Journal of Theoretical and Applied Information Technology
15th January 2012. Vol. 35 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# ON THE THREE LEVELS SECURITY POLICY COMPARISON BETWEEN SVM AND DECISION TREES

[1]A. RADI, [2]A. KARTIT, [3] D. ABOUTAJDINE, [4]B. REGRAGUI, [5]M. EL MARRAKI, [6]A.RAMRAMI,

[1, 2,3, 5, 6]Department of Physic, Faculty of Sciences, University Mohammed V, Rabat Morocco
[5]Department of Computer Sciences, Faculty of Sciences, University Mohammed V, Rabat Morocco
E-mail: alikartit@gmail.com

**ABSTRACT**

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. Lot of mechanisms have been developed to assure the computer systems security. Conventional intrusions detection systems "IDS" have shown their insufficiencies and limits. To improve computer systems security approach, in our 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], However, manual analysis of the huge volume of data generated, audit data, is 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 shows that our 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, Feature Selection, False Negative, False Positive.*

## 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. Also, more computer systems become safer; more 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, faster and more accurate detection that may result.
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 KDD [11].

Journal of Theoretical and Applied Information Technology
15th January 2012. Vol. 35 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

## 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 which 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.

### 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 characteristics: 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) (<,> is the inner product of two vectors) where $(< w, x > +b)=0$ can be viewed as a separating 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:

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

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. DECISION TREE

### 3.1 Introduction

If the goal of an analysis is to predict the value of some variable, then supervised learning is the recommended approach. Decision trees is example of supervised learning method used, in statistics, data mining and machine learning, as a predictive model which maps observations about an item to conclusions about the item's target value.

Journal of Theoretical and Applied Information Technology
15th January 2012. Vol. 35 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Decision trees methods, tracing their origins back to the work of Hunt [14], Breiman et al [15] and Quinlan [16], construct trees whose leaves are labeled with the predicted classification.

### 3.2 Decision Trees methods

A decision tree is a tree that has three main components: nodes, branches, and leaves. Each node of a decision tree is some attribute, with the branches representing alternative values of that attribute connected to children nodes the set that initially contains all rules can be considered as the tree's root node while its children are the direct subsets created by partitioning the rule set according to the first attribute, each subset is associated with a node in the tree. When a node contains more than one rule, these rules are subsequently partitioned and the node is labeled with the attribute that has been used for this partitioning step. An arrow that leads from a node to its child is annotated with the value of the attribute that is specified by all the rules in this child node, then continue making such choices at each node until a leaf node is reached.

The decision tree is constructed during the learning phase; it is then used to predict the classes of new instances. Most of the decision trees algorithms use a top down strategy; i.e. from the root to the leaves. Two main processes are necessary to use the decision tree:

**a) Building process:** It consists in building the tree by using the labeled training data set. An attribute is selected for each node based on how it is more informative than others which is measured using Shannon entropy. The selection of the best attribute node is based on the Gain or Gain ratio (equations (8, 9)). This gain defines the expected reduction in entropy due to sorting on attribute.

**b) Classification process:** A decision tree is important not because it summarizes the training set, but because we hope it will classify correctly new cases. New instances are classified by traversing the tree, already built, from up to down based on their attribute values and the node values until one leaf is reached that corresponds to the class of the new instance.

### 3.3 Formalization

If there are $n$ equally probable possible messages, then the probability $p$ of each is $1/n$ and the information conveyed by a message is

$$-\log_2(p)=\log_2(n) \quad (4)$$

In general, if we are given $P=(p_1, p_2, .., p_n)$ a probability distribution then the Information conveyed by this distribution, also called the Entropy of $p$, is: $\text{Entropy}(P)=-\sum_{i=1}^{n} p_i * \log_2(p_i)$ **(5)**

**Note that the more uniform is the probability distribution, the greater is its information.**

**If a set $T$ of records is partitioned into disjoint exhaustive classes $C_1, C_2, .., C_k$ on the basis of the value of the categorical attribute, then the information needed to identify the class of an element of T is $\text{Info}(T)=\text{Entropy}(P)$, where $p$ is the probability distribution of the partition $(C_1, C_2, .., C_k)$: $P=(C_1/T, C_2/T, .., C_k/T)$**

**If we first partition $T$ on the basis of the value of a non-categorical attribute $X$ into sets $T_1, T_2, .., T_n$ then the information needed to identify the class of an element of $T$ becomes the weighted average of the information needed to identify the class of an element of $T_i$, i.e. the weighted average** of $\text{Info}(T_i)$:

$$\text{Info}(X, T) = \sum_{i=1}^{n} \frac{T_i}{T} * \text{Info}(T_i) \quad (6)$$

Consider the quantity Gain(X, T) defined as:

$$\text{Gain}(X,T)=\text{Info}(T)-\text{Info}(X,T) \quad (7)$$

$$\text{Gain}(X,T)=\text{Info}(T)-\sum_{i=1}^{n} \frac{T_i}{T} * \text{Info}(T_i) \quad (8)$$

This represents the difference between the information needed to identify an element of $T$ and the information needed to identify an element of $T$ after the value of attribute $X$ has been obtained, ie, the gain in information due to attribute $X$.

We can use this notion of gain to rank attributes and to build decision trees where at each node is located the attribute with greatest gain among the attributes not yet considered in the path from the root.

The intent of this ordering are two fold:

- To create small decision trees so that records can be identified after only a few questions.
- To match a hoped for minimization of the process represented by the records being considered

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

If we consider only $Gain(X,T)$ then an attribute with many values will be automatically selected. For example, if we have an attribute $X$ that has a many distinct value for each record, then $Info(X,T)$ is 0, thus $Gain(X,T)$ is maximal. To compensate for this Quinlan [16] suggests using the following ratio (named GainRatio) instead of Gain defines as:

$$GainRatio(X,T) = \frac{Gain(X,T)}{SplitInfo(X,T)} \quad (9)$$

where SplitInfo(X,T) is the information due to the split of T on the basis of the value of the categorical attribute $X$. Thus $SplitInfo(X,T)$ is defined as:

$$SplitInfo(X,T) = -\sum_{i=1}^{n} \frac{T_i}{T} * \log_2(\frac{T_i}{T}) \quad (10)$$

where $T_i$ is a subset of $T$ induced by the value of $X$.

In the case of a discrete valued attribute, this strategy tests all possible values of the attribute under consideration. However, in the case of continuous-valued attributes a transformation technique, introduced in [16], consists in defining new discrete-valued attributes that partition the continuous attribute into a discrete set of intervals. The algorithm dynamically creates a new boolean attribute $A_t$ that is true if $A \leq t$ and false otherwise. A threshold $t$ is selected, based on the information gain (equation (8 or 9)), if it produces the greatest information gain.

### 3.4 The ID3 and C4.5 Algorithms

ID3 and C4.5 are algorithms introduced by Quinlan in [16], they are used to build decision trees, We are given a set of records, each record has the same structure, consisting of a number of attribute/value pairs. One of these attributes represents the category of the record. The problem is to determine a decision tree that on the basis of answers to questions about the non-category attributes predicts correctly the value of the category attribute, as explained previously (see the ID3 algorithm attached thereafter).

C4.5 introduces a number of improvements of the original ID3 algorithm:

- In building a decision tree we can deal with training sets that have records with unknown attribute values by evaluating the gain, or the gain ratio, for an attribute by considering only the records where that attribute is defined.
- In using a decision tree, we can classify records that have unknown attribute values by estimating the probability of the various possible results.

The decision tree built using the training set deals correctly with most of the records in the training set. In fact, in order to do so, it may become quite complex, with long and very uneven paths. Therefore, pruning of the decision tree is done by replacing a whole subtree by a leaf node. The replacement takes place if a decision rule establishes that the expected error rate in the subtree is greater than in the single leaf.

### 3.5 The enhanced C4.5 Algorithm

Decision trees C4.5 algorithm written by Quinlan [16] presents a drawback towards the set of instances that are not covered by any of the rules generated from the decision tree. For those instances he proposes a default class which is defined as the one with most items not covered by any rule, in the case of conflict, the most frequent class prevails. This solution could be interesting if we know that all classes are known a priori and there is no new class.

Since we are interested in detecting novel attacks this classification kind would not be able to detect new attacks that normally are not covered by any rule from the tree built during the learning step. To overcome this problem, instances that do not have a corresponding class in the training data set are assigned to a new class. Therefore, if any new instance does not match any of the rules generated by the decision tree then this instance is classified as a new class instead of assigning it to a default class. This algorithm is called enhanced C4.5 as presented in [5].

## 4. A THREE LEVELS SECURITY POLICY SYSTEM

### 4.1 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 [17]:

- 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).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

➢ 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 [18].
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.

### 4.2 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 [19]. 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.

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" and the use of ACL "Access Control Lists" (figure-1 ). 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.

### 4.3 Level 2: Functional Security Policies

The second level of detection consists to define functional security policies, which means policies according to tasks assigned to users within the enterprise by the segmentation of the network into VLAN "Virtual Local Area Network" 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 to users who belonged to the same VLAN (all other actions are forbidden) or inversely, other users won't have access to this VLAN. Also, VLAN will allows - in worse case - if an intruder has succeeded taking control on a host (few number of machines), the attack will be restricted to a small subnet 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.

#### 4.3.1 Security with Virtual LAN

A VLAN is a broadcast domain created by one or more switches. The network design in figure-1 shows three VLAN (named: VLAN1, VLAN2 and VLAN3) created by separate switches. The router routes traffic between VLANs using Layer 3 routing. Switches forwards frames to the router interfaces if it is a broadcast frame or if the destination is the router's MAC addresses.
The headline of frame is encapsulated or modified to include an identifier (ID) of VLAN before forwarding the frame on the link between switches (VLAN1-ID=100, VLAN2-ID=200 and VLAN3-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ID=300), and the original frame is re-established before forwarding it to the destination host.

#### 4.3.2 Security with Access Control List

Access Control Lists "ACLs" are lists of instructions that you apply to router's interfaces which indicate what kinds of packets to accept or to deny. The acceptance or dropping based on specified conditions in the ACL which could be source address, destination address, port number, protocol, or other information

There are many reasons to create ACLs. For example:

- Limit network traffic and increase network performance.
- Provide traffic flow control.
- Provide a basic level of security for network access.

The figure-4 shows mecanisme of access control list. Extended ACLs are more often used than standard ACLs because they provide a greater range of control. We can control, not only, the ip addresses of origin and destination of the packet, but also, We can verify the protocols and port numbers. This gives greater flexibility to describe what parameter checks the ACL.

ACL emplacement is very important. If the ACL are correctly placed, not only the traffic can be filtered, but the whole network becomes more effective. Accordding to [1], we can change ACL rules in a correct and safe way.

### 4.4 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.

### 4.5 General architecture of the proposed system

#### 4.5.1 Architecture

The figure 2 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.

#### 4.5.2 Diagram of the Proposed System

As shown in diagram of figure 3, 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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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 determining 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

### 5.1 Introduction

Different experiments thereafter, as those of other many researchers in intrusion detection area [21, 22, 23, 24, 25], will be based on the KDD99 data sets [20]. 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 environment, but being peppered with multiple attacks. In fact, a framework for constructing features for intrusion detection systems is performed in [26]. 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.

| 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 |

Table 1: The different attack types and their corresponding occurrence

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;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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.

### 5.2 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 classification 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, defines as:

$$PSP = \frac{\text{Number of Successful Instance classification}}{\text{Number of instances in the test set}} \quad (11)$$

#### 5.2.1 Results ranking and selection features using the enhanced C4.5 Algorithm

In this section, we present different results and experiments, ranking and selection features, given in [5] (tables 2 and 3) obtained when applying the methods discussed in section 3 on the KDD 99 cup data sets using decision trees Algorithms i.e. standard and enhanced C4.5 Algorithm proposed in [5].
From Table 2, the two classes R2L and U2R are badly predicted. On the other hand, many probing and DoS instances are misclassified within the normal category which increases the false negative rate. Most misclassified instances are predicted as normal, that assigns a default class as normal, as explained in Section 3.5. Hence, if a new instance is presented, even abnormal instances in the learning step, it is automatically classified as the default class normal.

| Predicted / Actual | % Normal | % Probing | % DoS | % U2R | % R2L |
|---|---|---|---|---|---|
| Normal | **99,47** | 0,40 | 0,12 | 0,01 | 0,00 |
| Probing | 18,24 | **72,73** | 2,45 | 0,00 | 6,58 |
| DoS | 2,62 | 0,06 | **97,14** | 0,00 | 0,18 |
| U2R | 82,89 | 4,39 | 0,44 | **7,02** | 5,26 |
| R2L | 81,60 | 14,85 | 0,00 | 0,70 | **2,85** |
| **PSP=92,30%** | | | | | |

Table 2: Confusion Matrix relative using the rules generated by the standard C4.5rules algorithm.

| Predicted / Actual | % Normal | % Probing | % DoS | % U2R | % R2L | % New |
|---|---|---|---|---|---|---|
| Normal | **99,43** | 0,40 | 0,12 | 0,01 | 0,00 | 0,04 |
| Probing | 8,19 | **72,73** | 2,45 | 0,00 | 6,58 | 10,06 |
| DoS | 2,26 | 0,06 | **97,14** | 0,00 | 0,18 | 0,36 |
| U2R | 21,93 | 4,39 | 0,44 | **7,02** | 5,26 | 60,96 |
| R2L | 79,41 | 14,85 | 0,00 | 0,70 | **2,85** | **2,20** |
| **PSP=92,87%** | | | | | | |

Table 3: Confusion Matrix relative using the rules generated from the enhanced C4.5 rules algorithm.

By using the enhanced C4.5 algorithm, the detection rate of the U2R class is increased by 60.96% which decreases the false negative rate of this class from 82.89% to 21, 93%. The detection rate of the Probing class is also enhanced by 10, 06%. We note that the different ratios presented in table 3 are the same as those in table 2 except the normal column where the corresponding ratios have decreased. This is expected since the normal class is the default class when using the standard C4.5 algorithm. The false positive rate is increased by a small ratio of 0.04%. However, the false negative rate of the R2L class remains stable. In addition, even if we count the detection ratio of the new instances that are classified as new attacks, the PSP (92.30% + 0.57% = 92.87%) ratio remains far from 100%.

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

In this section, we present different results and experiments, ranking and selection features, shown in tables 4 and 5 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 [27], Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java developed at the University of Waikato in New Zealand.
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 4- steps B, C and D), but for classification we will use only those classified as important **features (table 5- steps A, B, C and D).**

Journal of Theoretical and Applied Information Technology
15th January 2012. Vol. 35 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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 informa-tion 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 in steps D) instead of five (in steps D).

| Test mode | Number of IAS | Correctly Classified Instances | Number of Selected attributes | % of AR |
|---|---|---|---|---|
| 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 4: Results of features selection

| Test mode | Nbre 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 |

Table 5: Results of features ranking

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

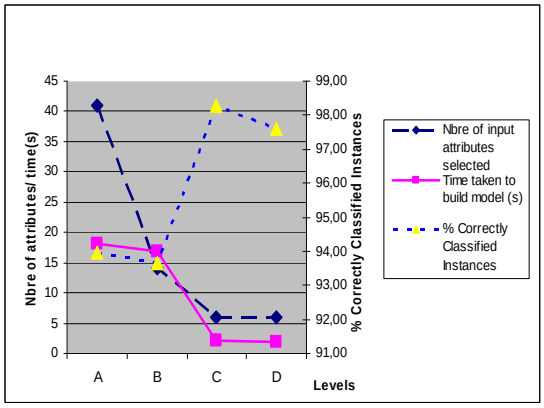


Figure -5: Performance of our approach using split 66% train algorithm

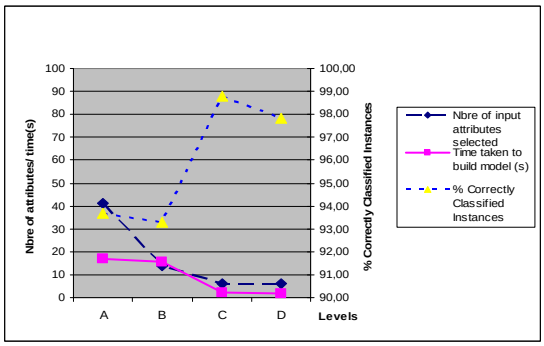


Figure -6: Performance of our approach using 10-fold cross-validation algorithm

| Predicted / Actual | % Normal | % DoS | % Probing | % Abs |
|---|---|---|---|---|
| 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%** | | | | |

Table 6: Confusion Matrix relative using our three levels security policies system

### 5.3 Experimental Analysis and comparison

Our results are presented in tables 4, 5 and 6 and figures 5 and 6. If we compare them with those shown in tables 2 and 3 found in [5], we see clearly that with our approach we have found better performance enhancement results than found in [5], and values of PSP are clearly performed.

According to Table 2 and 3, even applying decision trees algorithms in the two experiments, the two last classes R2L and U2R are not well detected. The PSP isn't well performed.

While using our three level security policies system approach, 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%))

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

become 56,77%. Furthermore, as shown in table 6, 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 decision trees algorithms. 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:

- Level 1: consist to apply an external protection;
- Level 2: consist to define functional security policies;
- Level 3: consist to define operational security policies.

The third part of this paper described the different experimental analysis and comparison.

In [5] using decision trees algorithms provide a slight difference between the use of standard and enhanced C4.5 Algorithm. The two last classes R2L and U2R are not well detected, in some cons, normal 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.

## REFRENCES:

[1] 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] E. B Hunt,. (1962). Concept Learning: An Information Processing Problem. Wiley.

[15] L. Breiman, J. H.Friedman, R. A.Olshen, & C. J. Stone, (1984). Classification and Regression Trees.

[16] J. R. Quinlan, (1986). Induction of decision trees. Machine Learning, 1, 1–106.

[17] Revue Mag Securs Novembre 2005.

[18] "Risks associated to the Internet uses" - http://www.filhot.com/vaucelles/, article seen the 20/07/04

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[19] A. Radi., B. Regragui and A. Ramrami, "Establishment of an Intrusion Prevention", University Mohammed V, Rabat, Morocco-VSST'2007 – Marrakech, Morocco.

[20] http://kdd.ics.uci.edu/databases/kddcup99/task.html.

[21] Pfahringer, B. (2000). Winning the KDD Classification Cup: Bagged Boosting. SIGKDD Explorations. ACM SIGKDD, 1, 65–66.

[22] 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.

[23] 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.

[24] 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).

[25] Hettich, S. & Bay, S. D. (1999). The UCI KDD Archive. Available at: http://kdd.ics.uci.edu/.

[26] W. Lee, & S. Stolfo. A Framework for Constructing Features and Models for Intrusion Detection Systems. ACM Transactions on Information and System Security, 3(4).

[27] http://weka.sourceforge.net/wekadoc/index.php/en:Download_(3.5.7).

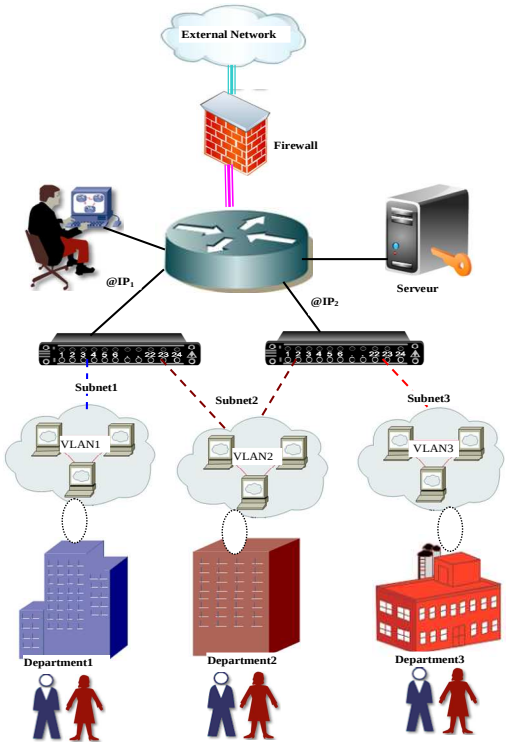


**Figure -1 Architecture network with VLAN**

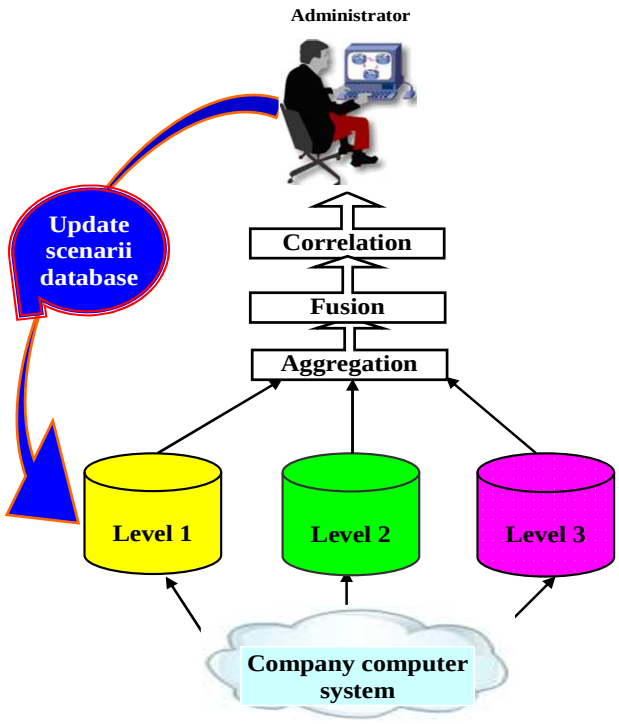


**Figure -2: Architecture of three levels security policies**

Journal of Theoretical and Applied Information Technology
15th January 2012. Vol. 35 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

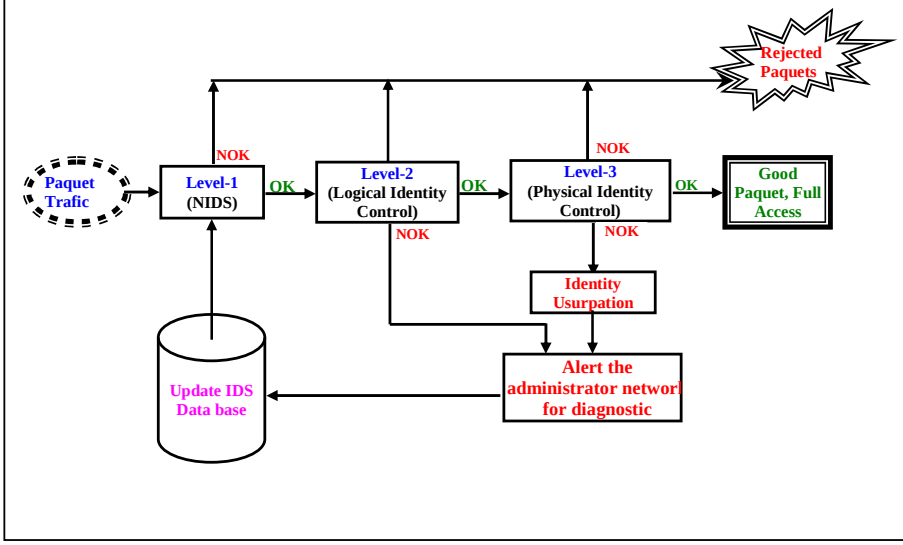


Figure-3: Diagram of three levels security policies algorithm

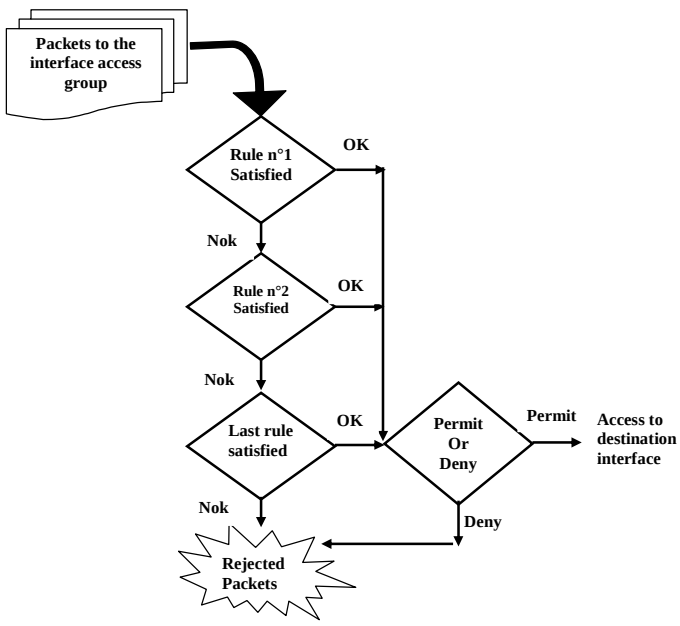


FIGURE -4: FUNCTIONAL ACL DIAGRAM

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

**The ID3 algorithm**

The ID3 algorithm is used to build a decision tree, given a set of non-categorical attributes $R_1, R_2, .., R_n$ the categorical attribute C, and a training set T of records.

function ID3 (R: a set of non-categorical attributes,
C: the categorical attribute,
T: a training set) returns a decision tree;
begin
If T is empty, return a single node with value Failure;
If T consists of records all with the same value for the categorical attribute,
return a single node with that value;
If R is empty, then return a single node with as value the most frequent of the values of the categorical attribute that are found in records of T; [note that then there will be errors, that is, records that will be improperly classified];
Let X be the attribute with largest Gain(X,T) among attributes in R;
Let $\{X_j | j=1,2, .., m\}$ be the values of attribute X;
Let $\{T_j | j=1,2, .., m\}$ be the subsets of T consisting respectively of records with value $X_j$ for attribute X;
Return a tree with root labeled X and arcs labeled $X_1, X_2, .., X_m$ going respectively to the trees ID3(R-{X}, C, T1), ID3(R-{X}, C, T2), .., ID3(R-{X}, C, Tm);
end ID3;
---