International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 31999-32014

http://www.ripublication.com

Layered Approach For Intrusion Detection Using Multiobjective Particle Swarm Optimization

B.Ben Sujitha and Dr.V.Kavitha

Department of CSE, Ponjesly College of Engineering, Nagercoil, India
E-Mail : bensujithaphdece@gmail.com
CSE Department,Universal College of Engineering, Nagercoil, Tamilnadu, India

Abstract:

Intrusion detection is one of the challenging tasks in today's networked world. It is necessary to formulate a new intrusion detection system, which can monitor the network to detect the malicious activities. The proposed work focuses the issues, namely accuracy and efficiency. One way to improve performance is to use a minimal number of features to define a model in a way that it can be used to accurately discriminate normal from anomalous behavior. So the new system uses an optimized feature selection algorithm to produce the reduced set of features and high attack detection accuracy can be achieved by using a layered approach. The feature selection algorithm used in the proposed system is a multi-objective particle swarm optimization algorithm which does the feature selection effectively. The layered approach is effectively applicable to detect anomaly attack. The proposed system is tested with the benchmark KDD '99 intrusion data set as well real time captured data set, which outperforms other well-known methods such as the decision trees, naive Bayes and Ant Colony optimization. The system is highly robust and efficient. It can deal with real-time attacks and detect them fast and quick response.

Keywords: Particle Swarm Optimization, KDD,layered, Fitness, Objective function, Attack.

1. INTRODUCTION

There is a steady increase in the number of internet users. The malicious behavior is increasing at a fast pace and can easily cause millions of dollars in damage to an organization. Hence, the development of intrusion detection systems has been set corporations. Network and information security is one of the highest priority and

Challenging tasks for network administrators and security professionals. Intrusion Detection is a process to identify suspicious activities in a monitored system, from authorized and unauthorized users. More sophisticated security tools mean that the attackers come up with newer and more advanced penetration methods to defeat the installed security systems [1] and [2]. Thus, there is a need to safeguard the networks from known vulnerabilities and at the same time take steps to detect new and unseen attack, it is necessary to develop more reliable and efficient intrusion detection systems. Its prime purpose is to detect as many attacks as possible with a minimum number of false alarms, i.e., the system must be accurate in detecting attacks. So, there is an urgent and pressing need for replacing them by automated systems for constant monitoring and quick responses [3]. The new system can detect most of the attacks and the reduced false alarm, which can cope with large amounts of data, and is fast enough to make real-time decisions. The research on the Intrusion detection has been initiated around 1980s by Anderson [4]. Intrusion detection systems are classified as network based, host based, or application based depending on their mode of deployment and data used for analysis [5]. Additionally, intrusion detection systems can also be classified as signature based or anomaly based depending upon the attack detection method. The signature based systems is being trained by extracting specific patterns (or signatures) from previously known attacks and the anomaly-based systems learn from the normal data collected when there is no anomalous activity [6].The hybrid system incorporates the advantages of both the signature-based and the anomaly-based systems. A hybrid system is efficient, which can also be used to label unseen or new instances as they assign one of the known classes to every test instance. This system learns features from all the classes while training is done. In this paper, a new intelligent intrusion detection system has been proposed using the hybrid feature selection technique and modified layered approach with generation of a new rule which can effectively detect a new type of attack.

The rest of this paper is organized as follows: In Section 2, discuss about the related work with emphasis on various methods and frameworks used for intrusion detection. Section 3 covers the multi objective particle swarm optimization feature selection algorithm and the Layered Approach in Section 4. In Section 5 deals with the integration of layered approach with MPSO. The experimental results and comparison of the proposed method with other approaches is given in section 6. It is observed that the proposed system, Layered based multi objective particle swarm optimization, performs significantly better than other systems.

2. RELATED WORK

INTRUSION DETECTION SYSTEM (IDS) are security management system is used to identify anomalous activities and incomplete signatures within computers or networks. The number of methods and frameworks has been proposed and many systems have been built to detect intrusions. The various existing techniques and frameworks are discussed as follows.

Lee et al, introduced data mining approaches for detecting intrusions in [8], [9], and [10]. Data mining approaches for intrusion detection are based on building

classifiers by discovering relevant patterns of program and user behaviour. Association rules [11] and frequent episodes are used to learn the record patterns that describe user behaviour. These methods can deal with symbolic data, and the features can be defined in the form of packet and connection details. However, mining of features is limited to entry level of the packet and requires the number of records to be large and sparsely populated; otherwise, they tend to produce a large number of rules that increase the complexity of the system [12]. Data clustering methods such as the k-means and the fuzzy c-means have also been applied extensively for intrusion detection [13] and [14].

One of the main drawbacks of the clustering technique is that it is based on calculating numeric distance between the observations. Observations with symbolic features cannot be easily used for clustering, resulting in inaccuracy. In addition, the clustering methods consider the features independently and are unable to capture the relationship between different features of a single record, which further degrades attack detection accuracy. Naive Bayes classifiers have also been used for intrusion detection [15]. However, they make strict independence assumptions between the features in an observation resulting in lower attack detection accuracy when the features are correlated. Bayesian network can also be used for intrusion detection [16]. However, they tend to be attack specific and build a decision network based on the special characteristics of individual attacks. Thus, the size of a Bayesian network increases rapidly as the number of features and the type of attacks modelled by a Bayesian network increases. To detect anomalous traces of system calls in privileged processes [17], hidden Markov models (HMMs) have been applied in [18], [19], and [20]. However, modelling the system calls alone may not always provide accurate classification as in such cases various connection level features are ignored. Further, HMMs are generative systems and fail to model long-range dependencies between the observations [21]. Decision trees have also been used for intrusion detection [22]. The decision trees select the best features for each decision node during the construction of the tree based on some well-defined criteria. One such criterion is to use the information gain ratio, which is used in C4.5. Decision trees generally have very high speed of operation and high attack detection accuracy. Debar et al. [23] and Zhang et al. [24] discuss the use of artificial neural networks for network intrusion detection. Though the neural networks can work effectively with noisy data, they require large amounts of data for training and it is often hard to select the best possible architecture for a neural network. Support vector machines have also been used for detecting intrusions [25].

Support vector machines map real valued input feature vector to a higher dimensional feature space through nonlinear mapping and can provide real-time detection capability, deal with large dimensionality of data, and can be used for binary-class as well as multiclass classification. Other approaches for detecting intrusion include the use of genetic algorithm and autonomous and probabilistic agents for intrusion detection [26] and [27]. These methods are generally aimed at developing a distributed intrusion detection system. To overcome the weakness of a single intrusion detection system, a number of frameworks have been proposed, which describe the collaborative use of network-based and host based systems [28].

Systems that employ both signatures based and behaviour-based techniques are discussed in [29] and [30]. In [10], the authors describe a data mining framework for building adaptive intrusion detection models. A distributed intrusion detection framework based on mobile agents is discussed in [31]. Maximum entropy principle [32] for detecting anomalies in the network traffic, use only the normal data during training and build a baseline system. The system fails to model long-range dependencies in the observations. The new system is designed to have the benefits of computational efficiency and high accuracy of detection in a single system.

3. FEATURE SELECTION

Feature Selection (FS) is one of the data pre-processing techniques used before classification in IDS [33]. This step is applied to remove the irrelevant, noisy and redundant features, which improves the classification and detection accuracy. FS methods generate a new set of features by selecting only a subset of the original features [34]. It has two main conflicting objectives of maximizing the classification performance and minimizing the number of features. Data in high dimensional space may lead to decrease the classification accuracy of the IDS. There are three major problems [36] in the collected network traffic database: problem of irrelevant and redundant features, the problem of uncertainty, and problem of ambiguity. These problems not only hinder the detection speed, but also decline the detection performance of intrusion detection systems. There are two main feature selection methods: filter methods [35] and wrapper methods [36]. Filter methods evaluate the relevance of the features depending on the general characteristics of the data, without using any machine learning algorithm to select the new set of features [37]. By employing the filter approach to intrusion detection work, Qu et al. [40] applied pairwise correlation analysis to uncover mutual information between each feature and the decision class. Filter method can work efficiently with large number of traffic records. Frequently used filter methods include Principal Component Analysis (PCA)[38], Gain Ratio and Information Gain (IG)[39], Relief [38] and Focus [39]. The wrapper method employs a predetermined induction algorithm to find a subset of features [40] with the highest evaluation by searching through the space of feature subsets and evaluating quality of selected features. For increasing the detection rate and decreasing the false alarm rate in a network intrusion detection task, Stein et al. [43] used a genetic algorithm to select a subset of features with ID3[41] C4.5 algorithm[42]. The work of Mukkamala and Sung [44] is another example of using wrapper method.

Our proposed system uses the Multi objective Particle Swarm Optimization [48] feature selection method increases the accuracy and speeds up the detection time. This technique encompasses the filter and wrapper approach .The PSO-based algorithms using multi-swarm strategy has more exploration and exploitation abilities due to the fact that different swarms have the possibility to explore different parts of the solution space [6]. The multi-objective PSO can sustain the diversity of swarms, and ensure its adaptability, modified multi-objective PSO [47] is used consists of a number of sub-swarms and a scheduling module.

3.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was developed by Kennedy and Eberhart in 1995 [24]. PSO is an evolutionary computation technique which simulates the social behavior of organisms, such as bird flocking. Particle swarm optimization has the strongest global search capability and initialized with a population of particles having a random position (solution). Each particle is associated with velocity. Particle velocities are adjusted according to the historical behavior of each particle and its neighbors while they fly through search space [42]. Members of swarm communicate each other and adjust their own position and velocity based on the good position. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *lbest*. When a particle takes all the population as its topological neighbors, the best value is a global best and is called *best*. In this concept, at each time step in changing the velocity of (accelerating) each particle toward its p*best* and *lbest* locations. Consider Swarm of particles is flying through the parameter space and searching for optimum. Each particle is characterized by Position vector ($xi\ (t)$) and Velocity vector ($vi\ (t)$)

During the process, each particle will have its individual knowledge *pbest*, i.e., its own best-so-far in the position and social knowledge *gbest* i.e., *best* of its best neighbor.The velocity will be updated using the formula (1).Thus, the particles have a tendency to fly towards the better and better search area over the course of the search process [41]. The calculation of velocity is:

$$V_i\ (t+1) = \alpha V_i + C_1\ x\ rand\ x\ \left(pbest(t) - x_i(t)\right) C_2\ x\ rand\ x\ \left(gbest(t)\ - x_i(t)\right) \qquad (1)$$

Where α is the inertia weight that controls the exploration and exploitation of the search space *c*1 and *c*2, the cognitive and social components, respectively are the acceleration constants which changes the velocity of a particle towards the *pbest* and *gbest*. *rand* is a random number between 0 and 1. Usually *c*1 and *c*2 values are set to 2. Performing the position update using eq .(2),

$$X_{id} = X_{id} + V_{id} \qquad (2)$$

Even though PSO is an efficient method to do the feature selection it has drawbacks

- The method suffers from the partial optimism, which leads to the less exact at the regulation of its speed and the direction.
- The method cannot work out the problems of scattering.
- The method cannot work out the problems with non-coordinate system. To overcome the drawback multi objective PSO algorithm is used.

The pre-processing steps are

- Convert Symbolic features to numeric value using the eq .(7)

- Normalize the feature values, since the data have significantly varying resolution and ranges. The feature values are scaled to the range [0, 1], using the following equation:

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} - 1 \qquad (3)$$

Where X_{max},X_{min} , are the minimum and maximum value of a specific feature . X_n is a normalized output.The algorithm is initialized with a random population (swarm) of individuals (particle) , where each particle of the swarm represents a candidate solution in the d- dimensional search space. To find the best solution, each particle changes its searching direction according to: the best previous position of its individual memory (pbest) , represented by P_i = (P_{i1}, P_{i2} ,… $P_{id)}$; the global best position gained by the swarm (gbest) G_i = (G_{i1}, G_{i2} ,… $G_{id})_n$

The d- dimensional position for the particle i at iteration t can be represented as:

$$\mathrm{x_i^t} = \mathrm{x_{i1}^t}, \mathrm{x_{i2}^t}, .. \mathrm{x_{id}^t}. \qquad (4)$$

While the velocity (the rate of the position change) for the particle *i* at iteration t is given by

$$V_i^t = V_{i1}^t, V_{i2}^t, .. V_{id}^t. \qquad (5)$$

All of the particles have function have fitness values, which are evaluated based on a function as in eq (6)

$$\text{Fitness} = \alpha.\gamma R(D) + \beta \frac{|C|+|R|}{|C|} \qquad (6)$$

Where $\gamma R(D)$ is the classification quality of condition attributed set R relative to decision D and |R| is the length of the selected feature subset. |C| is the total number of features. While, the subset parameters α and β are correspond to the importance of classification quality and subset length. α=[0,1] and β=[$1 - \alpha$].

3.2. IEM Discretization Phase

Discretization is a process of converting the continuous space of feature into a nominal space [30]. The goal of discretization process is to find a set of cut points, these cut points partition the range into a small number of intervals [31]. In this model, the 11 features output from the PSO where discretised by the Information Entropy Minimization (IEM) discretization method. The IEM discretization method was proposed by Fayyad et al. [32], where the cut points should be set between points with different class labels.

Let T partition set S into subsets *S*1 and *S*2, for k classes *C*1, ...,*Ck* the class entropy of a subset S is given by

$$Ent\ (S) = -\sum_{i=1}^{k} P(C_i, S)\log\left(P(C_i, S)\right) \qquad (7)$$

Where $P(C_i, S)$ is the proportion of examples in S that have class C_i. For an attribute A, the class information entropy of the partition induced by partition T is defined as

$$E(A,T,S) = \frac{|S_1|}{|S|}Ent(S_1) + \frac{|S_2|}{|S|}Ent(S_2) \quad (8)$$

3.3. F-score

F-score is a simple technique which measures the discrimination of two sets of real numbers. Given training vectors $\boldsymbol{X}_k$, k = 1.2,…,m, if the number of positive and negative instances are n+ and n−, respectively, then the F-score of the i^{th} feature is defined as follows[10]:

$$F_i = \frac{\sum_{j=1}^{l}(\bar{x}_i^{(j)} - \bar{x}_i)^2}{\sum_{j=1}^{l}\frac{1}{n_j - 1}\sum_{k=1}^{n_j}(x_{k,1}^{(j)} - x_i^{(j)})^2} \qquad (9)$$

Where $\bar{x}_i, \bar{x}_i^{(j)}$, are the average of the i^{th} feature of the whole dataset and the j^{th} data set respectively . $x_{k,1}^{(j)}$ is the i^{th} feature of the k^{th} instance in the j^{th} dataset . If the i^{th} feature is selected ("1" represents that feature i is selected and "0" represents that feature i is not selected. FS(i) equals the instance of feature i, otherwise FS(i) equals 0.

$$FS(i) = \begin{cases} instance\ i, if\ i\ is\ selected \\ 0, if\ i\ is\ not\ selected \end{cases} \quad (10)$$

3.4. Objective function

The Objective function is the evaluation criteria for the selected features. To get accuracy rate, it is necessary to train and test the dataset according to the selected features.

The fitness of individual feature was obtained and thus that feature can be decided to be added or removed from the feature subset used.

$$\text{Fitness}_i = \theta_a\ X\ accuracy_i + \theta_b X \left[\left[\frac{\sum_{j=1}^{N_b} F(FS(i))}{\sum_{k=1}^{N_b} F(k)}\right]\right] \quad (11)$$

In Eq. (11), θ_a is the weight for SVM classification accuracy rate, $accuracy_i$ the classification accuracy rate for the selected features, θ_b the weight of the score of selected features, $F(FS(i))$ the function for calculating the score of the current features, and the total score of the selected features and all features. MPSO is

proposed, which holds a number of swarms scheduled by the multi-swarm scheduling module. Each swarm controls its iteration procedure, position updates, velocity updates, and other parameters respectively. The scheduling module monitors all the sub-swarms and gathers the results from the sub-swarms. Each sub-swarm contains a number of particles. The multi-swarm scheduler can send commands or data to sub-swarms, and vice versa.

a) **The swarm request rule:** If the current sub-swarm meets the condition according to Eq. (12), it sends the results which correspond *pbest* and *gbest* values to the multi-swarm scheduler. If *Si* = 1, the current swarm sends records which contain the *pbest* and *gbest* values, otherwise the current swarm does not send the results

$$. \ S_{i=} \begin{cases} 1, if\ d_i < \frac{tit_i - it_i}{tit_i} x\ rand() x\ Fitness \\ 0,\ if\ d_i \geq \frac{tit_i - it_i}{tit_i} x\ rand() x\ Fitness \end{cases} \quad (12)$$

In Eq. (12), d_i represents a threshold, *tit* is the maximal iteration number, *it* is the current iteration number and rand ( ) is a random number uniformly distributed in U (0, 1).

b) **The multi-swarm scheduler request rule**: The multi-swarm scheduler monitors each subswarm, and sends a request in order to obtain a result from the current sub-swarm when the current sub-swarm is valuable. If sub-swarm has sent the swarm request rules more than $k \times n$ times, where $k = 3$, $n = 1, 2, 3, ..., 100$, the multi-swarm scheduler will send the r1ule. The multi-swarm scheduler request rule is touched off, according to evaluating the activity level of the current sub-swarm.

c) The multi-swarm collection rule: The multi-swarm scheduler collects results from the alive sub-swarm and updates *pbest* and *gbest* from storage table.

d) The multi-swarm destroying rule:

- If the swarm sends the swarm request rule *k* times and $k < fi$ according to Eq. (13), then the multi-swarm scheduler destroys the current sub-swarm.
- If the swarm does not change the *gbest* in p_n iterations, then the multi-swarm scheduler destroys the current sub-swarm. We set p_n in the initialization of PSO.

$$f_i = \frac{\sum_{l=1}^{n} ite(l) X\ m}{pl} \quad (13)$$

In Eq. (13), *ite*( ) is the function for calculating how many times the sub-swarm sends swarms request rule, *m* a threshold, *pl* the alive sub-swarm size, *l* is the swarm.

3.5. MPSO algorithm

Step 1 : initialize the PSO with the object format of data set from the text file . The positions xij and velocities vij of each swarm particle with random values.
Step 2: calculate the objective function and update pbest (local best) and gbest (global best) of each swarm from the table.
Step 3: set the lower and upper bounds of the velocity, the size of particles, the number of iterations.initially Set iteration number = 0, current particle number = 1, *titi* = size of particles, and *iti* = current particle number.
Step 4: execute multi-swarm collection rule.
Step 5: In each swarm, if the current particle number < particle size, go to Step 6, otherwise, go to Step 10.
Step 6: In each swarm, get *gbest* and *pbest* from the table and each particle updates its position and velocity.Go to Step7.
Step 7: Restrict position and velocity of each individual.Go to Step 8.
Step 8: Each particle calculates its fitness and updates *pbest* and *gbest*. Execute swarm request rule, and go to Step 9. If the current swarm needs to be destroyed according to multi-swarm destroying rule, dispose the current swarm, and exit.
Step 9: current particle number = current particle number + 1. Go to Step 4.
Step 10: current iteration number = current iteration number + 1. Go to Step 3.
Step 11: Execute multi-swarm collection rule, and exit.

Based on the iterative search and evaluation procedure 11 sets of important features are selected in the selection process.

4. LAYERED APPROACH

The layer-based intrusion detection system (LIDS) proposed by Gupta et al. [50] represents a sequential layered approach was developed for ensuring availability, confidentiality, and integrity of data and (or) services over a network. In this approach, the algorithm uses the selected features and check whether there is a probe attack in the first layer called probe layer. Similarly, at each layer, it checks for the occurrence of the corresponding attacks. If there is an attack, it informs the prevention system. The main advantage of the layered approach is that it reduces the computation time by using separate feature selected by CRF-based feature selection algorithm. They are Probe layer, DoS layer, R2L layer, and U2R layer. Each layer is then separately trained with a small set of relevant features. Layers behave like filters that block any anomalous connection, thereby eliminating the need of further processing at subsequent layers enabling quick response to intrusion. It is implemented with a small set of features for every layer rather than using all the 41 features. So the system is being with significant performance improvement during both the training and the testing of the system. In many situations, there is a trade-off between efficiency and accuracy of the system and there can be various avenues to improve system performance.

5. INTEGRATING LAYERED APPROACH WITH MPSO

The Integrated proposed system is given in the figure 3

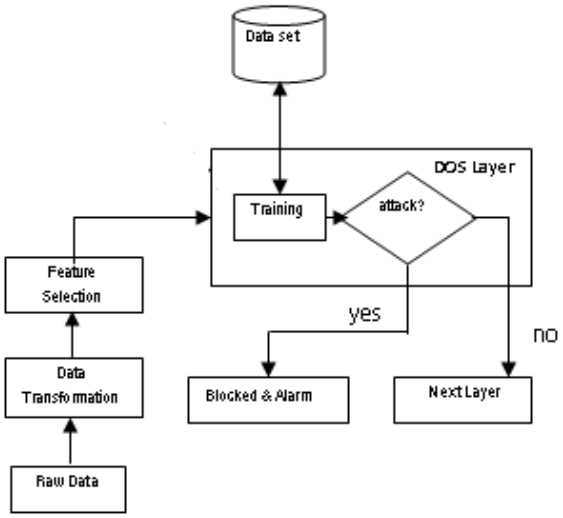

Figure 3. Architecture of proposed model

The proposed system is organized as four layers which can be specially designed to identify the respective class of attack. Each layer is composed with the training phase to identify the attack. The training of the data is done using BPN approach, Our proposed system can label the attack as well as the system is trained with the new dataset regularly. The important advantage of our proposed work is challenged with identifying new attacks without any difficulty .Once the training process is repeated and identification is performed. This approach has the great advantage of reducing false alarm and increase the detection rate. The effect of such a sequence of layers is that the anomalous events are identified and blocked as soon as they are detected.

6. EXPERIMENTS

The raw packets are collected using tool SNORT. The collected features are continuous and discrete stored in the text file. The next process is the conversion of continuous attribute to discrete using the eq .(7). Here the normalized features are taken as input for the selection process the features are selected with the help of the F Score eq.(9). The next important step is applying the objective function of the selected feature in the previous step. The objective function is given in eq, (11). Finally, the algorithm gives the output of less number of selected features by evaluating the fitness function and the swarm rule given in eq.(13). Feature selection is done automatically by MSPSO algorithm given in section 3. The algorithm works on the position and velocity of the particle and updated as given in eq (1).As the result of feature selection only important 11 features have been selected and fed in- to the

integrated, layered approach as discussed in section 4 in which each layer is separately trained to detect a specific type of attack category. This is a collaborative intrusion detection model. The benefit of this approach is the system can detect any type of attack includes the transmission control protocol (TCP) attacks, UDP attacks, ICMP attacks, and content-based attacks. When the classification takes place if the system finds difficulty to identify the label to mark, then the rule generation process is executed. Once the new rule is generated, which is automatically updated in the rule pool. Also the prescribed data set used for labelling is automatically updated for the next time training. Because of the updating capability, the new proposed system can identify any new type of attack without risk.

In the successive classification process, the computation time is so much reduced. The experiment is repeated for 125 times, it is observed the time taken to detect and report is gradually decreased. The first layer is designed to identify DOS attack, which is stopping the service .The Second layer is the probe attack identifier, which is based on connection level features such as the "duration of connection" and "source bytes" are significant while features like "number of files creations" and "number of files accessed". The Third layer is the identifier of R2L attacks are one of the most difficult to detect as they involve the network level and the host level features. So both the network level features such as the "duration of connection" and "service requested" and the host level features such as the "number of failed login attempts" are selected. The U2R attacks involve the semantic details that are very difficult to capture at an early stage. Such attacks are often content based and target an application. The features such as "number of file creations" and "number of shells prompt invoked," are selected in the fourth layer. The most significant benefit of this layered approach is reduced time taken for training. In each layer the features are identified and labelled. If the feature is identified an attack, it is blocked and discarded ,then by prevented to pass to the next layer. The features available at the output of the fourth layer are not attacked. As soon as an attack is detected, the system raises the alarm or gives alert about the attack. This method has increased the detection speed and accuracy

Table 1: Dataset taken from experiment

	Training set	Test set
Normal	97,277	65,500
Probe	4,107	4,500
DoS	391,458	229,813
R2L	1,126	16,391
U2R	52	78
Total	494,020	316282

The experiment is repeated several times to find the detection rate for one class Of the KDD dataset (R) , as follows :

$$R = \alpha S \; X \; \beta S \; X \; classes$$

In this study, the length of set αS is 5 and set βS is 5.

Table 2. Detection rate

	DOS	**Probe**	**U2R**	**R2L**
Multi-SVM	96.8	75	5.3	4.2
PN rule	96.9	73.2	6.6	10.7
Layered CRF	97.4	98.6	86.3	29.6
PSO with SVM	97.9	98.6	68.9	19.5
Layered with MPSO	98.0	98.9	88.0	32.70

Table 3. False Alarm rate

	DoS	Probe	U2R	R2L
Layered MPSO	**0.03**	**2.0**	**0.025**	**0.19**
Multi-SVM	0.1	11.7	47.8	35.4
PN rule	0.05	7.5	89.5	12.0
Layered CRF	0.07	0.91	0.05	0.35
PSO with SVM	0.07	3.1	0.05	0.35

The number of classes is 5 and these are : Normal , DOS, Probe ,R2L,U2R. The experiment is repeated 125 times. It is analyzed from the table 2 and table 3 the new proposed system has a high detection rate and reduced false alarm rate compared with other existing system. The figure 4 and figure 5 shows the advantage of the proposed method.

CONCLUSION

In this study, MPSO algorithm with discretization is proposed can deal with discrete and continuous attribute at the same time and keep the completeness of data information. An efficient objective function of which is designed by taking into consideration classification accuracy rate and F-score. Layered based MPSO model performs the classification tasks using the optimal parameter values and the subset of features. From the result, it is observed that the proposed approach performs significantly better over other methods in terms of the classification and accuracy rates. The experiment results show that for misuse detection and anomaly detection, the proposed method can provide high detection rate and low false positive rate, which is two important criteria for security systems.The results obtained showed the adequacy of the proposed network IDS by reducing the number of features from 41 to 11, which leads to high detection accuracy (98%) and speed up the time to 0.15 sec.

REFERENCES

[1]. Overview of Attack Trends, attack_trends.pdf, 2002.

[2]. K.K. Gupta, B. Nath, R. Kotagiri, and A. Kazi, "Attacking Confidentiality: An Agent Based Approach," Proc. IEEE Int'l Conf. Intelligence and Security Informatics (ISI '06), vol. 3975, pp. 285-296, 2006.

[3]. J Bartlett, Machine Learning for Network Intrusion Detection, 2009.

[4]. J.P. Anderson, Computer Security Threat Monitoring and Surveillance,2010.

[5]. R. Bace and P. Mell, Intrusion Detection Systems, Computer Security Division, Information Technology Laboratory, Nat'l Inst. of Standards and Technology, 2001.

[6]. R. Bace and P. Mell, Intrusion Detection Systems, Computer Security Division, Information Technology Laboratory, Nat'l Inst. of Standards and Technology, 2001.

[7]. K.K. Gupta, B. Nath, and R. Kotagiri, "Network Security Framework," Int'l J. Computer Science and Network Security, vol. 6, no. 7B, pp. 151-157, 2006.

[8]. W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," Proc. Seventh USENIX Security Symp. (Security '98), pp. 79-94, 1998.

[9]. W. Lee, S. Stolfo, and K. Mok, "Mining Audit Data to Build Intrusion Detection Models," Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD '98), pp. 66-72, 1998.

[10]. W. Lee, S. Stolfo, and K. Mok, "A Data Mining Framework for Building Intrusion Detection Model," Proc. IEEE Symp. Security and Privacy (SP '99), pp. 120-132, 1999.

[11]. R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM SIGMOD, vol. 22, no. 2, pp. 207-216, 1993.

[12]. T. Abraham, IDDM: Intrusion Detection Using Data Mining Techniques, 2008.

[13]. L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion Detection with Unlabeled Data Using Clustering," Proc. ACM Workshop Data Mining Applied to Security (DMSA), 2001.

[14]. H. Shah, J. Undercoffer, and A. Joshi, "Fuzzy Clustering for Intrusion Detection," Proc. 12th IEEE Int'l Conf. Fuzzy Systems (FUZZ-IEEE '03), vol. 2, pp. 1274-1278, 2003.

[15]. N.B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC '04), pp. 420-424, 2004.

[16]. C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Bayesian Event Classification for Intrusion Detection," Proc. 19th Ann. Computer Security Applications Conf. (ACSAC '03), pp. 14-23, 2003.

[17]. S. Forrest, S.A. Hofmeyr, A. Somayaji, and T.A. Longstaff, "A Sense of Self for Unix Processes," Proc. IEEE Symp Research in Security and Privacy (RSP '96), pp. 120-128, 1996.

[18]. Y. Du, H. Wang, and Y. Pang, "A Hidden Markov Models-Based Anomaly Intrusion Detection Method," Proc. Fifth World Congress on Intelligent Control and Automation (WCICA '04), vol. 5, pp. 4348-4351, 2004.

[19]. W. Wang, X.H. Guan, and X.L. Zhang, "Modeling Program Behaviors by Hidden Markov Models for Intrusion Detection," Proc. Int'l Conf. Machine Learning and Cybernetics (ICMLC '04), vol. 5, pp. 2830-2835, 2004.

[20]. C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting Intrusions Using System Calls: Alternative Data Models," Proc. IEEE Symp. Security and Privacy (SP '99), pp. 133-145, 1999.

[21]. J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proc. 18th Int'l Conf. Machine Learning (ICML '01), pp. 282-289, 2001.

[22]. N.B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC '04), pp. 420-424, 2004.

[23]. H. Debar, M. Becke, and D. Siboni, "A Neural Network Component for an Intrusion Detection System," Proc. IEEE Symp. Research in Security and Privacy (RSP '92), pp. 240-250, 1992.

[24]. Z. Zhang, J. Li, C.N. Manikopoulos, J. Jorgenson, and J. Ucles, "HIDE: A Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification," Proc. IEEE Workshop Information Assurance and Security (IAW '01),pp. 85-90, 2001.

[25]. D.S. Kim and J.S. Park, "Network-Based Intrusion Detection with Support Vector Machines," Proc. Information Networking, NetworkingTechnologies for Enhanced Internet Services Int'l Conf. (ICOIN '03), pp. 747-756, 2003.

[26]. Autonomous Agents for Intrusion Detection, http://www.cerias. purdue.edu/research/aafid/, 2010.

[27]. Probabilistic Agent Based Intrusion Detection, http://www.cse.sc. edu/research/isl/agentIDS.shtml, 2010.

[28]. Y.-S. Wu, B. Foo, Y. Mei, and S. Bagchi, "Collaborative Intrusion Detection System (CIDS): A Framework for Accurate and Efficient IDS," Proc. 19th Ann. Computer Security Applications Conf. (ACSAC '03), pp. 234-244, 2003.

[29]. L. Ertoz, A. Lazarevic, E. Eilertson, P.-N. Tan, P. Dokas, V. Kumar, and J. Srivastava, "Protecting against Cyber Threats in Networked Information Systems," Proc. SPIE Battlespace Digitization and Network Centric Systems III, pp. 51-56, 2003.

[30]. E. Tombini, H. Debar, L. Me, and M. Ducasse, "A Serial Combination of Anomaly and Misuse IDSes Applied to HTTP Traffic," Proc. 20th Ann. Computer Security Applications Conf. (ACSAC '04), pp. 428-437, 2004.

[31]. D. Boughaci, H. Drias, A. Bendib, Y. Bouznit, and B. Benhamou, "Distributed Intrusion Detection Framework Based on Mobile Agents," Proc. Int'l Conf. Dependability of Computer Systems (DepCoS-RELCOMEX '06), pp. 248-255, 2006.

[32]. Y. Gu, A. McCallum, and D. Towsley, "Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation," Proc. Internet Measurement Conf. (IMC '05), pp. 345-350, USENIX Assoc., 2005.

[33]. S. Dzeroski and B. Zenko, "Is Combining Classifiers Better than Selecting the Best One," Proc. 19th Int'l Conf. Machine Learning (ICML '02), pp. 123-129, 2002.

[34]. C. Ji and S. Ma, "Combinations of Weak Classifiers," IEEE Trans. Neural Networks, vol. 8, no. 1, pp. 32-42, 1997.

[35]. K.K. Gupta, B. Nath, and R. Kotagir i, "Conditional Random Fieldsfor Intrusion Detection," Proc. 21st Int'l Conf. Advanced Information Networking and Applications Workshops (AINAW '07), pp. 203-208, 2007.

[36]. T. S. Chou, K. K. Yen, and J. Luo Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms, International Journal of Computational Intelligence Volume 4 Number 3.

[37]. G. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," in Proceedings ML-94, pp. 121-129, Morgan Kaufmann, 1994.

[38]. K. Kira and L. A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," in Proceedings AAAI-92, pp. 129-134, MIT Press, 1992.

[39]. H. Almuallim and T. G. Dietterich, "Learning with Many Irrelevant Features," in Proceedings AAAI-91, pp. 547-551, MIT Press, 1991.

[40]. G. Qu, S. Hariri, and M. Yousif, "A New Dependency and Correlation Analysis for Features," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 9, pp. 1199-1207, September 2005.

[41]. J. R Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, pp. 81-106, 1986.

[42]. J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[43]. G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision Tree Classifier For Network Intrusion Detection With GA-based Feature Selection," in Proceedings of the 43rd ACM Southeast Conference, Kennesaw, GA, March 2005.

[44]. S. Mukkamala and A. H. Sung, "Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines", Journal of the Transportation Research Board of the National Academics, Transportation Research Record No 1822, pp. 33-39, 2003.

[45]. ang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley & Sons, Chichester (2010)

[46]. Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. Int. J.Math. Modelling Num. Optimisation 1(4), 330–343 (2010)

[47]. Yuanning Liu1,2, Gang Wang1,2, Huiling Chen1,2, Hao Dong1,2, Xiaodong Zhu1,2, Sujing Wang1,2 An Improved Particle Swarm Optimization for Feature Selection. Journal of Bionic Engineering 8 (2011)

[48]. Particle Swarm Optimization for Feature SelectioninClassification:AMulti-Objective approach**,**ieeexplore.ieee.org (Volume:43 ,Issue: 6)

[49]. Essam Al Daoud, Intrusion Detection Using a New Particle Swarm Method and Support Vector Machines, World Academy of Science, Engineering and Technology Vol:77 2013-05-27

[50]. KK Gupta, B Nath, R Kotagiri, Layered Approach using Conditional Random Fields for Intrusion Detection. IEEE Trans. Dependable Secure Comput 7, 1 (2010)