems. The anomaly detection based IDSs are utilized to determine patterns in data that do not match the expected performance, where those patterns can be anomalies, exceptions, contaminants, peculiarities or outliers. Those systems can be used in various applications, such as in detecting fraud of credit cards and intrusions. (Chandola & Kumar, 2009)

The pattern matching based IDSs are utilized to decide the number of times that an applicant pattern occurs and data concerning its frequency distribution through a text. Patterns can be defined as groups of strings, where ach sting is considered as a set of symbols. The most optimal pattern is the one that has the smallest number of strings. The protocol analysis based IDSs are utilized to decide locations and lengths of fields in protocol packets, which are used then with reverse engineering to explore the structure of responses and requirements. Those systems depend on using perceptions and protocol analyzer instruments, such as tcpdump. (Chandola & Kumar, 2009)

2.1.5.2. Protected systems based IDSs

IDSs that based on the protected systems can be categorized into Host based, network based and hybrid systems. The Host IDS (HIDS) are used to monitor the system calls; the Network IDS (NIDS) are used to monitor the system performance, while the hybrid systems are mixtures of both types. In the NIDS, the network activities are independent across various ports. In those systems, the dimensionality of information is decreased using the random projection schemes, while the abnormalities are discovered across various aggregation levels using the multi-resolution non Gaussian marginal distribution ((Leung, 2008; FIPS PUB 191, 1994). NIDS are utilized as the final defense line to permit various responses to events with the presence of insufficient intrusion avoidance mechanisms. Those systems depend on comparing the network traffic with a predefined dataset to discover illegal traffics. The main advantages of those systems are that they are easy to be used and have small numbers of false alarms. Conversely, those systems have not the ability to discover the whole types of attacks

The main works of HIDS can be classified into two types; sequence based and feature based works. The sequence based ones are dependent on the sequential orders of events, while the feature based ones take into account the calls as independent information elements. Those systems are utilized to detect intrusions based on analyzing various computing activities models, such as the CPU usage and memory. In addition, those systems examine the system settings, calls, local log inspections and more as well as they are utilized in a wide manner because of their efficiency in detecting known attacks. However, those systems cannot detect new attacks (Hu, 2010).

The hybrid systems combine among the benefits of both systems to provide forceful systems to be the foundation for monitoring and detecting misuses and filter alerts and notifications an optimal way to assist in monitoring and reacting misuses.

2.2 Overview of SVM and KNN Techniques

This section offers an overview concerning both techniques; SVM and KNN.

2.2.1. Overview of SVM

Support Vector Machine (SVM) is an advanced machine learning technique where it outperforms many other typical machine learning techniques in the various field. The SVM is a very efficient method for classifying where it determines the most optimal separating hyper-plane among classes depending on training cases. To understand this technique, suppose that there are two linearly separable classes in a certain d-dimensional space with the use of training vectors that related to two classes; $\{\mathrm{x_i}, \mathrm{y_i}\}$ in which $\mathrm{x_i} \in \mathrm{R^d}$ signifies vectors in the d-dimensional space, while $\mathrm{y_i} \in \{-1, +1\}$ represents a class label. The purpose is the design of a hyper-plane to simplify data in an accurate way where this hyper-plane is the one that leaves the extreme margin from both classes (Furey, Cristianini, Duffy, Bednarski, Schummer & Haussler, 2000).

The main idea of SVM technique is finding the hyper-plane which has the most extreme margin towards the sample object, where the margin value and the probability to inaccurately classify a feature vector are inversely related to each other. The following equation**(2.1)** can be used to define a hyper-plane (Furey, Cristianini, Duffy, Bednarski, Schummer & Haussler, 2000).

$$(\mathrm{w.x}) + \mathrm{b} = \mathrm{U}(2.1)$$

where w is a normal to the hyper-plane, x represents a feature vector that lies on that hyper-plane and b represents the bias in which $\frac{|\mathrm{b}|}{\|\mathrm{w}\|}$ represents a perpendicular distance among the origin and the middle point of the hyper-plane as shown in the **Figure 2.1** concerning the SVM basics.

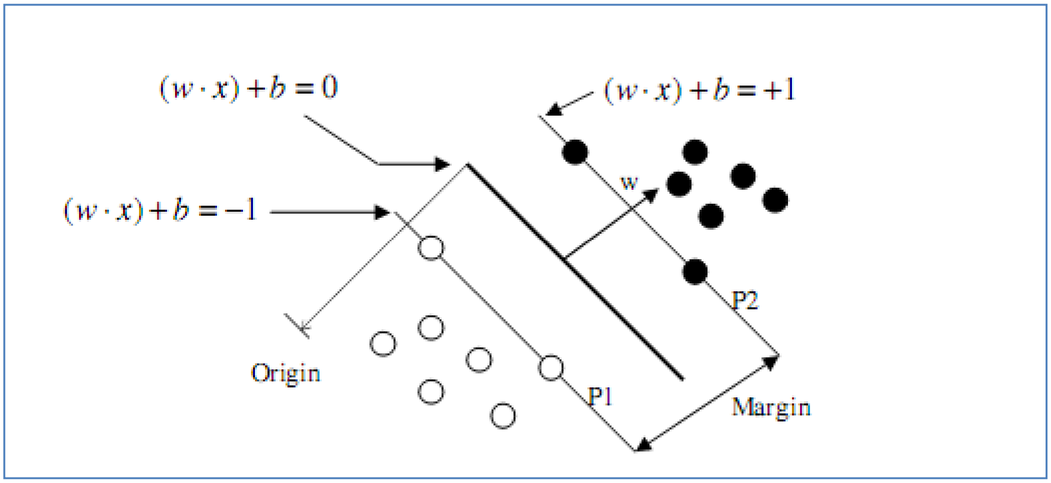


Figure 2.1 SVM classification basics (Bhavsar & Kalyani, 2013)

The purpose is to separate among two classes; open circle that stands for the class label -1 and the solid circle which stands for the +1. The lying circles on both planes; P1 and P2 represent the support vectors in which the optimal *hyper-plane* located among those two plans which are parallel to each other. The margin among those planes is $\frac{|2|}{\|w\|}$. The SVM technique should maximize the *hyper-plane* margin to get enhanced generalization. The following formulas**(2.2) (2.3)** can be used then to describe the hyper-plane of the two classes (Pedersen & Schoeberl, 2006).

$$(\mathrm{w.x}) + \mathrm{b} = +1 \quad \mathrm{forclassy} = +1 \qquad (2.2)$$

$$(\mathrm{w.x}) + \mathrm{b} = -1 \quad \mathrm{forclassy} = -1 \qquad (2.3)$$

Practically, classes are not linearly separated. Thus, the input space must be mapped into another feature space with high dimensionality. More clearly, input vectors, such as the low-level feature vectors are mapped into a feature space H using a nonlinear conversion, $\Phi: \mathrm{R}^{\mathrm{d}} \rightarrow \mathrm{H}$.Thus, the optimal *hyper-plane* is generated in that high dimensional feature space with the use of kernel function; $\mathrm{K}(\mathrm{x_i}, \mathrm{x_j})$ that generated among two input vectors; $\mathrm{x_i}$and $\mathrm{x_j}$. This formula **(2.4)**can be written as follows: (Lanckriet, Deng, Cristianini, Jordan, & Noble, 2004)

$$\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \Phi(\mathrm{x_i}).\Phi(\mathrm{x_j})(2.4)$$

Polynomials kernel is one of the most common mappings, where its formula**(2.5)** is described below: (Liu, Jun & Zhang, 1995).

$$\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = (\mathrm{x_i}.\mathrm{x_j} + 1)^{\mathrm{d}} (2.5)$$

where, d represents the polynomial degree. Another common mapping is the Radial Basis Functions (RBFs) kernel**(2.6)** as described below: (Lanckriet, Deng, Cristianini, Jordan, & Noble, 2004)

$$\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \mathrm{e}^{\frac{\|\mathrm{x_i} - \mathrm{x_j}\|^2}{2\sigma}} \quad (2.6)$$

where,σ stands for the Gaussian sigma. As described above, the SVM technique is developed to solve binary classification problems with two class labels only; +1 and -1. This technique can be enhanced more to be used for multi-class problems. Generally, there are two approaches for SVM multi-class classification; one against all and one against one. The one against all approach includes the construction of SVMs among each class and other classes). As an example, suppose that there are four classes; C1, C2, C3 and C4, thus, four SVMs must be generated in which C1 can be classified based on classifying C1 and on C1 by the corresponding SVM and the same for other classes.

The one against one approach includes the construction of SVMs among the whole pairs of classes. As an example, suppose that there are four classes; C1, C2, C3, and C4, thus, six SVMs must be generated where those six classifiers classify [C1 or C2], [C1 or C3], [C1 or C4], [C2 or C3], [C2 or C4] and [C3 or C4].

2.2.2. Overview of KNN

KNN is a machine learning technique that classifies data depending on their similarity with data in the training set. This technique makes decision depending on the whole training dataset. The KNN is a simple method, which saves all obtainable cases and categorizes new data depending on a certain similarity measure. This method has been applied in various pattern recognition and statistical estimation applications. In this method, data are classified via a majority vote of its neighbors, where data are assigned to the most common class between all its K-Nearest Neighbors that measured using a certain distanceformula. When the number of nearest neighbors is one, then the data are assigned to that class. This method does not depend on using training data points for generalization. This means that there is no clear training phase, thus the training phase is quick. This demonstrates that this method keeps the whole training data.

The KNN method is developed based on initially defining a group of notations: $S = (x_i, y_i)$, where i=1,2,…, N as a training set, x_i represents the d-dimensional vector of features and y_i is related to the obtained class labels. Based on considering a binary classification with supposing that all training data are random variables (X, Y) that have unknown distributions, variables are labeled as training samples. The KNN then creates a local sub region $R(x)$ that is located at x. this region includes the closest training points to x. Thus, it can be written as follows:$R(x) = \{\hat{x} | D(x, \hat{x}) \leq d_{(k)}\}$, where $d_{(k)}$ represents the k^{th} order statistic of $D(x, \hat{x})_1^N$ and $D(x, \hat{x})$ represents the distance metric. On the other hand, $k[y]$ represents the number of the region samples that labeled by y. The main purpose of the KNN technique is to estimate the posterior probability $p(y|x)$ using the following formula**(2.7)**: (He & Wang, 2007).

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{k[y]}{k} (2.7)$$

for a certain sample x, the decision that donated by $g(x)$ can be formulated based on assessing the $k[y]$ values and choosing the class with highest $k[y]$ value as follows**(2.8)**: (He & Wang, 2007).

$$g(x) = \begin{Bmatrix} 1, k[y = 1] \geq k[y = -1] \\ -1, k[y = -1] \geq k[y = 1] \end{Bmatrix} (2.8)$$

Therefore, the decision, which can maximize the related posterior probability can be then utilized in the KNN method.

In this work, the performance of PCA-SVM will be applied and compared with another classifier, which is related to PCA-KNN classifier in order to measure different tools for each system.

2.3 Related Works

From many years, and while the number of the alert messages increased in the system and networks, the Intrusion Detection System (IDS) have been developed to decrease the number of the alert messages, and the main mission of the Intrusion Detection System (IDS) is to keep the systems or networks save from the different intrusions, and analyzed it in addition to anticipated the users behaviors, after that, these behaviors can be classified by either an attack or a usual behavior.

2.3.1. Developed SVM Based IDSs

The Support Vector Machine (SVM) is a classification method that designed originally for binary categorization as well as it is used for solving multi-class problems. This method is applied widely in IDSs to enhance the detection of various types of attacks.

Mukkamala, Janoski & Sung (2002) conducted a study about the intrusion detection system and discussed how to reveal the intrusions and decrease the audit trail; they used two approaches: the support vector machines (SVM) and neural networks. They used a set of standards from the competition of Knowledge Discovery and Data Mining (KDD), which designed by Defense Advanced Research Projects Agency (DARPA). They show that the classifiers that are effective and very precise in the intrusion detection can be constructed by either the neural networks or the support vector machines (SVMs).

Wang, Hong, Ren & Li (2009) proposed the development of IDS that depends on using two techniques; SVM and Particle Swarm Optimization (PSO). The PSO is utilized to enhance the SVM practicability, which is influenced by the difficulty of choosing suitable SVM parameters. This method is an enhanced technique that has elevated global search ability with an easiness to be designed and implemented. The combined method; PSO-SVM is applied in the proposed research to IDS with the use of KDD Cup 99 dataset. The typical

PSO is mainly utilized to decide the SVM free parameters, while the binary one is utilized to get the best subset of features during implementing the system. Results demonstrated that the proposed PSO-SVM based IDS outperforms the typical SVM algorithms in terms of detection rate.

Chen, Cheng, Chen & Hsieh (2009) used to revealing the intrusions the Rough Set Theory (RST) in addition to using the Support Vector Machine (SVM), the importance of the RST comes from lessen the dimensions via making some pre-processes, and the SVM model after that is responsible for the learning and testing processes, respectively, furthermore, this method reduce the data space density, and they show that the RST and SVM can ameliorate the rate of the false positive and precision, accuracy 89.13%.

Mulay, Devale & Garje (2010) they suggested the decision tree based algorithm to build multi- class intrusion detection system, in addition, to investigating the tree-structured multiclass, the multiclass issues can be solved by using the applications of classification, also the Decision-tree-based support vector machine (SVM), which merges the support vector machines and decision tree helping in solving the multi-class issues, the SVM is one of the classifiers which are built for the binary classification, they show that this method able to decrease the testing time in addition to the training time. According to them, there are various methods to build the binary trees, and these methods divide the data set into two subsets from root to the leaf until every subset consists of only one class. Finally, they found that this structure have been participated in improving the performance of the classification.

Bhavsar & Waghmare (2013) According to the security is an essential issue in all of the systems and networks, and one of the main points in the network security is the intrusion detection system (IDS), and this system able to find the different types of attacks in the network , they suggested to implement the IDS by using the technique of data mining, and they suggested to use the Support Vector Machine (SVM) in the classification, because the SVM one of the most popular method using nowadays in the data mining's classification algorithm, even if it needs a large amount time in the training. But by making several experiments, they found that this disadvantage can be overcome by making some pre-processing for the data set, and by using an appropriate SVM kernel function just like the Gaussian Radial Basis Function, they can increase the rate of SVM attack detection, and decrease the False Positive Rate (FPR), and they conducted these experiments using the version of NSL-KDD Cup'99 data set, which defined by the NSL-KDD Cup'99 dataset in order to verify the effectiveness of the suggested system. .

Yao, Zhao & Fan (2015) proposed the design and implementation of IDS with the use of an improved SVM model as a classical pattern recognition method. The used SVM is combined with a weighted kernel function depending on the training data features for intrusion detection. In addition, a rough set theory is used to carry out the ranking of features and choosing of new model tasks. The designed system was evaluated using the KDD dataset. Based on comparing the developed system with IDS that depends on using a conventional SVM, it was demonstrated that the developed system outperformed the other one in terms of false negative rate, computation time and accuracy.

2.3.2. Developed KNN Based IDSs

In the recent years, various IDSs that depend on using KNN method alone or combined with other machine learning methods have been proposed. The KNN is one of the simple classification methods, which is used to compute the distance among a query point and all its neighbors and then select the closest one.

Li, Fang, Guo & Chen (2007) proposed a solution for the high false alarm rates of IDS and problems in getting precisely obvious data to model normal patterns and detection rate deterioration due to the presence of noisy data in the training dataset. Authors presented an advanced network anomaly detection technique depending on using an enhanced "Transductive Confidence Machines for K-Nearest Neighbors (TCM-KNN)" method. The KDD Cup 1999 dataset was utilized to perform experiments on the designed system. Results demonstrated that the presented system can efficiently detect anomalies with low false positive rate, high true positive rate and elevated confidence than the traditional anomaly detection techniques. Furthermore, the presented technique is robust and efficient in detecting noisy data. As well, it retains enhanced detection performance with the use of feature selection to prevent dimensionality curse.

Shailendra & Sanjay (2009) proposed a hybrid feature selection technique that consists of two stages: filter and wrapper. The first stage chooses features that have the highest information gain to be fed to the second stage to offer the final feature subsets. Those subsets are then inserted to the KNN classifier to categorize attacks. The DARPA KDDCUP99 dataset was used to evaluate the effectiveness of the technique.

Li & Guo (2007) developed an advanced supervised network IDS depending on using Transductive Confidence Machines for K-Nearest Neighbors (TCM-KNN). This method can efficiently discover abnormal data with high detection rate and minimum number of generated false alarms with the use of small number of data and features for the training stage. The system performance has been evaluated using the KDD Cup 1999 dataset, where results explored that the developed system is more efficient and robust than traditional IDSs and can be used in real applications.

Ming (2011) proposed a combined approach of KNN and Genetic Algorithm (GA) for choosing and weighting features. It includes two main phases; training and testing. In the training phase, the initial 35 features were weighted, where the features that have highest weights were then chosen for the second phase. Various DoS attacks were applied in this work to assess the performance of the system.

Li, Yi, Wu, Pan, and Li (2014) developed new IDS depending on using KNN method in wireless sensor network. The developed system can distinguish among normal and abnormal nodes based on discovering the abnormal performance. The work depends on analyzing parameter selection and IDS error rate with elaborating the development and design of the system. Results demonstrated that the system has effective, quick detection with high accuracy and speed.

Htun & Khaing (2015) proposed an advanced technique to implement an anomaly IDS with the use of misuse to train normal data and detect attacks. It depends on combining a random forest machine learning algorithm with the KNN pattern recognition technique to detect and classify the known classes from unknown or attacks ones. The KDD Cup 1999 dataset was used to evaluate the system. Results proved the efficiency of the developed system in detecting and classifying normal classes from abnormal ones.

2.4 Summary

This chapter reviews the security of computer systems and networks, which is one of the main issues recently because of the improvements of several types of attacks and threats that can expose a system or network and threat its security of data. Network threats can be classified into persons, events or objects, which can cause damages in a network or system. In addition, threats can be accidental, like errors in calculations or malicious, like data intended modification.

The main defense systems of networks are IDSs, firewalls, physical security and encryption. The IDSs can detect several types of attacks by monitoring the networks, the firewalls are utilized in the protection of large networks in large organizations to divide between networks via utilizing several rules to decide the allowable connections, the encryption is used to hide data by using a secret algorithm, while the physical security assists in the evaluation and understanding of several risks which in turn facilitates taking corrective actions. The IDSs in turn are divided into two types: NIDS and HIDS. The NIDS monitors the behavior of the system, while the HIDS monitors the calls of the system below. There are three types of detection methodologies used in IDS: pattern matching, protocol analysis and anomaly detection.

In this chapter, various IDSs that depend on using SVM and KNN methods, each one alone or combined with other machine learning methods to enhance their efficiency are presented, analyzed and discussed. It can be concluded that both the SVM and KNN are efficient methods that can effectively enhance the detection rate, accuracy and positive alarms of IDSs and decrease the negative alarms for various types of attacks.

Chapter Three: Methodology

Chapter Three: Methodology

A binary classifier based IDS is developed in this work using two types of classifiers; PCA-SVM and PCA-KNN with and without applying the control chart. The developed system is applied on the NSL-KDD to evaluate its performance and determine the optimal classifier that offers the highest detection rate with the lowest number of generated false alarms. This chapter defines the used dataset and determines its main features. In addition, it discusses the conducted methodology in details and demonstrates the stages of each classifier.

3.1 The proposed IDS

In this work, PCA-KNN is applied and compared with PCA-SVM In the KDD dataset based IDS, both the training and testing stages are prepared through a categorization process.

NSL-KDD dataset is used to measure the system performance. This dataset includes 41 features of the network connection. In this work, the MATLAB program is used to apply system with the use of this dataset.

The presented IDS includes two stages; training, testing. In the training stage, a training dataset is used to train the system to recognize the normal connections from the attacked ones. Thus, the training dataset should have adequate information concerning connections and attacks. In this stage, a SVM classifier is used to recognize the most important features to be used in detecting attacks. In the testing stage, a testing dataset that includes connections and attacks is used in the system to measure the IDS performance, where the high performance stands for the high accuracy in determining both connections and attacks. When the system performance level is not accepted, these two stages are executed again. In the running stage, the system is used to protect the network traffic. In both the testing and running stages, the system categorizes the network traffic depending on the requested service and then depending on the chosen features.

Recently, various researches have been conducted to find solutions for reducing the high dimensionality for feature vectors. It was found that the efficient solution for reducing the high dimensionality is the application of various dimensionality reduction approaches which are classified into linear and nonlinear dimension reduction approaches. The main linear dimension reduction approaches are random projection Singular Value Decomposition (SVD), and Principal Component Analysis (PCA). On the other hand, the main nonlinear dimension reduction approach is the Multi-Dimensional Scaling (MDS). Generally, the PCA approach is one of the most efficient and appropriate one for dimensionality reduction. It depends initially on computing both the mean vector (μ) and the covariance matrix (C) from datasets using the following formulas**(3.1) (3.2)**:(Mardia et al, 1979)

$$\mu = \frac{1}{n}\sum_{i=1}^{n} X_i \quad (3.1)$$

$$C = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)(X_i - \mu)^T \ (3.2)$$

where (n)represents the number of feature vectors. (X)Represents the features selected .After computing both the mean vector and the covariance matrix, both the eigenvectors and Eigen-values of the covariance matrix (C) are calculated. After that, Eigenvectors offers coefficients, which in turn give the principal components.

The KDD dataset includes 41 features, which results in a large dimension space for these data. Therefore, the PCA is applied in this work to reduce this dimension space based on selecting the most effective features from the 41 ones to be applied in the developed system. This in turn can speed up the system.

A reduction process has been used to reduce the number of features in order to decrease the complexity of the IDS. It is well known that PCA has been widely used in data compression and feature selection. Feature selection refers to a process whereby a data space is transformed into a feature space, which has a reduced dimension.

In this research, PCA is applied on the dataset for feature selection, then the classification is applied into normal and attack records, in training phase normal data will be taken as training data to both SVM and KNN classifiers in order to learning main features of normal data, then to filter the training data, control chart (CC) will be used as lower control chart (LCC) and upper control chart (UCC), by compute mean and standard deviation to each record. Control chart is used to control normal training data within a specific range, in order to apply testing data on same range.On the other hand, the performance of both classifiers will be compared to each other. The expected results can be shown and compared with different practical scenarios as shown in the **Figure 3.4**.

PCA: -PCA is applied on the NSL-KDD dataset to reduce dimension and feature selection in order to decrease the complexity of the IDS. It depends initially on computing the covariance matrix (C), Firstly, preprocessing to dataset from feature mapping and scaling by changing each character in the dataset to numerical numbers**,** then feature selection compute the covariance matrix (C) then data splitting to training and testing data. The following **Figure 3.1** shows the diagram of proposed IDS system based PCA

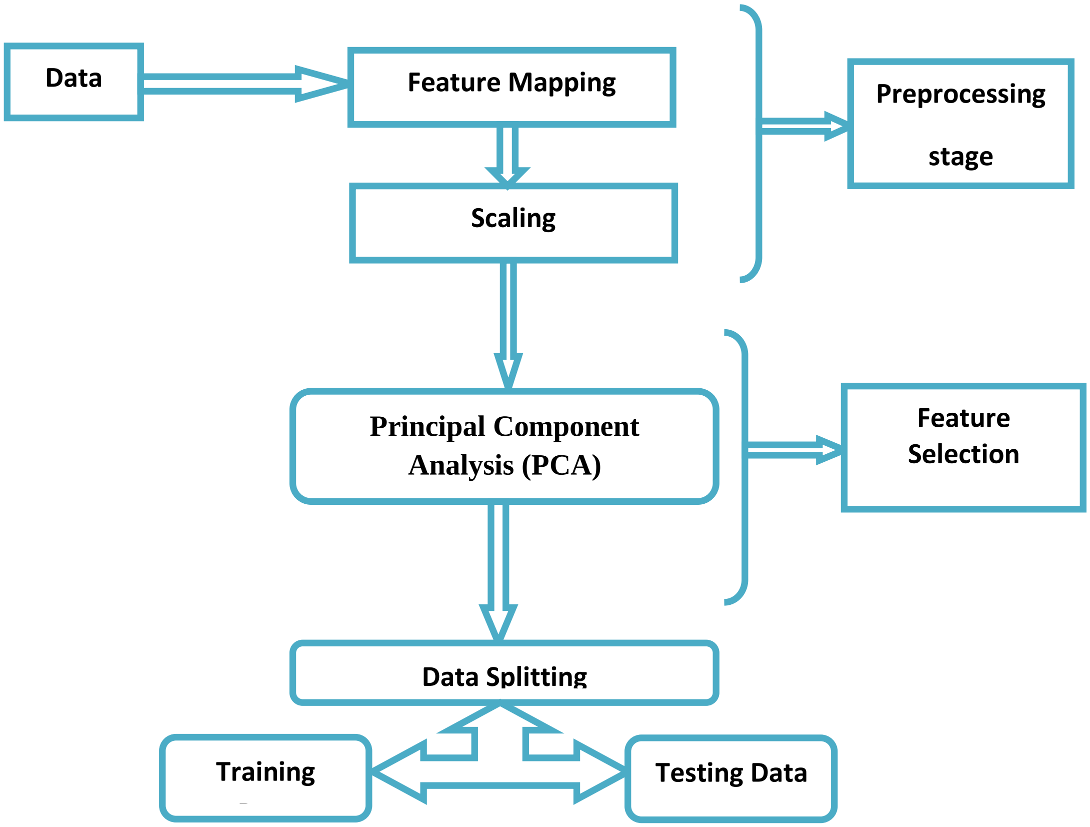


Figure 3.1Diagram of proposed IDS system based PCA

SVM:- In training stage, SVM classifier based feature selection method is used to recognize the features to be used in detecting attacks. In the testing stage, a testing dataset that includes connections and attacks is used in the system to measure the IDS performance, where the high performance stands for the high accuracy in determining both connections and attacks then compute the FP,FN,TP,TN,DR,CR. The following **Figure 3.2**shows the flowchart of proposed IDS system based PCA - SVM

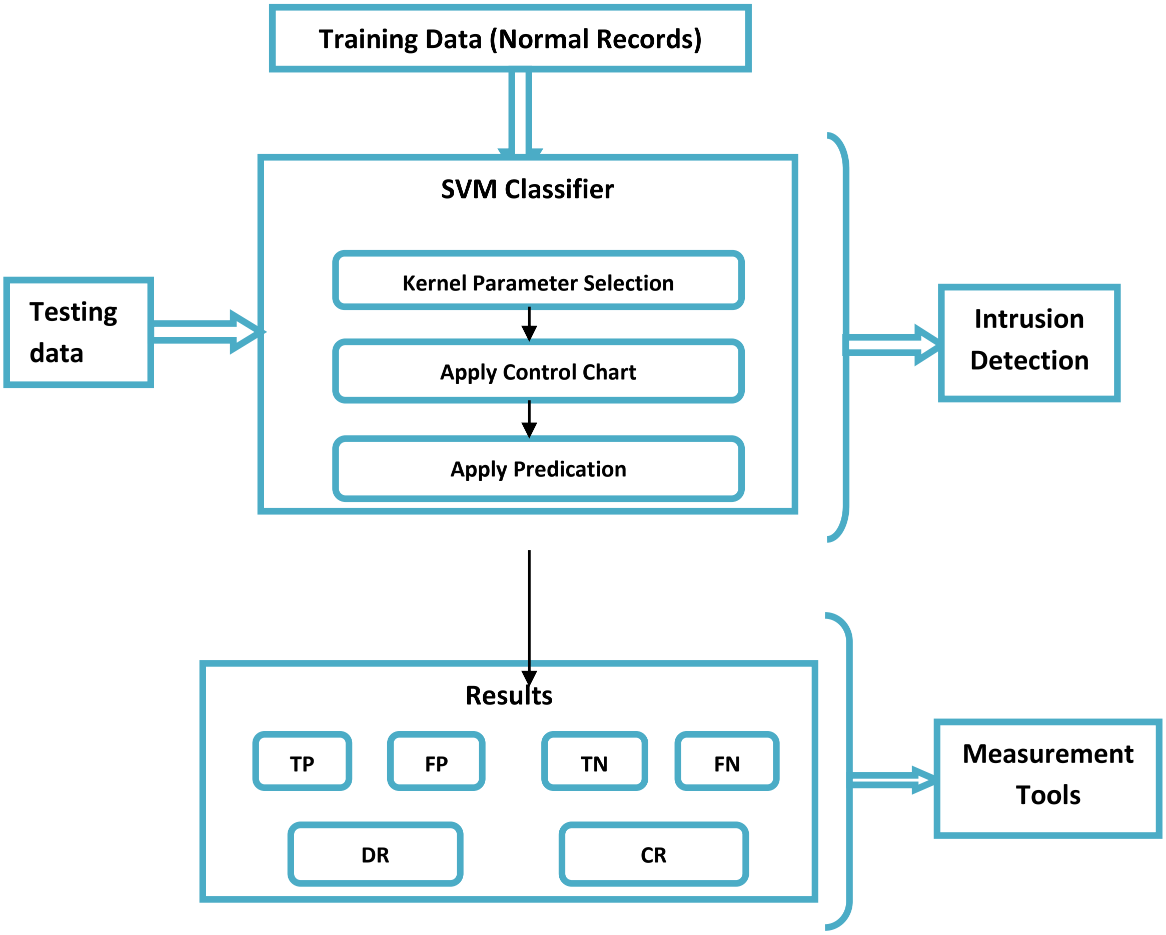


Figure 3.2Diagram of proposed IDS system based PCA - SVM

KNN:-KNN is a machine learning technique that classifies data depending on their similarity with data in the training set. This technique makes decision depending on the whole training dataset. The KNN is a simple method, which saves all obtainable cases and categorizes new data depending on a certain similarity measure by using several steps .firstly, Determine k then Compute the distances among new data and the training data then Sort the distances and decide the k nearest neighbors then Collect their classes and decide the optimal class after that compute FP, FN, TP, TN, DR, CR. The following Figure 3.3 shows the flowchart of proposed IDS system based PCA - KNN

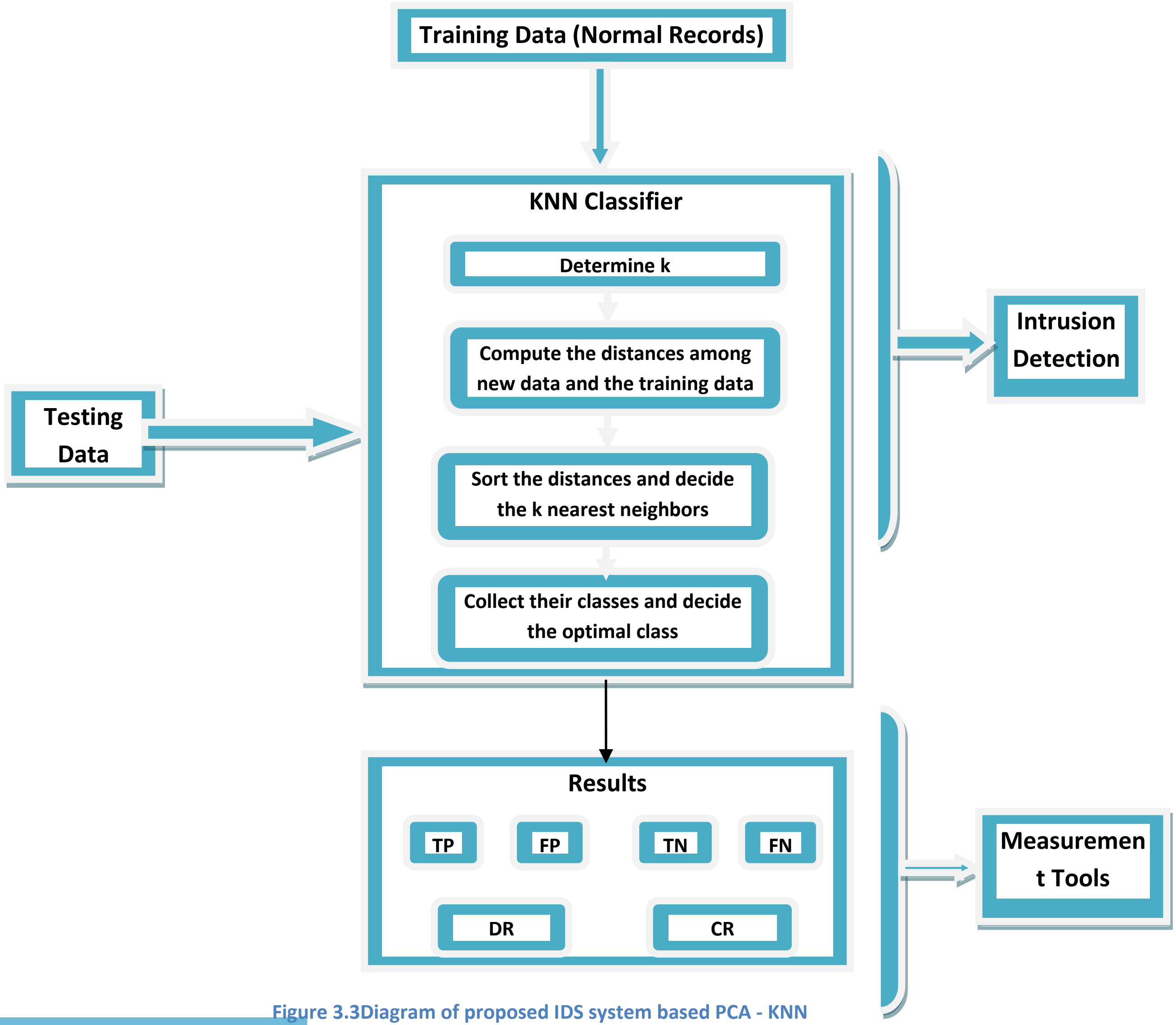

Training Data (Normal Records)
KNN Classifier
Determine k
Compute the distances among new data and the training data
Sort the distances and decide the k nearest neighbors
Collect their classes and decide the optimal class
Testing Data
Intrusion Detection
Results
TP
FP
TN
FN
DR
CR
Measurement Tools

Figure 3.3Diagram of proposed IDS system based PCA - KNN

The main meaning behind Support Vector Machines (SVMs) is to enable us to extract and accomplish a mixed component that maximizes the separating margin between two Confusion Matrix classes the negative and the positive. (Vapnik, 1998)

An introduction to SVM strategy founded by Lippmann et al (2000), the main goal of SVM is that it approximates the implementation of the Structure Risk Minimization (SRM) principle that in its basic structure based on statistical learning theory rather than the Empirical SRM, in the way that the classification function that SVM adopt it in the way of minimizing the Mean Square Error (MSE) all over the training data set records. Form the metrics that used in the aim to estimate the classification quality is by measuring the classification accuracy. Another important metrics that must be addressed is the need to measure the running time (computational complexity) of the intrusion detector. The computational complexity of related to linearity or nonlinearity depending on kernel function,proportion to the number of support vectors this considers as a problem in evaluating the computational complexity value since it is in a linear relation with the number of vectors.

The KNN is a simple method, which saves all obtainable cases and categorizes new data depending on a certain similarity measure. This method has been applied in various pattern recognition and statistical estimation applications. In this method, data are classified via a majority vote of its neighbors, where data are assigned to the most common class between all its K-Nearest Neighbors that measured using a certain distance function. When the number of nearest neighbors is one, then the data are assigned to that class. This method does not depend on using training data points for generalization. This means that there is no clear training phase, thus the training phase is quick. This demonstrates that this method keeps the whole training data.

By taking the average of the K-neighbors nearest to the testing process, it can smooth out the impact of isolated noisy training examples.

This research in a simple view provided a methodology that provides a security solution based on K-Nearest Neighbor method and SVM. For reaching a better level in evaluation on unknown attacks, in the proposed methodology the detection of suspicious traffic using the clustering strategy well be tested integrating the SVM filter on them. Following attractive points is interesting in proposed method

1. As a first step there is a process of classifying the network traffic using SVM (support Vector Machine)

2. Then as a second step by applying, clustering based detection as a stage and prevention of intrusion on real time traffic as another stage instead of KNN.

Proposed Work

Firstly, data from NSL-KDD are used, where the dimension and feature selection is reduced using the PCA. The data are then divided into two sets; training and testing sets Then apply the Control Chart on the training and testing dataset where both the SVM and KNN classifiers are applied on the training and testing datasetto measure the IDS performance. Where the FP,FN,TP,TN,DR, and CR metrics are computed. The performance of both classifiers is compared to determine the optimal classifier that offer the lowest FP and FN. The following **Figure 3.4** shows the flowchart of proposed IDS system based PCA – SVM and PCA – KNN

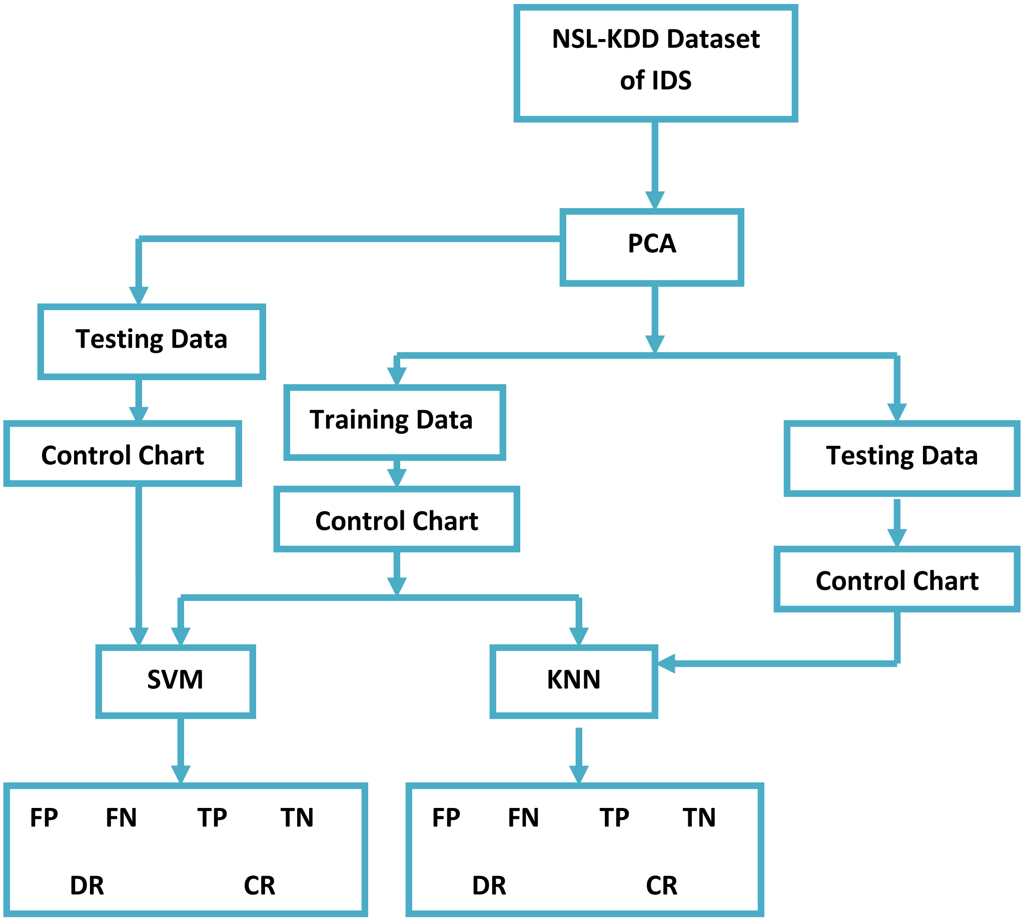


Figure 3.4Diagram of proposed IDS system based PCA – SVM and PCA – KNN

Chapter Four:
Results and Discussion

Chapter Four: Results and Discussion

4.1. Introduction

In this thesis, an improved IDS is implemented using two classifiers; PCA-SVM and PCA-KNN, where a comparison is conducted among them using the NSL-KDD dataset to determine the optimal classifier in terms of accuracy and the number of generated false alarms. The used dataset is divided into training and testing sets, training set records around 40000 that includes 22000 normal records and 1800 attack ones, while testing set records around 8000 where both classifiers are applied on them to evaluate the IDS performance. In the implemented system, data are classified into normal and attack ones usingsix evaluation metrics; FP, FN, TP, TN, DR, and CR are computed after using the control chart for three sets of features.

In this thesis, the PCA is applied to reduce the number of features in the presented dataset, which includes 41 features based on the Eigen values. The PCA then offers the most effective 11 features from the presented dataset. This number is then reduced to 9 features based on removing the least effective two features and then to 7 features in the same process.

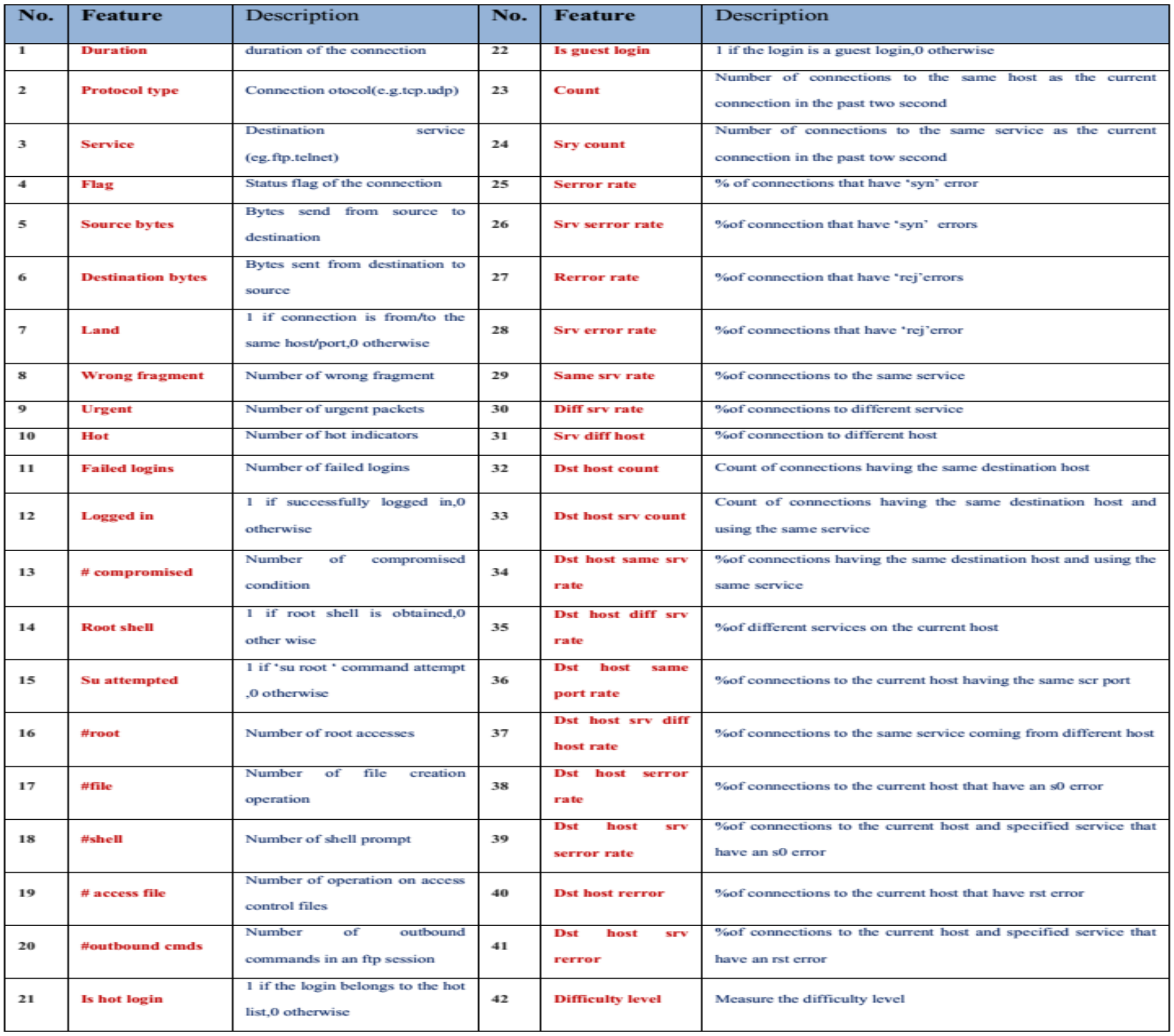
The process of reducing the data dimensionality based on the Eigen values of features depends on computing the covariance matrix for the training set of data. After that, the Eigen values are computed and sorted in a decreasing order. The first computed Eigen value is related to the first principal component, the second value is related to the second principal component and so on. The effectiveness of these features on the system is tested based on applying two types of tests; screen plot and critical Eigen value tests. In the screen plot, principal components are plotted against the difference among each two consecutive principal Eigen values. Sets of principal components that have decreasing differences between successive Eigen values are determined. But, this test results in more than one set, thus, the critical Eigen value test is applied to verify the screen plot test results. This test selects all principal components that have Eigen values bigger than a specific threshold.

After that, the percentage of how each set accounts for the total variation related to the 41 original features is computed, where the set with the highest percentage is then chosen.

4.2 Dataset

This dataset composed of 41 network connection features, where the names of those features are demonstrated in this research. The NSL-KDD dataset can be downloaded from (http://iscx.ca/NSL-KDD/). The proposed classification methods are applied on the proposed IDSs using NSL-KDD dataset.Where the training set records around 40000 that includes 22000 normal records and 1800 attack ones, while testing set records around 8000 where both classifiers are applied on them to evaluate the IDS performance. The following **Table 4.1** shows the dataset (kayaci et al, 2005).

Table 4.1Dataset features



No.	Feature	Description	No.	Feature	Description
1	**Duration**	duration of the connection	22	**Is guest login**	1 if the login is a guest login,0 otherwise
2	**Protocol type**	Connection otocol(e.g.tcp.udp)	23	**Count**	Number of connections to the same host as the current connection in the past two second
3	**Service**	Destination service (eg.ftp.telnet)	24	**Sry count**	Number of connections to the same service as the current connection in the past tow second
4	**Flag**	Status flag of the connection	25	**Serror rate**	% of connections that have 'syn' error
5	**Source bytes**	Bytes send from source to destination	26	**Srv serror rate**	%of connection that have 'syn' errors
6	**Destination bytes**	Bytes sent from destination to source	27	**Rerror rate**	%of connection that have 'rej'errors
7	**Land**	1 if connection is from/to the same host/port,0 otherwise	28	**Srv error rate**	%of connections that have 'rej'error
8	**Wrong fragment**	Number of wrong fragment	29	**Same srv rate**	%of connections to the same service
9	**Urgent**	Number of urgent packets	30	**Diff srv rate**	%of connections to different service
10	**Hot**	Number of hot indicators	31	**Srv diff host**	%of connection to different host
11	**Failed logins**	Number of failed logins	32	**Dst host count**	Count of connections having the same destination host
12	**Logged in**	1 if successfully logged in,0 otherwise	33	**Dst host srv count**	Count of connections having the same destination host and using the same service
13	**# compromised**	Number of compromised condition	34	**Dst host same srv rate**	%of connections having the same destination host and using the same service
14	**Root shell**	1 if root shell is obtained,0 other wise	35	**Dst host diff srv rate**	%of different services on the current host
15	**Su attempted**	1 if 'su root ' command attempt ,0 otherwise	36	**Dst host same port rate**	%of connections to the current host having the same scr port
16	**#root**	Number of root accesses	37	**Dst host srv diff host rate**	%of connections to the same service coming from different host
17	**#file**	Number of file creation operation	38	**Dst host serror rate**	%of connections to the current host that have an s0 error
18	**#shell**	Number of shell prompt	39	**Dst host srv serror rate**	%of connections to the current host and specified service that have an s0 error
19	**# access file**	Number of operation on access control files	40	**Dst host rerror**	%of connections to the current host that have rst error
20	**#outbound cmds**	Number of outbound commands in an ftp session	41	**Dst host srv rerror**	%of connections to the current host and specified service that have an rst error
21	**Is hot login**	1 if the login belongs to the hot list,0 otherwise	42	**Difficulty level**	Measure the difficulty level

The following table illustrates the values of Eigen value for 41 features. As shown, the shadow blocks includes highest Eigen value, in our study we take highest 11 value. The following**Table 4.2**shows the output of 41 features after applying the PCA

Table 4.2output of 41 features after applying the PCA

Features #	Name	Eigen value	Features #	Name	Eigen value
1	Duration	$5.96*10^{-19}$	22	is_guest_login	0.0246
2	protocol_type	$8.86*10^{-06}$	23	Count	$1.1297*10^{4}$
3	Service	$0.00017*10^{-4}$	24	srv_count	$6.188*10^{3}$
4	Flag	$1.043*10^{11}$	25	serror_rate	0.0447
5	src_bytes	$2.168*10^{9}$	26	srv_serror_rate	0.0615
6	dst_bytes	4.5431*10-4	27	rerror_rate	0.232
7	Land	4.950*10-4	28	srv_error_rate	0.4126
8	wrong_fragment	8.744*10-4	29	same_srv_rate	$5.020*10^{3}$
9	Urgent	0.0013	30	diff_srv_rate	1.176
10	Hot	$1.0168*10^{6}$	31	srv_diff_host_rate	$1.608*10^{3}$
11	num_failed_logins	$2.532*10^{5}$	32	dst_host_count	0.00225
12	logged_in	0.00246	33	dst_host_srv_count	$1.263*10^{3}$
13	num_compromised	0.00247	34	dst_host_same_srv_rate	$2.303*10^{-4}$
14	root_shell	0.0029	35	dst_host_diff_srv_rate	0.73915
15	su_attempted	0.00318	36	dst_host_same_src_port_rate	0.036074
16	num_root	0.0052	37	dst_host_srv_diff_host_rate	0.03055
17	num_file_creations	0.0059	38	dst_host_serror_rate	3.2156
18	num_shells	0.0067	39	dst_host_srv_serror_rate	0.00161
19	num_access_files	0.0068	40	dst_host_rerror_rate	4.180*10-4
20	num_outbound_cmds	0.0103	41	dst_host_srv_error_rate	2.8067
21	is_hot_login	0.0154			

The resultant most effective 11 features, which have the highest Eigen values after applying the PCA. Other features are considered as noisy ones. The following **Table 4.3** shows the best output 11 features after applying the PCA

Table 4.3Best output 11 features after applying the PCA

Features #	Name	Eigen value
4	Flag	$\mathbf{1.043*10^{11}}$
5	src_bytes	$\mathbf{2.168*10^{9}}$
10	Hot	$\mathbf{1.0168*10^{6}}$
11	num_failed_logins	$\mathbf{2.532*10^{5}}$
23	Count	$\mathbf{1.1297*10^{4}}$
24	srv_count	$\mathbf{6.188*10^{3}}$
29	same_srv_rate	$\mathbf{5.020*10^{3}}$
31	srv_diff_host_rate	$\mathbf{1.608*10^{3}}$
33	dst_host_srv_count	$\mathbf{1.263*10^{3}}$
38	dst_host_serror_rate	**3.2156**
41	dst_host_srv_error_rate	**2.8067**

Thus, the first set of features includes 11 features; F1: [4,5,10,11,23,24,29,31,33, 38,41]. As shown in the **Table4.3** above, the two features that have the less Eigen values are 38 and 41. Therefore, these two features are removed to have a second set of features; F2: [4,5,10,11,23,24,29, 31,33]. The same process is applied then in this set where the two features that have the less Eigen values are 31 and 33. Therefore, these two features are removed to have a third set of features; 7 features F3: [4,5,10,11,23, 24,29]. In order to compute main parameters without control chart and without PCA (41 features), the following **Table 4.4** illustrates initial results

Table 4.4Results of applying the SVM &KNN without PCA

	TN	FP	TP	FN	DR	CR
SVM	80.5%	19.5%	84.2%	15.8%	84.2%	82.35%
KNN	77.1%	22.9%	80.2%	19.8%	80.2%	78.65%

4.3. Results without Control Chart

The following subsections demonstrate the obtained results of both PCA-SVM and PCA-KNN without applying the control chart

4.3.1 Results of PCA-KNN Based IDS

The following subsections demonstrate the achieved results of the PCA-KNN based IDS for the proposed three sets of features without applying the control chart.

4.3.1.1 Results of Applying the PCA-KNNBased IDS on F1

The obtained results of applying the PCA-KNN based IDS without control chart on F1 that includes 11 features from the NSL-KDD dataset; [4,5,10,11,23,24,29,31,33, 38,41].As shown in the following **Figure 4.1**represents the results of applying the PCA-KNN based IDS on F1 without control chart andthe **Table 4.5** represents the Results of applying the PCA-KNN based IDS on F1 without control chart.

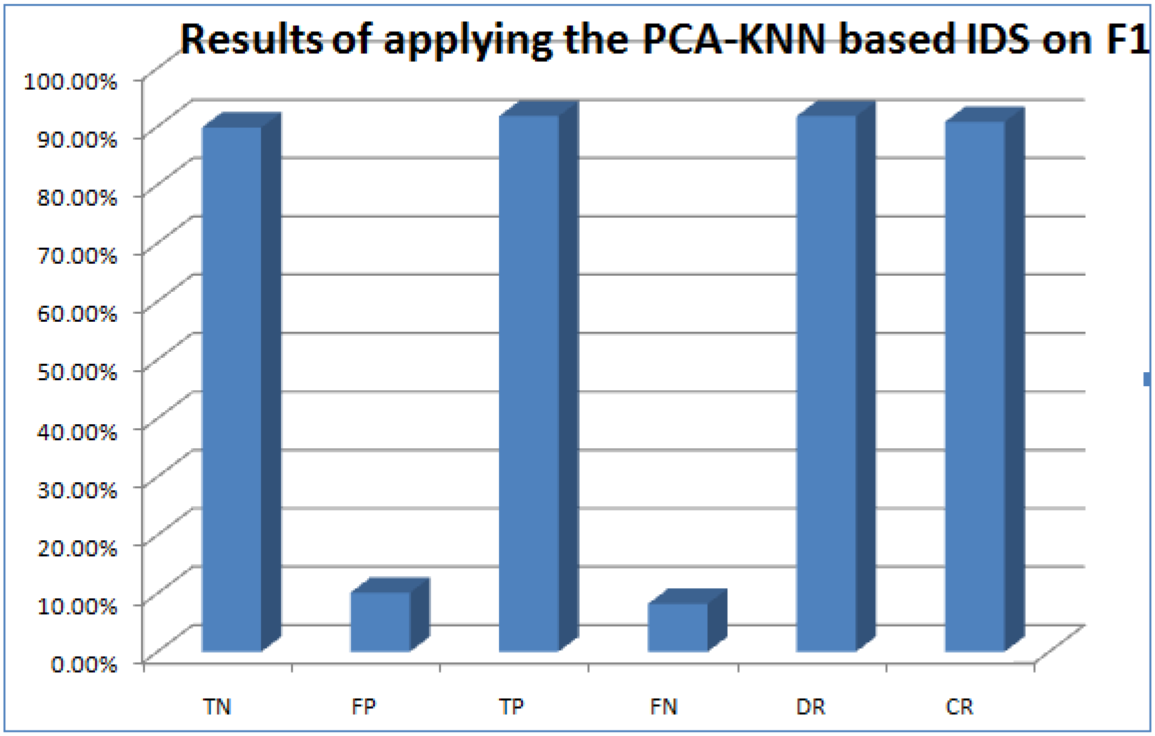


Figure4.1Results of applying the PCA-KNN based IDS on F1 without control chart

Table 4.5Results of applying the PCA-KNN based IDS on F1 without control chart

TN	FP	TP	FN	DR	CR
89.8545%	10.1455%	91.7786%	8.2214%	91.7786%	90.8165%

The results above demonstrate that the system has 8.2214% and 10.1455% FN and FP percentages; respectively, which stand for false alarms. Thus, the related records for these alarms should be removed from the dataset. Conversely, the system has 91.7786% and 90.8165% detection and classificationrates, respectively.

4.3.1.2 Results of Applying the PCA-KNNBased IDS on F2

The following **Figure4.2** and **Table4.6** show the obtained results of applying the PCA-KNN based IDS without control chart on F2 that includes 9 features from the NSL-KDD dataset;[4,5,10,11,23,24,29,31,33].

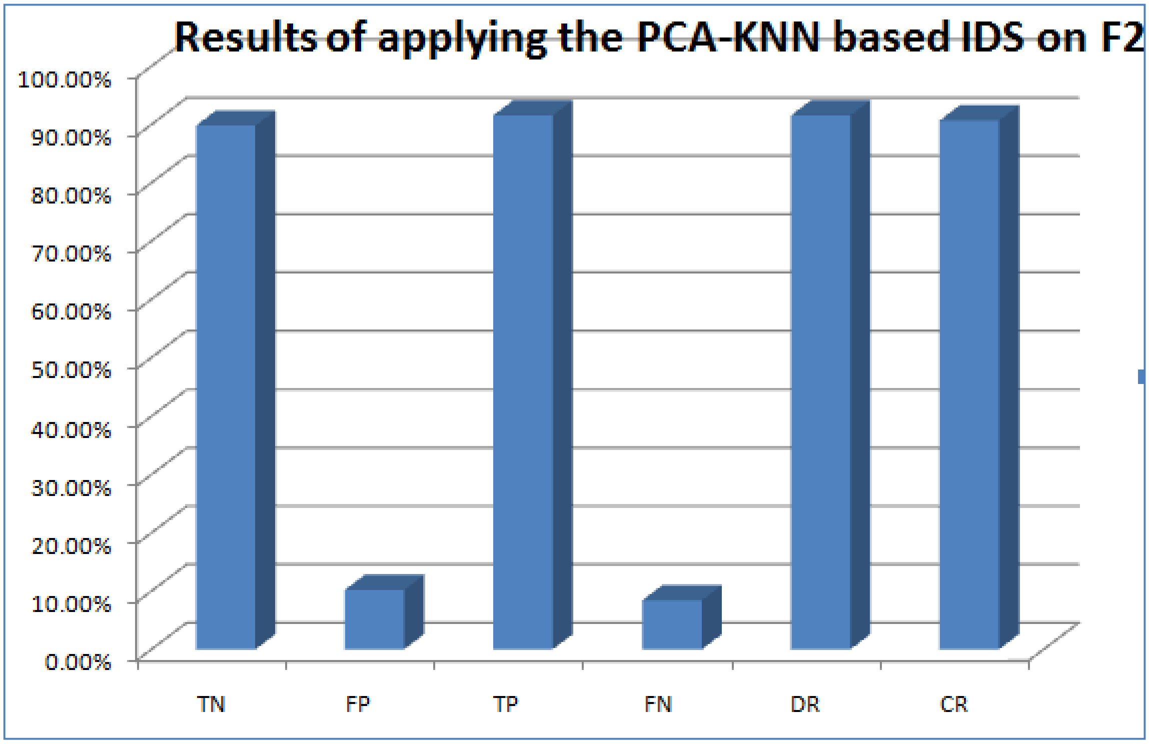


Figure 4.2Results of applying the PCA-KNN based IDS on F2 without control chart

Table 4.6Results of applying the PCA-KNN based IDS on F2 without control chart

TN	FP	TP	FN	DR	CR
89.8165%	10.1835%	91.5718%	8.4282%	91.5718%	90.6942%

It can be noticed that the system has 8.4282% and 10.1835% FN and FP percentages; respectively. Conversely, the system has 91.5718% and 90.6942% detection and classificationrates, respectively.

4.3.1.3 Results of Applying the PCA-KNNBased IDS on F3

This section demonstrate the obtained results of applying the PCA-KNN based IDS without control chart on F3, which includes 7 features from the NSL-KDD dataset;[4,5,10,11,23,24,29]. The following **Figure4.3** and **Table4.7** show the achieved outcomes.

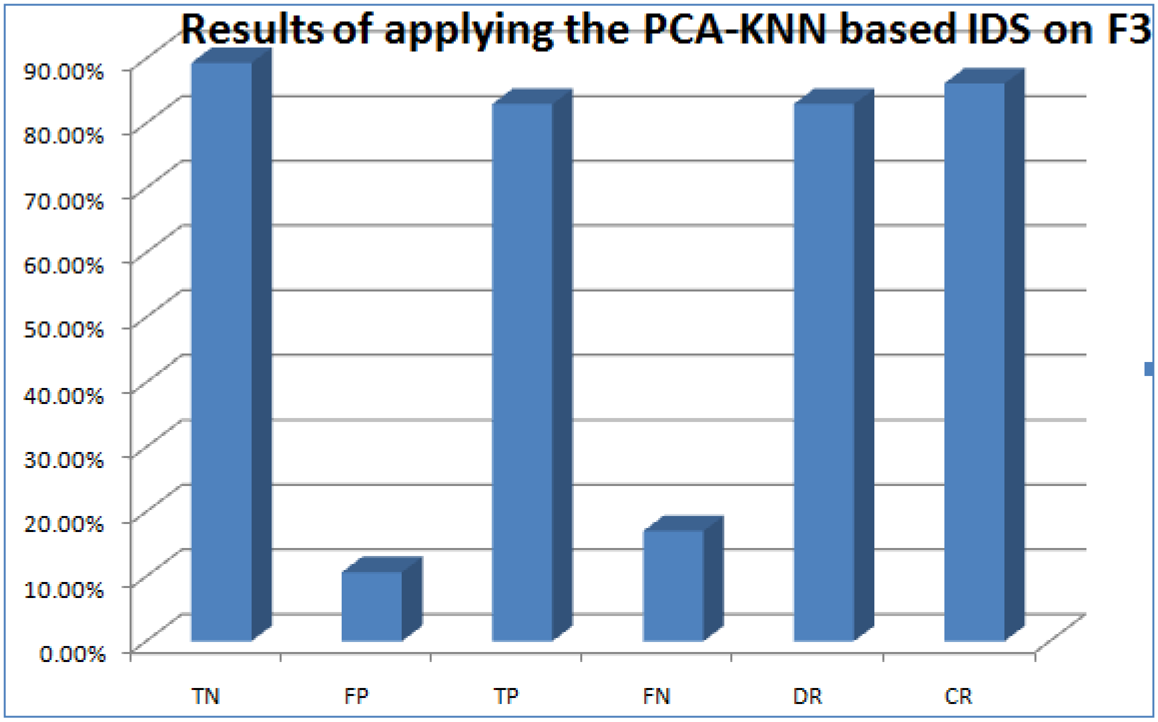


Figure 4.3Results of applying the PCA-KNN based IDS on F3 without control chart

Table 4.7Results of applying the PCA-KNN based IDS on F3 without control chart

TN	FP	TP	FN	DR	CR
89.3335%	10.6665%	82.9710%	17.0290%	82.9710%	86.1522%

4.3.1.4 Comparison between Three Cases

The **Table4.8** and **Figure4.4**below illustrate a comparison among the three cases in terms of FP and FN for the PCA-KNN based IDS without control chart.

Table 4.8Comparison between the presented three cases in terms of FP and FN for PCA-KNN based IDS without control chart

Features set	FP	FN
F1: [4,5,10,11,23,24,29,31,33, 38,41]	10.1455%	8.2214%
F2: [4,5,10,11,23,24,29,31,33]	10.1835%	8.4282%
F3: [4,5,10,11,23,24,29]	10.6665%	17.0290%

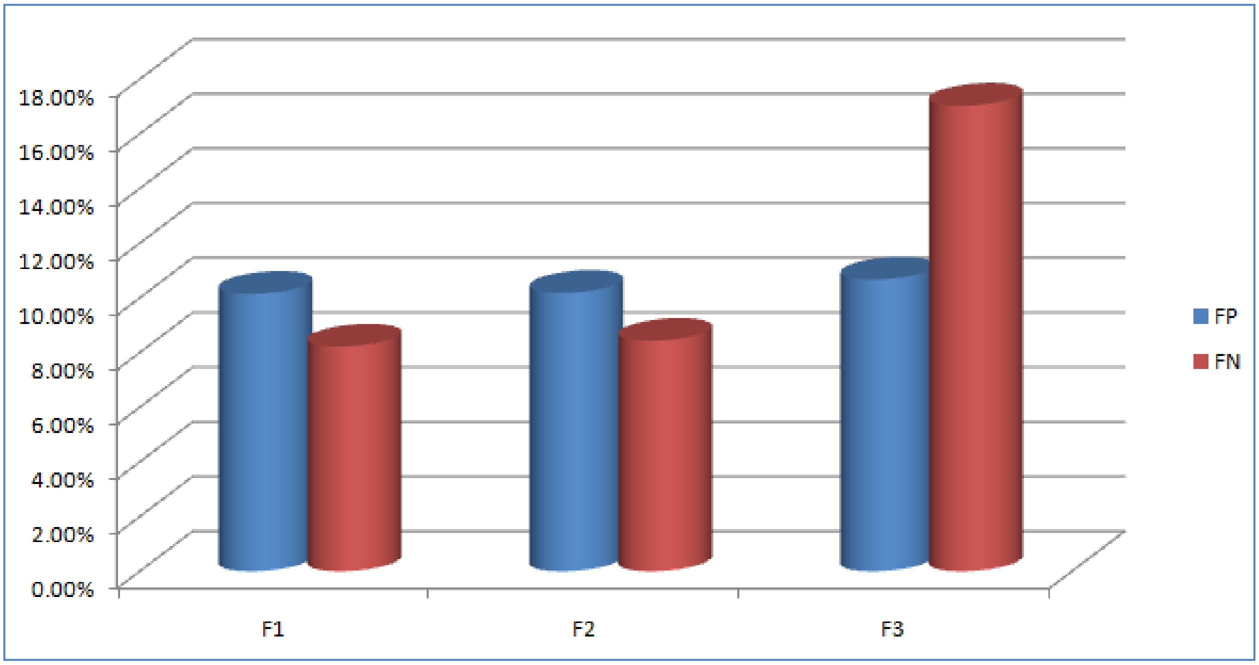


Figure4.4Comparison between the presented three cases in terms of FP and FN for PCA-KNN based IDS without control chart

As shown above, the lowest achieved FP and FN percentages are for F1. Thus, the PCA-KNN based IDS without control chart offers the minimum number of false alarms with using the first set of features

The **Table4.9** and **Figure4.5** below illustrate a comparison between the three cases in terms of DR and CR for the PCA-KNN based IDS without control chart.

Table 4.9Comparison between the presented three cases in terms of DR and CR for PCA-KNN based IDS without control chart

Features set	DR	CR
F1: [4,5,10,11,23,24,29,31,33, 38,41]	91.7786%	90.8165%
F2: [4,5,10,11,23,24,29,31,33].	91.5718%	90.6942%
F3: [4,5,10,11,23,24,29]	82.9710%	86.1522%

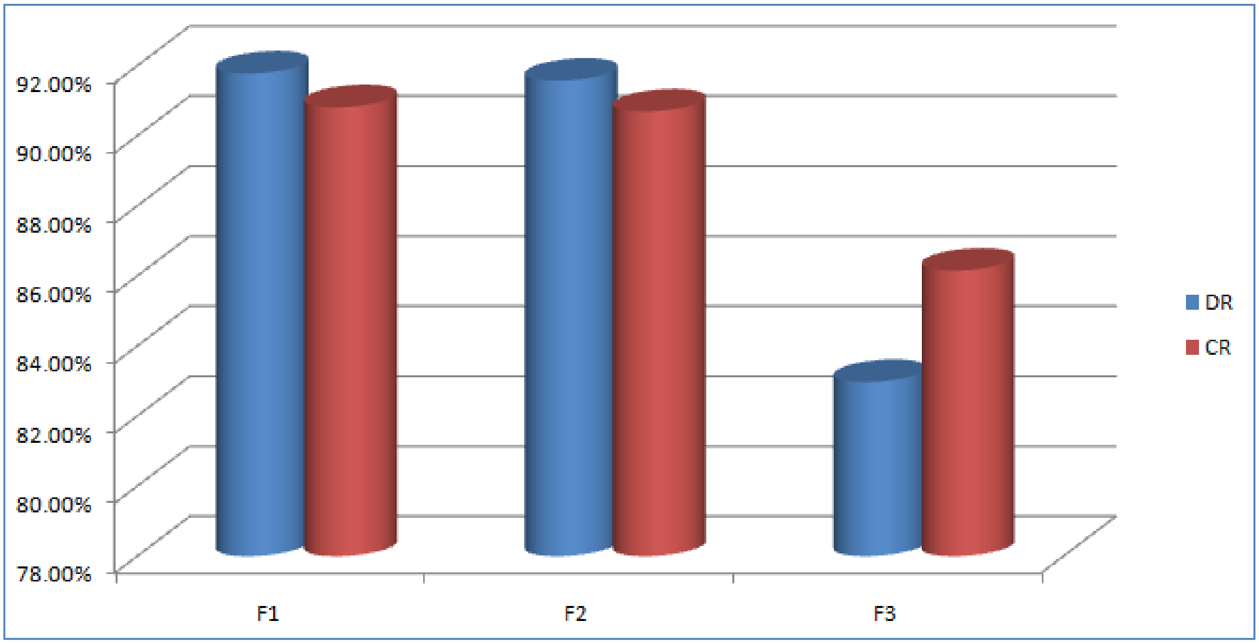


Figure 4.5Comparison between the presented three cases in terms of DR and CR for PCA-KNN based IDS without control chart

It can be seen that the highest percentages of DR and CR are for F1. Therefore, the PCA-KNN based IDS without control chart provides the minimum number of false alarms and the highest detection and classification rates with applying the first set of features.

4.4 Results of PCA-SVM Based IDS

The following subsections demonstrate the obtained results of applying the PCA-SVM based IDS without control chart on the presented three sets of features.

4.4.1 Results of Applying the PCA-SVMBased IDS on F1

This subsection illustrate the obtained results after applying the PCA-SVM based IDS without control chart on F1;[4,5,10,11,23,24,29,31,33, 38,41]. The obtained results are shown in the **Figure4.6** and **Table4.10** below.

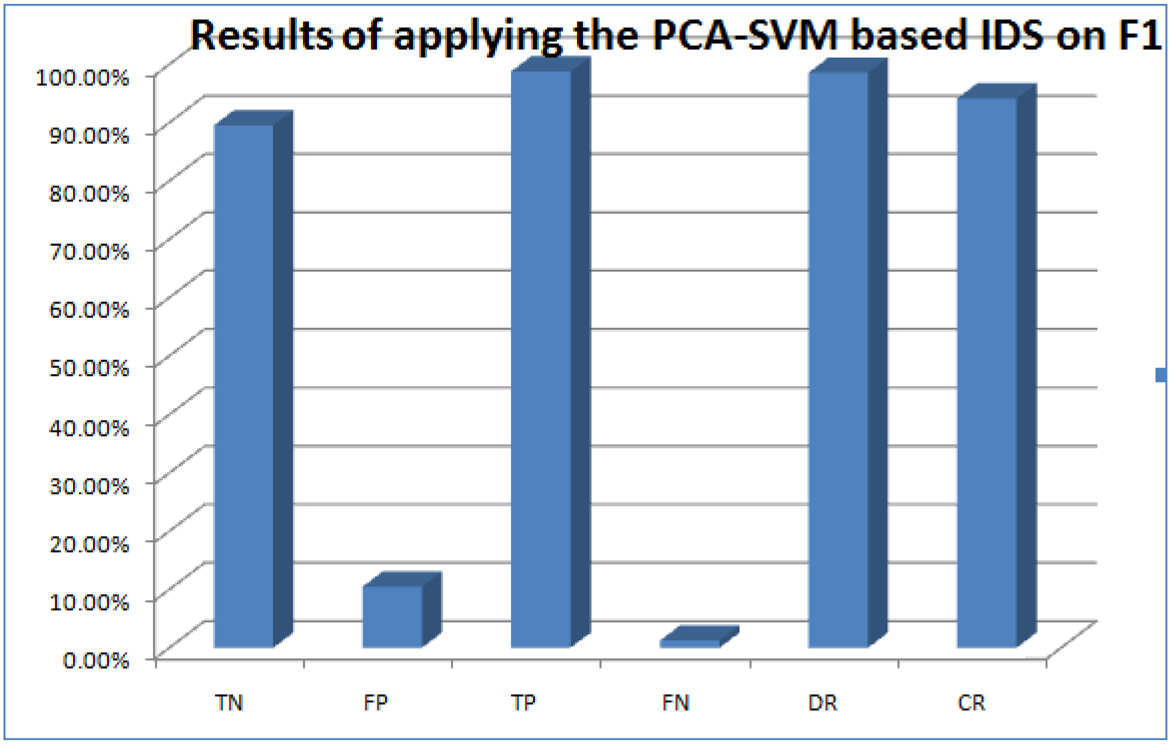


Figure 4.6Results of applying the PCA-SVM based IDS on F1 without control chart

Table 4.10Results of applying the PCA-SVM based IDS on F1 without control chart

TN	FP	TP	FN	DR	CR
89.5323%	10.4677%	98.7525%	1.2475%	98.5725%	94.1424%

As shown above, the system has 1.2475% and 10.4677% FN and FP percentages; respectively, where these percentages stand for false alarms. In contrast, the system has 98.5725% and 94.1424% detection and classificationrates, respectively.

4.4.2 Results of Applying the PCA-SVMBased IDS on F2

The obtained results of applying the PCA-SVM based IDS without control chart on F2;[4,5,10,11,23,24,29,31,33]are demonstrated in the following **Figure 4.7** and **Table4.11**.

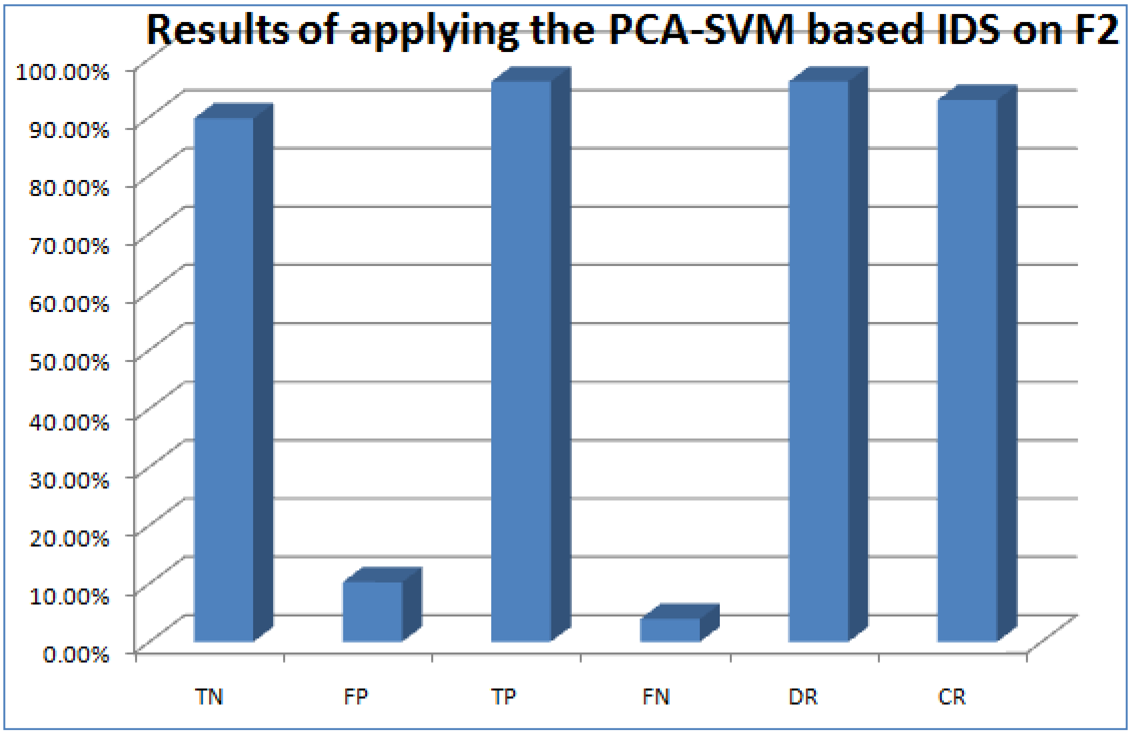


Figure 4.7Results of applying the PCA-SVM based IDS on F2 without control chart

Table 4.11Results of applying the PCA-SVM based IDS on F2 without control chart

TN	FP	TP	FN	DR	CR
89.7574%	10.2426%	96.0800%	3.9200%	96.0800%	92.9187%

As illustrated above, the system has 3.9200% and 10.2426% FN and FP percentages; respectively, where these percentages stand for false alarms, while it has 96.0800% and 92.9187% detection and classificationrates, respectively.

4.4.3 Results of Applying the PCA-SVM Based IDS on F3

This section illustrates the achieved results of applying the PCA-SVM based IDS without control chart on F3;[4,5,10,11,23,24,29]. The **Figure4.8** and **Table4.12** below show the obtained results.

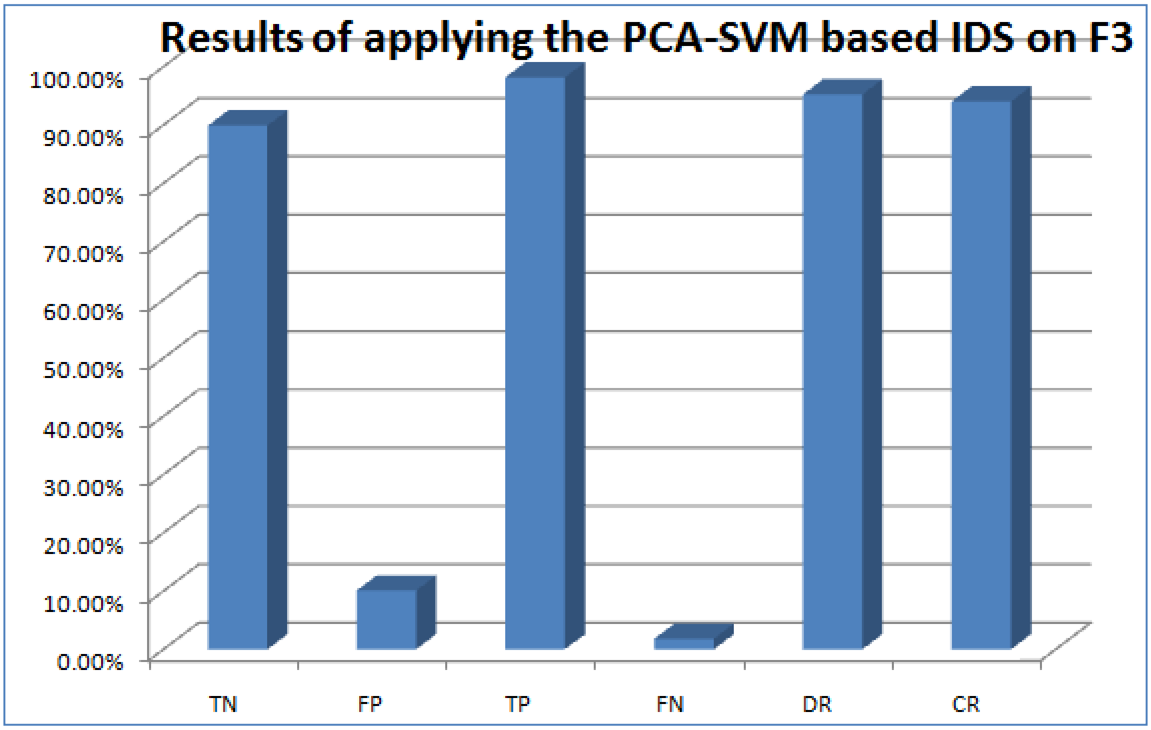


Figure 4.8Results of applying the PCA-SVM based IDS on F3 without control chart

Table 4.12Results of applying the PCA-SVM based IDS on F3 without control chart

TN	FP	TP	FN	DR	CR
89.9131%	10.0869%	98.1779%	1.8221%	95.2551%	94.0455%

It can be seen that the system has 1.8221% and 10.0869% FN and FP percentages; respectively, where these percentages stand for false alarms. In contrast, the system has 95.2551% and 94.0455% detection and classificationrates, respectively.

4.4.4 Comparison between Three Cases

The following**Table4.13** and **Figure4.9**show a comparison between the presented three cases in terms of FP and FN for the PCA-SVM based IDS without control chart.

Table 4.13Comparison between the presented three cases in terms of FP and FN for PCA-SVM based IDS without control chart

Features set	FP	FN
F1: [4,5,10,11,23,24,29,31,33, 38,41]	10.4677%	1.2475%
F2: [4,5,10,11,23,24,29,31,33].	10.2426%	3.9200%
F3: [4,5,10,11,23,24,29]	10.0869%	1.8221%

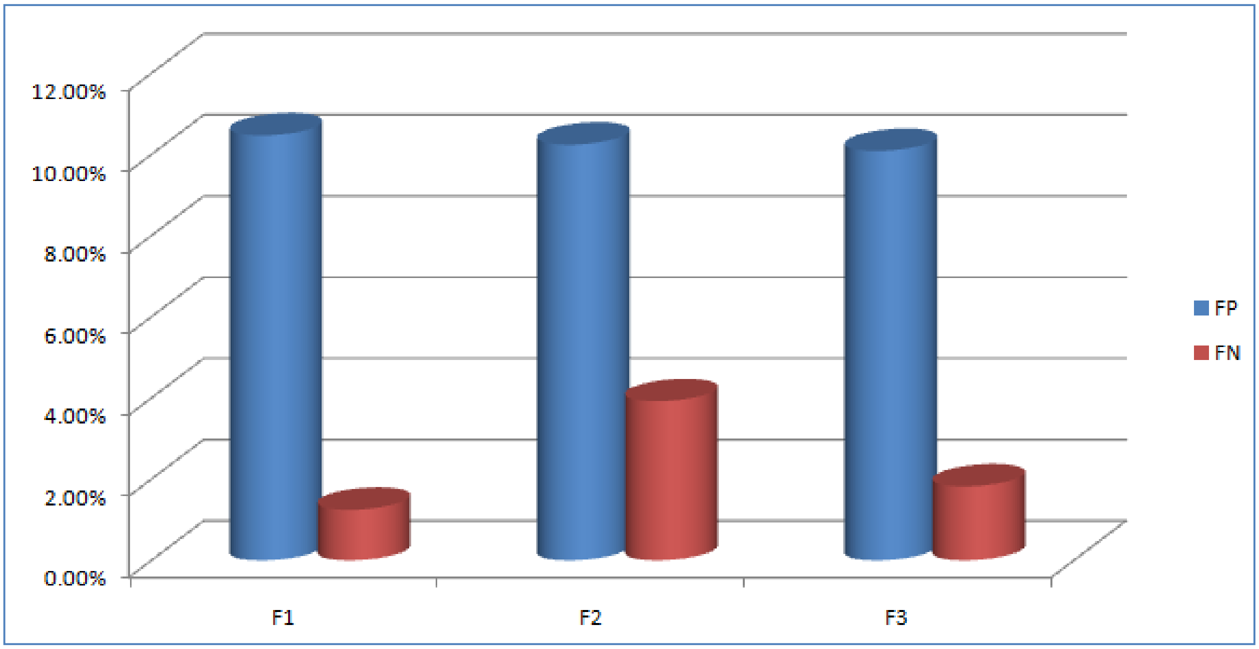


Figure 4.9Comparison between the presented three cases in terms of FP and FN for PCA-SVM based IDS without control chart

The following **Table4.14** and **Figure4.10** below show a comparison between the presented three cases in terms of DR and CR for the PCA-SVM based IDS without control chart

Table 4.14Comparison between the presented three cases in terms of DR and CR for PCA-SVM based IDS without control chart

Features set	DR	CR
F1: [4,5,10,11,23,24,29,31,33, 38,41]	98.5725%	94.1424%
F2: [4,5,10,11,23,24,29,31,33].	96.0800%	92.9187%
F3: [4,5,10,11,23,24,29]	95.2551%	94.0455%

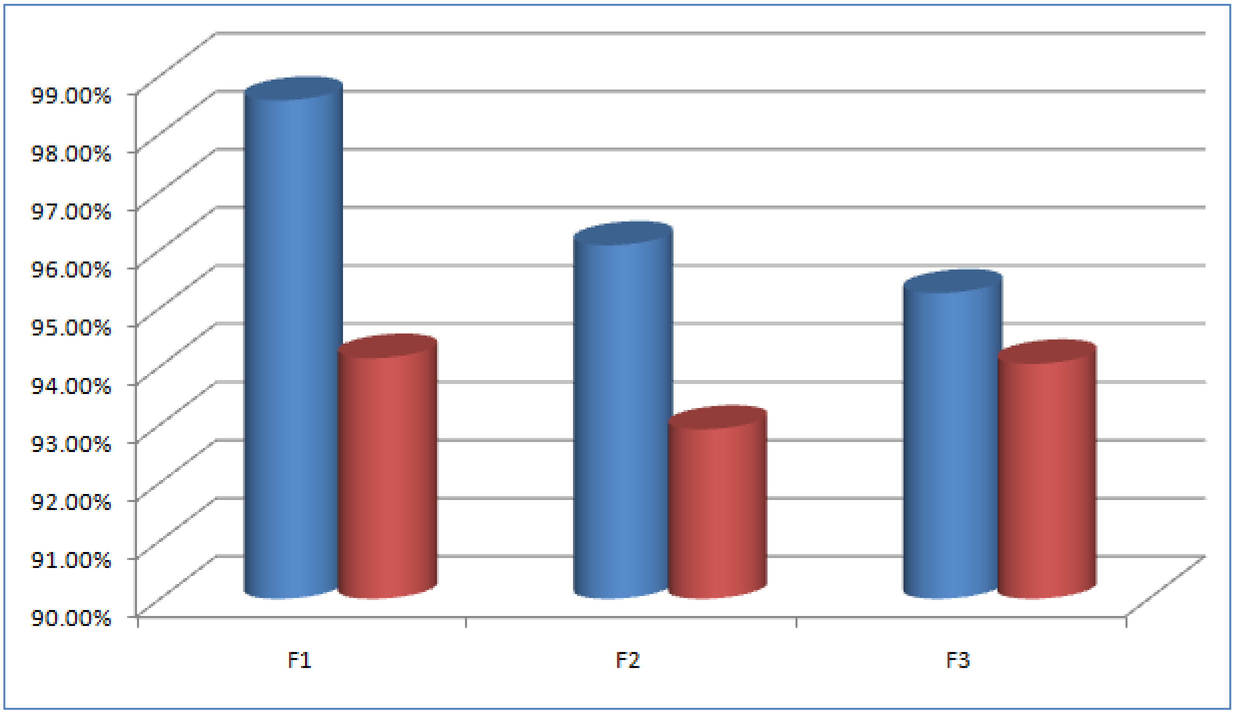


Figure 4.10Comparison between the presented three cases in terms of DR and CR for PCA-SVM based IDS without control chart

It can be noticed that the highest percentages of DR and CR are for F1. Thus, the PCA-SVM based IDS offers the minimum number of false alarms and the highest detection and classification rates for the first set of features.

4.5 Comparison without Applying PCA

The **Table4.15**below shows a comparison between the KNN without PCA and KNN-PCA without applying the control chart.

Table 4.15Comparison between the KNN without PCA and KNN-PCA without applying the control chart

	TN	FP	TP	FN	DR	CR
KNN without PCA	77.1%	22.9%	80.2%	19.8%	80.2%	78.65%
KNN-PCA without CC	89.8545%	10.1455%	91.7786%	8.2214%	91.7786%	90.8165%

As shown above the use of KNN-PCA without control chart offers higher detection and classification rates with smaller number of generated false alarms than the KNN without PCA.

The **Table4.16** below illustrates another comparison between the SVM without PCA and SVM-PCA without applying the control chart.

Table 4.16Comparison between the SVM without PCA and SVM-PCA without applying the control chart

	TN	FP	TP	FN	DR	CR
SVM without PCA	80.5%	19.5%	84.2%	15.8%	84.2%	82.35%
SVM-PCA without CC	89.5323%	10.4677%	98.7525%	1.2475%	98.5725%	94.1424%

It is obvious that the use of SVM-PCA without control chart offers higher detection and classification rates with smaller number of generated false alarms than the SVM without PCA.

4.6 Results with Control Chart

4.6.1 Application of Control Chart

The Control Chart is applied in both cases;PCA-SVM and PCA-KNNto enhance the results based on filtering the training data to remove the out-bound data and keep the data in the range from Mean-3sigma to Mean+3sigma, where sigma represents the standard deviation of the data. The following figure shows the probability of the training data before filtering using the control chart. Both red lines represent the requiredrange of data; Mean-3sigma to Mean+3sigma. As shown in the **Figure4.11**, the data exceeds the upper limit before applying the filtering. Where x-axis represents data record and y-axis amplitude of data or probability of each record.

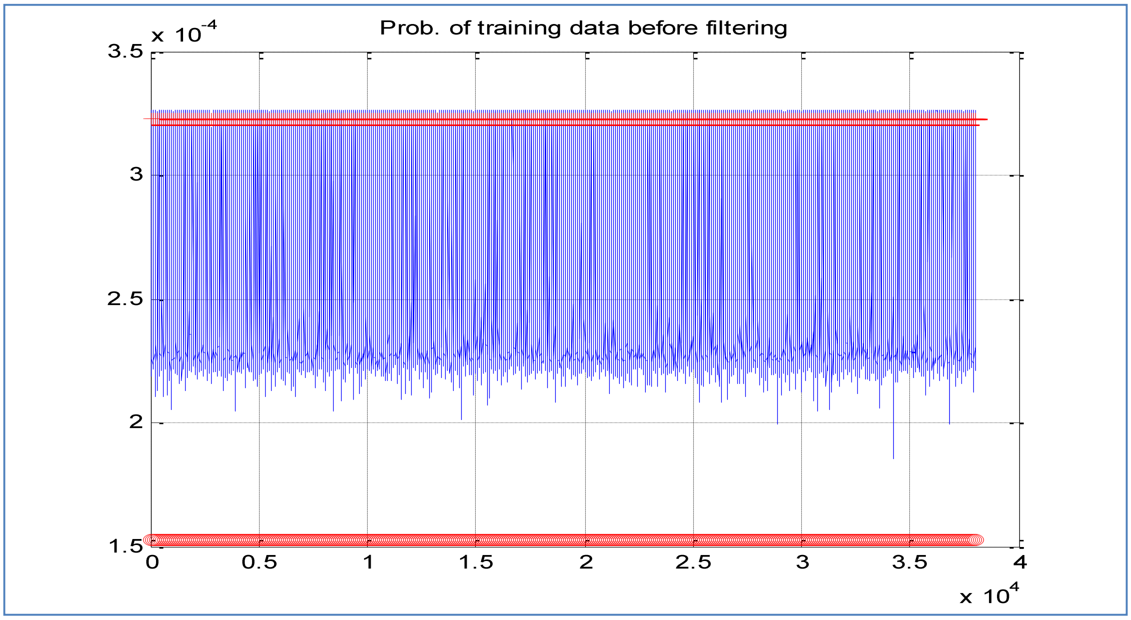


Figure 4.11Probability of the training data before filtering

The **Figure4.12** below shows the probability of the training data after filtering using the control chart. It can be clearly seen that the control chart keeps the data in the defined range.

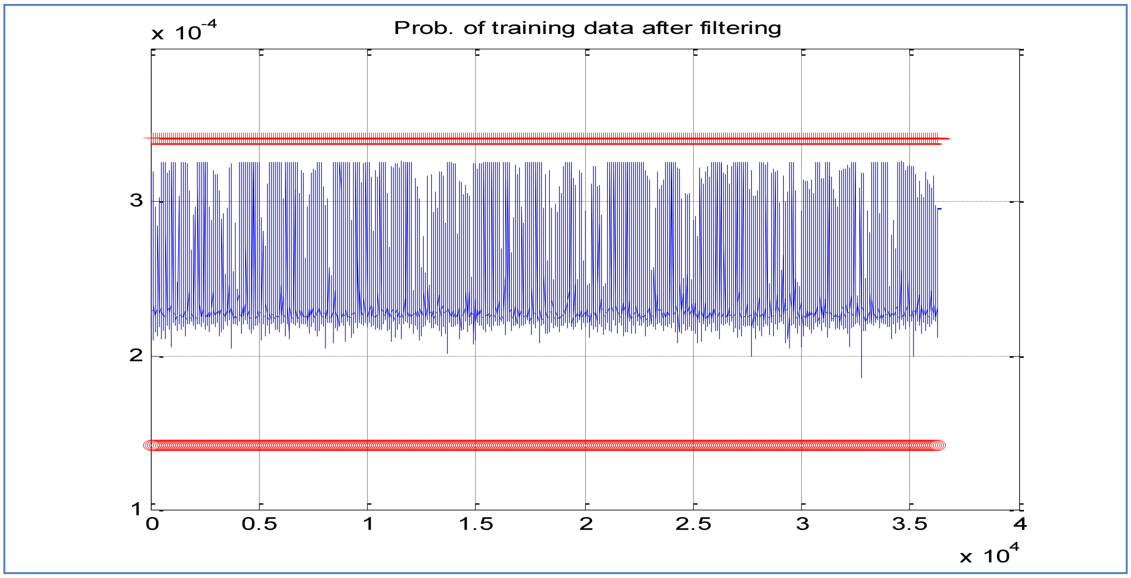


Figure 4.12Probability of the training data after filtering

The **Figure4.13** below shows the probability of the testing data.

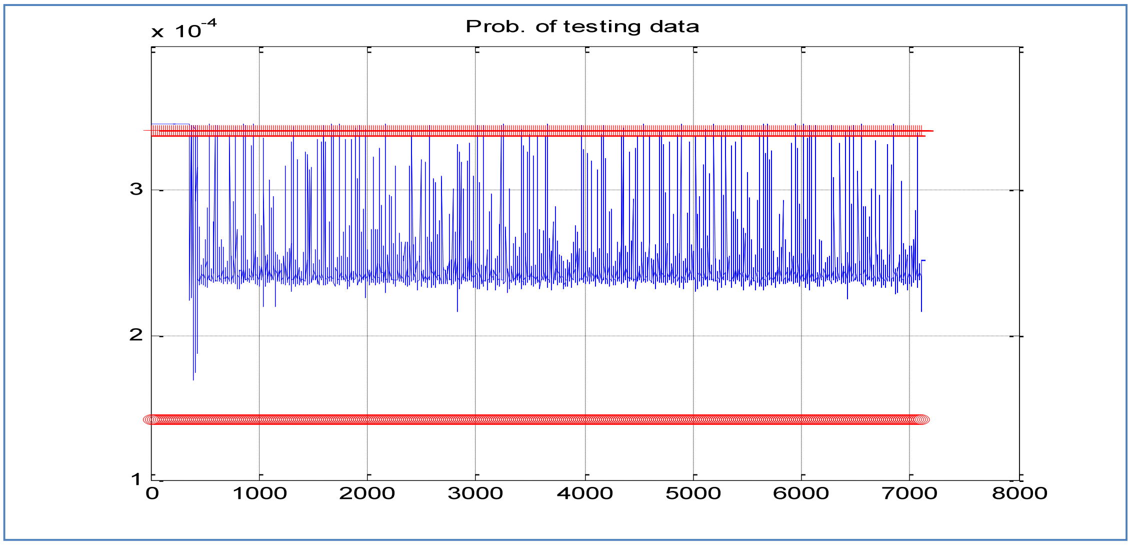


Figure 4.13Probability of the testing data

4.7 Results of PCA-KNN Based IDS

The following subsections demonstrate the obtained results of the PCA-KNN based IDS for the proposed three sets of features.

4.7.1 Results of Applying the PCA-KNNBased IDS on F1

In this subsection, the obtained results of applying the PCA-KNN based IDS with control chart on F1, which includes 11 features from the NSL-KDD dataset; [4,5,10,11,23,24,29,31,33, 38,41]. The following **Figure4.14** and **Table4.17** demonstrate the measured FP, FN, TP, TN, DR, and CR percentages for this case.

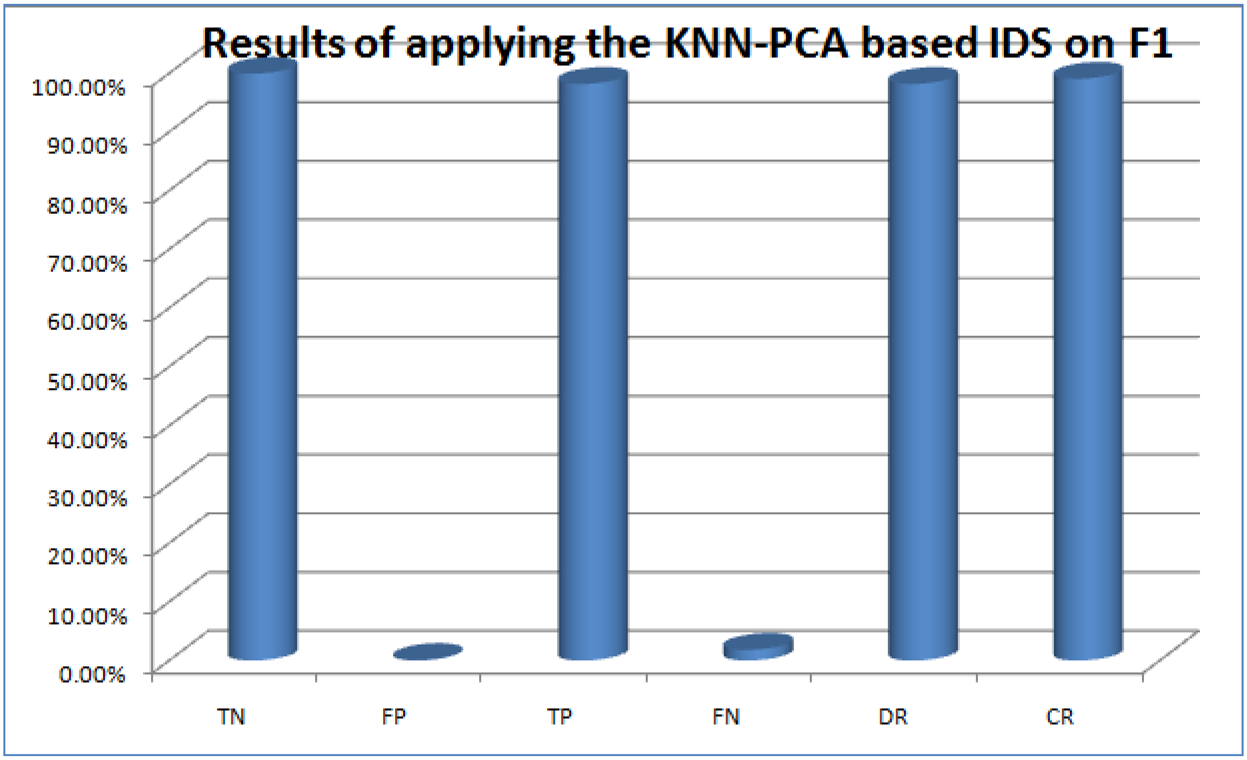


Figure 4.14Results of applying the PCA-KNN based IDS on F1 with control chart with control chart

Table 4.17Results of applying the PCA-KNN based IDS on F1 with control chart

TN	FP	TP	FN	DR	CR
99.925%	0.075%	98.1859%	1.8141%	98.1859%	99.0555%

As shown above, the system has 1.8141% and 0.075% FN and FP percentages; respectively. These percentages stand for false alarms, where the related records for these alarms must be removed from the dataset. On the other hand, the system has 98.1859% and 99.0555% detection and classificationrates, respectively.

4.7.2 Results of Applying the PCA-KNNBased IDS on F2

The obtained results of applying the PCA-KNN based IDS with control chart on F2, which includes 9 features from the NSL-KDD dataset;[4,5,10,11,23,24,29,31,33] are shown below. The following **Figure4.15** and **Table4.18** show the measured FP, FN, TP, TN, DR, and CR percentages for this case.

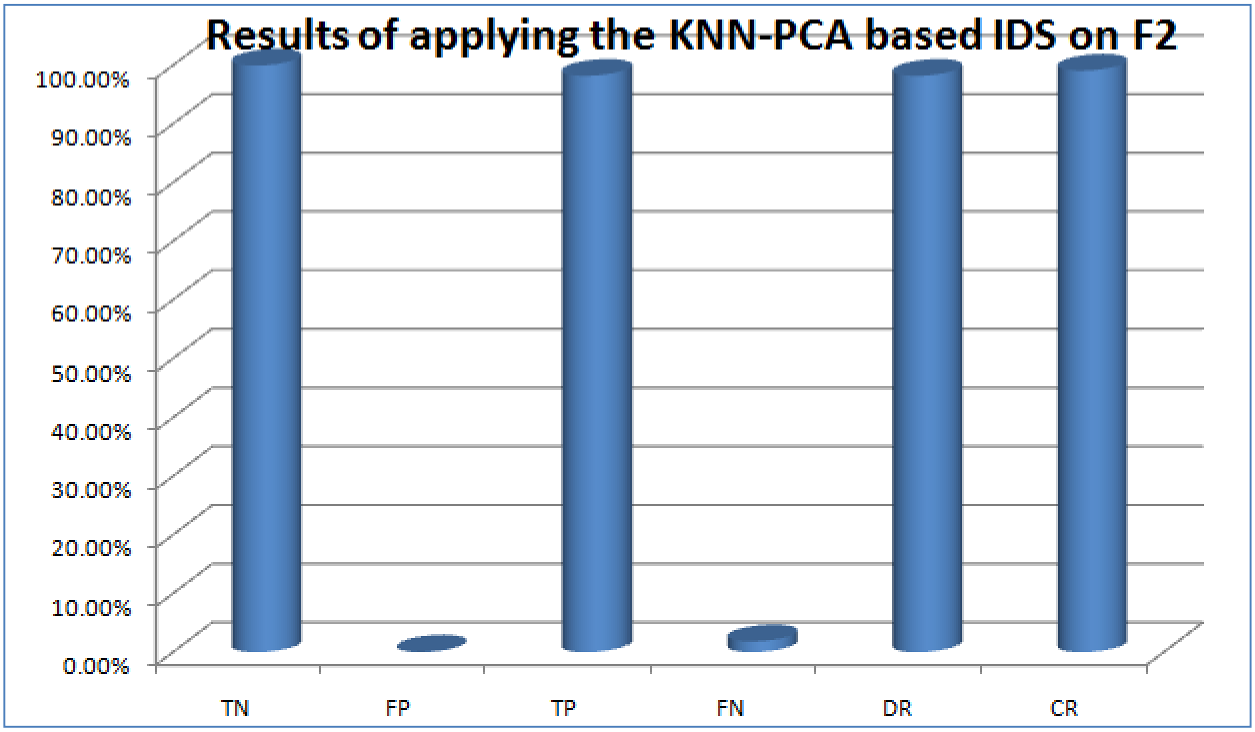


Figure4.15Results of applying the PCA-KNN based IDS on F2with control chart

Table 4.18Results of applying the PCA-KNN based IDS on F2 with control chart

TN	FP	TP	FN	DR	CR
99.94%	0.06%	98.19%	1.81%	98.19%	99.065%

The **Figure4.15** and **Table4.18** above demonstrate that the system has 1.81% and 0.06% FN and FP percentages; respectively, where these percentages stand for false alarms. In contrast, the system has 98.19% and 99.065% detection and classificationrates, respectively.

4.7.3 Results of Applying the PCA-KNNBased IDS on F3

This section illustrate the obtained results of applying the PCA-KNN based IDS with control charton F3, which includes 7 features from the NSL-KDD dataset;[4,5,10,11,23,24,29]. The following **Figure4.16** and **Table4.19** show the measured FP, FN, TP, TN, DR, and CR percentages for this case.

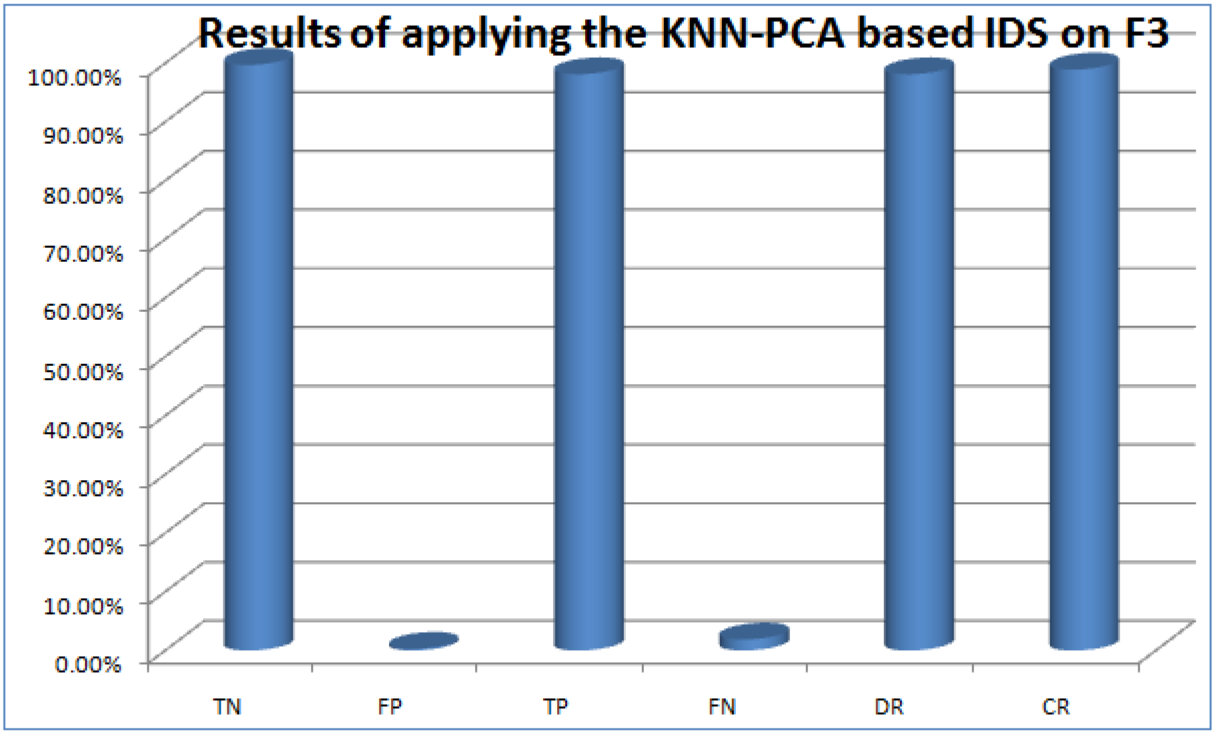


Figure 4.16Results of applying the PCA-KNN based IDS on F3 with control chart

Table 4.19Results of applying the PCA-KNN based IDS on F3 with control chart

TN	FP	TP	FN	DR	CR
99.7157%	0.2843%	98.1265%	1.8735%	98.1265%	98.9211%

It can be seen that the system has 1.8735% and 0.2843% FN and FP percentages; respectively, where these percentages stand for false alarms. In contrast, the system has 98.1265% and 98.9211% detection and classificationrates, respectively.

4.7.4 Comparison between Three Cases

The **Table4.20** and **Figure4.17** below show a comparison between the presented three cases in terms of FP and FN for the PCA-KNN based IDS with control chart.

Table 4.20Comparison between the presented three cases in terms of FP and FN for PCA-KNN based IDS with control chart

Features set	FP	FN
F1: [4,5,10,11,23,24,29,31,33, 38,41]	0.075%	1.8141%
F2: [4,5,10,11,23,24,29,31,33].	0.06%	1.81%
F3: [4,5,10,11,23,24,29]	0.2843%	1.8735%

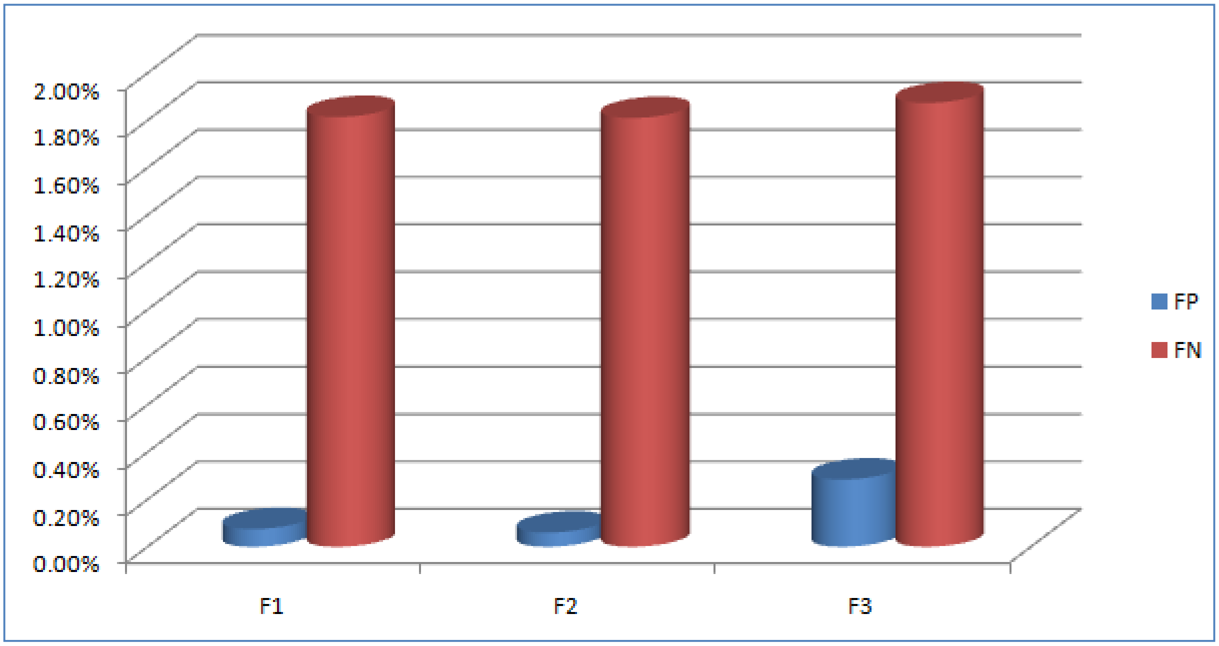


Figure 4.17Comparison between the presented three cases in terms of FP and FN for PCA-KNN based IDS with control chart

As shown above, the lowest achieved FP and FN percentages are for F2. Thus, the PCA-KNN based IDS offers the minimum number of false alarms with using the second set that has9 features.

The **Table4.21** and **Figure4.18** below show a comparison between the presented three cases in terms of DR and CR for the PCA-KNN based IDS with control chart.

Table 4.21Comparison between the presented three cases in terms of DR and CR for PCA-KNN based IDS with control chart

Features set	DR	CR
F1: [4,5,10,11,23,24,29,31,33, 38,41]	98.1859%	99.0555%
F2: [4,5,10,11,23,24,29,31,33].	98.19%	99.065%
F3: [4,5,10,11,23,24,29]	98.1265%	98.9211%

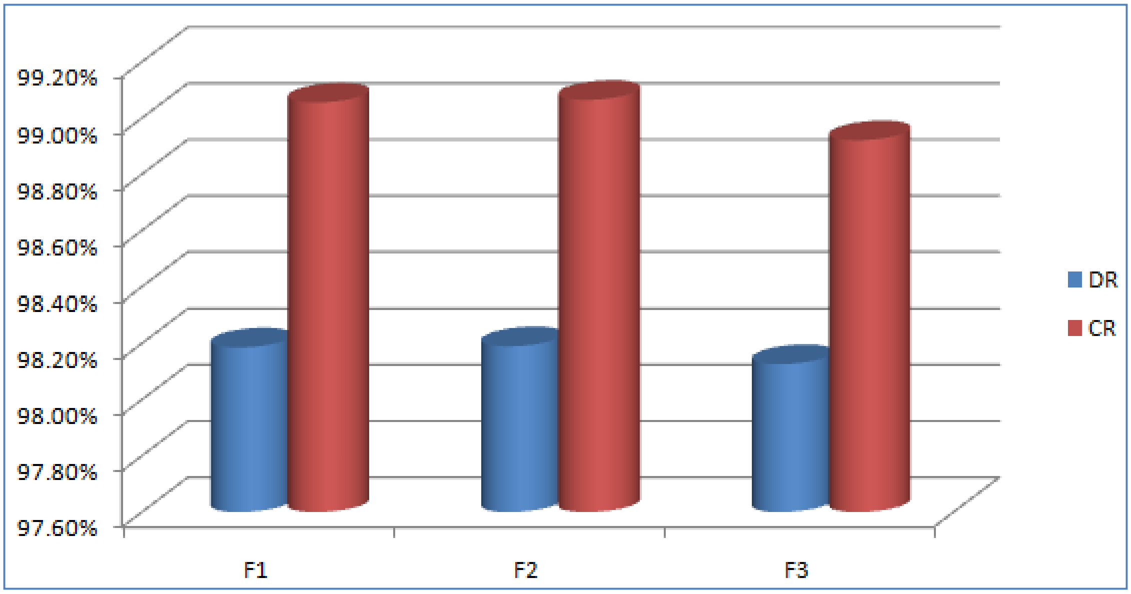


Figure 4.18Comparison between the presented three cases in terms of DR and CR for PCA-KNN based IDS with control chart

It can be noticed from the results above that the highest percentages of DR and CR are for F2. Thus, it can be summarized that the PCA-KNN based IDS offers the minimum number of false alarms and the highest detection and classification rates with using the second set that has 9 features.

4.8 Results of PCA-SVM Based IDS

4.8.1 Results of Applying the PCA-SVMBased IDS on F1

This subsection demonstrate the achieved results after applying the PCA-SVM based IDS with control charton F1, which includes 11 features from the NSL-KDD dataset;[4,5,10,11,23,24,29,31,33, 38,41]. The measured FP, FN, TP, TN, DR, and CR percentages are shown in the **Figure4.19** and **Table4.22** below.

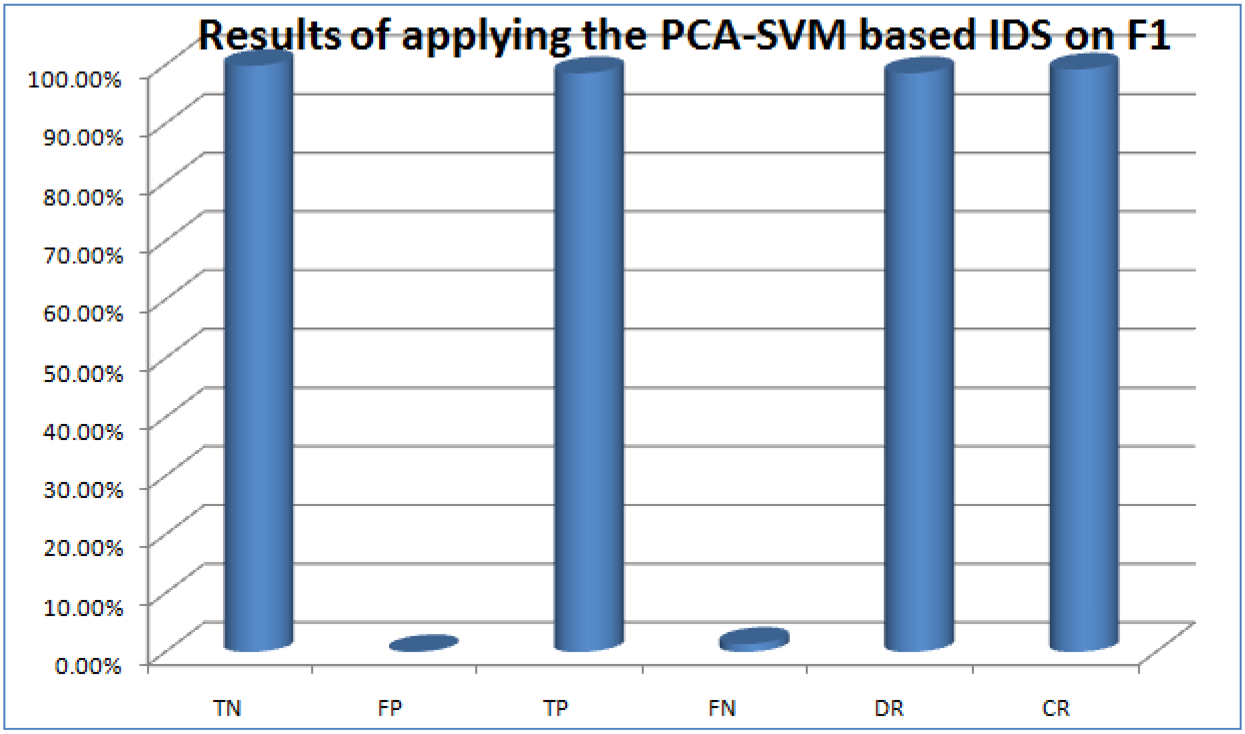


Figure 4.19 Results of applying the PCA-SVM based IDS on F1 with control chart

Table 4.22Results of applying the PCA-SVM based IDS on F1 with control chart

TN	FP	TP	FN	DR	CR
99.8801%	0.1199%	98.6239%	1.3761%	98.6239%	99.2520%

As demonstratedin the table and figure above, the system has 1.3761% and 0.1199% FN and FP percentages; respectively, where these percentages stand for false alarms. On the other hand, the system has 98.6239% and 99.2520% detection and classificationrates, respectively.

4.8.2 Results of Applying the PCA-SVMBased IDS on F2

The achieved results of applying the PCA-SVM based IDS with control charton F2, which includes 9 features from the NSL-KDD dataset;[4,5,10,11,23,24,29,31,33]are demonstrated in this subsection. The following **Figure4.20** and **Table4.23** demonstrate the measured FP, FN, TP, TN, DR, and CR percentages for this case.

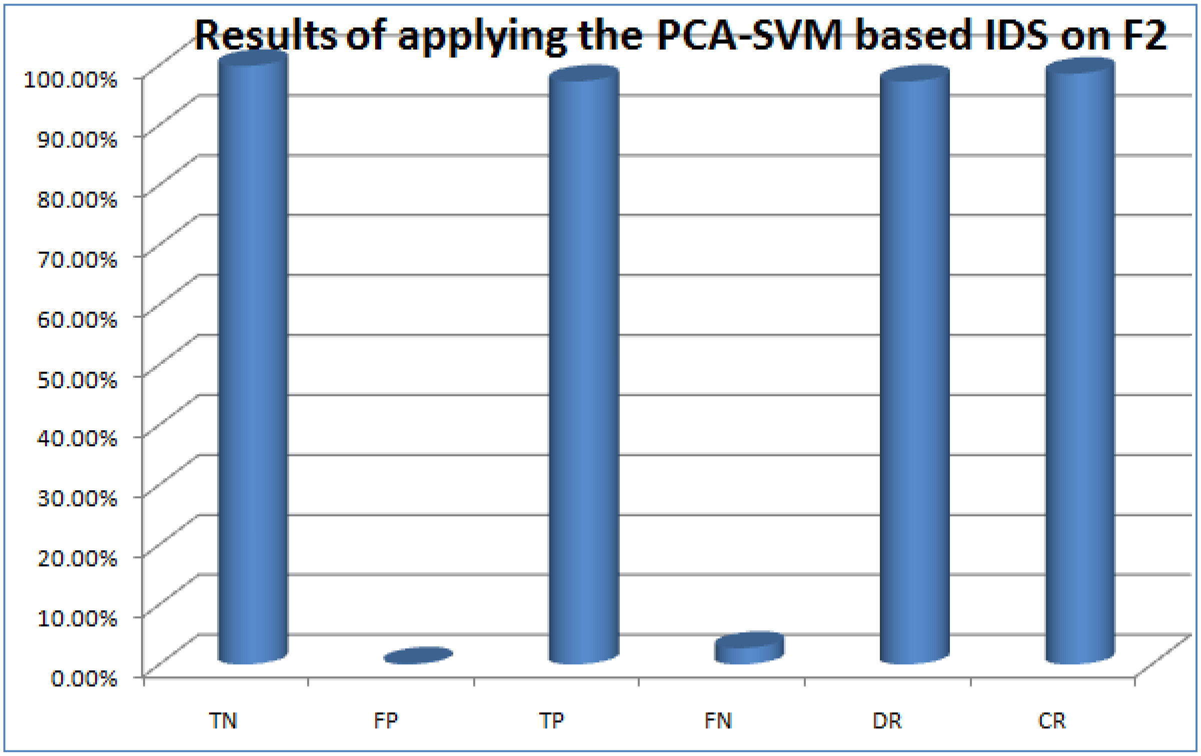


Figure 4.20 Results of applying the PCA-SVM based IDS on F2 with control chart

Table 4.23Results of applying the PCA-SVM based IDS on F2 with control chart

TN	FP	TP	FN	DR	CR
99.88%	0.12%	97.2851%	2.7149%	97.2851%	98.5826%

As shown above,the system has 2.7149% and 0.12% FN and FP percentages; respectively, where these percentages stand for false alarms. In contrast, the system has 97.2851% and 98.5826% detection and classificationrates, respectively.

4.8.3 Results of Applying the PCA-SVMBased IDS on F3

This section show the achieved results of applying the PCA-SVM based IDS with control charton F3, which includes 7 features from the NSL-KDD dataset;[4,5,10,11,23,24,29]. The **Figure4.21** and **Table4.24** below show the measured FP, FN, TP, TN, DR, and CR percentages for this case.

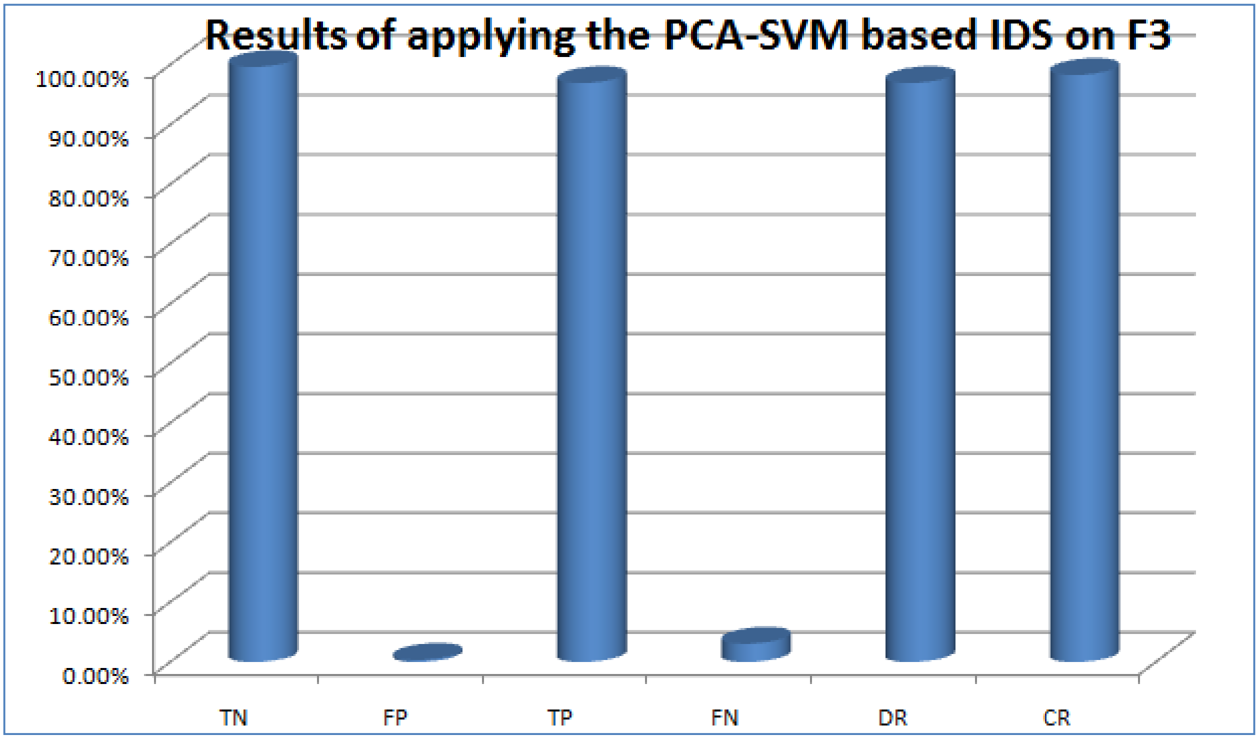


Figure 4.21 Results of applying the PCA-SVM based IDS on F3 with control chart

Table 4.24Results of applying the PCA-SVM based IDS on F3 with control chart

TN	FP	TP	FN	DR	CR
99.6409%	0.3591%	96.9555%	3.0445%	96.9555%	98.2982%

It can be seen that the system has 3.0445% and 0.3591% FN and FP percentages; respectively, where these percentages stand for false alarms. In contrast, the system has 96.9555% and 98.2982% detection and classificationrates, respectively.

4.8.4 Comparison between Three Cases

The following**Table4.25** and **Figure4.22**illustrate a comparison between the presented three cases in terms of FP and FN for the PCA-SVM based IDS with control chart.

Table 4.25Comparison between the presented three cases in terms of FP and FN for PCA-SVM based IDS with control chart

Features set	FP	FN
F1: [4,5,10,11,23,24,29,31,33, 38,41]	0.1199%	1.3761%
F2: [4,5,10,11,23,24,29,31,33].	0.12%	2.7149%
F3: [4,5,10,11,23,24,29]	0.3591%	3.0445%

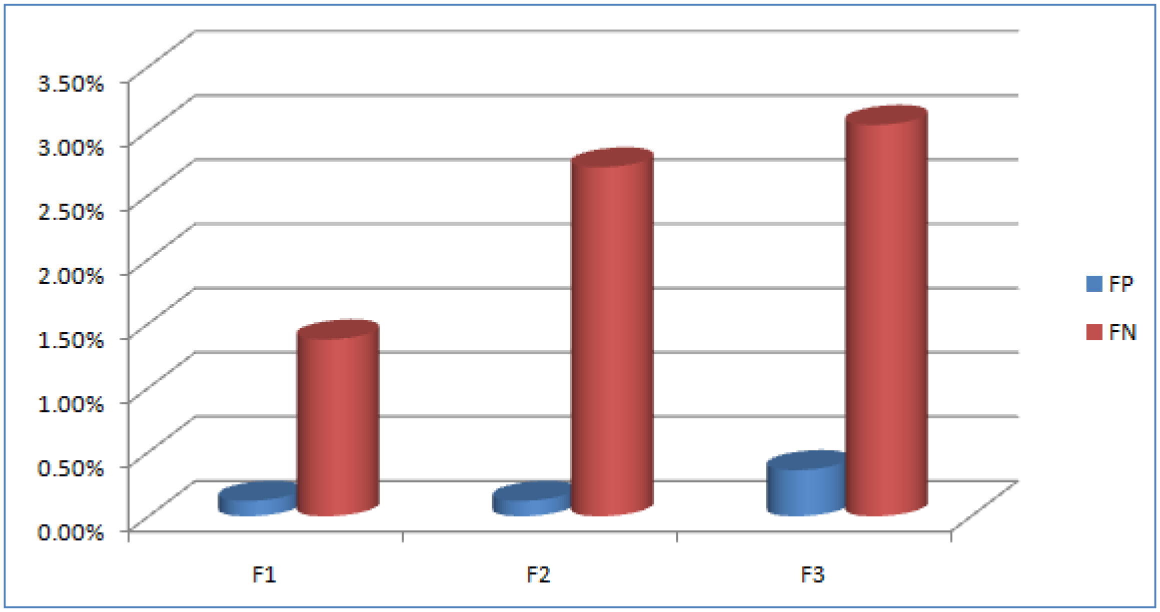


Figure 4.22Comparison between the presented three cases in terms of FP and FN for PCA-SVM based IDS with control chart

As shown above, the lowest achieved FP and FN percentages are forF1. Therefore, the PCA-SVM based IDS provides the minimum number of false alarms with using the first set that includes 11 features.

The following **Table4.26** and **Figure4.23** below illustrate a comparison between the presented three cases in terms of DR and CR for the PCA-SVM based IDS with control chart.

Table 4.26Comparison between the presented three cases in terms of DR and CR for PCA-SVM based IDS with control chart

Features set	DR	CR
F1: [4,5,10,11,23,24,29,31,33, 38,41]	98.6239%	99.2520%
F2: [4,5,10,11,23,24,29,31,33].	97.2851%	98.5826%
F3: [4,5,10,11,23,24,29]	96.9555%	98.2982%

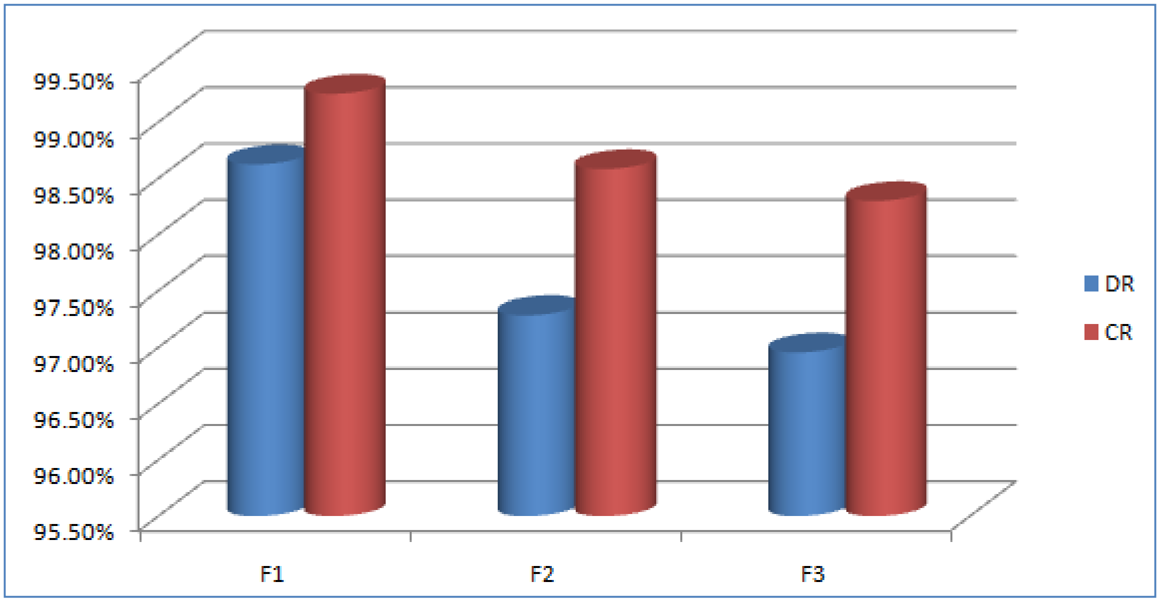


Figure 4.23Comparison between the presented three cases in terms of DR and CR for PCA-SVM based IDSwith control chart

It can be concluded from the results above that the highest percentages of DR and CR are for F1. Thus, the PCA-SVM based IDS provides the minimum number of false alarms and the highest detection and classification rates with using the first set that has 11features.

4.9 Comparison between PCA-KNN and PCA-SVM Classifiers without Control Chart

4.9.1 Comparison between Classifiers with Applying F1

The **Table4.27** and **Figure4.24** below illustrate a comparison between both the PCA-KNN and PCA-SVM based IDS without control chart using the first set of features; F1 in terms of DR and CR percentages.

Table 4.27Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F1 without control chart

F1	DR	CR
PCA-KNN	91.7786%	90.8165%
PCA-SVM	98.5725%	94.1424%

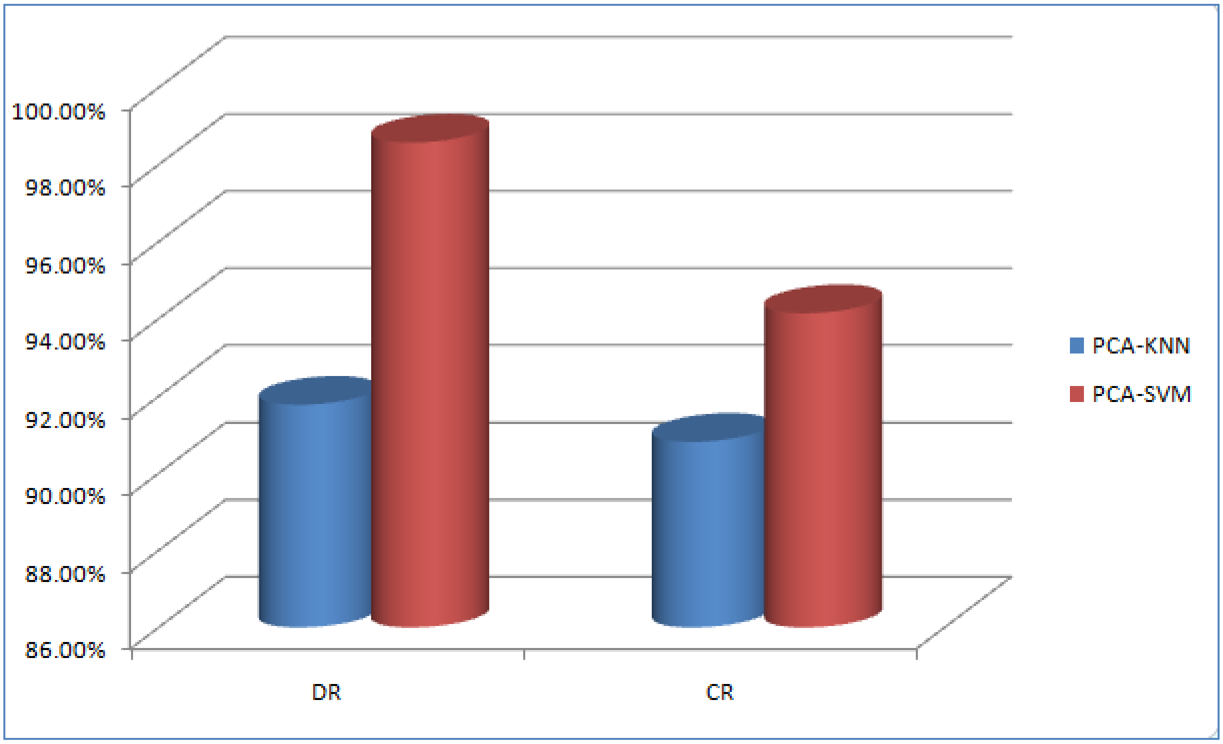


Figure4.24Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F1

It can be noticed that the PCA-SVM based IDS outperforms the PCA-KNN based IDS without control chart in terms of DR and CR percentages for F1.

4.9.2 Comparison between Classifiers with Applying F2

The following **Table4.28** and **Figure4.25**show a comparison between both the PCA-KNN and PCA-SVM based IDS without control chart using F2 in terms of DR and CR percentages.

Table 4.28Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F2 without control chart

F2	DR	CR
PCA-KNN	91.5718%	90.6942%
PCA-SVM	96.0800%	92.9187%

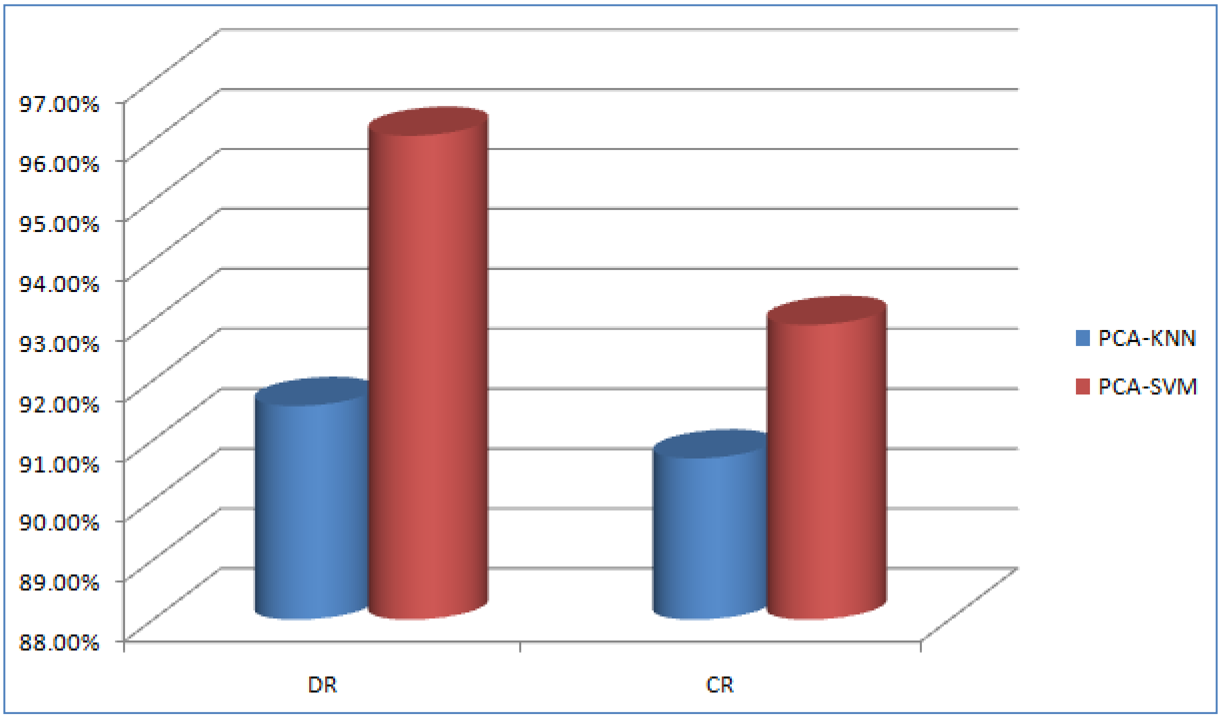


Figure 4.25Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F2 without control chart

It can be noticed from the figure and table above that the PCA-SVM based IDS outperforms the PCA-KNN based IDS without control chart in terms of DR and CR percentages for F2.

4.9.3 Comparison between Classifiers with Applying F3

The following **Table4.29** and **Figure4.26**show a comparison between both the PCA-KNN and PCA-SVM based IDS without control chart using the third set of features; F3 in terms of DR and CR percentages

Table 4.29Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F3 without control chart

F3	DR	CR
PCA-KNN	82.9710%	86.1522%
PCA-SVM	95.2551%	94.0455%

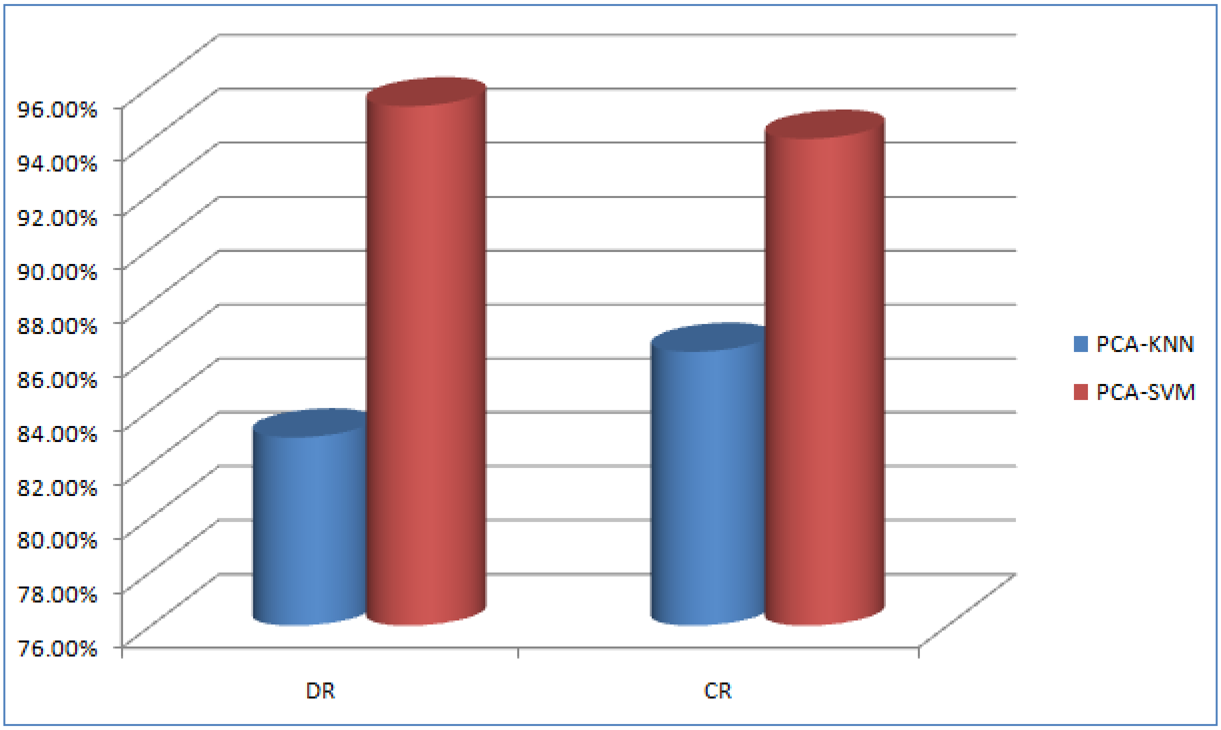


Figure 4.26Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F3 without control chart

It can be noticed for F3, the PCA-SVM outperforms the PCA-KNNin terms of DR and CR percentages.

The average achieved DR and CR percentages for the PCA-KNN based IDS are 88.7738% and 89.22097 %, respectively. On the other hand, the average achieved DR and CR percentages for the PCA-SVM based IDS are 96.63587% and 93.7022%, respectively. Thus, it can be summarized that the PCA-SVMclassifier offers more effective results than the PCA-KNN one when applied to the IDS without control chart.

4.10 Comparison between PCA-KNN and PCA-SVM Classifiers with Control Chart

The following subsections show a comparison between both classifiers for the three sets of features.

4.10.1 Comparison between Classifiers with Applying F1

The **Table4.30** and **Figure4.27** below show a comparison between both the PCA-KNN and PCA-SVM based IDS with control chart using the first set of features; F1in terms of DR and CR percentages.

Table 4.30Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F1 with control chart

F1	DR	CR
PCA-KNN	98.1859%	99.0555%
PCA-SVM	98.6239%	99.2520%

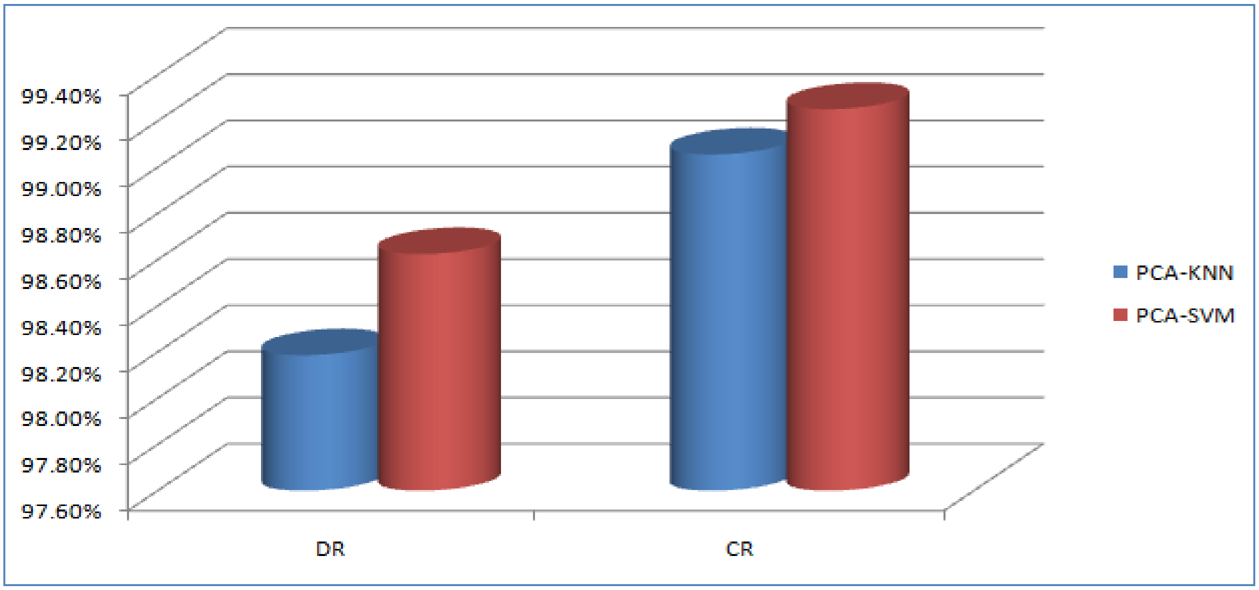


Figure 4.27Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F1 with control chart

As shown above, the PCA-SVM based IDS outperforms the PCA-KNN based IDS in terms of DR and CR percentages for the first set of features.

4.10.2 Comparison between Classifiers with Applying F2

The following **Table4.31** and **Figure4.28**illustrate a comparison between both the PCA-KNN and PCA-SVM based IDS with control chart using F2in terms of DR and CR percentages.

Table 4.31Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F2 with control chart

F2	DR	CR
PCA-KNN	98.19%	99.065%
PCA-SVM	97.2851%	98.5826%

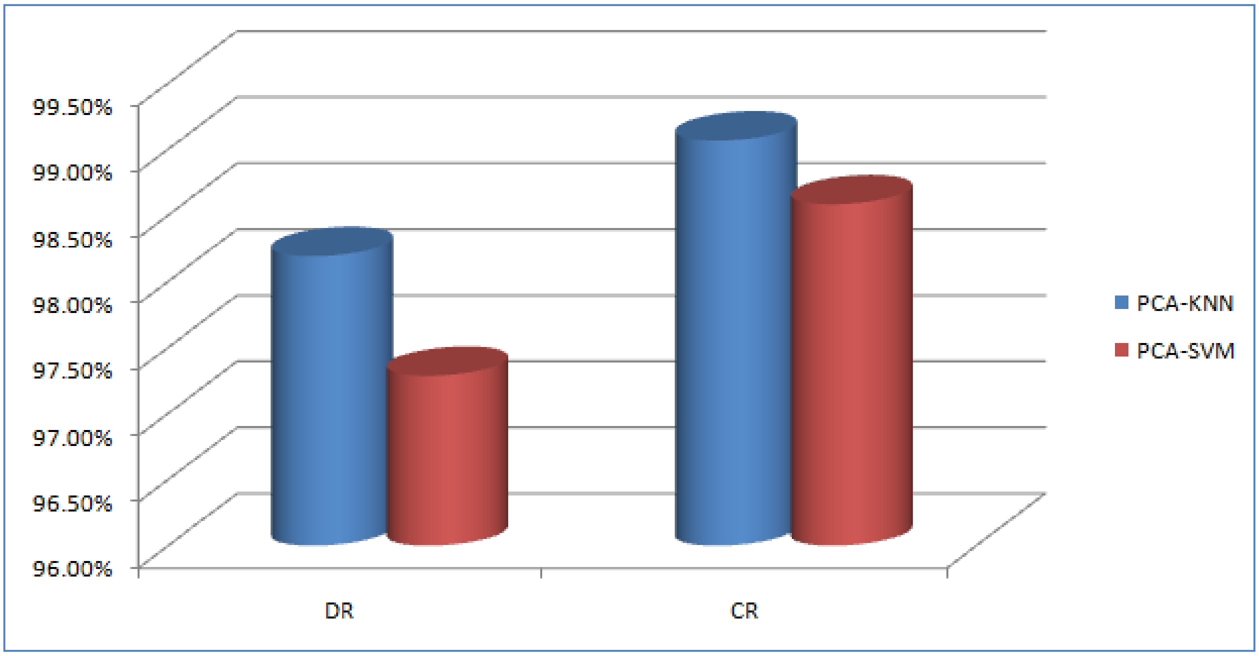


Figure 4.28Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F2with control chart

It can be noticed from the figure and table above that the PCA-KNN based IDS outperforms the PCA-SVM based IDS in terms of DR and CR percentages for the second set of features.

4.10.3 Comparison between Classifiers with Applying F3

The following **Table4.32** and **Figure4.29**illustrate a comparison between both the PCA-KNN and PCA-SVM based IDS with control chartusing the third set of features; F3 in terms of DR and CR percentages

Table 4.32Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F3 with control chart

F3	DR	CR
PCA-KNN	98.1265%	98.9211%
PCA-SVM	96.9555%	98.2982%

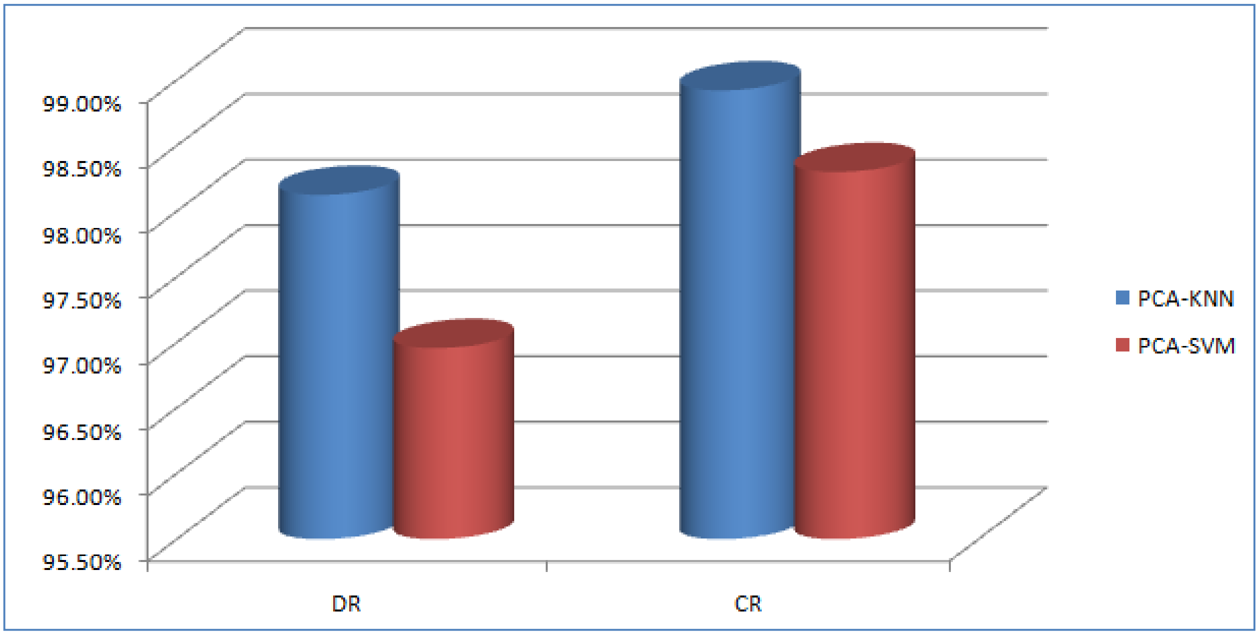


Figure 4.29Comparison between the PCA-KNN and PCA-SVM based IDS in terms of DR and CR for F3with control chart

It can be noticed from the figure and table above that the PCA-KNN based IDS also outperforms the PCA-SVM based IDS in terms of DR and CR percentages for the third set of features.

The average achieved DR and CR percentages for the PCA-KNN based IDS are 98.17% and 99.01%, respectively. On the other hand, the average achieved DR and CR percentages for the PCA-SVM based IDS are 97.62% and 98.71%, respectively. Thus, it can be summarized that the PCA-KNNclassifier offers more enhanced results than the PCA-SVM one when applied to the IDS.

4.11 Comparison between Classifiers with and without Control Chart

4.11.1 Comparison between Classifiers for F1

The following **Table4.33** shows a comparison among the classifiers with and without control chart for F1.

Table 4.33Comparison between the classifiers with and without control chart for F1

F1	DR	FN
PCA-KNN without control chart	91.7786%	8.2214%
PCA-KNN with control chart	98.1859%	1.8141%
PCA-SVM without control chart	98.5725%	1.2475%
PCA-SVM with control chart	98.6239%	1.3761%

For F1, it can be noticed thatthe PCA-SVM based IDS with control chart offers the best detection rate, but, it still needs some improvements to decrease the number of generated false alarms.

4.11.2 Comparison between Classifiers for F2

The following**Table4.34** below shows a comparison among the classifiers with and without control chart for F2.

Table 4.34Comparison between the classifiers with and without control chart for F2

F2	DR	FN
PCA-KNN without control chart	91.5718%	8.4282%
PCA-KNN with control chart	98.19%	1.81%
PCA-SVM without control chart	96.0800%	3.9200%
PCA-SVM with control chart	97.2851%	2.7149%

For F2, it can be noticed thatthe PCA-KNN based IDS with control chart offers the best detection rate with the minimum number of generated false alarms.

4.11.3 Comparison between Classifiers for F3

The following**Table4.35** below shows a comparison among the classifiers with and without control chart for F3.

Table 4.35Comparison between the classifiers with and without control chart for F3

F3	DR	FN
PCA-KNN without control chart	82.9710%	17.0290%
PCA-KNN with control chart	98.1265%	1.8735%
PCA-SVM without control chart	95.2551%	1.8221%
PCA-SVM with control chart	96.9555%	3.0445%

For F3, it can be concluded thatthe PCA-KNN based IDS with control chart offers the best detection rate with minimum number of generated false alarms.

4.11.4 Comparison between all Cases

The following **Table4.36** shows a comparison between all cases for both classifiers with and without using control chart.

Table 4.36Comparison between all cases for both classifiers with and without using control chart

	PCA-KNN						PCA-SVM					
	Without control chart			With control chart			Without control chart			With control chart		
	DR	CR	FN	DR	CR	FN	DR	CR	FN	DR	CR	FN
F1	91.7786%	90.8165%	8.2214%	98.1859%	99.0555%	1.8141%	98.5725%	94.1424%	1.2475%	98.6239%	99.2520%	1.3761%
F2	91.5718%	90.6942%	8.4282%	98.19%	99.065%	1.81%	96.0800%	92.9187%	3.9200%	97.2851%	98.5826%	2.7149%
F3	82.9710%	86.1522%	17.0290%	98.1265%	98.9211%	1.8735%	95.2551%	94.0455%	1.8221%	96.9555%	98.2982%	3.0445%

The **Table4.36** above shows that the highest achieved detection and classification rates with minimum false alarm rate is are for the application of PCA-SVM for the first set of features.

4.12 Measurement Tools

The testing engine is used to test the resultant training engine by using the NSL-KDD dataset and to determine if the record is an attack or not based on a specified threshold. Accuracy and results of tests depend on the datasets, features and threshold value. The following percentage expressions are used in the analysis of data. (Altajry and Algarny, 2011)

True Negative (TP): Normal records which are correctly classified,

True Positive (TP): Attack records which are correctly classified,

False Positive (FP): Normal records which are incorrectly classified as attacks,

False Negative (FN): Attack records which are incorrectly classified as normal.

By using these expressions, both the detection rate and classification rate can be represented as follows**(4.1) (4.2):**

$$Detection\ Rate\ (DR) = \frac{TP}{TP + FN} \quad (4.1)$$

$$Classification\ Rate\ (CR) = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.2)$$

Chapter Five:
Conclusion and Future Works

Chapter Five: Conclusion and Future Works

5.1. Conclusion

This work introduces the development of efficient IDSusing two classifiers; PCA-SVM and PCA-KNN to increase the detection rate and reduce both False Positive (FP) and False Negative (FN) rates using the MATLAB program. This is performed to determine the best classifier that decrease the number of generated false alarms, enhance the network security and improve the detection rate of various types of attacks.

The Principal Component Analysis (PCA) technique is combined with both classifiers to reduce the dimension space; PCA-SVM and PCA-KNN. The NSL-KDD dataset is used to evaluate and measure the system performance after applying both classifiers based on dividing it into two sets; training and testing. The use of the PCA technique offers an enhancement for these two sets based on reducing their dimensionalities and selecting the optimal features.

The implemented IDS consists of two main stages; training, testing. In the first stage, a training set is utilized to train the system in order to distinguish the normal records from the attacked ones, where the SVM or KNN classifier is applied in this stage to determine the most important features to be used in detecting attacks and the main features of normal data. The training data are then filtered using a control chart (CC) that has lower control chart (LCC) limit and upper control chart (UCC) limit based on computing the mean and standard deviation of each record. The CC filters the training data based on controlling them within a specific range in order to apply the testing data on the same range. After applying the CC, all records inside the specified range are considered as normal records, while those outside this range are considered as attacks.

In the testing stage, a testing set, which composed of normal records and attacks, is utilized to measure the IDS performance in which the high performance represents the higher accuracy in determining both normal records and attacks. In the last stage, the system is used to protect the network traffic. In the first two stages, the IDS determines the network traffic based on the requested service initially and then on the selected features.

The IDS performance is measured and evaluated based on computing six evaluation metrics; False Positive (FP), False Negative (FN), True Positive (TP), True Negative (TN), Detection Rate (DR) and Classification Rate (CR). These metrics are used to compare between both classifiers; SVM and KNN and then determine the best one based on the lowest FP and FN percentages and the highest DR and CR percentages.

The used NSL-KDD dataset includes 41 features. In this work, three sets of features from this dataset are used to choose the best set for each classifier; 11 features F1: [4,5,10,11,23,24,29,31,33,38,41], 9 features F2: [4,5,10,11,23,24,29,31,33] and 7features F3: [4,5,10,11,23,24,29].

The achieved results of applying the PCA-KNN based IDS without control chart on the three sets of features demonstrate that the system has 8.2214% and 10.1455% FN and FP percentages; respectively for the first set with 91.7786% detection rate, 8.4282% and 10.1835% FN and FP percentages; respectively for the second set with 91.5718% detection rate and 17.0290% and 10.6665% FN and FP percentages; respectively for the third set with 82.9710% detection rate. Thus, the PCA-KNN based IDSoffers the minimum number of false alarms and the highest detection rate with using the first set that has 11 features.

The obtained results of applying the PCA-KNN based IDSwith control chart on the three sets of features demonstrate that the system has 1.8141% and 0.075% FN and FP percentages; respectively for the first set with 98.1859% detection rate, 1.81% and 0.06% FN and FP percentages for the second set with 98.19% detection rate and 1.8735% and 0.2843% FN and FP percentages; respectively for the third set with 98.1265% detection rate. Thus, the PCA-KNN based IDS provides the minimum number of false alarms with the highest detection rate with using the second set that has 9 features.

The obtained results of applying the PCA-SVM based IDS without control chart on the three sets of features demonstrate that the system has 1.2475% and 10.4677% FN and FP percentages; respectively for the first set with 98.5725% detection rate, 3.9200% and 10.2426% FN and FP percentages; respectively for the second set with 96.0800% detection rate and 1.8221% and 10.0869% FN and FP percentages; respectively for the third set with 95.2551% detection rate. Thus, the PCA-SVM based IDSprovides the minimum number of false alarms and the highest detection rate with using the first set that has 11 features.

The obtained results of applying the PCA-SVM based IDSwith control chart on the three sets of features demonstrate that the system has 1.3761% and 0.1199% FN and FP percentages; respectively for the first set with 98.6239% detection rate, 2.7149% and 0.12% FN and FP percentages; respectively for the second set with 97.2851% detection rate and 3.0445% and 0.3591% FN and FP percentages; respectively for the third set with 96.9555% detection rate. Thus, the PCA-SVM based IDSprovides the minimum number of false alarms and the highest detection rate with using the first set that has 11 features.

Based on comparing the obtained results of both PCA-KNN and PCA-SVM based IDSs with and without control chart, it is noticed that the PCA-KNN based IDS with control chartoffers the best detection rate with minimum number of generated false alarms for sets F1 and F3. On the other hand, the PCA-SVM based IDS with control chart offers the best detection rate with minimum number of generated false alarms for F2. It can be concluded that the application of control chart enhances the detection rate and decreases the number of false alarms for both classifiers.

5.2. Future Works

In this thesis, IDS is implemented using two classifiers; PCA-KNN and PCA-SVM, where results demonstrated that the PCA-KNN based IDS offers the minimum number of false alarms with the highest detection rate. On the other hand, the current work can be enhanced in the future based on applying the following:

- Applying the current system on other dataset to evaluate its performance
- Combining the best classifier with other classifiers such as naïve Bayesian and Particle Swarm Optimization (PSO) classifiersto form a hybrid classification method that have the benefits of these combined classifiers

References

Alessandria.D.(2004).Attack-Class-Based Analysis of Intrusion Detection Systems.*University of Newcastle upon Tyne School of Computing Science.*

Bhavsar, Y.B,W & Kalyani, C.(2013).Intrusion Detection System Using Data Mining Technique: Support Vector Machine. *International Journal of Emerging Technology and Advanced Engineering.*

Chandola, V. & Kumar, V. (2009). Anomaly Detection: A Survey, *ACM Computing Surveys*,. 1-72,

Chen, R. Cheng, K. Chen, Y.and Hsieh, C. (2009). Using rough set and support vector machine for network intrusion detection**.** *International Journal of Network Security & Its Applications (IJNSA).*

Dacier, M, & Alessandri, D.(1999). A Vulnerability Dataset. *Computer Security Journal***.**

Dewaele, G., Fukuda K. & Borgnat, P. (2007). Extracting hidden anomalies using sketch and non Gaussian multi-resolution statistical detection procedures

Faria, D. (2006). Scalable location-based security in wireless networks khan.Federal Information Processing Standards Publication 191 (FIPS PUB 191). (1994). Guideline for the Analysis Local Area Network Security

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

Ghali, N. I. (2009). Feature Selection for Effective Anomaly-Based Intrusion Detection. *IJCSNS International Journal of Computer Science and Network Security*. **9** (3)

Gogoi, P. Bhattacharyya, D.K. Borah B. & Kalita, J. k. (2013). MLH-IDS: A Multi-Level Hybrid Intrusion Detection Method. The Computer Journal Advance.

He, Q. P., & Wang, J. (2007). Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. Semiconductor manufacturing. *IEEE*.345-354

Hofmeyr, A. Forrest S. & Somayaji, A. (1998). Intrusion detection using sequences of system calls. *Journal of Computer Security*. **6**: 151–180

Htun, P. T and Khaing, K. T.(2015). Anomaly Intrusion Detection System using Random Forests and k-Nearest Neighbor. *International journal ssrg.*

Hu, J. (2010). Host-Based Anomaly Intrusion Detection'', *Springer*. 235-255

Kayaci, H. S., &, Aybar S. (2005). Simulation of Natural Convection in a Differentially Heated Cavity. In ASME 2005 Fluids Engineering Division Summer Meeting (pp. 641-646). American Society of Mechanical Engineers.

Lakhina, A. Crovella, M. & Diot, C. (2005). Mining anomalies using traffic feature distributions. 21–26

Lanckriet, G. R., Deng, M., Cristianini, N., Jordan, M. I., & Noble, W. S.(2004). Kernel-based data fusion and its application to protein function prediction in yeast. *In Pacific symposium on biocomputin**g***. 300-311

Leung, J. (2008). Vulnerability Management – A Guide to Managing Internal and External Threats

Li, W., Yi, P., Wu, Y., Pan, L., and Li, J.(2014). A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network. *Hindawi Publishing Corporation Journal of Electrical and Computer Engineering.*

Li, Y and Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. *Computers & security***.** 459-467

Li, Y., Fang, B Guo, L & Chen, Y. (2007). Network Anomaly Detection Based on TCM-KNN Algorithm. 13-19

Liu, W. K., Jun, S., & Zhang, Y. F.(1995). Reproducing kernel particle methods. *International journal for numerical methods in fluids*.1081-1106

Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). Multivariate analysis. Academic press.

Ming, Y. (2011). Real Time Anomaly Detection Systems for Denial of Service Attacks by Weighted k-Nearest Neighbor Classifiers. *Expert Systems with Applications*.

Mukkamala, S., Janoski, G., and Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In Neural Networks. *IEEE*. *International Joint Conference on.* 1702-1707

Mulay, S. Devale, P. Garje, G. (2010). Intrusion detection system using support vector machine and decision tree. *International Journal of Computer Applications*.

Pedersen, R. & Schoeberl, M. (2006). An embedded support vector machine. In Intelligent Solutions in Embedded Systems, *IEEE* . 1-11

Porras, P, Schnacenberg, D, Chen, S. S & Wu. F. (2000).The Common Intrusion Detection Framework Architecture. *Journal of Computer Security*.

Shailendra and Sanjay.(2009). An ensemble approach for feature selection of Cyber Attack Dataset. *International Journal of Computer Science and Information Security*.

Tandon, G. & Chan, P.K. (2005). In the Florida Artificial Intelligence Research Society Conference. 405-410

Wang, J., Hong, X., Ren, R. R. & Li, T. H. (2009). A Real-time Intrusion Detection System Based on PSO-SVM. *Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009)*. 319- 321

Xu J. & Shelton, C.R. (2008). In European Conference on Machine Learning

Yao, J., Zhao, S., & Fan, L. (2015). An Enhanced Support Vector Machine Model

Ye, N., Emran, S.M Chen, Q. & Vilbert, S. (2002). Multivariate statistical analysis of audit trails for host-based intrusion detection", Transactions *of Computers*, **51** (7): 810–820

Vapnik, V. N. (1998). Statistical Learning Theory, John, Inc.

Lippmann, R. Fried, D. and Graf, I. (2000). Evaluating Intrusion Detection Systems: the '1998 DARPA off-line Intrusion Detection Evaluation. *In: Information Survivability Conference and Exposition, IEEE*, pp 12-26

Y. Liao and V. R. Vemuri. (2002). Use of K-Nearest Neighbor Classifier for Intrusion Detection. , *Computers and Security*, pp 439-448

Appendices

Name: Nafea Ali Majeed Alhammadi
Program: MATLAB 2014A

Appendix A: MATLAB Code for PCA-KNN Based IDS

```
clc
clear all
close all


[dos1]= xlsread('nsltrain-service.xlsx');
[dos2]= xlsread('nsltest-service.xlsx');

ATTACK_TYPE=[4,5,10,11,23,24,29,31,33];%%%% comment ref.



TestSet=xsT;
GroupTest=ysT;

u=unique(GroupTrain);
numClasses=length(u);
result = zeros(length(TestSet(:,1)),1);

 result = knnclassify(TestSet, TrainingSet, GroupTrain);


%%%%%%%%%%%%%%%%%% CONTROL CHART%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 yy=unique(ysO);%%
NN=length(yy);%%


MM=length(ysO);%%

att=find(ysO==1);
norma=find(ysO==1000);

for i=1:NN
    FY(i)=sum(double(ysO==yy(i)))/length(ysO); %%
end
```

```
 [z]=(find(ysO==1000));%

  ys=ysO(z);
  xs=xsO(z,:); %%%



MU=[];
SIGMA=[];



for i=1:length(xs)
    xi=xs(i,:);%%
    mu=mean(xi);
    sigma=std(xi);
    MU=[MU,mu];
    SIGMA=[SIGMA,sigma];

end



PRB=[];
for j=1:length(xs) %%%%



    FU=normcdf(xs(j,:),MU(j),SIGMA(j));%%
    prb=FY(1,2).*prod(FU);%
    PRB=[PRB,prb];
end



X1=PRB'; %%%

meanN=mean(X1);
stdN=std(X1);


LCC_N=meanN-(3*stdN);
UCC_N=meanN+(3*stdN);


figure(1)
plot(X1,'Linewidth',1)
hold on
plot(LCC_N*ones(1,length(X1)),'or','Linewidth',1)
plot(UCC_N*ones(1,length(X1)),'+r','Linewidth',1)
grid on
```

```
title('Prob. of training data before filtering')

X2=X1;%%%backup
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



    RR=[];%%
LCC_T=[];%%
UCC_T=[]; %%
for ii=1:152

    TT=X1((250*(ii-1)+1):(250*ii)); %%
    [seg,LCC,UCC]=make_seg(TT);
    RR=[RR;seg];
    LCC_T=[LCC_T,LCC];
    UCC_T=[UCC_T,UCC];

end

LCC_N=min(LCC_T);
UCC_N=max(UCC_T);%

figure(2)
plot(RR,'Linewidth',1)
hold on
plot(LCC_N*ones(1,length(RR)),'or','Linewidth',1)
plot(UCC_N*ones(1,length(RR)),'+r','Linewidth',1)
grid on
title(' Prob. of training data after filtering')

X_OUT=find(RR>UCC_N);

%
% % % % %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%



MUT=[];
SIGMAT=[];




for i=1:length(xsT)
    xii=xsT(i,:);%%
    mu=mean(xii);
    sigma=std(xii);
    MUT=[MUT,mu];
    SIGMAT=[SIGMAT,sigma];
```

```
end



PRBT=[];
for j=1:length(xsT) %%



    FUT=normcdf(xsT(j,:),MUT(j),SIGMAT(j));
    prbt=prod(FUT);%%
    PRBT=[PRBT,prbt];
end

X3=PRBT';


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%



figure(3)
plot(X3,'Linewidth',1)
hold on
plot(LCC_N*ones(1,length(X3)),'or','Linewidth',1)
plot(UCC_N*ones(1,length(X3)),'+r','Linewidth',1)
grid on
title(' Prob. of testing data ')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
```

Appendix B: MATLAB Code SVM-PCA Based IDS

```
clc
clear all
close all


[dos1]= xlsread('nsltrain-service.xlsx');
[dos2]= xlsread('nsltest-service.xlsx');

ATTACK_TYPE=[4,5,10,11,23,24,29,31,33];




models = 
svmtrain(TrainingSet,GroupTrain,'kernel_function','rbf','RBF_Sigma',3);


result = svmclassify(models,TestSet);




%%%%%%%%%%%%%%%%%% CONTROL CHART%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 yy=unique(ysO);%%
NN=length(yy);%%


MM=length(ysO);%%

att=find(ysO==1);
norma=find(ysO==1000);

for i=1:NN
    FY(i)=sum(double(ysO==yy(i)))/length(ysO); %%
end




 [z]=(find(ysO==1000));%

  ys=ysO(z);
  xs=xsO(z,:); %%%
```

```
MU=[];
SIGMA=[];



for i=1:length(xs)
    xi=xs(i,:);%%
    mu=mean(xi);
    sigma=std(xi);
    MU=[MU,mu];
    SIGMA=[SIGMA,sigma];

end



PRB=[];
for j=1:length(xs) %%%%



    FU=normcdf(xs(j,:),MU(j),SIGMA(j));%%
    prb=FY(1,2).*prod(FU);%
    PRB=[PRB,prb];
end



X1=PRB'; %%%

meanN=mean(X1);
stdN=std(X1);


LCC_N=meanN-(3*stdN);
UCC_N=meanN+(3*stdN);


figure(1)
plot(X1,'Linewidth',1)
hold on
plot(LCC_N*ones(1,length(X1)),'or','Linewidth',1)
plot(UCC_N*ones(1,length(X1)),'+r','Linewidth',1)
grid on
title('Prob. of training data before filtering')

X2=X1;%%%backup
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    RR=[];%%
LCC_T=[];%%
UCC_T=[]; %%
for ii=1:152

    TT=X1((250*(ii-1)+1):(250*ii)); %%
    [seg,LCC,UCC]=make_seg(TT);
    RR=[RR;seg];
    LCC_T=[LCC_T,LCC];
    UCC_T=[UCC_T,UCC];

end

LCC_N=min(LCC_T);
UCC_N=max(UCC_T);%

figure(2)
plot(RR,'Linewidth',1)
hold on
plot(LCC_N*ones(1,length(RR)),'or','Linewidth',1)
plot(UCC_N*ones(1,length(RR)),'+r','Linewidth',1)
grid on
title(' Prob. of training data after filtering')

X_OUT=find(RR>UCC_N);

%
% % % % %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%




MUT=[];
SIGMAT=[];



for i=1:length(xsT)
    xii=xsT(i,:);%%
    mu=mean(xii);
    sigma=std(xii);
    MUT=[MUT,mu];
    SIGMAT=[SIGMAT,sigma];

end
```

```
PRBT=[];
for j=1:length(xsT) %%



    FUT=normcdf(xsT(j,:),MUT(j),SIGMAT(j));
    prbt=prod(FUT);%%
    PRBT=[PRBT,prbt];
end

X3=PRBT';


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%




figure(3)
plot(X3,'Linewidth',1)
hold on
plot(LCC_N*ones(1,length(X3)),'or','Linewidth',1)
plot(UCC_N*ones(1,length(X3)),'+r','Linewidth',1)
grid on
title(' Prob. of testing data ')
```

Appendix C: PCA Function

```
function D=make_PCA(dos1)


dos1(:,3)=[];
[Rows, Columns] = size(dos1);              % find size of input matrix
m=mean(dos1);                              % find mean of input matrix
y=dos1-ones(size(dos1,1),1)*m;         % normalize by subtracting mean
c=cov(y);                         % find covariance matrix
[V,D]=eig(c);                 % find eigenvectors (V) and eigenvalues (D)
of covariance matrix
```