Evolutionary Approach for Network Anomaly Detection Using Effective Classification

[1]A.Chandrasekar,
Research Scholar,
Anna University,

[2] V. Vasudevan,
Professor,
A.K. College of Engineering,

[3] P. Yogesh,
Assistant Professor,
Anna University,

Summary
Intrusion Detection Systems (IDS) have become a necessary component of the computer and information security framework. Due to the increase in unauthorized access and stealing of internet resources, internet security has become a very significant issue. Network anomalies in particular can cause many potential problems. This work discusses about the ways of implementing Evolutionary Approach for Network based Anomaly Detection Systems using Effective Classification. It involves characterizing the network traffic and detecting intrusion through observation of deviation from normal behavior patterns. This work aims at providing a potential solution to the problem of Network intrusion by using effective classification technique Support Vector Machines, Evolutionary approaches namely genetic algorithm(GA) and Particle Swarm Optimization (PSO). These evolutionary approaches are used for feature selection and SVM is used for classification. We tested this technique using KDD Cup99 dataset, and analyzed its performance. The experimental results show that the PSO-SVM is an effective approach in network intrusion detection.

Key words:
IDS, PSO, PSO-SVM, Anomaly detection

1. Introduction

With the development of network and information technology, network is becoming increasingly important in politics, economy, military affairs and daily life. But we can't turn a blind eye to a fact that more and more network attacks have seriously threatened our networks. It includes attempting to destabilize the network, gaining unauthorized access to files with privileges, or mishandling and misusing of software. The intrusion detection is to automatically scan network activity and detect attacks. In order to protect the network and information, intrusion detection is applied to network. Intrusion Detection is a kind of security technology to protect the network against the intrusion attacks. Intrusion Detection means "the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions. It is also defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network".

There have been many techniques for modeling anomalous and normal behaviors for intrusion detection. The signature-based and supervised anomaly detections are widely deployed and commercially available. The signature-based detection extracts features from the network data. It detects intrusions by comparing the feature values to a set of attack signatures provided by human experts. However, it can only detect previously known intrusions with a signature. The signature database has to be manually revised for each new type of discovered attacks. On the other hand, the supervised anomaly detection trains models on labeled data (i.e., data pre-classified as an attack or not) and checks how well new data fit into the model. Obviously, it cannot be quickly adapted to new types of intrusion and do not have enough labeled data available. In general, a very large amount of network data needs to be handled and classified. Hence, it is impractical to classify them manually.

Intrusion detection techniques can be categorized in misuse detection and anomaly detection. Misuse detection systems find intrusions by matching sample data to known intrusive pattern. Anomaly detection systems find intrusion by analyzing the deviation from normal activities profiles that are retrieved from historical data. Intrusion detection is a critical component of secure information systems.

Many Support Vector Machines (SVM) have been successfully applied to gene expression data classification problems. Since they are not negatively affected by high dimensionality; hence they can obtain a higher accuracy than a general classification methods, optimize the obtained support vector machine. This avoids a common disadvantage of general classification methods, namely the long operation time, and can reduce training errors of the SVMs.

In this paper, Genetic algorithm (GA) and Particle Swarm Optimization (PSO) are used to implement feature

Manuscript received January 5, 2009
Manuscript revised January 20, 2009

selection and Support Vector Machine is used for Classification.

2. Literature Survey

There are many methods that have been applied to intrusion detection such as wavelet analysis [3], fuzzy data mining [4] and intelligent Bayesian classifier [5]. But many experiments [6], [7], [8] show that there is a high detection accuracy when Support Vector Machine (SVM) is used in intrusion detection. But it is very critical to select features in intrusion detection based on SVM. In a classification problem, the number of features can be quite large, many of which can be irrelevant or redundant. Since the amount of audit data that an IDS needs to examine is very large even for a small network, classification by hand is impossible. Feature reduction and feature selection improves classification by searching for the subset of features, which best classifies the training data. Most intrusion occurs via network using the network protocols to attack their targets. For example, during a certain intrusion, a hacker follows fixed steps to achieve his intention. First he sets up a connection between a source IP address to a target IP, and sends data to attack the target.

Generally, there are four categories of attacks. They are: 1) DoS (denial-of-service), for example ping-of-death, teardrop, smurf, SYN flood. 2) R2L (Remote to Local): unauthorized access from a remote machine, for example guessing password. 3) U2R (User to Root): unauthorized access to local super user (root) privileges, for example, various "buffer overflow" attacks, 4) PROBING: surveillance and other probing, for example, port-scan, ping-sweep, etc. Some of the attacks (such as DoS, and PROBING) may use hundreds of network packets or connections, while on the other hand attacks like U2R and R2L typically use only one or a few connections.

We mainly study intrusion detection based on SVM, and use GA and PSO to select and optimize features at the same time.

3. System Architecture

The figure 1 illustrates the over all structure of the system. Network Intrusion Detection is a kind of security technology to protect the network against the intrusion attacks. It includes two main categories. One is misuse detection and the other is anomaly detection. Misuse detection can exactly detect the known attacks, but it can do nothing against the unknown attacks.

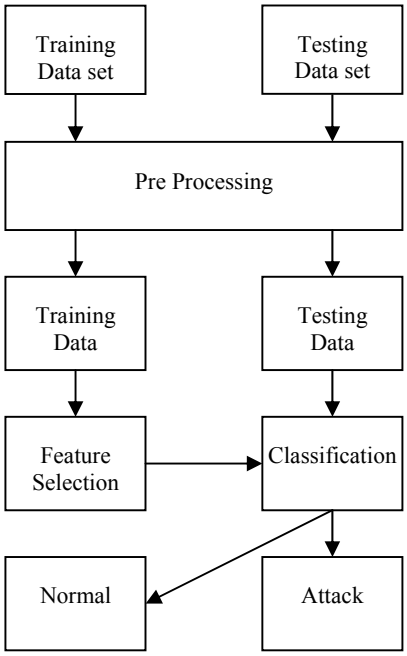

Figure 1. Architecture Diagram

However, anomaly detection can detect the unknown and new attacks. So it is an all-round method to detect attacks. SVM is based on the structural risk minimization principle from the statistical learning theory. Its kernel is to control the empirical risk and classification capacity in order to maximize the margin between the classes and minimize the true costs. A support vector machine finds an optimal separating hyper-plane between members and non-members of a given class in a high dimension feature space. The processes of intrusion detection based on SVM are depicted as follows:

Capturing network data: We set the Network Interface Card (NIC) on promiscuous mode, and use libpcap or tcpdump to collect the data stream of a given network.

Data preprocessing: The data applied to SVM have different types. Some are symbolic such as user's command, and some are numeric such as the number of connections. It is necessary to preprocess the data and transform them to the same format. This format must be recognized and dealt with by SVM.

SVM training and test: We first train the selected data with SVM. Then, we can get a set of support vectors after training and put them into support vector database (SVB). Finally, the support vectors in SVB are used to detect the actual network events and the decisions are made.

*Responding to network events:*If the intrusion detection system (IDS) detects the attack, it will adopt some measures to hold them back such as giving an alert, cutting off connections and so on.

A. Feature Selection and optimization using GA

GA is an adaptive method of global-optimization searching and simulates the behavior of the evolution process in nature. It maps the searching space into a genetic space. That is, every possible key is encoded into a vector called a chromosome. One element of the vector represents a gene. All of the chromosomes make up of a population and are estimated according to the fitness function.

A fitness value will be used to measure the "fitness" of a chromosome. Initial populations in the genetic process are randomly created. GA then uses three operators to produce a next generation from the current generation: reproduction, crossover and mutation. GA eliminates the chromosomes of low fitness and keeps the ones of high fitness. Thus more chromosomes of high fitness move to the next generation. This whole process is repeated until a good chromosome (individual) is found. The figure 2 illustrates the feature selection using genetic algorithm.

Encoding is the first step in GA. For a data record, we convert each value of its feature into a bit binary gene value, 0 or 1. In our experiments, we choose the subsets of KDD Cup99, 1999 kddcup.data_10_percent and corrected [9], as the training dataset and test dataset. Because the values of feature No.2 (Protocol type), No.3 (Service), and No.4 (Flag) are all symbols, it is not necessary to optimize these three features. It is just all right to eliminate them before encoding and add them to the record after optimization. For feature that has a numeric value, other than 0, we convert it to 1; For example, a record (0, tcp, http, SF, 181, 5450, 0, 0, 0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00, 0.00,9,9, 1.00,0.00,0.11,0.00,0.00,0.00,0.0,0.00.) can be converted to (01100000100000000011000010 01110100000) after encoding.

$$F(X) = A(X) + \beta N0 \text{ -------- } (1)$$

$$\text{Where } A = (\alpha 1, \alpha 2 \ldots \alpha n)$$

$$X = (x1, x2 \ldots xn)T$$

$$\alpha n = Nn/Nall \text{ --------- } (2)$$

We adopt the value of fitness function to decide whether a chromosome is good or not in a population. The equation (1) is used to evaluate the fitness function of every chromosome. In this equation N0 is the number of 0 in a chromosome, β is the coefficient, xn means the nth gene (0 or 1), and αn means the weight of the nth gene. We use equation (2) to calculate the weight.

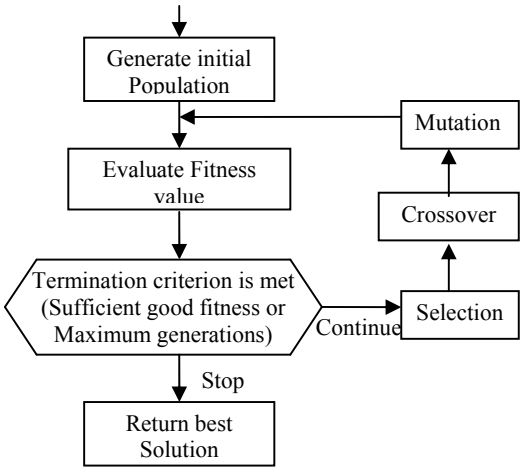

Figure 2. Feature Selection using GA

B. Feature selection and optimization using particle swarm optimization

PSO is a new branch in evolutionary algorithms, which were inspired in group dynamics and its synergy and were originated from computer simulations of the coordinated motion in flocks of birds or schools of fish. As these animals wander through a three-dimensional space, searching for food or evading predators, these algorithms make use of particles moving in an n-dimension space to search for solutions for an n-variable function optimization problem. In PSO, individuals are called particles and the population is called a swarm. [10]

The initial swarm is generally created in such a way that the population of the particles is distributed randomly over the search space. At each iteration, each particle is updated by following two "best" values, called *pbest* and *gbest*. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) the particle has achieved so far. This fitness value is stored, and called *pbest*. When a particle takes the whole population as its topological neighbor, the best value is a global "best" value and is called *gbest*. The figure 3 and figure 4 illustrate the flow graph and pseudo code of the PSO procedure is given below.

Based on the rules of particle swarm optimization, we set the required particle number first, and then the initial coding alphabetic string for each particle is randomly produced, in our case we coded each particle to imitate a chromosome in a genetic algorithm. Each particle was coded to a binary alphabetic string $S=F_1\ F_2\ K\ F_n$, n=1, 2,...m; the bit value {1} represents a selected feature, whereas the bit value {0} represents a non-selected feature.

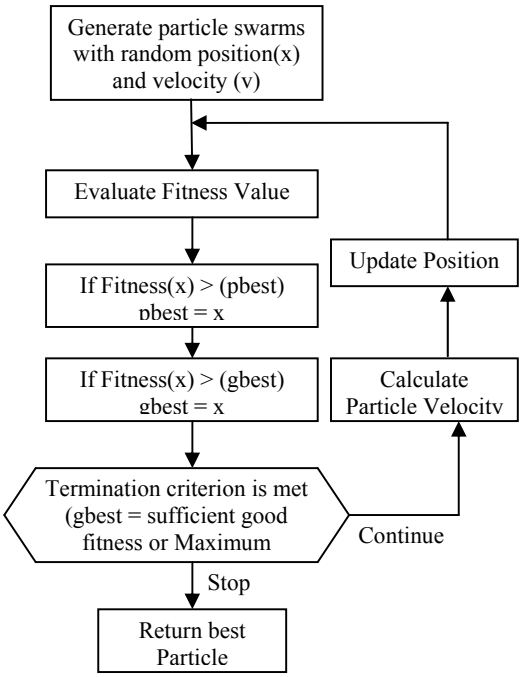

Figure 3. Feature Selection using PSO

The adaptive functional values were data based on the particle features representing the feature dimension; this data was classified by a support vector machine (SVM) to obtain classification accuracy; the SVM serves as an evaluator of the PSO fitness function. v_{pd}^{new} and v_{pd}^{old} are the particle velocities, x_{pd}^{old} is the current particle position (solution), and x_{pd}^{new} is updated particle position (solution). The values $pbest_{pd}$ and $gbest_d$ are defined as stated above. The two factors rand1, and rand2 are random numbers between (0, 1), where cl and c2 are acceleration factors, usually cl=c2=2. Particle velocities of each dimension are tried to a maximum velocity v_{max}. If the sum of velocities causes the total velocity of that dimension to exceed Vmax, Vmax is a user-specified parameter

Initialize population
While (number of iterations, or the stopping Criterion is not met)
For p=I to number of particles
Segment training data and testing data
Initialize super parameter α
$k(x_i, y_j) = exp^{-r||xi - yj||xp}$
While (number of iterations, or the stopping criterion is not met)
For i=1 to number of training data

$$zi = \sum_{j=1}^{n} \alpha_j y_j k(x_i, y_j)$$

$\delta\alpha_i = \eta(1 - z_i y_j)$
if $(\alpha_i + \delta\alpha_i) \leq 0$ *then* $\alpha_i = 0$
if $(\alpha_i + \delta\alpha_i) > 0$ *then* $\alpha_i = (\alpha_i + \delta\alpha_i)$
Next i
Next iteration until criterion
For i=1 to number of testing data

$$zi = \sum_{j=1}^{n} \alpha_j y_j k(x_i, y_j)$$

If $z_i > 0$ *then* $class_i = +1$ *else* $class_i = -1$
If $class_i$ = *real class of testing data then*
right = right + 1
Next i
$fitness_p$ = *right/ number of testing data*
if the fitness of xp is greater than the fitness of $pbest_p$
then Update pbestp = x_p
for k Є Neighborhood of x_p
if the fitness of x_k *is greater than that of gbest*
Update gbest = xk
Next k
For each dimension d

$$v_{pd}^{new} = w \times v_{pd}^{old} + c1 \times rand1 \times (pbest_{pd} - x_{pd}^{old}) + c2 \times rand2 \times (gbest_d - x_{pd}^{old})$$

$$S(v_{pd}^{new}) = 1/(1+e^{-v_{pd}^{new}})$$

If $(rand < S(v_{pd}^{new}))$ *then* $x_{pd}^{new} = 1$; *else* $x_{pd}^{new} = 0$

Next d
Next p
Next generation until stopping criterion

Figure 4. Pseudo code for Feature Selection using PSO

For example, when a 10-dimensional data set (n=10) Sn= ($F_iF_2F_3F_4F_5F_6F_7F_8F_9F_{10}$) is analyzed using particle swarm optimization to select features, we can select any number of features smaller than n, i.e. we can chose a random 6 features, here Sn = (F F3F5F7F9EF7O). When the adaptive value is calculated, these 6 features in each data set represent the data dimension and are evaluated by the SVM. The fitness value for the SVM evolves according to the K-fold Cross-Validation Method [11] for small sample sizes, and according to the Holdout Method [11] for big sample sizes. Using the K-Fold Cross-Validation Method, we separated the data into 10 parts {D_1, D_2,…D_{10}}, and carried out training and testing a total of 10 times. If ever part Dn, n=1, 2,…,10 is processed as a test set, the other 9 parts will be training sets. Following 10 times of training and testing, 10 classification accuracies are produced, and the averages of these 10 accuracies are used as the classification accuracy for the data set. When the Holdout

Method is used, the data can be divided into two parts, a training set part, which contains a larger amount of data, and a test set part, which contains relatively fewer data. We assumed that the obtained classification accuracy is an adaptive functional value.

Each particle renewal is based on its adaptive value. The best adaptive value for each particle renewal is *pbest*, and the best adaptive value within a group of *pbest* is *gbest*. Once *pbest* and *gbest* are obtained, we can keep track of the features of *pbest* and *gbest* particles with regard to their position and speed. In this study, a binary version of a PSO algorithm is used for particle swarm optimization [12]. The position of each particle is given in a binary string that represents the feature selection situation. Each particle is updated according to the following equations.

$$v_{pd}^{new} = w \times v_{pd}^{old} + c1 \times rand1 \times (pbest_{pd} - x_{pd}^{old}) + c2 \times rand2 \times (gbest_{d} - x_{pd}^{old}) \quad (3)$$

$$S(v_{pd}^{new}) = 1/(1+e^{-v_{pd}^{new}}) \quad (4)$$

If $(rand < S(v_{pd}^{new}))$ then $x_{pd}^{new}=1$; else $x_{pd}^{new}=0$

The feature after renewal is calculated by the function $S(v_{pd}^{new})$, in which the speed value is v_{pd}^{new}. If $S(v_{pd}^{new})$ is larger than a randomly produced disorder number that is within (0, 1), then its position value Fn n=1, 2, …,m is represented as {1} (meaning this feature is selected as a required feature for the next renewal). If $S(v_{pd}^{new})$ is smaller than a randomly produced disorder number that is within {0-1}, then its position value Fn, n=l, 2, …, m is represented as {0} (meaning this feature is not selected as a required feature for the next renewal).

C. Support Vector Machine

Support Vector Machine (SVM) is based on the structural risk minimization principle from the statistical learning theory. Its kernel is to control the empirical risk and classification capacity in order to maximize the margin between the classes and minimize the true costs [8]. A support vector machine finds an optimal separating hyper-plane between members and non-members of a given class in a high dimension feature space [6]. Although the dimension of feature space is very large, it still shows good generalization performances. The basic SVM theory is as follows.

First, we are given a set of training examples *S=((X , y),.....(X , y))*, *l = 1, 2,.....n* , $X_n \in R^n$, and $y \in \{+1,-1\}$ where *X* is the input data and *y* is output. If *y* is "1", it means the input example is normal. If *y* is "-1", it means the input example is abnormal. Suppose this set can be separated by a hyper-plane $W X + b = 0$. That is, all the training examples satisfy:

$$y_i(\langle w.x_i \rangle + b) \geq 1, \text{ for all } i=1\ldots\ldots l \quad (5)$$

W is an adjustable weight vector, and *b* is the bias term. In Figure.5, the margin between two hyper-planes

$$H_1: w.x_1+b=1 \text{ and } H_2: w.x_1+b=-1 \text{ is } 2/\|w\| \quad (6)$$

And the hyper-plane that maximizes the margin is the optimal separating hyper-plane. Thus, the optimization is now a convex quadratic programming problem.

$$\underset{w,b}{\text{Minimize}} \quad \Phi(w) = (1/2)\|w\| \quad (7)$$

Subject to $y_i(\langle w.x_i \rangle + b) \geq 1$, for all i=1……l

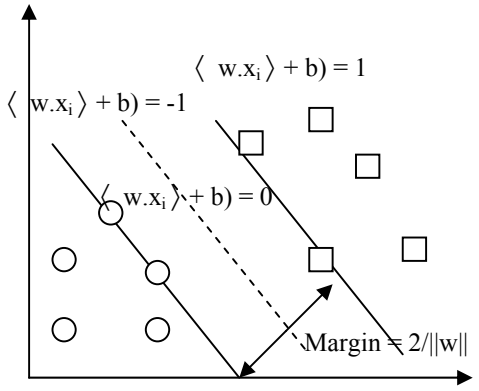

Figure 5. separating hyper-planebetween two classes

Where α_i is Lagrange multiplier. When the set is non-linearly separable, $K(X_i, X)$ is kernel function, and it must satisfy the Mercer condition. When the set is linearly separable, $K(X_i,X)$ means inner product $\langle X_i.X \rangle$.

4. Implementation and Results

In this experiment, we use a standard dataset the raw data used by the KDD Cup 1999 intrusion detection contest [9]. This database includes a wide variety of intrusions simulated in a military network environment that is a common benchmark for evaluation of intrusion detection techniques. In general, the distribution of attacks is dominated by probes and denial-of-service attacks. The data set has 41 attributes for each connection record plus one class label. There are 24 attack types, but we treat all of them as an attack group. A data set of size N is processed. The nominal attributes are converted into linear discrete values (integers).

We ran our experiments on a system with a 1.6 GHz Pentium M processor and 1024 MB DDR2 RAM running Windows XP. All the preprocessing was done using MATLAB.

A. Preprocessing

Preprocessing consists of two steps. The first step involved mapping symbolic-valued attributes to numeric-valued attributes and the second step implemented non-zero numerical features. We reduce the dimensionality of this data set from 41 to 10 attributes are duration, service, src bytes, dst byte, count, srv_count, serror rate, dst_host_srv count, dst host diff srv rate, and dst host same srcport rate.

B. Feature Selection

Optimization theory has been used to define the necessity of features. Feature selection is an optimization process in which one tries to find the best feature subset, from the fixed set of the original features, according to a given processing goal and a feature selection criterion. A pattern's features, from the point of view of processing goal and type, may be irrelevant (having no effect on processing performance) or relevant (having an impact on processing performance). Features can be redundant (correlated, dependent)]. When we process volumes of data, it is necessary to reduce the large number of features to a smaller set of features. There are 42 fields in each data record and it is hard to determine which fields are useful or which fields are trivial. GA and PSO allow us to determine (for a discrete attribute data set) a set called a core, containing strongly relevant features, and reducts, containing core plus additional weakly relevant features, such that each reduct is satisfactory to determine concepts in the data set. Based on a set of reducts for a data set some criteria for feature selection can be formed, for example a selecting feature from a reduct containing the minimal set of attributes

C. Performance measures

Standard measures for evaluating IDSs include detection rate, false alarm rate, trade-off between detection rate and false alarm rate, performance (Processing speed + propagation + reaction), and Fault Tolerance (resistance to attacks, recovery, and subversion). Detection rate is computed as the ratio between the number of correctly detected attacks and the total number of attacks, while false alarm (false positive) rate is computed as the ratio between the numbers of normal connections that are incorrectly misclassified as attacks. These are good indicators of performance, since they measure what percentage of intrusions the system is able to detect and how many incorrect classifications are made in the process.

Table 1: Performance of GA-SVM

Class Type	No. of Records	Detection Rate	False Alarm rate
Normal	5000	98.22%	1.18%
Probe	2000	97.46%	2.54%
DoS	3000	97.57%	2..43%
U2R	70	46.85%	53.15%
R2L	6000	39.29%	60.71%
Summary	16,070	75.878%	24.002%

Table 2 Performance of PSO-SVM

Class Type	No. of Records	Detection Rate	False Alarm rate
Normal	5000	99.49%	0.51%
Probe	2000	99.35%	0.65%
DoS	3000	99.58%	0.42%
U2R	70	48.64%	51.36%
R2L	6000	42.72%	57.28%
Summary	16,070	77.956%	22.044

5. Conclusions

In this paper, we first apply Genetic Algorithm for feature selection and optimization and apply SVM for Classification. Next, we apply Particle Swarm optimization for feature selection and optimization and apply SVM for Classification. Because, it is very critical to select features in intrusion detection based on SVM. In a classification problem, the number of features can be quite large, many of which can be irrelevant or redundant. Feature selection and Optimization improves classification by searching for the subset of features, which best classifies the training data. The experimental results show that PSO-SVM can achieve good classification accuracy. PSO has the advantage of easy implementation and the ability to converge optimal solution quickly. Therefore, it is effective to apply PSO and SVM to Network intrusion detection.

References

[1] Hua Zhou, Xiangru Meng, Li Zhang, “Application of Support Vector Machine and Genetic Algorithm to Network Intrusion Detection “The Telecommunication Engineering Institute, IEEE 2007

[2] Surat Srinoy, “Intrusion Detection Model Based On Particle Swarm Optimization and Support Vector Machine“,Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2007)

[3] Sanjay Rawat, Challa S. Sastry. Network Intrusion Detection Using Wavelet Analysis. In:G. Das and

V.P.Gulati. Ed. CIT 2004, LNCS 3356, 2004, pp. 224-232.
[4] Jian Guan, Da-xin Liu, Tong Wang. Applications of Fuzzy Data Mining Methods for Intrusion Detection Systems. In: A. Lagana et al. Ed.ICOIN 2004, LNCS 3045, 2004, pp. 706-714.
[5] Andrea Bosin, Nicoletta Dessi, Barabara Pes. Intelligent Bayesian Classifiers in Network Intrusion Detection. In:M.Ali and F.Esposito. Ed.IEA/AIE 2005, LANI 3533, 2005, pp. 445-447.
[6] Dong Seong Kim, Jong Sou Park. Network-based intrusion detection with support vector machines. In: Kahng H-K. Ed. ICOIN 2003, LNCS 2662, 2003, pp. 747-756.
[7] RAO Xian, DONG Chun-xi, YANG Shao-quan. An intrusion detection system based on support vector machine. Journal of Software. 4 (2003), pp. 798-803.
[8] ZHANG Kun, CAO Hong-xin, YAN Han. Application of support vector machines on network abnormal intrusion detection. Application Research of computers. 5 (2006) , pp. 98-100.
[9] KDD Cup99 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup 99.html.
[10] J. Kennedy, R.C. Eberhart, Particle Swarm Optimisation, in: Proceedings of the IEEE, International Conference on Neural Networks, Piscataway, 1995.
[11] Stone, M., "Cross-Validation choice and assessment of statistical predictions," journal of the Royal Statistical Society B, vol. 36, pp. 111-147, 1974.
[12] Kennedy, J., Eberhart, R.C., "A discrete binary version of the particle swarm algorithm", Systems, Man, and Cyernetics, 1997.'Computational Cybernetics and Simulation]., 1997 IEEE international Conference on Volume 5, 12-15 Oct. 1997. pp.4104-4108.

A. ChandraSekar, M.E., Ph.D Scholar, Anna University, Chennai. He has published four International Journals and five International Conferences. Now he is Currently working as Assistant Professor in Computer Science & Engineering Department in St.Joseph's College of Engineering, Chennai. His area of interest includes network security, analysis of algorithms.

V. Vasudevan, Ph.D, Professor & Head of Information Technology, A.K College of Engineering, Srivilliputhur. He has published more than thirty International Journals and many international conferences. His area of interest includes grid computing, image processing, agent technologies, and multicasting and network security.

P. Yogesh, Ph.D, Assistant Professor working in Computer Science & Engineering Department, Anna University, Chennai. He has published more than ten International Journals and many international conferences. His area of interest includes computer networks, mobile computing, multimedia communications, web technologies and artificial intelligence.