On the Three Levels Security Policy Comparison between PCA and SVM

[1]A. RADI, [2]A. Kartit, 3B. REGRAGUI, 4M. El Marraki, [5] D. ABOUTAJDINE, [6]A.RAMRAMI,

[1, 3, 5, 6]Department of Physic, Faculty of Sciences, University Mohammed V, Rabat Morocco
[2,4]Department of Computer Sciences, Faculty of Sciences, University Mohammed V, Rabat Morocco

Summary
The omnipresence of the computer system tools intensified every year in all companies. They integrate equipments, data and services that constitute a wealth to protect. A lot of mechanisms have been developed to assure the computer systems security. Conventional intrusions detection systems "IDS" have shown their insufficiencies and limits. In the previous articles, we have proposed an exact algorithm for the deployment of security policies for single computer systems [1] and an enhanced three levels security policy for complex computer systems [2] to improve computer systems security approach. However, manual analysis of the huge volume of data generated and audit data are usually impractical. To overcome this problem and evaluate our system proposed in [2], we use Support Vector Machines (SVM) which becomes one of the most important techniques for anomaly intrusion detection due to their good generalization nature and the ability to overcome the curse of dimensionality [3, 4] with applications involve large number of events as well as large number of features.
Experimental analysis and comparison show that the proposed system in [2] outperformed other recent systems [5, 6] in precision, computation time, false positive and false negative rate.

Keywords:
Intrusions detection, Security policy, Support vector machine, Principal component, Classification,

1 1. Introduction

Intrusion detection is a critical component for computer systems security. Various intrusion detection systems are proposed to meet the challenges of a vulnerable internet environment and rough attackers. In addition, more computer systems become safer and the problem of events and features becomes difficult to treat. Having a large number of events and input features helps to understand better the system behavior, but before that. It is necessary to eliminate the insignificant and/or useless input features to simplify the problem and to make detection that may result faster and more accurate.
Various artificial intelligence techniques have been increasingly used for intrusion detection systems to overcome dimensionality problems. Qiao and al. [7] presented an anomaly detection method by using a hidden Markov model to analyze the UNM dataset (related to University of New Mexico). Lee and al. [8] established an anomaly detection model that integrates the association rules and frequency episodes with fuzzy logic to produce patterns for intrusion detection. Mohajerani and al. [9] developed an anomaly intrusion detection system that combines neural networks and fuzzy logic to analyze the KDD dataset (Knowledge Discovery in Databases). Wang and al. [10] applied genetic algorithms to optimize the membership function for mining fuzzy association rules. Yao and al. [6] proposed a new SVM algorithm for considering weighting levels of different features and the dimensionality of intrusion data.
In this paper, we use Support Vector Machines which becomes one of the most important techniques for anomaly intrusion detection due to their good generalization nature and the ability to overcome the curse of dimensionality with applications involve large number of events as well as large number of features.
Experiments results and comparisons are conducted through intrusion datasets: the KDD Cup 1999 dataset [11].

2. Support vector machines

2.1 Introduction

The support vector machines or maximum separators margin are a set of supervised learning techniques, based on statistical learning theories used to solve problems related to classification and regression analysis. The original SVM algorithms have been developed in the 1990s by VLADIMIR VAPNIK and the current standard incarnation (soft margin) was proposed by CORINNA CORTES -VLADIMIR VAPNIK [12]. The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high dimension feature space. In this feature space a linear decision surface is constructed. Special properties of the decision surface ensure high generalization ability of the learning machine. The fact that they are well founded theoretically, and have good results in practice; SVMs have been applied to many fields (bio-informatics, information retrieval, computer vision, finance [13]). According to data type, the performance of support vector machines is similar or even superior to that of a neural network or a Gaussian mixture model.

Manuscript received February 5, 2011
Manuscript revised February 20, 2011

2.2 Formalization

A binary (two-class) classification problem can be described as follows: given some training data D, represented by a set of n labelled points of the form: $D=\{(x_i,y_i)\mid x_i\in R^n,\ y_i\in\{-1,1\},\ i=1,.....n\}$ Where x_i, are vectors of features, and y_i, are class labels, construct a rule that correctly assigns a new point x to one of the classes.

The vectors x_i correspond to objects, and the dimensions n of the space are the features or characteristics of these objects. For example, a vector may represent:
-A person, with individual features corresponding to measurements given by some medical tests (blood group and pressure, cholesterol level, white cell count,…);
-A flower, with its morphological characteris-tics: leaf shape, stem length, colour, fruit, …;
-Traffic flow events with attributes or features: date/time, protocol, IP source/destination, Port source/destination, packet Size;
and so on.
In general, instead of a binary (two-class) classification, we have a multi-class problem with l classes (l labels: $l=0,\ .....,\ l-1$). A classification method or algorithm is a particular way of constructing a rule, also called a classifier, from the labelled data and applying it to the new data.
A general binary classification problem is to find a discriminant function $f(x)$, such that $y_i=f(x_i)$ with $i=1,\ .....,\ n$. Otherwise, we want to find the maximum-margin hyperplane that divides the points having $y_i=1$ from those having $y_i=-1$. A possible linear discriminate function can be formulated as $f(x)=\mathrm{sgn}(<w,x>+b)$ (1) (<,> or dot is the inner product of two vectors) where $(<w,x>+b)=0$ can be viewed as a separate-ing hyperplane in the data space. Therefore, choosing a discriminate function is to find a hyperplane having the maximum separating margin with respect to the two classes. The final linear discriminate is formulated as $f(x)=\mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i(x_i.x+b)\right)$ (2), where n is the number of training records, $0\le\alpha_i\le C$ (constant C > 0), and x_i is the support vectors.
When the surface separating two classes is not linear, we can transform the data points to another higher dimensional space, using the so-called “kernel trick” SVM methodology, such that the data points will be linear separable. The non-linear discriminate function of SVM is formulated:

$$f(x)=\mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i K(x_i.x)+b\right) \quad (3),$$

where $K(x_i.x)$ is the kernel function that we used to transform data points.
The main idea behind the “kernel trick" is to map the data into a different space, called feature space, and to construct a linear classifier in this space. It can also be seen as a way to construct non-linear classifiers in the original space.

3. Principal Components Analysis

1.1 Introduction

Principal Components Analysis (PCA) is an effective statistical technique for reducing the dimensions of a given unlabeled high-dimensional dataset while keeping its spatial characteristics as much as possible by performing a covariance analysis between factors. As such, it is suitable for data sets from multiple dimensions field of application, such as image compression, pattern recognition (face recognition in particular), gene expression, data clustering and traffic flow events intrusion detection. One of the main advantages of PCA is that you can compress the data, i.e. by reducing the number of dimensions, without much loss of information.
However, it is generally accepted that the earliest descriptions of the technique now known as PCA were given by K. Pearson (1901) and H. Hotelling (1933) [14]. Now it is mostly used as a tool in exploratory data analysis and for making predictive models. PCA can be done by eigenvalue decomposition of a data covariance matrix or singular value decomposition of a data matrix. PCA is also known as the discrete Karhunen-Loeve transformation, or the Hotelling transformation.

1.2 Formalization

Principal component analysis involves many steps in order to transform a set of correlated variables into a number of uncorrelated variables called principal components. Like other transformation techniques, such as Fourrier transform, PCA transforms data into another representation where new variables are considered as the basis of this new presentation. While in the Fourrier Transformation deals with frequencies aspect, in PCA the variance in the data set is considered.

PCA is a linear transformation and the new basis vectors are subject to an orthonormality constraint. If we denote e_i these new basis vectors then: $e_i^T e_j = \delta_{ij} = \begin{cases} 1, & if \ i = j \\ 0, & if \ i \neq j \end{cases}$ (4)

Since PCA is a linear transformation with linear orthonormal basis vectors, it can be expressed by a translation and rotation. Then, such transformation may be expressed as the following: $y = B(x - \mu_x)$ (5) where y is output data of the input data x, B is the new base vectors matrix (i.e. $B = [e_1 e_2 .. e_n]^T$) and $\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i$, is the standard mean of x.

If we consider the two dimensional case then Figure 3.1 illustrates the basic principle of this transformation.

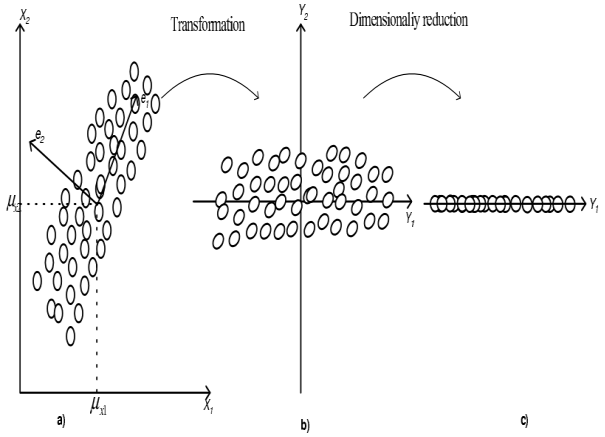

Figure .1: PCA basic principle transformation

Figure (a) presents each ith sample, denoted $x_i = [x_{i1},\ x_{i2}]^T$, of the initial data set which is transformed into another representation (Figure (b)), denoted $y_i = [y_{i1},\ y_{i2}]^T$, calculated using Equation (5).

The main portion of the variance is stored in the first variable Y_1. This means that if we ignore the second variable Y_2, as in figure (c), the main variance of the data is kept. Therefore, representing an initial data set with a new more compact space keeping much of the variance of the data in the new compact representation offers many facilities to interpret the data in a new reduced space.

This example illustrates a reduction from two dimensions into one dimension. However, in reality the reduction might be performed over hundreds or thousands of variables into only 2 or 3 variables as shown by the different experiments we conduct thereafter.

The most used transformation for generating the new compact space having less dimension axes is that of Hotelling [14] where for a set of N observed d−dimensional data line vectors $v_i,\ i \in \{1,.....,N\}$, the q principal axes $u_j,\ j \in \{1,.....,q\}$, are those orthonormal axes onto which the retained variance under projection is maximal. It can be shown that the vectors u_j are given by the q dominant eigenvectors (i.e. those with the largest associated eigenvalues) of the simple covariance matrix:

$$C = \sum_i \frac{(v_i - \bar{v})(v_i - \bar{v})^T}{N} \quad (6)$$

Such that: $Cu_j = \lambda_j u_j$ (7)

and where $\bar{v}$ is the simple mean and λ_j the eigenvalue corresponding to the eigenvector u_j.

The vector $u_j u_i = u^T (v_i - \bar{v})$ (8), where $u = \{u_1, u_2, ..., u_j\}$, is thus a q−dimensional reduced representation of the observed vector v_i.

Generally the task of how much the dimensionality can be reduced is a matter of representing as much information as possible in as small a space. In other words, to determine how many eigenvectors to ignore is a trade-off between the wanted low dimension and the unwanted information loss. Since the ith the eigenvalue is, by definition, equal to the variance of the ith variable and while we consider $\lambda_i \geq \lambda_{i+1}$, then this trade-off may be defined as the inertia, such as:

$$I_{q'} = \frac{\sum_{i=1}^{q'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} . 100 \quad \% \quad (9)$$

where q' represents the number of axes considered in the new subspace, d denotes the dimension of the input data.

The quantity denoted by $\sum_{i=1}^{q'} \lambda_i$ is called inertia explained by the subspace generated by the first q' eigen-vectors corresponding to the first q' highest eigenvalues, $\sum_{i=1}^{d} \lambda_i$ is the total inertia (variance) of the initial input data and $I_{q'}$

represents the percentage of information that is kept after transformation that corresponds to the inertia ratio explained by the new subspace.

1.3 The eigenprofiles approach

3.3.1. Introduction

Much of the previous work on anomaly intrusion detection has ignored the issue of the selection of measures of the user profile and/or application behaviour stimulus. This suggested us that an information theory approach to coding and decoding user behaviours may give new information about the user behaviours, emphasizing the most significant features to perform comparing user behaviour profiles.

In mathematical terms, we wish to find the principal components of behaviours distribution, or the eigenvectors of the covariance matrix of user's profiles set, treating behaviour as a vector in a space which dimension equal to the number of the different metrics used (equation (10)).

These eigenvectors can be thought of as a set of features that together characterize the variation between user behaviours. Each behaviour location contributes more or less to each eigen-vector which we call eigenprofile.

Each profile can also be approximated using only the best eigenprofiles, those that have the largest eigenvalues, and which therefore account for the most variance within the set of user profiles [5].

So if Γ is a profile that corresponds to behaviour of a certain user, then we can write:

$$\Gamma = (m_1 m_2 ... m_n)^T \quad (10)$$

where $m_i, i = 1,....., n$ correspond to the measures characterizing a user profile.

3.3.2. General architecture of the eigen-profiles approach

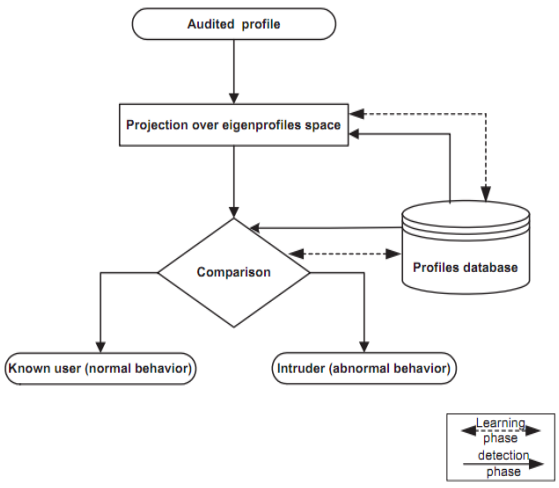

Figure 2: The general architecture of the eigen-profiles approach.

The general architecture of the proposed eigenprofiles approach is described in Figure 2.

It has two main steps:

i- The first step consists in learning the different user profiles with respect to different considered measures. A statistical knowledge database, representing different observed normal profiles during the first step, is then stored for further detection;

ii- The second step consists in observing a new profile then comparing it to different correspond-ing profile knowledges stored in the profiles database.

4. A three levels security policy system

1.4 Introduction

Traditional intrusion detection systems have shown their insufficiencies when protecting computer systems, in particular, from the inside. They permit to secure the network only on its entry point against the attacks coming from external network based on a model of a normal behaviour or database of attacks. However, according to several achieved studies [15]:

- 60 to 70% of attacks come from the inside of the computer systems.
- 70% of attacks that cause damages come from the inside of the network (Garthner Inc).
- Enterprises recorded a rise of 44% for the attacks coming from the inside between 2004 and 2005 (IDC and Pricewaterhouse Coopers).

As a result of the global economic crisis, the number of unsatisfied or dismissed employees increases each year. Some time, they can abuse their privileges they had during their period of activity, try sometimes to steal information deserve to be sold to competitor. In 1993, a British Airways company employee was smuggled over the Internet in Virgin Atlantic Airways computer system reservation to obtain the list of passengers who bought first class tickets. These passengers were then contacted by British Airways to cancel their reservations and travel on their own lines with lower price [16].

From where comes the idea to look for solutions providing protection of the computer systems from both non-authorized users (outsiders' attacks) as well as attacks from authorized users who abuse their privileges (insiders' attacks). The solution, we proposed in [2], summarised thereafter consists in setting up a three levels global security policies approach. It is a new interesting method that will offer adequate new techniques to the security managers and enhance network security.

1.5 Level 1: External Protection Policies

The first level of intrusion detection consists to use a well known intrusion detection systems using a mono or hybrid classic approach. It will be placed therefore in the firewall to prevent attacks from the outside network by denying malicious connection attempts from unauthorized parties located outside. For our case we propose a network-based intrusion detection system (NIDS) using a database of attacks [17]. The main advantage of a misuse-based detection system is that it usually produces very few positive false, its limitation is that it can not detect possible new intrusions not exist in the attacks database; this disadvantage will be improved by level 2 and level 3, which help us to detect new attacks, and the analysis of these attacks will help to update our system database.

1.6 Level 2: Functional Security Policies

The second level of detection consists to define functional security policies, which means policies according to users' tasks within the enterprise by segmentation the computer net-work into VLANs "Virtual Local Area Network" (figure 3) and the use of ACL "Access Control Lists". So:

- Users who are susceptible to communicate and share some computer system resources will be put in the same VLAN.
- Gateway machines of the different VLAN will be configured with ACL defining lists of the actions allowed only to users who belonged to the same VLAN (all other actions are forbidden) or inversely. Also, VLAN will allows, in worse case if an intruder has succeeded taking control on a host, to restrict the attack within a small subnet (few number of machines) and can't contaminate the whole computer network.

The main objective of this level is to protect inside network from the internal malicious users who can abuse their privileges (insiders' attacks) and from outside attackers who manage to infiltrate in the computer systems by usurpation.

1.7 Level 3: Operational Security Policies

The third level of intrusion detection consists on the definition of an operational security policy by a mechanism that correlate information from the physical access control list to the company and information from the logical access control list to the users' hosts. That means to deny access network to users who aren't really operational (i.e. who are absent or definatly dismissed) within the company at that time. This control will stop identity usurpation from the inside or from outside to the internal computer network.

These levels of our intrusion detection system permit to detect automatically violations security policies.

The analysis of the behavior of the computer network in this approach will permit to know the abnormal traffic from a host as:

- Connection attempts to the server network by user who isn't present in the enterprise.
- Connection attempts on a machine or non authorized resource by internal or external users.
- Detect attempting access to computer network or to some resources by non-authorized users.

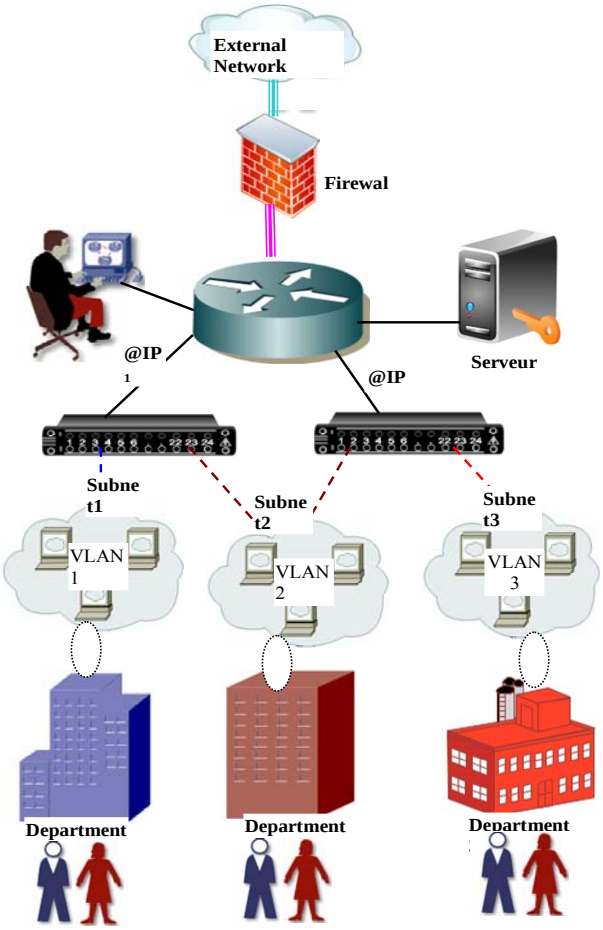

FIGURE -3: ARCHITECTURE NETWORK WITH VLAN

1.8 General architecture of the proposed system

Architecture

The figure 4 summarizes the important steps of our approach base on a three level security policy system. We need to gather events logs from the three different levels, then we can aggregate them, filter out the chronic alerts and finally we can correlate our data in order to reduce its volume for easy analysis and optimization of processing time looking for some intrusions.

In the case of an intrusion from the level L2 or L3, the administrator can piece data together in order to find out

how events have exactly happened. This method is called "Event Reconstruction" and it is really useful for administrators, because they can, thus:

- Have a better understanding of there system network needs.
- Identify system weaknesses and perform the security policies.
- Prevent the abuse of these weaknesses by insiders and outsiders attackers.
- Update the knowledge base in level 1.
- Help us to solve the problem of positive and negative false, and reduce its number, and therefore reduce the number of alerts and accelerate the processing thereafter, because we can correlate data according to context there are.
- Improve continuously the performance of our system.

Diagram of the Proposed System

As shown in diagram of figure 5, when packet traffic arrives, it passes through the first level were the IDS is installed. If it is an intrusive packet and its scenario is included in the data-base's IDS, the packet will be rejected, if it isn't, it passes through the 2nd level where we check the type of service performed or requested by the user behind this machine, if he is authorized to use the requested service or not. If he does not have rights to access the requested services and / or resources, the request will be rejected and the network administrator will be notified by an alert to start the diagnostics, if the packet is safe, it goes through the 3rd level. In this level, we check if that user is present within the company or not. If yes, the user will have full access to services and/or to requested resources. If he is absent, and not allowed to remotely access, the packet will be rejected and the network administrator will be notified by an alert to start the diagnostics. Intrusive packets analysis provides the network administrator to determine the origin of the attack using event reconstruction in order to highlight what have exactly happened and implemented counter-measures for this new attack and, thereafter, update the IDS database in the 1st level.

5. EXPERIMENTS

1.9 Introduction

Different experiments thereafter, as those of other many researchers in intrusion detection area [19, 20, 21, 22, 23], will be based on the KDD99 data sets [18]. It is considered a benchmark for intrusion detection evaluations, these data sets are the result of a transformation of raw tcpdump traffic into connection records originated from MIT's Lincoln Lab, developed by DARPA,

In the 1998 DARPA intrusion detection evaluation program, Lincoln Labs set up an environment to acquire raw tcpdump data for a network by simulating a typical U.S. Air Force LAN which was operated like a true environ-ment, but being peppered with multiple attacks. In fact, a framework for constructing features for intrusion detection systems is performed in [24]. Therefore we assume, in the following, that different features construction for intrusion detection (as a part of the data mining process in intrusion detection) are free of errors and we conduct our experiments for building classifiers over different KDD99 data sets.

The training data set contained about 5, 000, 000 connection records, and the training 10% data set consisted of 494, 021 records among which there were 97, 278 normal connections (i.e. 19.69%). Each connection record (about 100 bytes) consists of 41 different attributes that describe different features of the corresponding connec-tion, the value of the connection is labelled either as an attack with one specific attack type, or as normal. The 39 different attack types present in the 10% data sets and their corresponding occurrence numbers in the training and test data sets are given in table 1.

After analysis and correlation, each attack type can be grouped into one of the four following categories, as shown in table 1:

1- Probing: surveillance and other probing;
2- DoS: denial of service;
3- U2R: unauthorized access to local super-user (root) privileges;
4- R2L: unauthorized access from a remote machine.

The task was to predict the value of each connection (one of the five attack categories) for each connection record of the test data set containing 311, 029 connections.

It is important to note, from table 1, that:

1. The test data set has not the same probability distribution as the training data set;
2. The test data includes some specific attack types that are not present in the training data.

There are 22 different attacks types out of 39 present in the training data set.

1.10 Ranking and selection features

Ranking and selection features, therefore, are an important issue in intrusion detection, because we need to know, from the whole features, which are truly useful and which may be useless? Thus, the elimination of useless features (or audit trail reduction) enhances the accuracy of detection while speeding up the computation then improving the performance.

We performed experiments to rank the importance of input features for each of the five classes (normal, probe, DOS, U2R, and R2L) of patterns in the DARPA data set. It is shown that using only the important features for classifica-

tion gives well accuracies and, in certain cases, reduces the training time and testing time of the classifier.
The accuracy of each experiment is based on the percentage of successful prediction (PSP) on the test data set:

$$PSP = \frac{\text{Number of SIC}}{\text{Nber of inst. in the test set}} \quad (11)$$

(SIC= Successful instance classification)

Table 1: The different attack types and their corresponding occurrence

Number occurrence in data sets.	Training data	Testing data
Categories: Probing	**4107**	**4176**
ipsweep	1247	306
mscan	0	1053
nmap	231	84
portsweep	1040	364
saint	0	736
satan	1589	1633
Categories: DoS	**391458**	**229853**
apache2	0	794
back	2203	1098
land	21	9
mailbomb	0	5000
neptune	107201	58001
pod	264	87
processtable	0	759
smurf	280790	164091
teardrop	979	12
udpstorm	0	2
Categories: R2L	**1126**	**16189**
ftp write	8	3
guess passwd	53	4367
imap	12	1
multihop	7	18
named	0	17
phf	4	2
sendmail	0	17
snmpgetattack	0	7741
snmpguess	0	2406
spy	2	0
warezclient	1020	0
warezmaster	20	1602
worm	0	2
xlock	0	9
xsnoop	0	4
Categories: U2R	**52**	**228**
buffer overflow	30	22
httptunnel	0	158
loadmodule	9	2
perl	3	2
ps	0	16
rootkit	10	13
sqlattack	0	2
xterm	0	13

5.2.1 Results ranking and selection features using PCA approach
In this section, we present different results and experiments, ranking and selection features, given in [5] (tables 2, 3, 4 and 5) obtained when directly applying the methods discussed in section 3 on the KDD 99 cup data sets using PCA with a combination with two methods, namely the nearest neighbour (NN-rule) and decision trees (C4.5 Algorithm). For the combination with the PCA all data set are projected onto the new space generated by the few number PCA's principal axes. The two supervised algorithms are then applied to these projected data in the new reduced PCA's space.

* Nearest neighbour with/without PCA

Table 2: Confusion matrix obtained with the NN algorithm on 125 coordinates

Predicted / Actual	% Normal	% Probing	% DoS	% U2R	% R2L
Normal	**99.50**	0.26	0.24	0.00	0.00
Probing	17.21	**72.01**	10.28	0.00	0.50
DoS	2.87	0.12	**97.01**	0.00	0.00
U2R	39.96	18.80	32.01	**6.60**	2.63
R2L	96.12	2.65	0.00	0.02	**1.21**
PSP=92,05%					

Table 3: Confusion matrix obtained with the NN on 4 coordinates after performing PCA.

Predicted / Actual	% Normal	% Probing	% DoS	% U2R	% R2L
Normal	**99,50**	0.27	0.23	0.00	0.00
Probing	13.87	**74.40**	11.37	0.00	0.36
DoS	2.68	0.18	**97.14**	0.00	0.00
U2R	35.96	14.47	39.03	**7.91**	2.63
R2L	97.49	1.71	0.00	0.00	**0.80**
PSP=92,22%					

* Decision trees with/without PCA

Table 4: Confusion matrix relative to five classes using the C4.5 algorithm.

Predicted / Actual	% Normal	% Probing	% DoS	% U2R	% R2L
Normal	**99,42**	0.39	0.15	0.00	0.03
Probing	15.75	**78.80**	5.45	0.00	0.00
DoS	2.58	0.46	**96.96**	0.00	0.00
U2R	56.58	28.51	0.88	**5.26**	8.77
R2L	94.63	0.07	0.00	0.03	**5.27**
PSP=92,35%					

Table 5: Confusion matrix relative to five classes using the C4.5 algorithm after data set projection onto two principal component axes.

Predicted / Actual	% Normal	% Probing	% DoS	% U2R	% R2L
Normal	**98.99**	0.84	0.12	0.00	0.04

Probing	30.20	**66.30**	3.50	0.00	0.00
DoS	2.42	0.33	**97.25**	0.00	0.00
U2R	91.23	0.00	0.00	**8.33**	0.44
R2L	97.69	0.00	0.01	0.00	**2.30**
PSP=92,16%					

According to Table 3 and 5, in all experiments, even the computing time is clearly performed but the PSP of the two last classes R2L and U2R still weak, they are not well detected.

5.2.2 Results ranking and selection features using our three level security policies system

Table 6: Results of features selection

Test mode	Number of IAS	Correctly Classified Instances	Number of Selected attributes	% of Attribute Reduction
B- Attributes selection with 23 Class and 41 attributes				
split 66% train	41	93,5%	14	66
10-fold cross-validation	41	100%	14	66
C- Attributes selection with 05 Class and 41 attributes				
split 66% train	41	77,2%	6	85
10-fold cross-validation	41	100%	6	85
D- Attributes selection with 04 Class and 41 attributes				
split 66% train	41	77,3%	6	85
10-fold cross-validation	41	100%	6	85

Table 7: Results of features ranking

Test mode	Number of IAS	Time taken to build model (s)	Correctly Classified Instances	% of AR	% of TR
A- Classifier model with 23 Class and 41 attributes					
split 66% train	41	18,16	93,94%	-	-
10-fold cross-validation	41	16,72	93,68%	-	-
B- Classifier model with 23 Class and 41 attributes					
split 66% train	14	16,75	93,65%	66	8
10-fold cross-validation	14	15,61	93,28%	66	7
C- Classifier model with 05 Class and 41 attributes					
split 66% train	6	2,06	98,29%	85	88
10-fold cross-validation	6	2,31	98,78%	85	85
D- Classifier model with 04 Class and 41 attributes					
split 66% train	6	1,92	97,59%	85	89
10-fold cross-validation	6	1,91	97,84%	85	88

Note: (% AR= % of attributes reduction
% TR= % of time reduction
Nbre of IAS=Nber of input attributes selected)

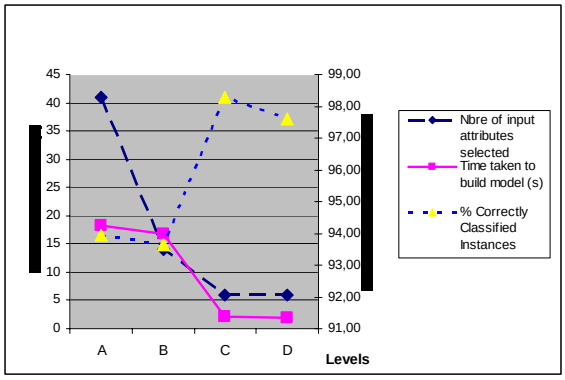

Figure -6: Performance of our approach using split 66% train algorithm

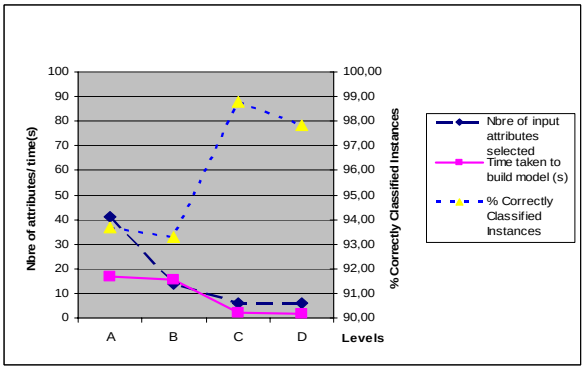

Figure -7: Performance of our approach using 10-fold cross-validation algorithm

Table 8: Confusion Matrix relative using our three levels security policies system

Predicted / Actual	% Normal	% DoS	% Probing	% Absent
normal	**99,11**	0,89	0,00	0,00
dos	0,16	**99,76**	0,08	0,00
probe	4,41	1,76	**92,65**	1,18
Abs	43,23	0,00	0,00	**56,77**
PSP=97,86%				

In this section, we present different results and experiments, ranking and selection features, shown in tables 6 and 7 obtained when directly applying the methods discussed in section 2 and 4 on the KDD 99 cup data sets using support vector machines (SVMs) algorithm, offered in data mining tool weka 3.5.7 freeware [25],

To simplify data set, our experiments based on a sample of data record attack existing in training and test data sets.
While ranking and selection features to create our model, we will use two SVM's methods: split 66% train, which seems to be less costly in time, and 10-fold cross-validation in order to compare and have good results.
The selection features will be monitored among the 41 variables (table 6- steps B, C and D), but for classification we will use only those classified as important features (table 7- steps A, B, C and D).
The third level of intrusion detection consists on definition of an operational security policy system, i.e. deny access network to users who aren't really operational within the company (i.e. who are absent or definatly dismissed). In general, an intruder who want to steal information from internal computer network, passe by a remote access to hosts whose users are absent. Thus, they use U2R and R2L attacks. Therefore, to simulate this level, we will merge the two classes attacks namely U2R and R2L in one class that we call Abs. Thus, for our approach experiments', we have only four classes (normal, probe, DOS, and Abs) instead of five.

1.11 Experimental Analysis and comparison

Our results are presented in tables 6, 7 and 8 and figures 6 and 7. If we compare them with those shown in tables 2, 3, 4 and 5 found in [5], we see clearly that with our approach we have found good performance enhancement results then found in [5] (tables 3 and 5), and values of PSP are clearly performed.
According to Table 3 and 5, even applying PCA in the two experiments, the two last classes R2L and U2R are not well detected. The PSP isn't well performed, the maximum for U2R class passes from 4.39% to 8.33%, but it decreases from 5.27% to 2.30% for R2L.
While using our three level security policies system approach, the detection rate of all classes is increased, especially for the classes U2R and R2L, the rate of our new class Abs (which is a fusion of the two classes U2R (7,02%) and R2L (2,85%)) become 56,77%. Furthermore, as shown in table 8, the false negative rate of this class decreases considerably from 21, 93% to 2,12% and the PSP is increased from 92,30% to 97, 86%.

6. Conclusion

The first part of this paper provided an overview of support vector machines and principal component analysis methods. Also, we have seen that they can perform network security mechanisms through mathematics formulations.
The second part of this paper described the different steps of our proposed approach based on a three level security policies system.
The third part of this paper described the different experimental analysis and comparison.
In [5] using PCA with a combination with two methods, namely the nearest neighbour and decision trees provide a slight difference between the use of decision trees on rough data and their combination with PCA on the new feature space. The two last classes R2L and U2R are not well detected, a slight enhancement for U2R class, in some cons, R2L detection rate decreases.
While using our three level security policies system approach, the detection rate is well performed for all classes, especially for the U2R and R2L class. The rate of class our new class Abs (which is a fusion of the two classes:U2R and R2L) become 56,77%. Furthermore, the false negative rate of this class decreases considerably and the PSP is increased from 92,30% to 97, 86%.
This new approach, aiming the protection of the network from the inside and the outside, will bring a very important improvement intrusion detection area. It can help network administrators to implement proactive response for the detected new attacks. Also, using intelligent agents to reduce the administrator daily tasks and choose the adequate answer to likely attacks.

References

[1] A. Kartit, M. El Marraki, A. Radi and B. Regragui "On the security of Firewall Policy Deployment", Journal of Theoretical and Applied Information Technology, ISSN: 1817-3195, Volume 22, n°2, pages 84 – 92, 2010.
[2] A. Radi, A. Kartit, B. Regragui, M. El Marraki and A. Ramrami "An Enhanced a three levels security Policy", Journal of Theoretical and Applied Information Technology, ISSN: 1817-3195, Volume 23, n°1, pages 39 – 50, 2011.
[3] Burge, C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data mining and knowledge discovery journal. 2(2) (1998) 121-167.
[4] V.N. Vapnik, "The Nature of Statistical Learning Theory". Springer edition (1995).
[5] Y. Bouzida "Principal Component Analysis for Intrusion Detection and Supervised Learning for New Attack Detection" thesis doctoral, ENST in Bretagne, March 24th 2006.
[6] JingTao Yao, Songlun Zhao and Lisa Fan "An Enhanced Support Vector Machine Model for Intrusion Detection ", vol. 4062, pp. 538-543 ISBN 3-540-36297-5 ; Springer, Berlin, ALLEMAGNE (2006) (Monographie)
[7] Qiao, Y., Xin, X.W., Bin, Y., Ge, S. "Anomaly Intrusion Detection Method Based on HMM". Electronics Letters. 38(13) (2002) 663-664.
[8] Lee, W., Stolfo, S.J. "Data Mining Approaches for Intrusion Detection". The 7th USENIX Security Symposium.(1998)79-94.
[9] M. Mohajerani, A. Moeini, M. Kianie, "NFIDS: A Neuro-fuzzy Intrusion Detection System ". Proc. of the 10th IEEE Int. Conf. on Electronics, Circuits and Systems. (2003) 348-351.
[10] W.D. Wang, S. Bridges. "Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association

Rules". Proc. of the 7th Int. Conf. on Fuzzy Theory & Technology. (2000) 131-134.
[11] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[12] C. CORTES and V. VAPNIK "Support-Vector Networks", AT&T Bell Labs. Hohndel, NJ 07733, USA. Springer edition 1995.
[13] B. Schölkopf, A. J. Smola, "Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond", 2002, MIT Press.
[14] I.T. Jolliffe Principal Components Analysis, Series: Springer series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus.
[15] Revue Mag Securs Novembre 2005.
[16] "Risks associated to the Internet uses" - http://www.filhot.com/vaucelles/, article seen the 20/07/04
[17] A. Radi., B. Regragui and A. Ramrami, "Establishment of an Intrusion Prevention", University Mohammed V, Rabat, Morocco- VSST'2007 – Marrakech, Morocco.
[18] http://kdd.ics.uci.edu/databases/kddcup99/task.html.
[19] Pfahringer, B. (2000). Winning the KDD Classification Cup: Bagged Boosting. SIGKDD Explorations. ACM SIGKDD, 1, 65–66.
[20] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2003). A Geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Applications of Data Mining in Computer Security, Kluwer Publishers.
[21] Fan, W., Miller, M., Stolfo, S. J., Lee, W., & Chan, P. K. (2004). Using artificial anomalies to detect unknown and known network intrusions. Knowledge and Information Systems, 6(5), 507–527.
[22] Shyu, M. L., Chen, S. C., Sarinnapakorn, K., & Chang, L. W. (2003). A Novel Anomaly Detection Scheme Based on Principal Component Classifier. In Proceedings of ICDM Foundation and New Direction of Data Mining workshop (pp. 172–179).
[23] Hettich, S. & Bay, S. D. (1999). The UCI KDD Archive. Available at: http://kdd.ics.uci.edu/.
[24] W. Lee, & S. Stolfo. A Framework for Constructing Features and Models for Intrusion Detection Systems. ACM Transactions on Information and System Security, 3(4).
[25] http://weka.sourceforge.net/wekadoc/index.php/en:Download_(3.5.7).

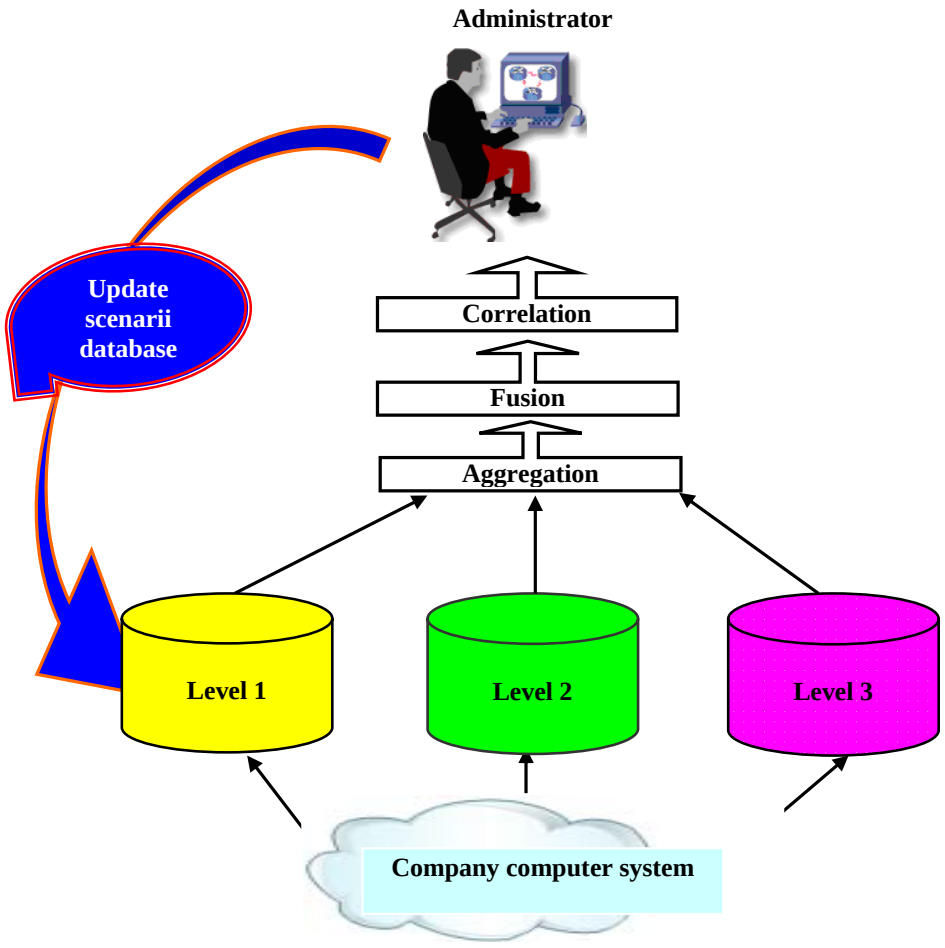

FIGURE -4: ARCHITECTURE OF THREE LEVELS SECURITY POLICIES

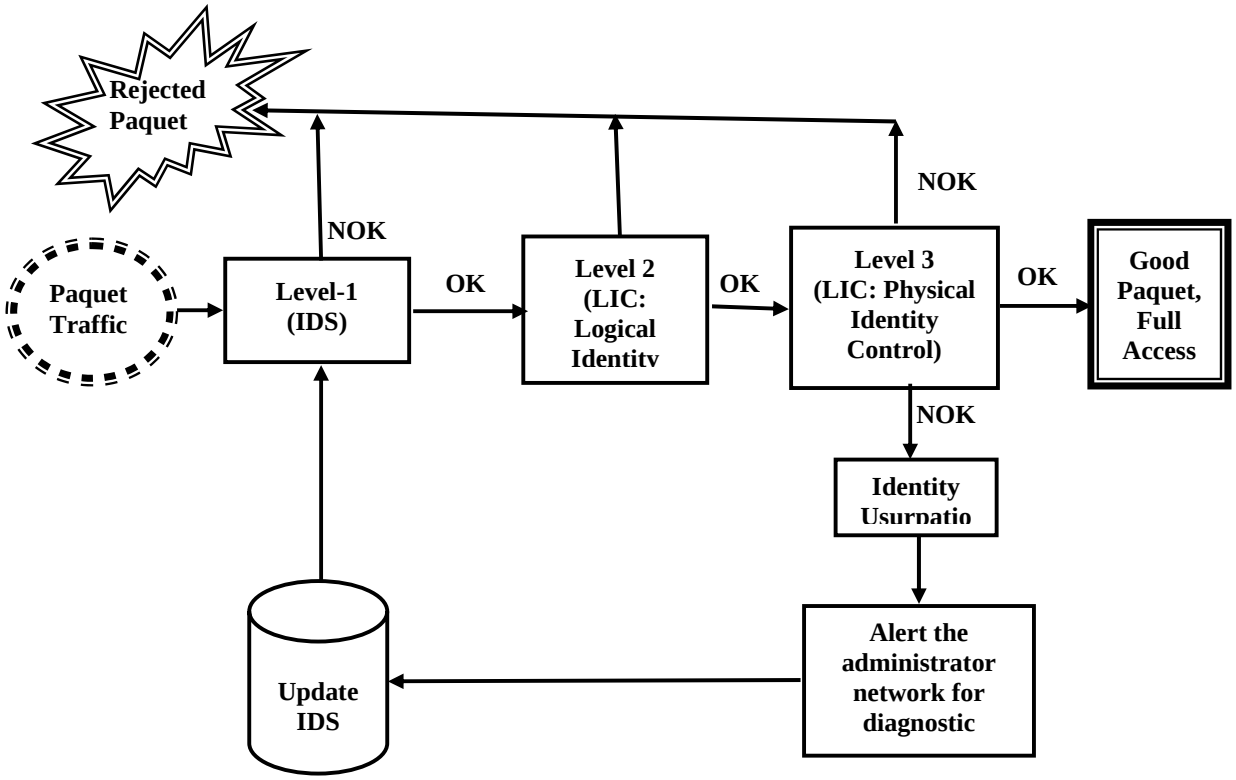

FIGURE-5: DIAGRAM OF THREE LEVELS SECURITY POLICIES ALGORITHM