

# Using An Improved Data Reduction Method in Intrusion detection system

Rasha Thamer Shawe, Safana H. Abbas

<sup>1</sup>Research Scholar, <sup>2</sup>Assistant Professor

College of Education in Computer Science

AL-Mustansiriya University

Baghdad, Iraq.

## ABSTRACT

*The information security is an issue of serious global concern. has become a very important and critical issue in network, data and information security. An Intrusion detection system collects and analyzes information from different areas within a computer or a network to identify possible security threats that include threats from both outside as well as inside the organization. The Intrusion detection system deals with large amount of data which contains various irrelevant and redundant features resulting in increased processing time and low detection rate. Therefore feature reduction plays an important role in intrusion detection..In this paper an improvement is suggested to SVD data reduction method and implemented with Back-Propagation algorithm to detect different types of attacks .the performance is evaluated using detection rate, error rate and accuracy*

**Key Words:** *Intrusion Detection, KDDCup99 dataset, Feature reduction, Classification..*

---

## 1. INTRODUCTION

In the interrelated world of computer networks, there is an augmented need to provide transaction security and safety systems [1]. Intruders have made many successful attempts to bring down high-profile Web services and company networks . Many methods have been developed to network infrastructure, secure computers, and communication over the Internet, among these methods the use of firewalls, encryption, and intrusion detection systems(IDS) [2].intrusion detection systems have been based Traditionally on the characterization of an attack and the tracking the activity of the system to see if it matches that characterization. Intrusion detection system based on data mining is making their appearance more efficacy[3].

The application of Data Mining techniques for intrusion detection systems have been widely used these days. The problem of intrusion detection has been reduced to a Data Mining task of classifying data. Briefly, given a set of data points belonging to different normal activity (Classes, different attacks) and aims to separate them as accurately as possible by means of a model. Many different data mining techniques exist for intrusion detection classification[4].

1.**KDD 99 dataset** Since 1999, SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) of ACM (Association for Computing Machinery) has been organizing an annual KDD (Data Mining and Knowledge

Discovery) CUP competition. Later on, KDD 99 became the most popular dataset used for evaluation of anomaly detection methods. This dataset consists of the data generated from DARPA'98 IDS evaluation program, which comprises about 4 gigabytes of compressed raw (binary) tcp dump data of 7-week network traffic, and can be processed into around 5 million connection records with 100 bytes each. The test data of two weeks includes around 2 million connection records [5],[6].the definition of a connection is a TCP data packet sequence including data from source IP address to destination IP address in a predefined protocol (such as TCP or UDP) from beginning to end in a period of time. Each connection is classified as either attack or normal. An attack can be sub classified into four categories of 39 types. The 7-week training dataset only contains 22 types of attacks, and the test dataset includes other unknown 17 types [7]. It is notable that the probability distribution of the test data is not the same as the one of training data, and also that the test data contains certain attack types which do not appear in the training data. It is believed by some intrusion experts that most of the novel attacks are variants of known attacks, the signature of which is sufficient to capture novel variants [6].

The training dataset is composed by about 4,900,000 single connection vectors. Each of those vectors includes 42features and one class label An attack can be classified into one of the four categories as below[8],[9]:

1. Denial of Service Attack (DoS): Some computing or memory resources are made too busy or too full to accept legitimate requests, or to allow legitimate users to access a machine. e.g., ping-of-death, syn flood, smurf.
2. User to Root Attack (U2R): An attacker gains access to a normal user account (perhaps by a dictionary attack, sniffing passwords or social engineering) and exploit the vulnerability in the system to gain root access to it. e.g., guessing passwords.
3. Remote to Local Attack (R2L): By sending packets to a machine over a network, an illegitimate user exploits vulnerability of the machine to gain local access to it as a user. e.g., buffer overflow attacks
4. Probing Attack: In order to circumventing the security controls of a computer network, the attacker attempts to gather information of the network. e.g., port-scan, ping-sweep[10].

The 42 features of the KDD 99 regarding to an attack can be divided into four categories [11], [12] :

1. Basic features: This category contains all the attributes extracted from a TCP/IP connection. The monitoring of these features will cause a fixed delay in detection.
2. Content features: The features of suspicious behavior in the data portion should be captured in order to detect attacks. e.g. number of failed login attempts. Those features are called content features. The R2L and U2R attacks normally don't appear in intrusion frequent sequential patterns, as they have been embedded in the data portions of packets and only request a single connection. While the DoS and Probing attacks involve many connections to hosts and show the attribute of intrusion frequent sequential patterns.
3. Time-based traffic features: Only the connections in the past 2 seconds are examined, which have the same destination host/service as the current connection, and of which the statistics related to protocol behavior, service, etc. are calculated.
4. Connection-based traffic features: Some slow probing attacks scan the hosts/service at an interval much longer than 2 seconds, e.g. once in every minute, which cannot be detected by the time-based traffic features, as it only examines the connections in the past 2 seconds. In such case, the features of same

destination host/service connections can be re-calculated at an interval of every 100 connections rather than a time window [13-14].

## 2. INTRUSION DETECTION SYSTEM

Intrusion Detection System (IDS) is software that automates the intrusion detection process and detects possible intrusions. IDS serve three essential security functions: they monitor, detect, and respond to unauthorized activity by insiders and outsider intrusion. An ID is a system for detecting intrusions and reporting them precisely to the suitable authority.

### 2.1 Types of IDS

These are the following types of Intrusion detection systems:

#### 2.1.1. Network Intrusion Detection System

NIDS examines the behaviour of a specified environment and make a decision whether these activities are malicious (intrusive) or legitimate (normal) based on system integrity, confidentiality and the availability of information resources [15]. NIDS does this by reading all incoming packets and endeavouring to find number of TCP connection demands to a huge number of different ports is detected, one could suppose that there is someone conducting a port scans of some or

all of the computers in the network. It typically tries to detect incoming shell codes in the same approach that a usual intrusion detection system does. Frequently examining precious information about an ongoing intrusion can be learned from outgoing or local traffic and also work with other systems as well. For example renew some firewalls blacklist with the IP address of computers used by intruder.

**2.1.2. Host-based intrusion detection system (HIDS)** examines elements of the dynamic behaviour and the status of computer system, vigorously inspects the network packets [15]. A HIDS also check that proper regions of memory have not been modified, for example- the system-call table comes to mind for Linux and various v table structures in Microsoft Windows. For each object in question typically remember its attributes and create a checksum of some kind for the substances, this information gets stored in a protected database for later comparison (checksum-database). At installation time- whenever any of the observed objects change legitimately- a HIDS have to initialize its checksum database by examining the proper objects. Persons in charge of computer security need to control this process tightly in order to prevent intruders making unauthorized changes to the database.

**2.2 IDS Techniques** There are two complementary trends in intrusion detection [16]:

**2.2.1. Misuse detection** The search for evidence of attacks based on the knowledge collected from known attacks and is referred to as misuse detection or detection by appearance.

**2.2.2. Anomaly detection** The search for deviations from the model of unusual behaviour based on the observations of a system during a normal state and is referred to as anomaly detection or detection by behaviour

### 3. DATA MINING TECHNIQUES FOR INTRUSION DETECTION SYSTEM

Data mining is extracting facts, secret information in large degrees of raw data. Typical tasks of data mining are detecting fraud and abuse in insurance and finance, predict peak load of a network. Hence Data Mining-based anomaly detection is become widespread in essence. Intrusion is an action that tries to destroy that secrecy, integrity and usability of network information, which is unlicensed and exceed authority. Data mining can be supervised, unsupervised supervised or reinforcement learning is to use the available data to build one particular variable of interest in terms of rest of data. Anomaly detection refers to discovering patterns in a given dataset that deviates from an established normal behaviour. The patterns as a result detected are called anomalies and turn to critical and actionable information in several application domains. Anomalies are also known as outlier, surprise deviation etc. Anomaly detection algorithms require a set of normal data to train the model and implicitly consider that anomalies can be treated as patterns not observed before. An outlier may be defined as a data point which is very different from the rest of the data, based on some measure; we use several detection methods in order to see how efficiently these methods may deal with the problem of anomaly detection. The statistics community has studied the concept of outliers widely. In these techniques, the data points are modelled using a stochastic distribution and points are verified to be outliers depending upon their relationship with this model. On the other hand with increasing dimensionality[17].

### 3.PROPOSED SYSTEM

The proposed system is consisting mainly of two major tasks which are:

1. Feature Reduction.
2. Attack Detection.

The proposed intrusion detection system is illustrated in figure (1) which consisting of the following stages:

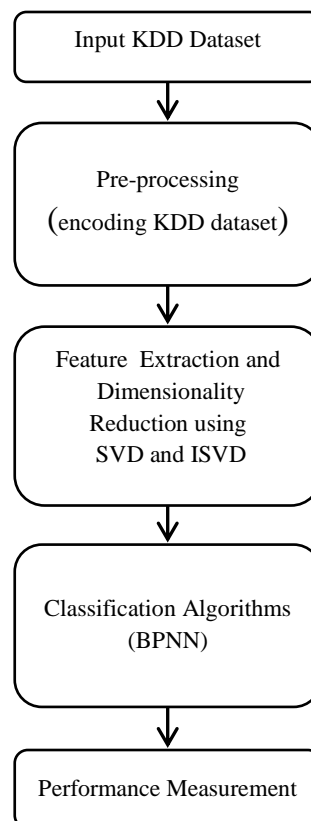


Figure .1 :Proposed Intrusion Detection System

1. Input Dataset Stage: In the first step of the proposed system takes the KDD Cup 99 dataset as an input. And it is given to the next pre-processing step.
2. Pre-processing Stage: KDDCUP1999 dataset contains number of features and these are in different format. Some are number format and others are in character format. So, different format dataset is converted into similar format to be used in the next phase.
3. Feature Extraction and Dimensionality Reduction Stage: Dimensionality reduction step is used for Feature Extraction phase by extracting suitable features from dataset, and reduce the KDD dimensions as well. In this step, different algorithms used such as Singular value decomposition (SVD), and proposed Improved Singular value decomposition (ISVM) techniques for features reduction. This step reduces the dimensionality of dataset and extracted features to be given as an input to the next step.
4. Detection Model Stage: Neural Network back propagation (BPNN) is used as a classifier in this step. It is important to take previous step output dataset as an input and trains the network using BPNN model is used both linear and non-linear classifier. For the non-linear classified stage different kernels is used like Gaussian, linear, and polynomial kernel as a transformation and mapping approach.
5. Performance Measurement Stage: Evolution for the results of the classification outputs using different criteria.

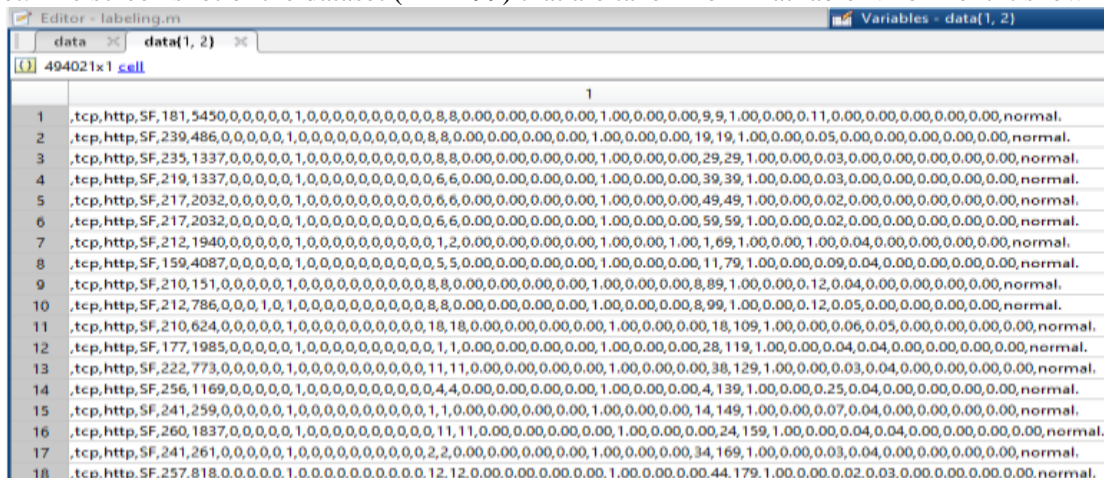
#### 4.PROCEDURE USING IN PREPROCESSING

KDD'99 as an input dataset contains number of features and these are in different format. Some are number format and others are in character format. So, these different format datasets are converted into similar format to be extracted to the next phase.

Since there are some features of KDD CUP1999 datasets are continuous, thus a process for normalizing these features have been done in order to become more convenient with the data mining, different dimensionality algorithms and classification algorithms. Normalization is used for data pre-processing, where the features data are scaled so as to fall within a small specified range such as -1.0 to 1.0 or 0.0 to 1.0. Normalizing the input values for each feature measured in the training samples will help to speed up the learning phase.

##### 4.1. Dataset Labelling

The dataset should be labelled by using 10% of the corrected dataset should be labelled by using the whole feature space in the KDD 10% corrected dataset as it shown in the screen shot which is located in the feature of the whole dataset. The screen shot of the dataset (KDD 99) that are taken from mat lab environment it shown in Figure(2).



	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1

Figure.2 :Sample data of 10% correction KDD cup dataset.

The dataset records contain 42 features (e.g., protocol type, service, and Flag) and is labelled as either normal or an attack with one specific attack type as shown in Figure (3), if we take a sample from the dataset before doing the scaling (normalization), first row as an example. We have noticed that the feature (42) has the normal type of attack as we describe that before in Table (1).

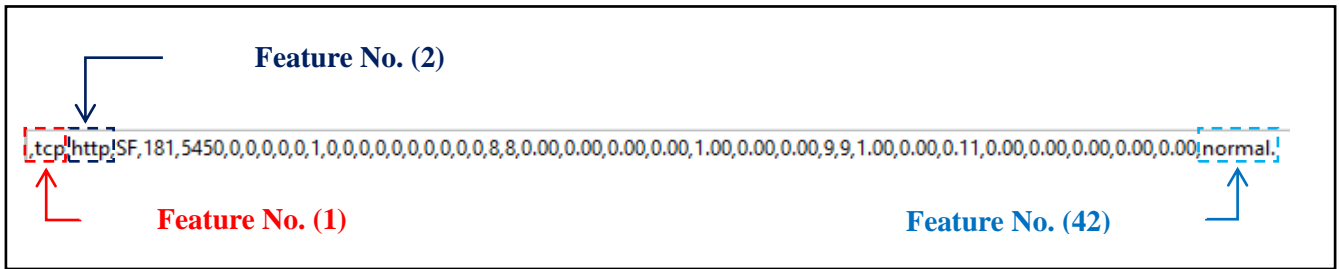


Figure. 3: First row (data sample) of 10% correction KDD cup dataset.

Table.1. Class labels and number of records of "10% KDD'99" dataset[82]

Attack Type	Original Number of Records	Number of Records after removing duplicated instances	Attack Category
Back	2203	994	DoS
Land	21	19	DoS
Neptune	107201	51820	DoS
Pod	264	206	DoS
Smurf	280790	641	DoS
Teardrop	979	918	DoS
Satan	1589	908	Probe
Ipsweep	1247	651	Probe
Nmap	231	158	Probe
Portsweep	1040	416	Probe
Normal	97277	87831	Normal
guess_passwd	53	53	R2L
ftp_write	8	8	R2L
Imap	12	12	R2L
Phf	4	4	R2L
Multihop	7	7	R2L
Warezmaster	20	20	R2L
Warezclient	1020	1020	R2L
Spy	2	2	R2L
buffer_overflow	30	30	R2L
Loadmodule	9	9	R2L
Perl	3	3	R2L
Rootkit	10	10	R2L

So, the dataset is labelled according to the following attacks which are fall into one of five categories listed below in Table (2):

Table .2 .Our Class labelling of "10% KDD'99" dataset.

Attack Type	Description	Sub Types	Label
(DoS) Denial of Service	Attacker tries to prevent legitimate users from using a service	Smurf	1
		Neptune	
		Back	
		Teardrop	
		Pod	
		Land	
Normal	data with no attack	normal	2
Probe	Attacker tries to prevent legitimate users from using a service.	Satan	3
		Ipsweep	
		PortswEEP	
		Nmap	
(R2L) Remote to Local	Attacker does not have an account on the victim machine, hence tries to gain access	WareZclient	4
		guess_passwd	
		WareZmaster	
		Imap	
		ftp_write	
		Multihop	
		Phf	
		spy.	
User to Root (U2R)	Attacker has local access to the victim machine and tries to gain super user privileges	buffer_overflow	5
		Rootkit	
		Loadmodule	
		Perl	

The algorithm steps that is used for doing the class labelling is shown in Algorithm (1).

**Algorithm 1.** KDD 99 Class Labelling

**Input:** 10% KDD data set  $T = D (F, C)$   
Normalized 10% KDD Dataset

**Output:** Class labels  $C$

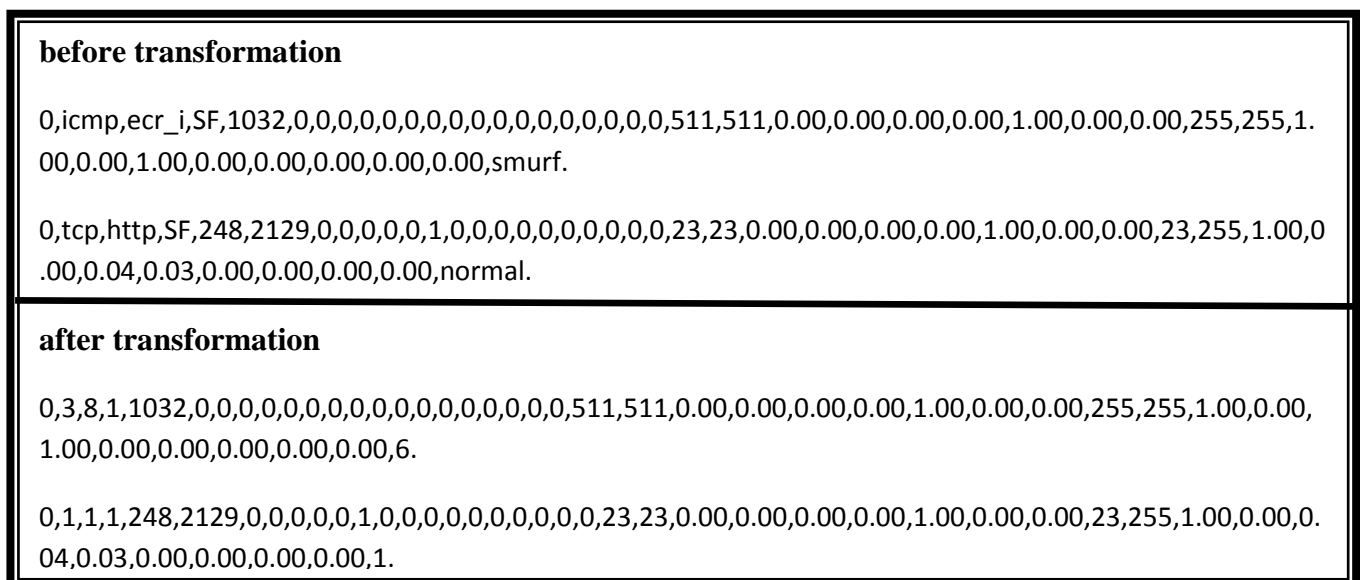
1. Initialize class labels  $class[i] \leftarrow 0, F \leftarrow$  'inital set of all features',  $C \leftarrow$  'class label',  $Class_{labes}[1..5], Du = D, Dl = ; ;$
2. **Repeat**
3.     **For** each column of feature  $f \in F$
4.     Choose feature number (42)
5.      $Tage \leftarrow f(42)$
6.     If  $Tage == class\_Labels[1..4]$
7.      $C \leftarrow Class\_label[1..4]$
8.     Obtain new labeled instances old one from  $Du$  induced by  $f$ ;
9.     **until**  $F = ;$  or  $Du = I_T$ ;
10. Return Selected features:  $S$ .

" There are many nominal values like HTTP, ICMP,SF in the dataset. therefore we have to transform these nominal values to numeric values in advance . For example, the service type of "tcp" is mapped to 1,"udp" is mapped to 2,"icmp" is mapped to 3 and we will follow table(3)to transform the nominal values of dataset features into the numeric values".

**Table .3 .Transformation Table.**

Type	Feature Name	Numeric value
Protocol-type	TCP	1
	UDP	2
	ICMP	3
Flag	SF	1
	S1	2
	REJ	3
	S2	4
	S0	5
	S3	6
	RSTO	7
	RSTR	8
	RSTOS0	9
	OTH	10
	SH	11
Service	All services	1 to 66
Attack	All attack	1 to 23

The transformation the original KDDCUP1999 dataset will become as shown in figure(4).



**Figure .4 :Pre-processing Original KDDCUP1999 dataset before and after transformation.**

#### 4.1.1 Mean range

The second step of the pre-processing KDD’99’s dataset is to find the mean range between [0,1]. We do that by finding the maximum and minimum value of a given feature, then it will be transformed the feature into a range of value [0,1] by using



$$x_i = \frac{v_i - \min(v_i)}{\max(v_i) - \min(v_i)} \quad (1)$$

#### 4.1.2 Normalization

The third step is to normalized those features depending on the minim and the maximum values that have been calculated in the previous step. It is estimated depending on the statistical normalization that has been described in Equation

The statistical normalization is defined as

$$v_i = \frac{v_i - \mu}{\sigma} \quad (2)$$

where  $\mu$  is mean of  $n$  values for a given attribute

$$\mu = \frac{1}{n} \sum_{i=1}^n v_i \quad (3)$$

and  $\sigma$  is its stand deviation

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (v_i - \mu)^2} \quad (4)$$

#### 4.2. Feature Extraction and Dimensionality Reduction

Feature extraction and Dimensionality reduction is defined as follows: given a set of candidate features, Select a subset or a feature that performs the best under some classification algorithms. This process can reduce not only the cost of recognition by reducing the number of features, but also provide a better classification accuracy due to finite dataset size effects..

##### 4.2.1 Singular-Value Decomposition(SVD)

It is a powerful computational tool and commonly used in the solution of matrix rank estimation Singular-Value Decomposition (SVD) algorithm steps are described in the algorithm (2) [86].

---

#### Algorithm 2. Singular-Value Decomposition (SVD)

---

**Input:** Generate Data matrix  $X$

**Output:** New Dimensions  $C$

**1.Repeat**

2.Applying SVD to the matrix  $X$  as  $X = USV^T$

$X \rightarrow$  is an  $m \times n$  matrix

$-m \rightarrow$  no. of sessions (vectors)

---

- $n \rightarrow$  is no. of attributes)

$U \leftarrow XX^T$  matrix of the eigenvectors

$S$  is matrix which is diagonal

$V \leftarrow$  is matrix the eigenvectors.

3. Construct the covariance matrix from this decomposition by

$$XX^T XX^T \leftarrow (USV^T)(USV^T)^T = (USV^T)(VSU^T)$$

4.  $V \rightarrow$  an orthogonal matrix ( $V^T V = I$ ),  $XX^T = US^2 U^T$

5. square roots of the eigenvalues of  $XX^T$  are the singular values of  $X$

6. until Represent every transaction  $l_i$  over the time interval  $t$  as a vector  $x(t)_i$

1. Return  $U^T X$

### 4.2.2 Improved Singulars Values Decompositions (ISVD)

The propose improvements to SVD algorithms was base on the thought of the improve Singulars Values Decompositions SVD which produced a diagonals matrixes  $S$ , for the dimensions as the ranks of  $X$  and by means nonnegative diagonals element in decrease orders , and Unitarians matrices  $U$  and  $V$ .

The idea of the proposed (ISVD) is shown in Figure (5).

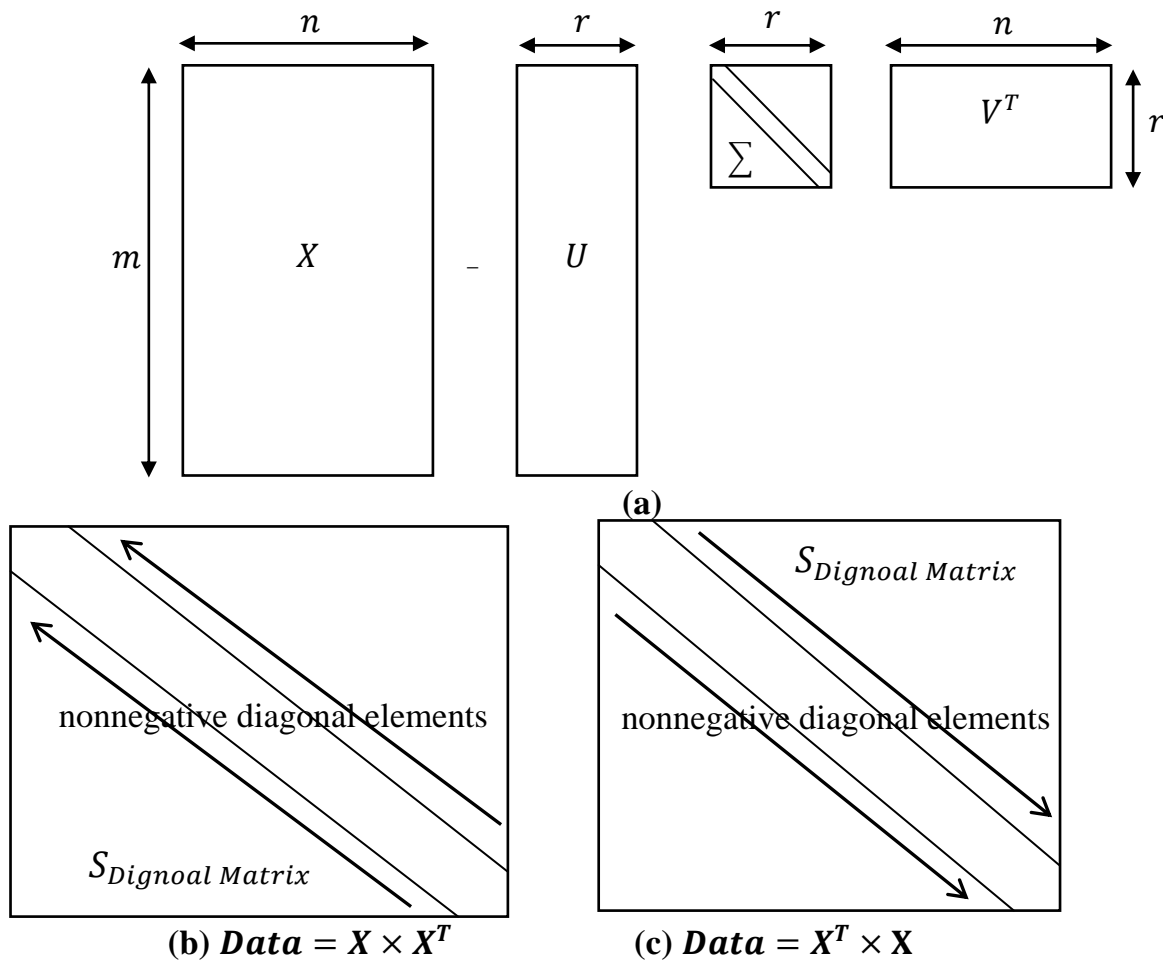


Figure. 5 :The form matrix of an (ASVD) Accelerated singular-value decomposition.

The best approximation (with respect to  $F$  norm) of  $X$  among all the matrices with rank no larger than the reduced dimension that we would like. base on the sizes off data matrixes  $X$  , ISVD computed the eigenvector o f  $X * X^T$  or  $X^T * X$  base o n the ratios among the numbers of feature in addition to the numbers for samples in the whole data matrix  $X$ , and then convert them to the eigenvectors of the other as it shown in Figure (5).

The proposed improved algorithm steps are described in algorithm (3)

#### Algorithm 3. Dimensionality Reduction Algorithm Improved SVD (ISVD)

**Input:** Generate Data matrix  $X$ (KDD 99)

**Output:** New Dimensions  $C$

1. Initialize
2. Repeat
3. Construct the covariance matrix ( $X$ ) from this decomposition depend on

- 
4. If  $\frac{\text{No.Features}}{\text{No.of Samples}}$
  5. **Data**  $\leftarrow \mathbf{X} \times \mathbf{X}^T$
  6. else
  7. **Data**  $\leftarrow \mathbf{X} \times \mathbf{X}^T$
  8. Compute the d-dimensional mean vectors for the different classes from  $\mathbf{X}$ .
  9.  $\mathbf{XX}^T \mathbf{XX}^T = (\mathbf{USV}^T)(\mathbf{USV}^T)^T = (\mathbf{USV}^T)(\mathbf{VSU}^T)$
  10.  $\mathbf{V}$  is an orthogonal matrix ( $\mathbf{V}^T \mathbf{V} = \mathbf{I}$ ),  $\mathbf{XX}^T = \mathbf{USU}^T$
  11. Compute the scatter matrices (in-between-class and within-class spread  $\mathbf{X}$ ).
  12. The squares rooted off the Eigen value of  $\mathbf{XX}^T$  were the singulars value of  $\mathbf{X}$ .
  13. Compute the eigenvectors ( $e_1, \dots, e_d$ ) and corresponding eigenvalues ( $\lambda_1, \dots, \lambda_d$ ).
  14. Sort the eigenvectors by
  15. decreasing eigenvalues
  16. choose  $k$  eigenvectors with the largest eigenvalues to form a
  17.  $d \times k$  dimensional matrix  $\mathbf{W}$
  18. (where every column represents an eigenvector).
  19. Use this  $d \times k$  eigenvector matrix to transform the samples
  20. onto the new subspace.  $\mathbf{Y} \leftarrow \mathbf{X} \times \mathbf{W}$
  21. where ( $\mathbf{X}$  is a  $n \times d$ -dimensional matrix representing the  $n$
  22.  $\mathbf{y}$  are the transformed  $n \times k$ -dimensional samples in the new
  23. subspace)
  24. **Until** Convergence
- 

### 4.3. Back Propagation (BP)

As mentioned before, the mathematical model of the Biological Neural Network is defined as Artificial Neural Network. One of the Neural Network models which are used almost in all the fields is Back Propagation Neural Network [90]. The back propagation algorithm is used in multi layered feed-forward ANNs. This means that the artificial neurons are organized in layers, then send their signals "forward", and then the errors are propagated backwards. The network receives the inputs signal by neurons in the input layer, and the output of the network is given by the neurons on an output layer. Then, may be one or more intermediate hidden layers [96]. The back propagation algorithm uses supervised learning, which means that the algorithm is provided by examples of the inputs and outputs that the network must be compute, and then the error which is the difference between actual and expected results is calculated. The idea of the back propagation algorithm which is used to reduce this error, until the ANN has been learned the training data. The training begins with initial (random) weights, and the goal is to adjust (update) them so that the error will be minimal.

The following Pseudo coding in Algorithm (4) describes the BP algorithm [90]

---

#### Algorithm 4. Back propagation Neural Network

---

**Input:** Input features or domain

**Output:** Class Type

1. **Initialize** all weights with small random number between [-1,1]
  2. **Repeat**
  3. **For** every pattern in the Training set
  4. Present the pattern to the network
  5. Propagation the input forward through the network
  6. **For** each layer in the network
  7. **For** each neuron in the layer
-

8. Calculate the weight sum of the input to the neuron
9. Add the threshold to the sum
10. Calculate the activation function for the neuron
11. **End**
12. **End**
13. Propagation the input forward through the network
14. **For** each layer in the network
15. **For** each neuron in the layer
16. Calculate the neuron's signal error
17. Update each neuron's weight in the network
18. **End**
19. **End**
20. Calculate global error
21. Calculate the error function
22. **End**
23. **Until** (Maximum number of iteration< specific) and (error> specific)

#### 4.4. Evaluation

To assess the validation and accuracy of the intrusion detection and classification system based on feature selection and dimensionality reduction, in this case we need to introduce the measures of validation the results of classification.

**4.4.1 Confusion matrix** for intrusion detection is defined as an  $m \times m$  matrix, where  $m$  denotes the number of classes. A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The confusion matrix shows the classes which are correctly classified and the classes that are misclassified. Confusion matrix is used to evaluate these parameters as shown in Table (4).

**Table .4. Confusion Matrix**

Attack		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

The performance of neural network can be evaluated using various parameters. Standard parameters include classification accuracy, detection rate and false positive rate, the given parameter calculated using True Positive (TP), False Negative (FN), False Positive (FP) and True Negative (TN).

- **True Positive (TP):** true positive results refer to correct classifications of positive cases.
- **True Negative (TN):** true negative results refer to correct classifications of negative cases.
- **False Positive (FP):** false positive results refer to incorrect classifications of positive cases into negative class.
- **False Negative (FN):** false negative results refer to incorrect classifications of negative cases into class positive.

#### 4.4.2 Accuracy

Recognition Rate is defined as the ratio between the numbers of correct recognition decision to the total number of attempts as it is given in equation(1.5).

$$Accuracy = \frac{TP + FN}{Total\ number\ of\ test\ samples} * 100 \quad (5)$$

The evaluating performance of face retrieval system is measured through how many predictions for specific query are truly relevant to are a query. The retrieval efficiency is generally evaluated through two well- known metrics, precision and recall. The formula for calculating these measures are given as in equations and respectively.

#### 4.4.3 Detection Rate

Detection rate is defined as the ratio between the numbers of true positive divided by the total number positive detection samples as it described in (1.6).

$$Detection\ Rate = \frac{TP}{TP + FP} \quad (6)$$

#### 4.4.4 False Alarm

False alarm is defined as the ratio between the numbers of false positive detection divided by the total number the false positive and true negative test samples as it described in (1.7).

$$False\ Alarm = \frac{FP}{FP + TN} \quad (7)$$

the category of data behavior in intrusion detection for binary kind classes (Normal and Attacks) in term of true negative, true positive, false positive and false negative.

## 5.RESULTS

Show the overall performances results of the Neural Network (BPNN) on KDD Cup 99 depending on training and testing datasets by using algorithm (SVD and The Improved for SVD) that we have proposed in our system. intrusion detection classification system based feature reduction using different algorithms on the KDD Cup 99 dataset. algorithms are used for reducing the 42 features of the KDD data set, and classification algorithms to detect the four type of ID attacks.

**5.1. Singular-Value Decomposition(SVD)**used for dimensionality reduction with Neural Network(BPNN) as an attack detection for intrusion detection system. We used the SVD depending on different number of selected features which called k-feature section (the reduced no. of feature as a dimensionality reduction that we want to select). usually after mean centering (normalizing) the data matrix for each feature . That means that there is no majority to give us

Rasha Thamer Shawe et al., Using An Improved Data Reduction Method in Intrusion detection system  
any clue about the k selection in this case we test the SVD using three different k starting from (k=21,11, and 7) from the original (42) feature space by using trial and error.

Tables (5) shows the confusion matrix results after applying the Neural Network (BPNN) classification algorithm on KDD Cup 99 training and testing datasets depending on selection just (k=21) features.

**Table .5. confusion matrix when using SVD with K=21 and BPNN in the training and testing dataset**

Original Feature	Dimensionality Reduction		Training Dataset						
	Feature Selection	Training Time	Confusion matrix				Accuracy		
			TP	TN	FP	FN			
42	21	90.521816	96.4787	33.1344	3.5213	66.8656	Detection	0.7444	
							False	0.0961	
							Accuracy	81.6722	
	Testing Time	Testing Dataset						Accuracy	
		Confusion matrix							
		TP	TN	FP	FN				
14.646747	75.9007	3.5883	24.0993	96.4117	Detection	0.9549			
					False	0.8704			
					Accuracy	86.1562			

Table (6) show different results depending on (k=11) and applied on training, and testing dataset.

**Table .6.confusion matrix when using SVD with K=11 and BPNN in the training and testing dataset**

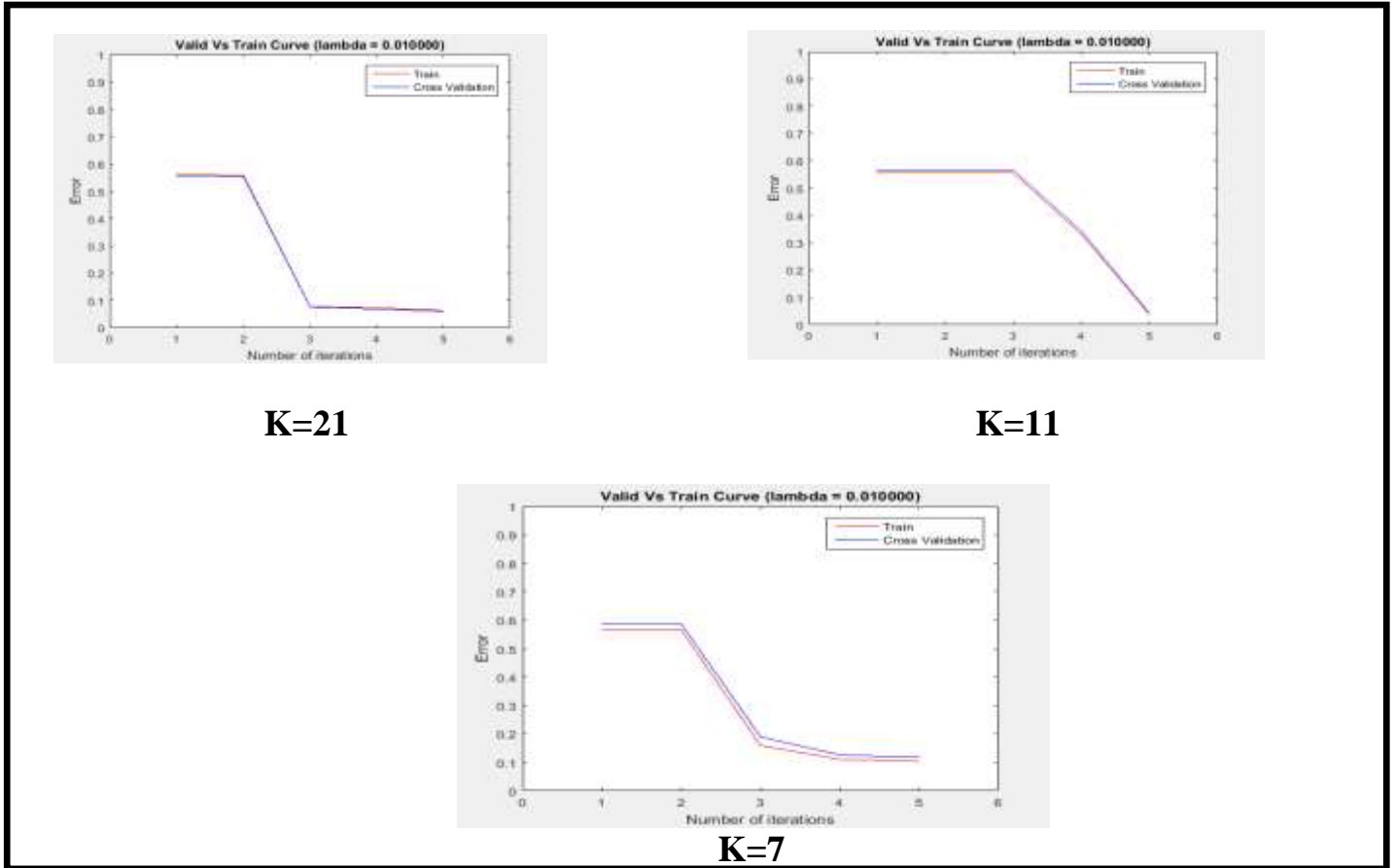
Original Feature	Dimensionality Reduction		Training Dataset						
	Feature Selection	Training Time	Confusion matrix				Accuracy		
			TP	TN	FP	FN			
42	11	92.203391	93.6704	7.2423	6.3296	92.7577	Detection	0.9282	
							False	0.4664	
							Accuracy	93.2141	
	Training Time	Testing Dataset						Accuracy	
		Confusion matrix							
		TP	TN	FP	FN				
8.060197	57.5514	5.9335	42.4486	94.0665	Detection	0.9065			
					False	0.8774			
					Accuracy	75.8090			

Table (7) show different results depending on different k-selection (k=7) and applied on training and testing dataset.

**Table .7.confusion matrix when using SVD with K=7 and BPNN in the training and testing dataset**

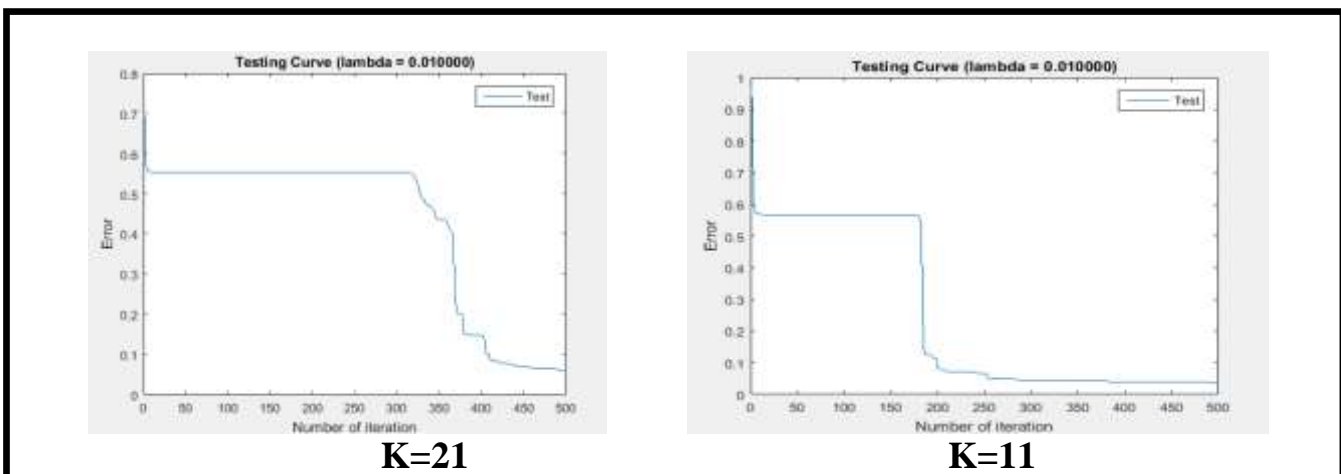
Original Feature	Dimensionality Reduction		Training Dataset						
	Feature Selection	Training Time	Confusion matrix				Accuracy		
			TP	TN	FP	FN			
42	7	45.154621	89.5418	11.0414	10.4582	88.9586	Detection	0.8902	
							False	0.4864	
							Accuracy	89.2502	
	Testing Time	Testing Dataset						Accuracy	
		Confusion matrix							
		TP	TN	FP	FN				
10.289327	43.6686	56.2010	56.3314	43.7990	Detection	0.4373			
					False	0.5006			
					Accuracy	43.7338			

Figure (6) shows the training error curve depending on  $k=21,11,7$  using (BPNN) and (SVD), and Figure (4.2) show the testing error curve result respect with the iteration number by using (SVD) with  $k=21,11,7$  and (BPNN) classification algorithm.



**Figure. 6: the training error curve result with respect the iteration number by using SVD with ( $k=21,11,7$ ) and Neural Network (BPNN) classification algorithm.**

Figure (7) shows the testing error curve result respect with the iteration number by using SVD with  $k=21,11,7$  and (BPNN) classification algorithm.



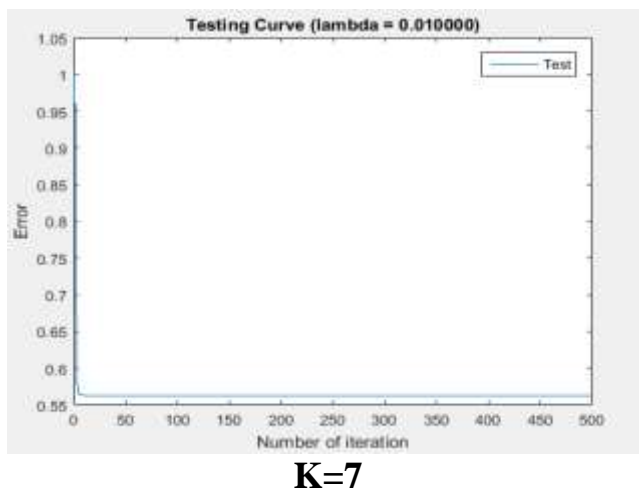


Figure.7:testing error curve depending on (BPNN)and (SVD) using k=21,11,7.

### 5.2. Improved Algorithm (ISVD)

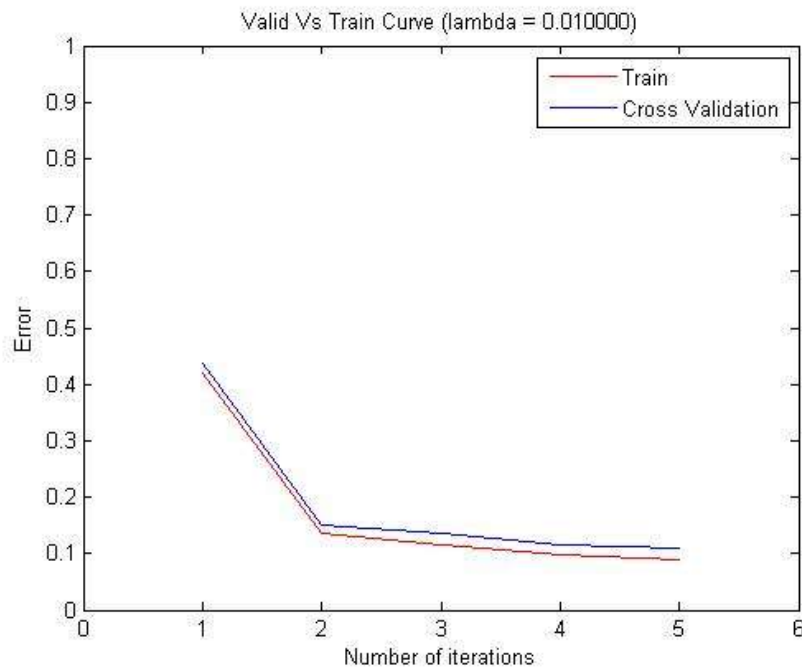
The proposal for the SVD development and according to the features ranking which biased by the reconstruction value, the largest best possible feature reduction domain is (k=7).. Table (8) shows the confusion matrix results after applying the Neural Network (BPNN) classification results by using seven features only (k=7) from (ISVD)algorithm on KDD Cup 99 datasets using training and testing dataset.

Table .8. confusion matrix when using ISVD and BPNN in the training and testing dataset.

Original Feature	Dimensionality Reduction		Training Dataset					Accuracy	
			Confusion matrix						
	Feature Selection	Training Time	TP	TN	FP	FN			
42	7	189.2327	90.9726	9.7238	9.0274	90.2762	Detection	0.9034	
							False	0.4814	
							Accuracy	90.6244	
	7	66.72870	92.8636	4.1755	7.1364	95.8245	Detection	0.9539	
							False	0.5473	
							Accuracy	94.3441	

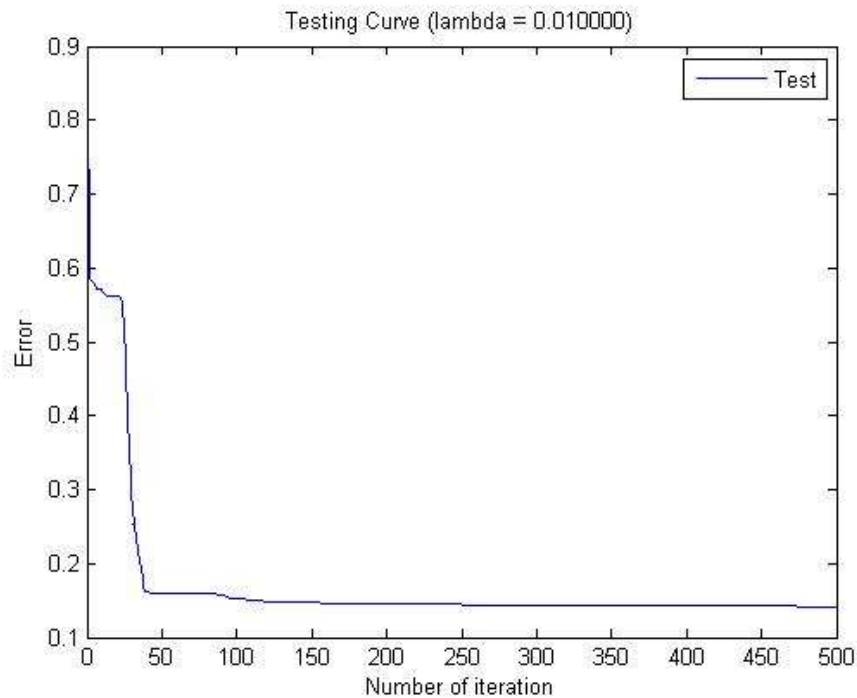
Figure (8) shows the training error curve depending on (k=7) using (NN) using (BPNN)with (ISVD).





**Figure.8: The Training error curve depending on (ISVD).**

Figure (9) show the testing curve error result respect with the iteration number by using (ISVD) with (k=7) and (BPNN) classification algorithm.



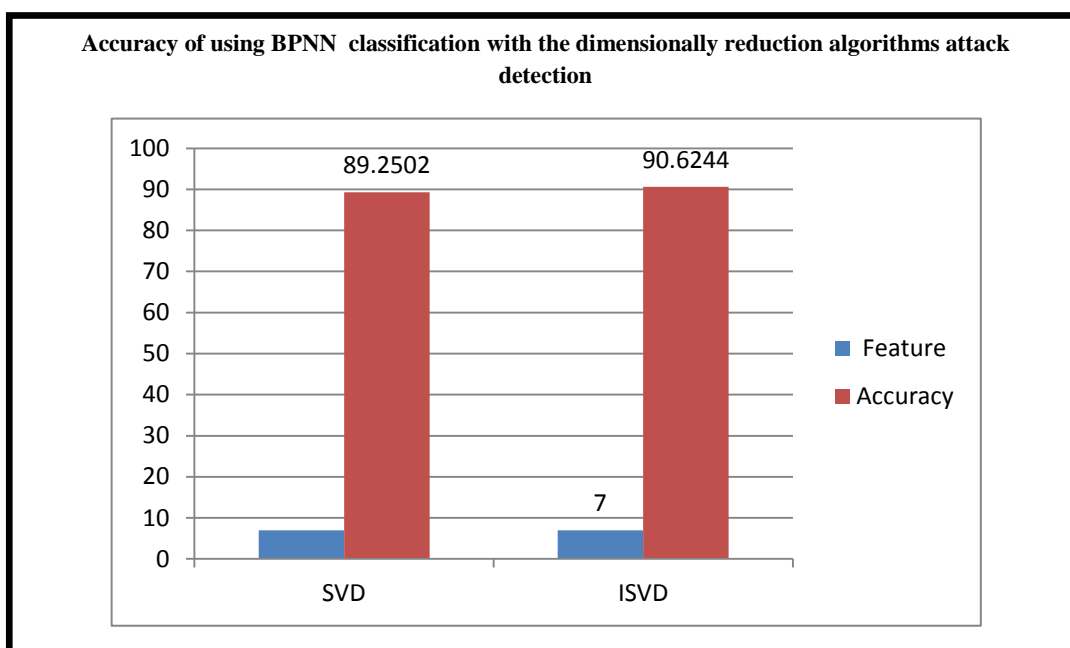
**Figure. 9: The Testing error curve depending on (ISVD).**

Tables (9) and (10) show the overall performances results of the Neural Network (BPNN) on KDD Cup 99 depending on training and testing datasets by using all algorithms (SVD and The Improved for SVD) that we have proposed in our system.

**Table .9. Accuracy for using BPNN with Dimensionality Reduction Algorithm on training dataset.**

Dataset Features No	Dimensionality Reduction Algorithm		Accuracy
	Algorithm	Feature No.	
42	SVD	7	89.2502
42	ISVD	7	90.6244

Figure (10) illustrate the performance results using training dataset and depending on feature extraction and dimensionality reduction algorithm that we used with the Neural Network (BPNN) classification algorithm for attack detection.

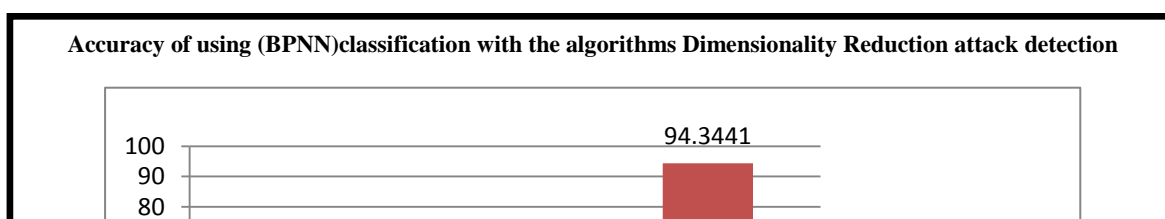


**Figure.10 :Accuracy of using BPNN classification with the dimensionally reduction algorithms attack detection.**

**Table .9. Accuracy for using BPNN with Dimensionality Reduction Algorithm on testing dataset.**

Dataset Features No	Dimensionality Reduction Algorithm		Accuracy
	Algorithm	Feature No.	
42	SVD	7	43.7338
42	ISVD	7	94.3441

Figure (11) illustrate the performance results of the feature extraction and dimensionality reduction algorithm that we used with the Neural Network (BPNN) classification algorithm for attack detection.



**Figure.11: Accuracy of using (BPNN)classification with the algorithms Dimensionality Reduction attack detection.**

## 6.Conclusions

The aim of this paper is to improve SVD algorithm and get better performance in IDS. dimensionality reduction are used including the improved SVD and one classification algorithms are used which are the Back propagation Neural network (BPNN) to detect the four types of attack addition to normal system .It is obvious from the obtained results ,that ISVD has achieved better results than SVD algorithm. The main advantage for using ISVD to speeding the execution time in extracting and reducing the IDS features. Future Work Using IDS the reduction number feature used dimension reduction algorithm use the particle of swarm optimization algorithm and genetic algorithm.

## REFERENCES

- [1]. B. D. Caulkins, J.Lee, and M.Wang, "A Dynamic Data Mining Technique for Intrusion Detection Systems", ACM Proceeding of the 43rd Annual Southeast Regional Conference, pp. 148-153, 2005.
- [2].R.U. Rehman, "Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID", Pearson Education, Inc. Publishing as Prentice Hall PTR, 2003.
- [3].H.NashatGabra, Dr. Ayman M. Bahaa-Eldin, Prof. Huda Korashy," Classification of IDS Alerts with Data Mining Techniques", nternational Journal of Electronic Commerce Studies Vol.5, No.1 , pp.1-6, 2014.
- [4].O.Maimon, L.Rokach," INTRODUCTION TO KNOWLEDGE DISCOVERY IN DATABASES", Department of Industrial Engineering Tel-Aviv University maimon@eng.tau.ac.il, Department of Industrial Engineering Tel-Aviv University liorr@eng.tau.ac.il.
- [5].M. Tavallae, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of theKDD CUP 99 Data Set, 2009" IEEE Int. Conf. Comput. Intell. Security DefenseAppl., 2009, Page. 53-58.

- [7].Lee, W., S.J. Stolfo, and K.W. Mok, A Data Mining Framework for Building Intrusion Detection Models. IEEE Symposium on Security and Privacy, 1999.
- [8].Mohammad Sazzadul Hoque, Md. Abdul Mukit, Md. Abu Naser Bikas, An Implementation of Intrusion Detection System Using Genetic Algorithm, International Journal of Network Security & Its Applications, Volume 4, Number 2, pages 109 - 120, March 2012.
- [9].C. Cranor, T. Johnson, and O. Spatschek. Gigascope: A stream database for network applications. In SIGMOD, pages 647–651, 2003.
- [10].J. Georges and A. H. Milley, “Kdd’99 competitions: Knowledge discovery contest,” ACM SIGKDD Explorations Newsletter, vol. 1, no. 2, pp. 79–84, 2000.
- [11]. Knowledge discovery in databases DARPA archive. Task Description.
- [12]. MIT Lincoln Lab., I.S.T.G., The 1998 intrusion detection off-line evaluation plan.
- [13]. Paxson, V., Bro: a system for detecting network intruders in real time Computer networks, 1999. 31(23-24): p. 2435-2463.
- [14]. Kayacik, H.G., A.N. Zinic-Heywood, and M.I. Heywood. Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets. 2005: Citeseer.
- [15]. A. Macgregor, M.Hall, P.Lorier and J.Bruskil, “Flow clustering using machine learning techniques”, In PAM 2004, Antibes-Juan-Les-Pins, France, LNCS. pp. 205-214, 2004.
- [16]. S. Kumar, Classification and Detection of Computer Intrusions, Ph.D. Thesis, Purdue University.
- [17]. White paper, Intrusion Detection: A Survey, ch.2, DAAD19-01, NSF, 2002