AN EFFECTIVE INTRUSION DETECTION MODEL BASED ON SVM WITH FEATURE SELECTION AND PARAMETERS OPTIMIZATION

[1]EL MOSTAPHA CHAKIR, [2,3]MOHAMED MOUGHIT, [4]YOUNESS IDRISSI KHAMLICHI

[1]IR2M Laboratory, FST, Univ Hassan 1, Settat, Morocco

[2]IR2M Laboratory, ENSA, Univ Hassan 1, Settat, Morocco

[3]EEA&TI Laboratory, FST, Univ Hassan 2, Mohammedia, Morocco

[4]LERSI Laboratory, ENSA, Univ Sidi Mohamed Ben Abdellah, FES, Morocco

E-mail: [1]e.chakir@uhp.ac.ma, [2,3] mohamed.moughit@uhp.ac.ma, [4]youness.khamlichi@usmba.ac.ma

ABSTRACT

With the growth of the internet, network attacks have increased severely in a substantial number in the last few years. Therefore, Intrusion Detection Systems (IDSs) have become a necessary addition to the information security of most organizations. An IDS monitors a network or a single host looking for suspicious activity and reports them. Many intrusion detection types of research have focused on the feature selection because some characteristics are irrelevant or redundant which result in a lengthy detection process and degrades the performance of IDS. For this purpose, we have used in this work an algorithm based on Information Gain technique. This algorithm selects an optimal number of features from NSL-KDD Dataset. In addition, we have combined the feature selection with a machine learning technique named Support Vector Machine (SVM) using Radial-basis kernel function (RBF) and a Particle Swarm Optimization algorithm to optimize the parameters of SVM for effective classification of the dataset. We have also compared the proposed method and other methods. Tests on the NSL-KDD Dataset have proved that our proposed method can reduce the number of features and obtain good results in terms of accuracy, attack detection rate and false positives rate, even for unknown attacks.

Keywords: *Intrusion Detection System, NSL-KDD, Feature selection, PSO, SVM, Information Gain.*

1. INTRODUCTION

The explosive increase in the number of networked computers in the world and the wide spread use of the Internet have led to an increase in the number and severity of intrusions, not only by external attackers but also by internal sources. An intrusion can be defined as any action that aims at compromising the goals of security which are: Integrity, confidentiality, and availability. As a result, intrusion detection systems (IDS) have become the mainstream of security infrastructure. IDSs as originally introduced by Anderson [1] and later formalized by Denning [2].

The main objective of IDS is to monitor a single or a network of computers looking for a suspicious activity and reports its results to an administrator. A large IDS can be placed on a backbone network to monitor all traffic, or smaller systems can be set up to monitor traffic for a particular server, switch, gateway, or router. However, a big weakness with IDS systems is that they generate a large number of false positives, which are alerts that mistakenly indicate security issues and draw attention from the intrusion detection analyst [3]. Most techniques used in today’s IDS are not able to deal with the dynamic and complex nature of cyber-attacks on computer networks. Traditional intrusion detection and prevention techniques such as firewalls, access control mechanisms, and encryptions have several limitations in fully protecting networks and systems from increasingly sophisticated attacks like DDoS [4].

Using various techniques of feature selection and machine learning can result in higher True Positive Rate, lower False positives, and better Accuracy. In this work, we will investigate the feature selection and machine learning methods that have been proposed by researchers in the few past years, and we propose a new Intrusion Detection System model, which combines a feature selection algorithm using information Gain and a Particle Swarm Optimization algorithm to optimize the parameters of SVM classifier. There are many algorithms used with IDS to improve detection such as genetic algorithm with SVM, PCA with SVM, Bat Algorithm with SVM... The accuracy of IDS Classifier depends on these algorithms. So that's why we have opted for using PSO with SVM to improve IDS. To test the effectiveness of our proposed model, we will use NSL-KDD Dataset.

The rest of this paper is organized as follow: Section 2 introduces the background and related works. Section 3 illustrates the preliminaries of SVM and the influence of the parameters over the performance of the SVM classifier. Section 4 describes in details the proposed PSO algorithm and how it can be applied to optimize the SVM parameters. Section 5 portrays the proposed IDS model in details using Information Gain as a Feature Selection method, Section 6 shows experiments results and presents analyses. Finally, the paper is concluded with our future work in Section 7.

2. BACKGROUND AND RELATED WORKS

In recent times, the use of the Internet has become an important part in our daily life. Along with the rapid development and widespread use of the internet, many types of intrusions have extensively grown in the recent years. Many protection techniques have been used in order to protect the networks (firewalls, encrypting data, access control, Web application firewalls and so on). These techniques are not sufficient, as each of them has proven to be inefficient. Therefore, the use of intrusion detection systems as an additional defense tool is almost indispensable. An Intrusion Detection System (IDS) monitors the events from a network or a single host and decides whether these events are normal or not [4]. IDSs can be classified into two approaches: misuse detection (or signature-based detection) and anomaly detection [5]. The first approach examines both the network and system activity for the known attacks using signature matching algorithms included in the database. This approach is effective at detecting all attacks that are already known. The anomaly detection compares every instance to what normal is to the network and thus looks for an abnormal behavior of the monitored system. This approach is effective at detecting new types of attacks. Moreover, IDS can also be classified according to its source of analyzed information (host-based IDS and network-based IDS) [6].

When building an intrusion detection model, many challenges need to be considered, such as obtaining a high True Positive Rate (TPR), a lower False Positive Rate (FPR) and a high accuracy as well. Multiple techniques have been proposed in order to improve the performances of IDSs. Recently, researchers have proposed several machine learning approaches. Machine learning algorithms may offer a possible solution that could resolve most of the challenges such as: handling noisy data, detecting new types of attacks with a low false positives rate or managing a large amount of input data and offering real-time responses. Because our proposed detection model uses SVM, we will focus on the recent IDS approaches based on SVM. Support Vector Machines (SVM) is a supervised machine learning algorithm that has become a popular research method in intrusion detection [7], [8].

There are many IDS models based on feature selection and classification algorithms on SVM that are proposed for IDS. In [7] Wang et al. presented an IDS based on SVM combined with Particle Swarm Optimization. They used two different PSO algorithms: Standard PSO to seek optimal SVM parameters and Binary PSO to extract the best feature subset. Their model improved detection rate with a high accuracy.

In [8] Pu et al. used Ant Colony Algorithm to optimize SVM parameters. They tested their model on the KDD CUP99 Dataset and reported that the anomaly detection rate can reach a high accuracy.

In [9] Iftikhar et .al proposed a genetic algorithm to search the genetic principal components that offer a subset of features with optimal sensitivity and the highest discriminatory power. They used SVM for classification. The results show that the proposed model improves SVM performance in Intrusion Detection.

In [10] Zhou et al. proposed an approach based on a Culture Particle Swarm Optimization algorithm (CPSO). They used the algorithm to optimize the parameters of SVM classifier. They used the colony aptitude of particle swarm and the ability to conserve the evolving knowledge of the culture algorithm to construct the population space based on the particle swarm and the knowledge space.

In [11] Horng et al. proposed a SVM-based IDS model, which used hierarchical clustering algorithm and SVM classifier. It was able to minimize the training time and improve the performance of SVM classifier. They applied a simple feature selection procedure to eliminate irrelevant features from the training set. The results showed that the SVM model could classify attacks more accurately.

In [12] Gaspar et al. reviewed strategies that are used to improve the classification performance of SVMs in terms of accuracy and performed some experimentation to study the influence of features and hyper parameters in the optimization process, using kernel functions.

In [13] Kim et. Al tested the effectiveness of SVM Classifier in detecting masquerade activities. The results of their experiments showed that their model could detect attacks with a high accuracy. Thus, they demonstrated that SVM is an effective solution for masquerade detection.

In [14] Ma et al. proposed a new hybrid detection model based on Binary Particle Swarm Optimization (BPSO) and Support Vector Machine (SVM). Their model performs two tasks in one step: reducing features in the dataset and selecting the optimum parameters for SVM. They used KDDCUP99 to test their model. The results indicate that their approach is more accurate.

3. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a linear machine learning technique that can perform binary classification and regression estimation tasks [16]. It becomes popular as a new paradigm of classification and learning technique. SVM shows good generalization skill. Unlike the other classifier, SVM minimizes the expected error rather than minimizing the classification error. It does not suffer from the local minimum and it can handle noisy datasets.

Given a training sample of instances (x_i,y_i), where $x_i \in \mathcal{R}^n$, and $y_i \in \{-1,1\}^l$, the support vector machines (SVM) require the solution of the following optimization problem (Boser et al., 1992; Cortes and Vapnik, 1995):

$$\begin{aligned} &\min_{w,b,\mathcal{E}} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ &subject\ to \quad y_i(w^T \phi(x_i) + b) \\ &\qquad\qquad \geq 1 - \xi_i \end{aligned} \tag{1}$$

The dataset is not always linearly separable. In these case, we can introduce a slack variable $\xi_i \geq 0$ for each x_i (, $i = 1, \dots N$).

The training vectors x_i, are mapped into a higher-dimension feature space by the function ϕ, and try to find the hyperplane that linearly separates the mapped vectors i, in other words, SVM tries to find a linear hyperplane with the maximal margin in this higher-dimension feature space. $C > 0$ is called the penalty parameter of the error term. Furthermore, the kernel function is:

$$K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_i) \tag{2}$$

There are many types of kernel functions in SVM which are being proposed by researchers: Linear kernel function, polynomial kernel function, radial-basis kernel function (RBF) and sigmoid function. In this paper, we will use SVM with radial basis function (RBF) [19].

- **Linear kernel function:**
$$K(x_i, x_j) = x_i^T x_j \tag{3}$$
- **Polynomial kernel function:**
$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \tag{4}$$
- **Radial-basis kernel function (RBF):**
$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0 \tag{5}$$
- **Sigmoid kernel function:**
$$K(x_i, x_j) = tan(\gamma x_i^T x_j + r) \tag{6}$$

$\boldsymbol{\gamma}$, **r** and d are kernel parameters.

In this work, we have chosen the SVM RBF kernel, because of the following reasons [20]:

- It has fewer controllable parameters than the polynomial kernel.

- o The RBF kernel maps samples into a higher dimensional.
- o Has less numerical difficulties.

For the SVM based on RBF as the kernel function, there are two parameters, (C and γ) to be optimized. The goal is to identify the best (C and γ):

- ***C*** is a regularization parameter that controls the "flexibility" of the hyperplane. If C is finite, it allows misclassifying some points and changes the problem of perfectly separable data to finding a "soft-margin" classifier, and if C is lower, it allows softer constraints and corresponds to a larger margin. If C is Larger, it will force the creation of a more accurate model, with a narrow margin.
- ***γ*** is the kernel parameter that controls the correlation among support vectors. selecting an improper value for y may cause an over fitting. So, it is difficult to realize enough accuracy

Many solutions have been proposed to optimize the SVM parameters. In the recent years, intelligent optimization has been applied for SVM and has shown great results, involving Particle Swarm Optimization (PSO) algorithm [21], [22], genetic algorithm [23], Artificial bee Colony (ABC) algorithm [24] etc... In this paper, we propose to apply Particle Swarm Optimization (PSO) to optimize parameters (C and γ). This method is a popular algorithm used for optimization problems, and it is a swarm intelligence algorithm based on a population of individuals. One important key factor when implementing these algorithms is to choose the right fitness function.

4. PARTICLE SWARM OPTIMIZATION

4.1 Overview

Particle swarm optimization, PSO, is a popular nature-inspired heuristic optimization algorithm developed by Eberhart and Kennedy [25]. The algorithm was inspired by the swarming behavior and natural flocking of birds and insects. A set of randomly generated solutions called initial swarm used to explore the space. Thus, each particle (a bird, an insect or a fish) makes use of its own memory. Besides, the knowledge gained by the whole swarm is used to find the best solution (a safe place or a rich source of food and to avoid predators). In this work, it will search for the best parameters of SVM classifier (C and γ) based on the accuracy of the SVM algorithm.

PSO performs searches using population (or swarm) of agents (called particles). Each particle *i* has an initial population and initial velocity of the population of size N and dimension d. The initial population (swarm) of size *n* and dimension *d* is denoted as: $Pos= [Pos_1, Pos_2 ..., Pos_n]^T$, where *T* is the transpose operator. Each individual (particle) Pos_i ($i = 1..., n$) is given as: $Pos_i = [Pos_{i,1}, Pos_{i,2},\dots Pos_{i,d}]^T$.

The initial velocity of the population is denoted as: $Vel=[Vel_1, Vel_2,\dots, Vel_n]^T$. Thus, the velocity of each particle: Vel_i ($i=1\dots s,,n$) is given as $Vel_i =[Vel_{i,1}, Vel_{i,2},\dots Vel_{i,d}]^T$. The index *i* varies from *1* to *n* whereas the index j varies from *1* to *d.*, in our case *d* is *2*.

To discover the optimal solution, each particle moves in the direction of its previous best position ($Pbest^t_{i,j}$) and its best global position ($Gbest$), according to the following equations:

$$\begin{aligned} Vel^{t+1}_{i,j} = w \times Vel^{t+1}_{i,j} + c_1 \times r_1 \\ \times \left(Pbest^t_{i,j} - Pos^t_{i,j}\right) \\ + c_2 \times r_2 \\ \times \left(Gbest^t_{i,j} - Pos^t_{i,j}\right) \end{aligned} \quad (7)$$

$$Pos^{t+1}_{i,j} = Pos^t_{i,j} + Vel^{t+1}_{i,j} \quad (8)$$

In the above equation, w is the inertia factor weight varying between $[w_{min}\ ,\ w_{max}]$ that is used to balance the global exploration and local exploration, c_1 and c_2 are social learning factors (or acceleration constants), r_1 and r_2 are random numbers between $\{0,1\}$.

In eqn. (7) $Pbest^t_{i,j}$ represents personal best jth component of ith individual, whereas $Gbest^t_{i,j}$ represents jth component of the best individual of population up to iteration t.

4.2 Proposed PSO-SVM classification method

In this section, we describe the proposed PSO-SVM system for intrusion detection classification. The aim of this model is to optimize the SVM classifier accuracy by estimating the best values of the regularization and kernel parameters (C and γ).

To implement our proposed model, the RBF kernel function is used for the SVM classifier. Thus, the parameters (C and γ) must be optimized using PSO-SVM model. For the fitness function, we used the classification accuracy obtained after training SVM with the user parameters (C and γ) as described in [26]. The detailed steps of PSO algorithm are defined as follows:

```
Step 1.  Set parameters $\boldsymbol{w_{min}}$ , $\boldsymbol{w_{max}}$ , $\boldsymbol{c_1}$ and
         $\boldsymbol{c_2}$ of PSO
Step 2.  Initialize population of particles having
         positions Pos and velocities Vel
Step 3.  Set iteration t = 1
Step 4.  Calculate the fitness of particles $\boldsymbol{F_i^t}$=
         $f(\boldsymbol{Pos_i^t})$; $\forall \boldsymbol{i}$ and find the index of best particle
         k
Step 5.  Select $\boldsymbol{Pbest_i^t}$=$\boldsymbol{Pos_i^t}$ ; $\forall \boldsymbol{i}$ and
         $\boldsymbol{Gbest^k}$=$\boldsymbol{Pos_k^t}$
Step 6.  $\boldsymbol{w} = \frac{\boldsymbol{w_{max}} - \boldsymbol{t} \times (\boldsymbol{w_{max}} - \boldsymbol{w_{min}})}{\mathbf{max\_it}}$
Step 7.  Update velocity and position of
         particles:
         $\boldsymbol{Vel_{i,j}^{t+1}} = \boldsymbol{w} \times \boldsymbol{Vel_{i,j}^{t+1}} + \boldsymbol{c_1} \times \boldsymbol{r_1} \times$
         $(\boldsymbol{Pbest_{i,j}^t} - \boldsymbol{Pos_{i,j}^t}) + \boldsymbol{c_2} \times \boldsymbol{r_2} \times$
         $(\boldsymbol{Gbest_{i,j}^t} - \boldsymbol{Pos_{i,j}^t})$ ; $\forall \boldsymbol{i}$ and $\forall \boldsymbol{j}$
         $\boldsymbol{Pos_{i,j}^{t+1}} = \boldsymbol{Pos_{i,j}^t}$+$\boldsymbol{Vel_{i,j}^{t+1}}$ ; $\forall \boldsymbol{i}$ and $\forall \boldsymbol{j}$
Step 8.  Evaluate the fitness of particles $\boldsymbol{F_i^{t+1}} =$
         $\boldsymbol{f}(\boldsymbol{Pos_i^{t+1}})$ $\forall \boldsymbol{i}$ and find the index of best
         particle k1
Step 9.  Update $\boldsymbol{Pbest}$ of population $\forall \boldsymbol{i}$
         If ($\boldsymbol{F_i^{t+1}} < \boldsymbol{F_i^t}$) then $\boldsymbol{Pbest_i^{t+1}}$=$\boldsymbol{Pos_i^{t+1}}$
         else $\boldsymbol{Pbest_i^{t+1}}$=$\boldsymbol{Pbest_i^t}$
Step 10. Update $\boldsymbol{Gbest}$ of population
         If ($\boldsymbol{F_{k1}^{t+1}} < \boldsymbol{F_k^t}$) then $\boldsymbol{Gbest^{t+1}} = \boldsymbol{Pos_k^{t+1}}$
         and k=k1
         else $\boldsymbol{Gbest^{t+1}} = \boldsymbol{Gbest^t}$
Step 11. If $t$ < $\boldsymbol{max\_it}$ then $t$=$t$+$1$ and return to
         Step 6
         else go to step 12
Step 12. Print optimum solution as $\boldsymbol{Gbest^t}$.
```

A detailed flowchart of PSO-SVM steps is shown in figure1:

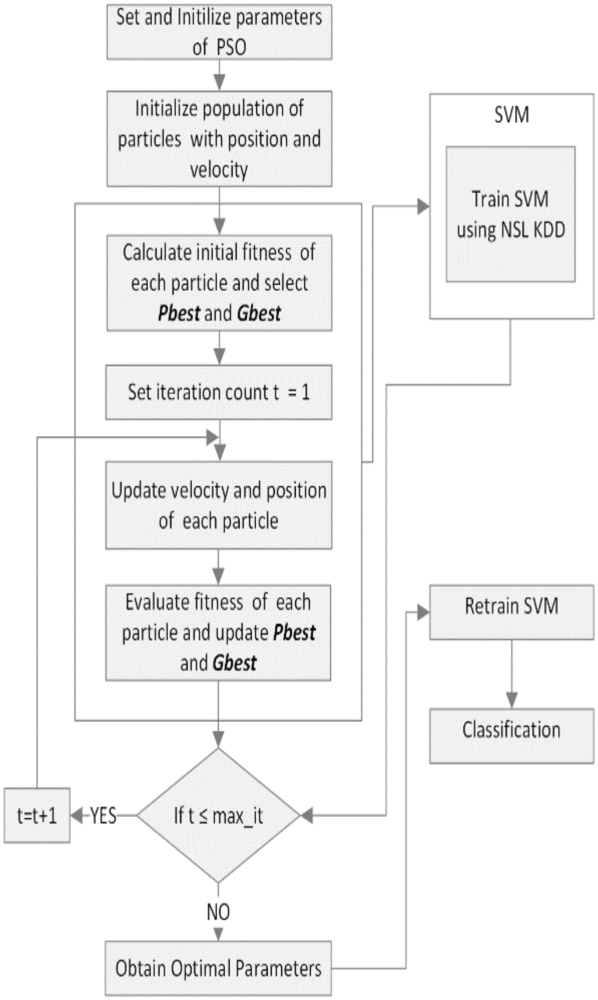

Set and Initilize parameters of PSO
Initialize population of particles with position and velocity
Calculate initial fitness of each particle and select Pbest and Gbest
Set iteration count t = 1
Update velocity and position of each particle
Evaluate fitness of each particle and update Pbest and Gbest
SVM
Train SVM using NSL KDD
Retrain SVM
Classification
t=t+1
YES
If t ≤ max_it
NO
Obtain Optimal Parameters

Figure 1: Flowchart of PSO-SVM

5. PROPOSED MODEL

5.1 Overview

Our proposed IDS model encompasses three phases:

- **Pre-processing phase**: In this unit, the NSL KDD Dataset is preprocessed by transforming the symbolic valued attributes to numeric and applying the discretization algorithm.
- **Feature selection phase:** In this unit, Information Gain is employed for feature selection.
- **Post-Processing phase**: or Classification phase, here, SVM is used for classification. The SVM parameters are selected by the particle swarm optimization algorithm PSO. Figure 2 shows the diagram of the proposed system.

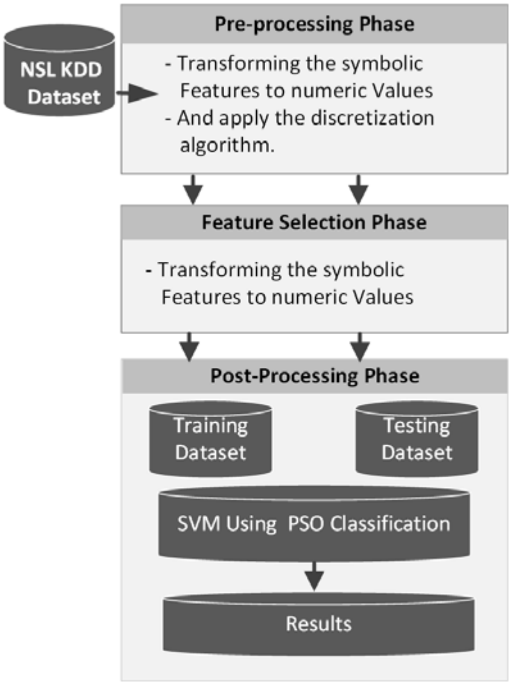

NSL KDD Dataset
Pre-processing Phase
- Transforming the symbolic Features to numeric Values
- And apply the discretization algorithm.
Feature Selection Phase
- Transforming the symbolic Features to numeric Values
Post-Processing Phase
Training Dataset
Testing Dataset
SVM Using PSO Classification
Results

Figure 2 Architecture of the proposed model

5.2 NSL KDD Dataset and pre-processing phase

To evaluate our proposed model, we will use NSL-KDD Dataset [27]. This Dataset is an improved version of KDD-Cup 99 that does not suffer from issues such as redundancy and complexity level of data. NSL-KDD data includes **41** features, **125973** instances for training set and **22544** instances for Testing set, **5** classes that are normal and 4 types of attacks. The attacks fall into four classes**: Denial of service (DoS), Remote-to-Local(R2L), User-to-Root(U2R)** and **Probing**, Attacks can be categorized in table 1 as the following:

Table 1: Attack classes type and their related attack names

Classes	Description	Attacks
Denial of Service Attack (DoS)	Is an attack in which the attacker takes action that prevents legitimate users from accessing targeted computer, devices or other network resources.	back, neptune, pod, smurf, teardrop, process table, warezmaster, apache2, mail bomb.
User to Root Attack (U2R)	Occurs when an attacker gain root access on a system by accessing as a normal user to the system and exploiting some vulnerability.	ipsweep, nmap, port sweep, satan, mscan, saint
Remote to Local Attack (R2L)	Occurs when an attacker gains local access on a machine that not have the access by exploiting some vulnerability.	guess_passwd, named, snmpgetattack, xlock, send mail
Probing	Is an action taken to gather information about the state of the network for the apparent purpose of circumventing its security controls.	http tunnel, ftp_write, multihop,buffer overflow, root kit, xterm, ps.

Table 2 shows the distribution of the four types of attacks in NSL-KDD Training Dataset:

Table 2: Attack classes of NSL-KDD

Classes	Nbre of events	Average
Normal	67343	53.46%
Denial of Service Attack (DoS)	45927	36.46%
Probing	11656	9.26%
Remote to Local Attack (R2L)	995	0.8%
User to Root Attack (U2R)	52	0.04%

NSL-KDD data has three features types: Numeric, Nominal, and Binary. Table 3 show the distribution of features according to their types:

Table 3: Type of features in NSL -KDD

Type	Features
Nominal	Protocol_type (2), Service (3), Flag (4)
Binary	Land (7), logged in (12), root shell (14), unattempt (15), is_host_login (21), is_guest_login (22)
Numeric	Duration (1), src_bytes (5), dst_bytes (6), wrong_fragment (8), urgent (9), hot (10), num_failed_logins (11), num_compromised (13), num_root (16), num_file_creations (17), num_shells (18), num_access_files(19), num_outbound_cmds (20), c ount (23), srv_count (24), serror_rate (25), srv_serror_rate (26), rerror_rate (27), srv_rerror_rate (28), same_srv_rate (29), diff_srv_rate (30), srv_diff_host_rate(31), dst_host_count (32), dst_host_srv_count (33), dst_host_same_srv_rate (34), dst_host_diff_srv_rate (35), dst_host_same_src_port_rate (36), dst_host_srv_diff_host_rate (37), dst_host_serror_rate (38),

	dst_host_srv_serror_rate (39), dst_host_rerror_rate (40), dst_host_srv_rer ror_rate (41)

5.3 Feature selection phase

In this phase, we reduce the number of features by eliminating the irrelevant ones using Feature Selection (FS) techniques. Feature selection is important to improve the efficiency of classification algorithms. Most of the data include irrelevant, redundant, or noisy features. FS methods reduce the number of features from a noisy Dataset and select only a subset of relevant features which best describe the problem to be solved and fewer downgrades the performances of the system [28]. These methods can offer several advantages such as: enhancing the performance of the classifier, creating a less complex Dataset easily interpretable or reducing processing costs in terms of execution time etc. There are two common approaches for feature reduction [29]:

- **Wrapper approach** requires a predictive model used to evaluate the Dataset and rate its relevance. This approach produces better feature subsets but needs more time than a filter approach to the process features.
- **Filter approach** evaluates features according to heuristics based on general characteristics of the data. The filter approach is simpler and more independent than the classifier.

In this work, we used Information Gain (IG) techniques for feature selection. This method belongs to the filter-based approach. The filter-based approach is currently applied, including Correlation Feature Selection (CFS) [30], Information Gain (IG) or Gain Ratio (GR). Though simple and rapid, these methods do not always offer an improved detection rate or accuracy of the detection stage but help the classifier to seek a better accuracy.

Let's suppose we have a dataset D with n classes $\{C_1, C_2,\ldots,C_n\}$. Suppose further that we have a possible test with m outcomes that partitions D into m subsets {D1, D2, …, Dm}. For a numeric attribute, m=2, since we only perform binary split. The probability that is selected one record from the set D of data records and decide if that belongs to some class Ci is given by:

$$\sum_{i=1}^{m} \left[\frac{(C_i, D)}{|D|}\right] \tag{9}$$

(C_i,D) represents the number of data records of the class C_i in Dataset D. $|D|$ is the total number of data records in Dataset D, the information that is convey is:

$$-log_2 \left[\frac{(C_i, D)}{|D|}\right] \tag{10}$$

Expected information needed to classify a given sample is calculated by:

$$IG(D_1, \ldots, D_m) = -\sum_{i=1}^{m} \left[\frac{(C_i, D)}{|D|}\right] \times log_2 \left[\frac{(C_i, D)}{|D|}\right] \tag{11}$$

A feature F with values $\{f_1, f_2, \ldots, f_v\}$ can divide the Dataset **D** into v subsets $\{D_1, D_2, \ldots, D_v\}$ where D_j is the subset which has the value f_j for feature F. Furthermore, let D_j contain D_{ij} samples of class C_i. Entropy of the feature F is given by:

$$E(F) = \sum_{j=1}^{v} \left[\frac{F_j}{|F|}\right] \times IG(F_j) \tag{12}$$

Where $|F_j|$ represents the number of records in the subset D_i after dividing the dataset D. The information gain is calculated by this formula:

$$Gain(F) = IG(D_1, \ldots, D_m) - E(F) \tag{13}$$

The steps of basic feature selection algorithm are as follows:

Input: Set of 41 features from NSL KDD Dataset.

Output: Reduced set of features.

Step 1. Select the attributes which have variation in their values.

Step 2. Calculate the IG(D) values for the selected attributes using (11).

Step 3. Select the attributes which have maximum number of nonzero values.

Step 4. Calculate the E(F) value for the attributes selected in step 3 using (12).

Step 5. Depend on the Gain value, using (13) select the attributes.

In order to apply the IG method for feature selection, the continuous attributes of NSL-KDD dataset, must be first discretized using the method introduced in [31]. For this, we used the

InfoGainAttributeEval with Ranker as the search method from Weka. Using the ranked list, the top 20 features were selected; these features are listed in table 4, where Symbolic values are denoted as S and Continuous values are denoted as C.

Table 4: Selected features after Applying Feature Selection Using IG

Feature name	Type	Rank
src_bytes	C	0.806
Service	S	0.632
dst_bytes	C	0.631
Flag	S	0.519
diff_srv_rate	C	0.515
same_srv_rate	C	0.507
dst_host_srv_count	C	0.472
dst_host_same_srv_rate	C	0.439
dst_host_diff_srv_rate	C	0.412
dst_host_serror_rate	C	0.403
logged_in	C	0.401
dst_host_srv_serror_rate	C	0.396
serror_rate	C	0.390
count	C	0.382
srv_serror_rate	C	0.377
dst_host_srv_diff_host_rate	C	0.268
dst_host_count	C	0.194
dst_host_same_src_port_rate	C	0.192
srv_diff_host_rate	C	0.144
srv_count	C	0.093

5.4 Post-processing phase

As explained in section 4, we used SVM based RBF kernel combined with PSO algorithm to optimize SVM parameters. The parameters of PSO algorithm are described as follows:

- Population: 20 particles
- Problem dimension is 2.
- Inertial weight reduced from 0.9 to 0.4
- Maximum iteration (max_it): 100

For parameter C we set the range between 1 and 240000 and for γ we set the range between 0.001 and 50. The fitness function is the accuracy of the SVM classifier.

5.5 Performance Metrics

In machine learning, many different measures metrics are used to evaluate the classification models [32]. In order to evaluate the effectiveness of our IDS model, we use the following performance measures: False Positive Rate (FPR), False Negative Rate (FPR), True Positive Rate (TPR), Accuracy and Precision.

These performance measures are computed based on the confusion matrix, where:

- **True Negative (TN):** Events which are actually normal and are successfully labeled as normal.
- **True Positive (TP):** Events which are actually attacks and are successfully labeled as attacks.
- **False Positive (FP):** A normal events being classified as attacks.
- **False Negative (FN):** Are attack events incorrectly classified as normal events.

So, True Positive Rate (TPR), also called Recall (R) or Attack Detection Rate (ADR) is the ratio between total numbers of attacks detected by the system to the total number of attacks present in the Dataset. TPR will show if our proposed model is capable of detecting attacks, TPR is calculated using the equation:

$$TPR = \frac{TP}{TP + FN} \tag{14}$$

The **False Positive Rate (FPR)** refers to the proportion that normal information is mistakenly detected as attack behavior. A high FPR will cause the low performance of the IDS, FPR is calculated using the equation:

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

False Negative Rate (FNR), refers to the proportion that normal attack events are incorrectly classified as normal events, a high FNR will leave the system vulnerable to intrusions, FNR is evaluated using the equation:

$$FNR = \frac{FN}{FN + TP} \tag{16}$$

Accuracy (A) is the ratio between total number of correctly classified instances to the total number of samples from the Dataset. The accuracy will show if our model is capable of raising proper alarms, when it detects attacks and not generating false alarms when the network traffic is normal, is calculated using the equation:

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{17}$$

Precision (P) is the proportion of attack cases that were correctly predicted relative to the predicted size of the attack class, is calculated using the equation:

$$P = \frac{TP}{TP + FP} \tag{18}$$

So, to have an IDS with high performances; both FPR and FNR rates should be minimized, together with maximizing Accuracy, TPR and TNR.

6. EXPERIMENTAL RESULTS

In order to evaluate our proposed model, we compared it with the default SVM classifier model without feature selection and with Feature selection using IG. We also compared our model with other proposed models. Our platform for the experiment is described as follows:

- Processor: Intel (R) Core (TM) i7-6500U CPU@ 2.50GHZ 2.59 GHZ,
- Memory: 6 GB
- System (OS): Linux Ubuntu Server 16.04 64-bit

To build the SVM Classifier and for feature selection using Information Gain, we used Weka 3.8.1 [34], and we implemented the PSO algorithm in JAVA using NetBeans IDE 8.2 to optimize the regularization and learning parameters of SVM classifier.

The evaluation of the classifier was performed by a ten-folds cross-validation for the NSL-KDD Dataset in order to avoid overfitting. As our evaluations imply cross validations, we simplify the dataset by randomly selecting 10% of records from the training file which is 12613 instances and 20% from testing file which is 4524 records.

For the feature selection we used the training. Having the training file with the subset of feature selected, we train the classifier and evaluate it for the reduced test file. In this manner we can determine if our model is capable to identify new attacks, as the intrusions in the test file are not included in the training. Moreover, to enhance classification, we convert the symbolic valued attributes (service, flag, class) to numerical values.

As we explained before, we compared our model with standard SVM using RBF kernel and PSO-SVM (we trained SVM with the proposed Particle Swarm Optimization algorithm) and with FS-PSO-SVM after applying feature selection using Information Gain.

Results from table 5 show that our proposed model (IG with PSO and SVM) improves attack detection rate (True Positive Rate TPR) with almost 2 % and reduces false positives rate (FPR) with almost 6.4 % and improves accuracy with 2 % when compared with standard SVM.

Table 5: Performance Measures of SVM, FS-SVM, PSO-SVM and FS-PSO-SVM using NSL -KDD

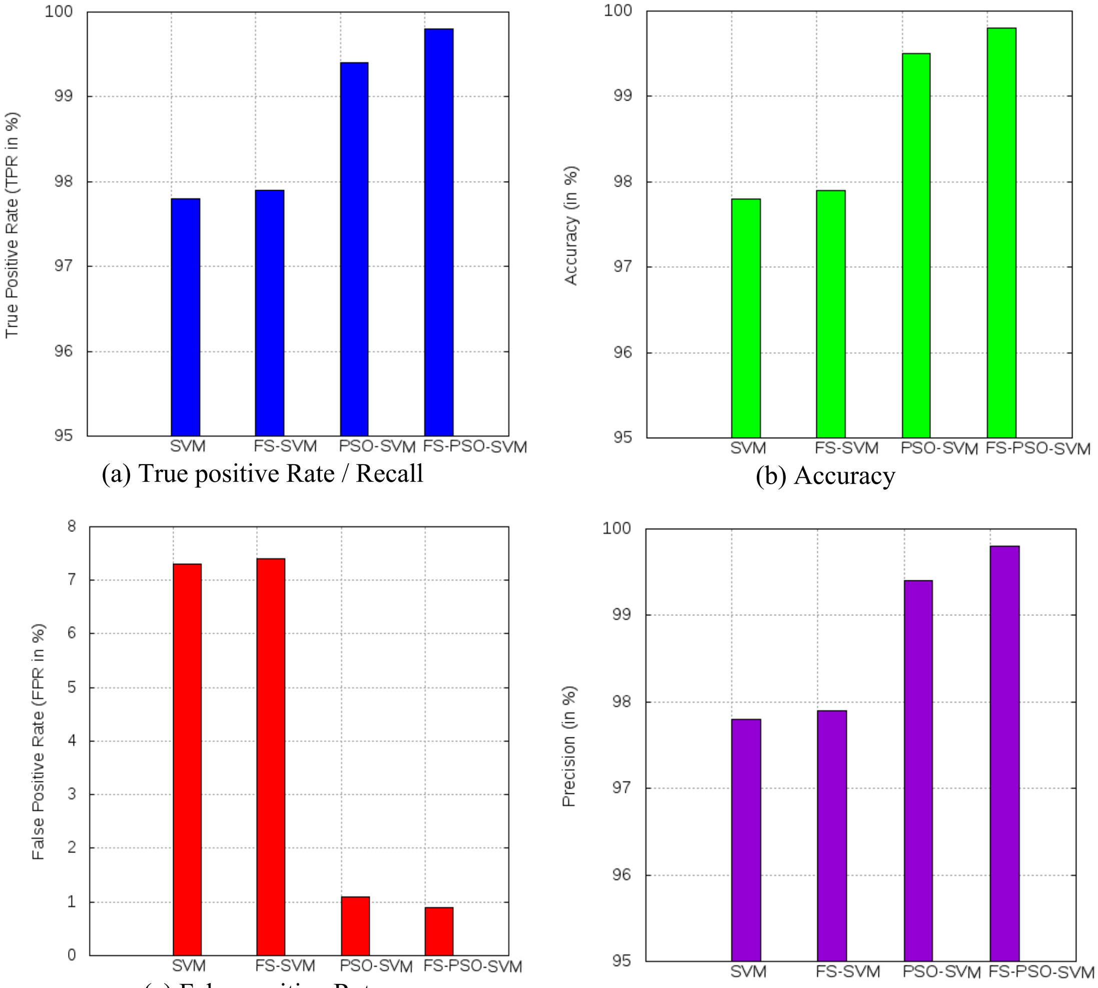
Classifier	**Nb. features**	**TPR %**	**FPR %**	**Accuracy %**	**Precision %**
SVM	41	97.8	7.3	97.8	97.8
FS -SVM	20	97.9	7.4	97.9	97.9
PSO-SVM	41	99.4	1.1	99.5	99.4
FS-PSO-SVM	**20**	**99.8**	**0.9**	**99.8**	**99.8**

The best parameters obtained of C and γ of SVM classifier using RBF kernel with consideration of PSO after 100 iterations are C = 240007.0 and $\boldsymbol{\gamma}$= 0.0016326.

From table 5, it is observed that the classification accuracy in the case of PSO-SVM using the entire feature (41) is 99.5%, whereas the classification accuracy in the case of using Feature Selection technique is found to be 99.8%. This demonstrates the efficiency of the proposed model in which PSO optimization and Feature selection are applied.

All 20 features are auto selected from the corresponding input, and the testing success rate has been improved significantly. Furthermore, it should be noted that when using the PSO technique, the process of classification takes a relatively shorter computational time for training.

The above four performance measures (TPR, FPR, Accuracy and Precision) of the four models (SVM, FS-SVM, PSO-SVM and FS-PSO-SVM) are plotted in Figure 3.



(a) True positive Rate / Recall

(b) Accuracy

(c) False positive Rate

(d) Precision

Figure 3: The Performance Measures of SVM, FS-SVM, PSO-SVM and FS-PSO-SVM

In the table 6, we show some recent IDS models based on SVM and Swarm Intelligence techniques in order to range our proposed IDS Model in the current context. Results from table 5 reveal that our model is comparable with other proposed IDS in term of Accuracy, TPR and even generates fewer false positives than other proposed models.

Table 6: Comparison of the proposed IDS model with other models using Swarm Intelligence technique

Classifier	**Authors**	**Dataset**	**TPR/Recall**	**FPR**	**Accuracy**
BPSO-SVM	Ma et al. (2008) [14]	KDD Cup 99	96.7%	8.0%	N/A
PSO-SVM (BPSO, SPSO)	Wang et al. (2009) [9]	KDD Cup 99	99.8%	N/A	N/A
ACO-SVM	Pu et al. 2012) [8]	KDD Cup 99	99.2%	N/A	N/A
IG-BA-SVM	Enache and Patriciu (2014) [16]	NSL-KDD	95.7%	4.08%	94.16%
IG-ABC-SVM	Enache and Sgârciu (2014) [15]	NSL-KDD	98.5%	0.0374	98.89%
FS-SVM, MSVM-PSO	GuiPing et al. (2015) [33]	KDD Cup 99	N/A	2.02%	97.64%
FS-PSO-SVM	**Chakir et al. (2017)**	**NSL-KDD**	**99.8%**	**0.9%**	**99.8%**

However, the other IDS models enumerated in table 6 use the KDD Cup 99 dataset which has some well-known flaws and therefore the results might be biased but, the comparison is still relevant as it offers a current status review.

7. DISCUSSION

The proposed intrusion detection model in this work is an integration of feature selection technique using Information Gain IG and Support Vector Machine SVM optimized by parameter tuning technique using Particle Swarm Optimisation PSO. This approach is different from the prior traditional approaches as the problem of dimensionality is high in large data sets. Hence, an integration of feature selection and classification results in a better classification accuracy of the attacks in comparison to other approaches discussed in section 2. SVM is one of the best learning algorithms. The reason for selecting SVM is because the unknown pattern is determined according to the maximum result obtained from all SVMs which results in a negligible error rate. The SVM model parameters are tuned by the parameter tuning technique using PSO algorithm discussed in Section 4.2, which is an additional optimization task performed to yield a better prediction. The training of the classifier with optimized parameters using PSO assures the prediction label is accurate for the testing phase. The novelty of this approach is this kind of optimization tuning and the reduction of the useless or irrelevant features from the dataset. Furthermore, when using the Feature Selection and the PSO technique, the process of classification takes a relatively shorter execution time for training. Nonetheless, it should be noted that obtaining a good parameters optimization for SVM Classifier is very difficult process, any small change in the parameters can lead to a big difference in the classification results, then the PSO algorithm must be well implemented.

8. CONCLUSION

This paper proposes an intrusion detection model using Information Gain feature selection and Support Vector Machine. A parameter tuning technique is adopted for optimization of RBF kernel parameter of the SVM Classifier gamma and overfitting constant (C and γ) using Particle Swarm Intelligence. The advantage of combining feature selection and parameter optimization for SVM is to reduces training and testing time and improve the effectiveness of the SVM Classifier. The investigational results on NSL-KDD dataset which is an enhanced version of KDD Cup 1999 dataset shows that our proposed model FS-PSO-SVM results in obtained the highest detection rate (99.8%) and the lowest false positive rate (0.9%) in comparison to other traditional approaches.

For future enhancements, we will further improve our optimization algorithm using other algorithms of swarm intelligence, we may develop some algorithms combining kernel methods with other classification methods for pattern analysis and optimization techniques for SVM parameter optimization. Also, we intend to perform further tests on different datasets in order to validate our proposed model.

REFRENCES

[1] James P. Anderson, 1980. Computer security threat monitoring and surveillance. Technical report, JamesP. AndersonCo.

[2] Dorothy E. Denning, 1987. An intrusion detection model. IEEE Transactions on Software Engineering, SE-13(2):222-232.

[3] Stefan Axelsson, 1999. The base rate fallacy and its implications for the intrusion detection. In Proceedings of the 6th ACM conference on Computer and Communications Security, pp.1-7, Kent Ridge Digital Labs, Singapore.

[4] Debar, H., Dacier, M., Wespi, A., 1999. Towards a taxonomy of intrusion-detection systems. Computer Networks: The International Journal of Computer and Telecommunications Networking - Special issue on computer network security, vol. 31, pp. 805–822.

[5] Koliasa, C., Kambourakisa, G., M. Maragoudakisa, 2011. Swarm intelligence in intrusion detection: A survey, Computers and Security, Vol.30(8), pp. 625–642.

[6] Kukielka, P., Kotulski, Z., 2014. New Unknown Attack Detection with the Neural Network-Based IDS. In State of the art in intrusion prevention and detection, Auerbach Publications, pp.259-284.

[7] Wang, J., Hong, X., Ren, R., Li, T., 2009. A real-time intrusion detection system based on PSO-SVM. in Proc. of the International Workshop on Information Security and Application (IWISA 2009), Qingdao, China, pp. 319–321

[8] Pu, J., Li, Y., Xiao, L., Dong, X., 2012. A Detection Method of Network Intrusion Based on SVM and Ant Colony Algorithm. inProc.

National Conference on Information Technology and Computer Science (CITCS 2012), Lanzhou, China, pp.153–156.

[9] Iftikhar, A., Muhammad, H., Alghamdi, A., Alelaiwi, A., 2014. Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components. Springer, Neural Computing and Applications pp. 1671-1682.

[10] Zhou, J., Tao, B., Jiming, T., Aiguang, Z., 2008. The study of SVM optimized by Culture Particle Swarm Optimization on predicting financial distress. Automation and Logistics. ICAL 2008. IEEE International Conference on. IEEE.

[11] Horng, S.J., Su, M.Y, Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L., Perkasa, C.D., 2011. A novel intrusion detection system based on hierarchical clustering and support vector machines. Elsevier, Expert systems with Applications pp. 306-313.

[12] Gaspar, P., Jaime, C., Oliveira, J.L, 2012. On the parameter optimization of Support Vector Machines for binary classification. Journal of Integrative Bioinformatics Vol 9(3): 201.

[13] Kim, H.-S., Cha, S.-D., 2005. Empirical evaluation of SVM-based masquerade detection using UNIX commands. Computers and Security, vol. 24 (2), pp.160–168.

[14] Ma, J., Liu, X., Liu, S., 2008. A New Intrusion Detection Method Based on BPSO-SVM. In Proceedings of the International Symposium on Computational Intelligence and Design (ISCID2008), vol. 1, pp.473–477.

[15] Enache, A.C, Sgârciu, V., 2014. Anomaly Intrusions Detection Based on Support Vector Machines with Bat Algorithm. In Proceedings of the 18th International Conference on System Theory, Control and Computing, Sinaia, Romania. IEEE, pp. 856-861

[16] Enache, A.-C., Patriciu, V. V., 2014. Intrusions detection based on Support Vector Machine optimized with swarm intelligence. In in Proceedings of the 9th International Symposium on Applied Computational Intelligence and Informatics (SACI2014), IEEE, pp.153 –158

[17] Boser, B. E., Guyon, I., Vapnik, V., 1992. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144-152. ACM Press.

[18] Cortes, C. Vapnik, V., 1995. Support-vector network. Machine Learning. Kluwer Academic Publishers, Boston. Manufactured in The Netherlands. Machine Learning, 20, 273-297.

[19] Vanitha, AR, Venmathi, L., 2011. Classification of Medical Images Using Support Vector Machine. In Proceedings of International Conference on Information and Network Technology (IPCINT 2011). Vol (4), pp. 63-67.

[20] Byun, H., Lee. S-W., 2003. A survey on pattern recognition applications of support vector machines. International Journal of Pattern Recognition and Artificial Intelligence 17.03: pp. 459-486.

[21] Yaghini, M., Khoshraftar, M., Fallahi, M., 2013. A hybrid algorithm for artificial neural network training. Journal, Engineering Applications of Artificial Intelligence, vol. 26, no. 1, pp. 293–301.

[22] Garro, B. A., Sossa, H., Vazquez, R., 2011. Back-propagation vs. particle swarm optimization algorithm: which algorithm is better to adjust synaptic weights of a feed-forward ANN. International Journal of Artificial Intelligence, vol. 7, no. A11, pp. 208–218.

[23] Jung, H. G., Yoon, P. J., Kim, J., 2007. Genetic algorithm-based optimization of SVM-based pedestrian classifier. In The 22nd International technical conference on circuits/systems, computers and communications(ITC-CSCC2007), Busan, Korea, pp.783–784.

[24] Ozturk, C., Karaboga, D., 2011. Hybrid Artificial Bee Colony algorithm forneural network training. IEEE Congress on Evolutionary Computing, New Orleans, LA, USA, pp.84–88.

[25] Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In Proceeding of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp.39–43.

[26] Huang, C.L., Dun, J.F., 2008. A distributed PSO–SVM hybrid system with feature selection and parameter optimization, Appl. Soft Comput. 8, PP.1381–1391.

[27] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A., 2009. A Detailed Analysis of the KDD CUP 99 Dataset. In Proceeding of the 2009 IEEE symposium on computational Intelligence in security and defense application (CISDA), Ottawa, ON, Canada, pp.1–6.

[28] Sammut, C., Webb, G. I, 2010. Feature selection. In Encyclopedia of Machine Learning, pages 429–433. Springer, New York.

[29] Nguyen, H., Franke, K., Petrovic, S., 2010. Improving effectiveness of intrusion detection by correlation feature selection. In ARES. In proceding of the 10 International Conference on Availability, Reliability, and Security, pages 17–24, 2010

[30] Liu, H., Motoda, H., Setiono, R., Zhao, Z., 2010. Feature Selection: An Ever-Evolving Frontier in Data Mining. Journal of Machine Learning Research, Proceedings Track 10, pp. 4- 13.

[31] Fayyad, U. M., Irani, K. B., 1993. Multi-interval discretization of continuous-valued attributes. in Proceeding of the 13th International Joint Conference on Artificial Intelligence, 1993, pp. 1022–1027

[32] Chakir, E.M., Moughit, M., Chancerel, C., Khamlichi. Y.I., 2015. False positive reduction in intrusion detection system using alert correlation and datamining techniques. International Journal of Advanced Research in Computer Science and Software Engineering, volume 5 issue 4, pp.77-85.

[33] Wang, G.P, Chen, S., Liu, J., 2015. Anomaly-based Intrusion Detection using Multiclass -SVM with Parameters Optimized by PSO‖. International Journal of Security and Its Applications Vol. 9, No. 6, pp. 227-242.

[34] Witten, H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2nd edition, 2005.