Ant Colony Optimization with Classification Algorithms used for Intrusion Detection

[1]Namita Shrivastava [2]Vineet Richariya

[1] Department of Computer Science and Engg.
Lakshmi Naraian College Of Technology, Rgpv
University Bhopal (M.P.), India.
namitas_10@yahoo.co.in

[2]Department of Computer Science and Engg.
Lakshmi Naraian College Of Technology, Rgpv
University Bhopal (M.P.), India.
Vineet_rich@yahoo.com

Abstract

IDS which are increasingly a key part of system defense are used to identify abnormal activities in a computer system. In general, the traditional intrusion detection relies on the extensive knowledge of security experts, in particular, on their familiarity with the computer system to be protected. To reduce this dependence, various data-mining and machine learning techniques have been used in the literature. During recent years, number of attacks on networks has dramatically increased and consequently interest in network intrusion detection has increased among the researchers. In this paper we have used the terms detection rates and false alarm rates to compare the results of Naïve Bayes algorithm and Support Vector Machine algorithm to find out the results for intrusion detections and by using Naïve Bayes method with Ant Colony Optimization technique try to improve the rates for better detection. The proposed algorithm is used for comparative study we have done in this paper on the basis of which we measure the performance and usefulness of particular methods in detecting specific class of attacks. Experimental results performed using the KDD99 benchmark network intrusion detection dataset indicate that it can significantly reduce the number and percentage of false positives and scale up the balance detection rates for different types of network intrusions.

Keywords: *Intrusion Detection System, Naïve Bayes, Support Vector Machine, Ant Colony Optimization, KDDCup99 dataset, Detection Rates, False alarm Rates, Fuzzy Logic, Fuzzy If-Then Rules.*

1. Introduction

Intrusion Detection refers to the process of monitoring the system for unauthorized access incidents which can be the violation of the security policy, system use policy, or any other security standards [1]. An Intrusion Detection System (IDS) is software that implements the intrusion detection process. On the other hand, an Intrusion Prevention System (IPS) prevents unauthorized access incidents from being successful. To better protect the system from any attacks, Intrusion Detection and Prevention System (IDPS)[2] which provides a completely automated monitoring services, is deployed on the systems. Most of the IDPS systems log the incident every time an attack on the system is detected and notifies the administration of the system so that all necessary actions can be taken to avoid such incidents again in the future. The administrators of the system can also configure the IDPS to monitor the violations of the end user policies. Figure 1 illustrates where an IDPS is placed and how it functions.

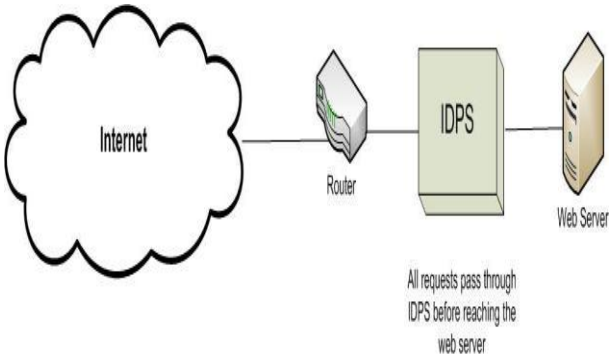

Fig 1: Intrusion Detection and Prevention System

An IDPS is composed of the following components:

Agent: Agent is the module that listens for the events and analyzes the system activities. In case of IDPS used for network attacks, the agent is called "sensor" [1].

Management Server: Management server is responsible for analyzing the received information from the current activity to decide if an attack is in progress. It would use the information from other elements also such as signatures and profiles to complete the analysis.

Database Server: Database server is used to store the information received from the agents and the management server. The management server also uses the database server to complete it operations.

Console: Console or management interface serves as the interface between the IDPS and the administrator. The console is used to monitor the system events produced by the IDPS. Some consoles are also used to configure the agents and perform software upgrades [1].

Figure 2 illustrates all the required components that an IDS is composed of. The models that the IDPS follows to detect attacks can be divided into two categories: Signature-Based Detection Model: It relies on the patterns of the known intrusions in order to match and identify intrusion which is a very efficient detection model due to its low complexity in implementation and detection. Signature recognition techniques are used in most existing intrusion detection systems, commercial or freeware. The drawback of a signature-based detection model is its inability to detect new unknown attacks since the system does not have any signature entry in the system for the new attacks.

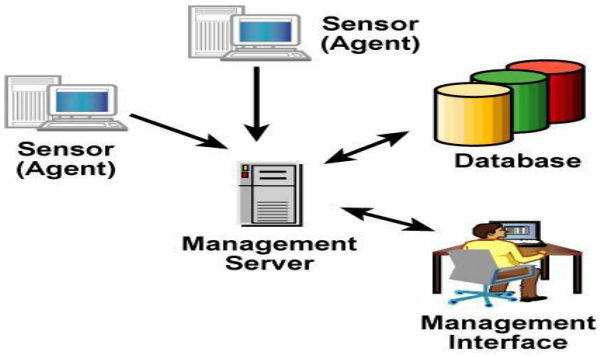

Fig 2: IDPS Components

Anomaly-Based Detection Model: It is used to identify an intrusion when the observed activities in computer systems demonstrate a large deviation from the norm profile built on long-term normal activities and it is able to detect even unknown attacks by comparing the current abnormal events with something that is considered normal.

In such cases, the anomalies may be showing false positives means classifying a normal behavior as an abnormal, and hence as possible attack instances. This discussion points out that the tradeoff between the ability to detect new attacks and the ability to generate a low false alarms rate is the key point to develop an effective IDS. In this, if fuzzy linguistic IF-THEN rules[3] are formulated and a process of fuzzification, inference, and defuzzification leads to the final decision of the system. Although sometimes the fuzzy rules can be directly derived from expert knowledge, different efforts have been made to obtain an improvement on system performance by incorporating learning mechanisms guided by numerical information to define the fuzzy rules. This issue, known as fuzzy rule learning(FRL) which is used to generate results obtained by applying on classification algorithms such as Naïve Bayes & Support Vector Machine in the KDD data set. Several data mining algorithms, such as decision tree, naïve Bayesian classifier, neural network, Support Vector Machines, and fuzzy classification, etc. [5] have been widely used by the IDS community for detecting known and unknown intrusions, from which we used Naïve Bayes and SVM in our work. Data mining based intrusion detection algorithms aim to solve the problems of analyzing the huge volumes of audit data and realizing performance optimization of detection rules [6]. But there are still some drawbacks in currently available commercial IDS, such as low detection accuracy, large number of false positives, unbalanced detection rates for different types of intrusions, long response time, and redundant input attributes. To overcome, in this contribution, a novel way of increasing efficiency Ant Colony Optimization (ACO) algorithms [7][8] is used with it.

The main emphasis of this paper is to find efficiently the values of detection rates (DR) and false alarms rate (FAR) of the various intrusion attacks detected in the data set and compare them to measure their performances. The experiments and evaluation are performed using the KDD-Cup99 benchmark dataset which contains information on computer networks, during normal and intrusive behaviors. This dataset is available at the University of California, Irvine web site.

The paper is organized as follows: in section 1 we outline introduction of the intrusion detection models, FRL concepts , architecture of data mining based IDS ,section 2 includes related work done ,section 3 includes introduction of various methods used ,section 4 we introduce the proposed algorithms. Section 5 we apply the

proposed algorithms to the area of intrusion detection using KDD99 benchmark network intrusion detection dataset, section 6 shows the experiments analysis and results and Section 7 is conclusions and future works.

2. Related Work

In 1980, the concept of intrusion detection began with Anderson's seminal paper [9]; he introduced a threat classification model that develops a security monitoring surveillance system based on detecting anomalies in user behavior. In 1986, Dr. Denning proposed several models for commercial IDS development based on statistics, Markov chains, time-series, etc [10]. In 2000, Valdes et al. [11] developed an anomaly based IDS that employed naïve Bayesian network to perform intrusion detecting on traffic bursts. In 2003, Kruegel et al. [12] proposed a multisensory fusion approach using Bayesian classifier for classification and suppression of false alarms that the outputs of different IDS sensors were aggregated to produce single alarm. In 2000, Dickerson at al. [13] developed the Fuzzy Intrusion Recognition Engine (FIRE) using fuzzy logic that process the network data and generate fuzzy sets for every observed feature and then the fuzzy sets are used to detect network attacks. In 2005, an efficient and biologically inspired learning model called Ant Colony Clustering Model for anomaly intrusion detection in the multi-agent IDS [14] is designed. In 2009, Wu and Yen [6] applied DT and support vector machine (SVM) algorithm to built two classifiers for comparison by employing a sampling method of several different normal data ratios. In 2004, Amor et al. [15] conducted an experimental study of the performance comparison between NB classifier and DT on KDD99 dataset. In 2010, Md. Abadeh and J. Habibi, [16] proposed a hybridization of evolutionary fuzzy systems and ant colony optimization which is used for Intrusion Detection.

3. Introduction to Various Methods used

3.1 Fuzzy If- then Rules

An FRBS presents two main components: 1) the Knowledge Base (KB), representing the knowledge about the problem being solved in the form of fuzzy linguistic IF-THEN rules, and 2) the Inference Engine, which puts into effect the fuzzy inference process needed to obtain an output from the FRBS when an input is specified. The structure of a linguistic FRBS is shown in Figure 3. Fuzzy IF-THEN rule is used to increase the interpretability and accuracy of intrusion detection model The essential part of fuzzy rule based systems (FRBSs) is a set of IF-THEN linguistic rules [3], whose antecedents and consequents are composed of fuzzy statements, related by the dual concepts of fuzzy implication and the compositional rule of inference.

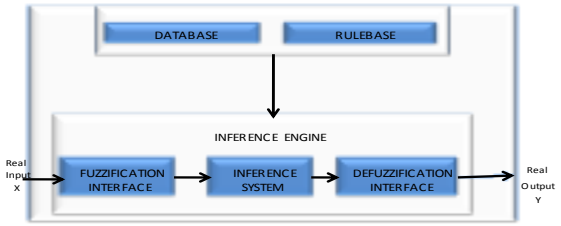

Fig 3: Generic structure of a linguistic Fuzzy Rule-Based System.

An FRBS is composed of a knowledge base (KB), that includes the information in the form of IF-THEN fuzzy rules;

IF a set of conditions are satisfied

THEN a set of consequents can be infered

and an inference engine module that includes a fuzzification interface, which has the effect of transforming crisp data into fuzzy sets; an inference system, that uses them together with the KB to make inference by means of a reasoning method; and a defuzzification interface, that translates the fuzzy rule action thus obtained to a real action using a defuzzification method.

FRBSs[4] can be broadly categorized into different families. One of the linguistic models based on collections of IF-THEN rules, whose antecedents are linguistic values, and the system behaviour can be described in natural terms. The consequent is an output action or class to be applied. For example, we can denote them as:

$\mathbf{R_1}$: IF X_1 is A_1 and X_2 is . A_2..... and X_n is A_n THEN Y_1 is B_1 and Y_2 is B_2 ... ... and Y_m is B_m

or,

$\mathbf{R_1}$: IF X_1 is A_1 or X_2 is . A_2..... or X_n is A_n THEN Y_1 is B_1 or Y_2 is B_2 ... ... or Y_m is B_m

If X={X_1, X_2, ….,X_n} and Y={Y_1, Y_2, ….,Y_m} are the universes of X_1, X_2, ….,X_n and Y_1, Y_2, ….,Y_m respectively, then A_1, A_2, ….,A_n and B_1, B_2, ….,B_m are fuzzy sets in these universes, and X_1, X_2, ….,X_n and Y_1, Y_2, ….,Y_m are fuzzy variables. For increased comprehensibility of the rules these variables may be linguistic variables taking linguistic terms.

3.2 Support Vector Machines (SVM)

Support Vector Machines (SVM) [17] have been a promising tool for data classification which is one of the most robust and accurate methods among all well-known algorithms. Its basic idea is to map data into a high dimensional space and find a separating hyper plane with the maximal margin. SVM are of two types: *Linear and Non-linear* [18]. For a linearly separable dataset, a linear classification function corresponds to a separating hyper plane f(*x*) that passes through the middle of the two classes, separating the two. Once this function is determined, new data instance x_n can be classified by simply testing the sign of the function f (x_n); x_n belongs to the positive class if f(x_n) > 0. Whereas a non-linearly separable dataset unlike the linear kernel, can handle the case when the relations between class labels and attributes is nonlinear.

We used linear machines trained on separable data for classification of data. Here label the training data {x_i, y_i}, i=1, ⋯ , *l*, $y_i \in \{-1, 1\}$, $x_i \in R^d$. Suppose we have some hyper plane which separates the positive from the negative examples (a "separating hyper plane"). The points x which lie on the hyper plane satisfy w · x + b = 0, where w is normal to the hyper plane, |b|/‖w‖ is the perpendicular distance from the hyper plane to the origin, and w is the Euclidean norm of w. Let d_+ (d_-) be the shortest distance from the separating hyper plane to the closest positive (negative) example. Define the "margin" of a separating hyper plane to be d_++d_-. For the linearly separable case, the support vector algorithm simply looks for the separating hyper plane with largest margin. This can be formulated as follows:
Suppose that all the training data satisfy the following constraints:

$$x_i \cdot w + b \geq +1 \text{ for } y_i = +1 \quad (1)$$

$$y_i \cdot w + b \leq -1 \text{ for } y_i = -1 \quad (2)$$

These can be combined into one set of inequalities:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \; \forall_i \quad (3)$$

Now consider the points for which the equality in Eq. (1) holds (requiring that there exists such a point is equivalent to choosing a scale for w and b). These points lie on the hyper plane H_1: $x_i \cdot w + b = 1$ with normal w and perpendicular distance from the origin $|1 - b|/\|w\|$. Similarly, the points for which the equality in Eq. (2) holds lie on the hyper plane H_2 : $x_i \cdot w+b= -1$, with normal again w, and perpendicular distance from the origin $|-1-b|/\|w\|$. Hence $d+ = d- = 1/\|w\|$ and the margin is simply $2/\|w\|$. Note that H_1 and H_2 are parallel (they have the same normal) and that no training points fall between them. Thus we can find the pair of hyper planes which gives the maximum margin by minimizing $\|w\|^2$, subject to constraints(3). Thus we expect the solution for a typical two dimensional case to have the form shown in Figure 4. Those training points for which the equality in Eq. (3) holds (i.e. those which wind up lying on one of the hyper planes H_1, H_2), and whose removal would change the solution found, are called support vectors; they are indicated in Figure 4 by the extra circles.

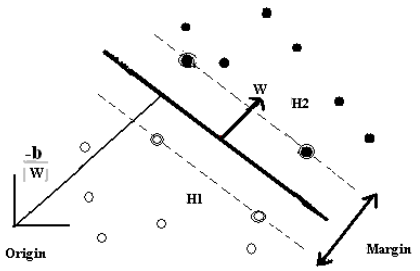

Fig 4. Linear separating hyper planes for the separable case. The support vectors are circled.

3.3 Naïve Bayesian Classifier (NB)

NAÏVE BAYES[19] method based on the 'Bayes rule' for conditional probability as this rule provides a framework for data assimilation. All data assimilation methods can be deduced from it. The principle of Bayesian analysis is "The probability of A given B is not the same as the probability of B given A".

$$P(A|B) \neq (B|A)$$

Two types of Bayes rule are:Non-Bayesian probabilities for data knowing the parameters and Bayesian probabilities for parameters knowing the data.

The Bayes theorem[20], by definition of conditional probabilities:

$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A) = [P(A)P(B)]^*$

Thus the Bayes theorem as given in eq(4).

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{P(A|B)P(B)}{P(A|B)P(B)+P(A|!B)P(!B)} \quad (4)$$

$$P(B|A) \propto P(A|B)P(B)$$

Where, if A and B are independent.

Naïve Bayesian classifier [21] is a simple classification scheme, which estimates the class- conditional probability by assuming that the attributes are conditionally independent, given the class label c. The conditional independence assumption can be formally stated as given in Eq.(5):

$$P(A|C = c) = \prod_{i=1}^{n} P(A_i|C = c) \quad (5)$$

Where each attribute set A = $\{A_1, A_2, \ldots, A_n\}$ consists of n attribute values. With the conditional independence assumption, instead of computing the class-conditional probability for every combination of A, only estimate the conditional probability of each A_i, given C. The latter approach is more practical because it does not require a very large training set to obtain a good estimate of the probability. To classify a test example, the naïve Bayesian classifier computes the posterior probability for each class C as given in Eq. (6):

$$P(C|A) = \frac{P(C)\prod_{i=1}^{n} P(A_i|C)}{P(A)} \quad (6)$$

Since P(A) is fixed for every A, it is sufficient to choose the class that maximizes the numerator term, as given in Eq. (7):

$$P(C)\prod_{i=1}^{n} P(A_i|C) \quad (7)$$

3.4 Ant Colony Optimization (ACO)

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs based on the strategies of real ants [24]. It was initially proposed in 1992 by Colorni, Dorigo and Maniezzo [22][23]. In ACO, each artificial ant is considered as a simple agent, communicating with other ants only indirectly and by effecting changes to a common environment.

Ant colonies (AC) Algorithm: Ant colonies (AC) Algorithm **[25]** is given below:

1: Initialize pheromone trail
2: **while** stopping criteria not met **do**
3: **for** all ants **do**
4: Deposit ant randomly
5: **while** solution incomplete **do**
6: Select next element randomly according to pheromone trail
7: **end while**
8: **end for**
9: Update pheromone trail
10: **end while**

Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants. The main features are high-precision solution, fast search speed, convergence to global optimum and greedy heuristic search.

Probabilistic transition rule: The probabilistic transition rule [26] which is also called as random proportional transition rule as given in Eq. (8):

$$P_{ij} = \frac{[\eta_{ij}]^{\alpha} \cdot [\tau_{ij}(t)]^{\beta}}{\sum_{i=1}^{a}\sum_{j=1}^{b_i}(\eta_{ij}\cdot\tau_{ij}(t))}, \quad \forall i \in I \quad (8)$$

where: P_{ij} is the probability rule antecedent , η_{ij} is the heuristic value , $\tau_{ij}(t)$ is the amount of pheromone at iteration t , a is the total number of attributes , b_i is the number of domain values of the i-th attribute , I are the attributes not yet used by the ant and (α and β) are two adjustable parameters that control the relative weight of the heuristic and pheromone values respectively.

Quality computation: Quality computation includes that the quality of a rule in Ant-Miner is computed according to as given in Eq. (9)

$$Q = \frac{TP}{(TP+FN)} \times \frac{TN}{(TN+FP)} \quad (9)$$

TP: true positives, FP: false positives, FN: false negatives, TN:true negatives.

As it is mentioned, the Quality is used for selecting the best rule between discovered ones. Besides, Ant-Miner uses the quality as a factor for pheromone updating.

Pheromone updating

Pheromone updating[27] includes that, in each iteration t, the pheromone will be increased for all the terms including in the constructed rule as given in Eq. (10):

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) * Q \quad \forall i,j \in R \qquad (10)$$

where R is the set of all the terms included in the rule. Furthermore, the pheromone should be decreased for every $\tau_{ij}(t)$ not in the antecedent part of the rule. By normalizing the pheromone this objective would be satisfied.

4. Proposed Work

Our aim is to increase level of performance of intrusion detection of the most using classification techniques nowadays by using optimization method like ACO. In this fuzzy IF-THEN rule is used to increase the interpretability and accuracy of intrusion detection model for better results. We choose naive bayes classifier (NB) and Support vector machine (SVM) in our work. Here in this paper we can measure the performance based on train values of these methods by comparing the generated values of DR and FAR respectively. In Naïve bayes classifier using ant colony optimization technique the values of DR and FAR can be improve for detecting intrusions in the given dataset.

In NB-ACO method, the values of parameter assumed in this paper for ant colony optimization includes total no. of cycles NC=100, no. of ants=30, maximum iterations =250, transaction data=30, values of $\alpha = 0.5$, $\beta = 0.5, \rho = 0.2$. Figure 5. shows flowchart for proposed algorithm.

The proposed algorithm can be summarized as follows:

STEP-1: Start with to generate input dataset from KDDCup99 to compare the performance of various intrusion detection methods.

STEP-2: Apply fuzzy IF-THE rule to increase the interpretability and accuracy of intrusion detection model.

STEP-3: Select the training dataset by using train values from the given input dataset to train our model.

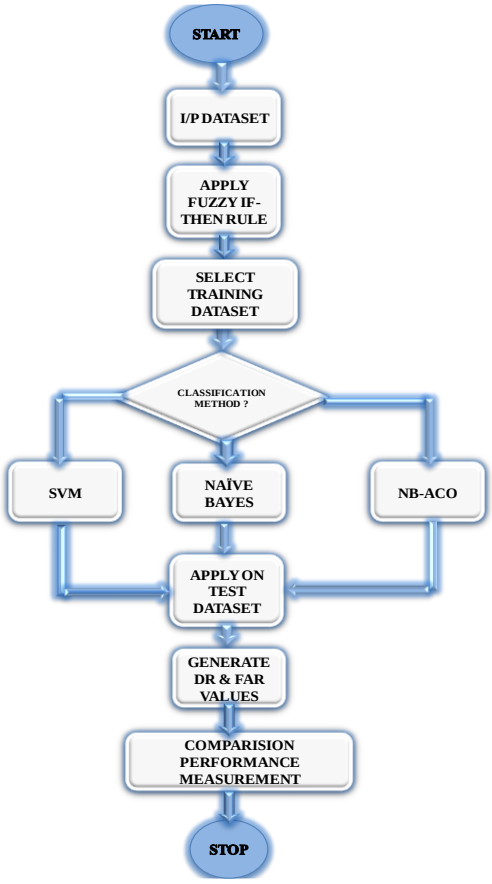

Fig 5. Flowchart for Proposed method.

STEP-4: Choose one of the following classification methods which is used to find out the various intrusion to be detect in test dataset.

a) Support Vector Machines (SVM)
b) Naïve Bayes Classifier (NB)
c) Naïve Bayes with Ant Colony Optimization (NB-ACO)

STEP-5: Apply it on the test dataset.

STEP-6: Generate detection rate (DR) and false alarm rate (FAR) values.

STEP-7: compare these values obtained in previous step and finally obtain the results by measure of performance of the model.

STEP-8: Stop.

5. Kdd Cup99 Dataset

The KDD cup 1999 dataset was used in the 3rd International Knowledge Discovery and Data Mining Tools Competition for building a network intrusion detector, a predictive model capable of distinguishing between intrusions and normal connections [28]. In 1998, DARPA intrusion detection evaluation program, a simulated environment was set up to acquire raw TCP/IP dump data for a local-area network (LAN) by the MIT Lincoln Lab to compare the performance of various intrusion detection methods. It was operated like a real environment, but being blasted with multiple intrusion attacks and received much attention in the research community of adaptive intrusion detection. The KDD99 dataset contest uses a version of DARPA98 dataset. In KDD99 dataset [29], each example represents attribute values of a class in the network data flow, and each class is labeled either normal or attack. Examples in KDD99 dataset are represented with a 41 attributes and also labeled as belonging to one of five classes as follow:1)Normal, (2) Denial of Service (DOS), (3) Remote to User (R2L), (4) User to Root (U2R), (5) Probing (Probes).
Table 1 presents the classification of different types of attacks in the four different Intrusion classes and the distribution of each class in the 10% of the KDD-Cup 99 data set.Hence, 41 numeric features are constructed and normalized to the interval [0, 1]. They are given in table 2. This section consists of two subsections.

As we know for better intrusion detection the rate of detection must be higher and rate of false alarm must be lower, in our next section the experimental analysis is shown**.** To estimate the performance of the system, two important formulas are used to evaluate system accuracy: 1) detection rate (DR) Eq.(11), 2) false alarm rate (FAR) Eq.(12).

$$DR=\frac{(Total\ number\ of\ detected\ attacks)}{(Total\ number\ of\ attacks\ detection)}\times 100\% \qquad (11)$$

$$FAR=\frac{(Total\ number\ of\ normal\ processes)}{(Total\ number\ of\ misclassified\ processes)}\times 100\% (12)$$

Table 1. Different Attacks Types in 10% Kdd99 Dataset

5 Main Attack Classes	22 Attacks Classes	Samples
Normal		97277
Denial of Service(DoS)	back, land, Neptune , pod, smurt , teardrop	391458
Remote to User (R2L)	ftp _write, guess _passwd , imap ,multihop, phf, spy, warezclient, warezmaster	1126
User to Root (U2R)	buffer_overflow,perl,load module, rootkit	52
Probing	ipsweep, nmap, portsweep, satan	4107

6. Experimental Analysis and Results

In order to evaluate the performance of proposed algorithm for network intrusion detection, we performed 5-class classification using KDD99 intrusion detection benchmark dataset. All experiments were performed using an Intel core 2 Duo Processor 2.0 GHz processor (2 MB Cache, 800 MHz FSB) with 512 MB RAM, and implemented on a Windows XP Professional operating system. Figure 6 shows results as obtained by using various classification techniques.The results of the comparison of proposed algorithm with SVM, with naive Bayes classifier and with NB-ACO are in Table 3.

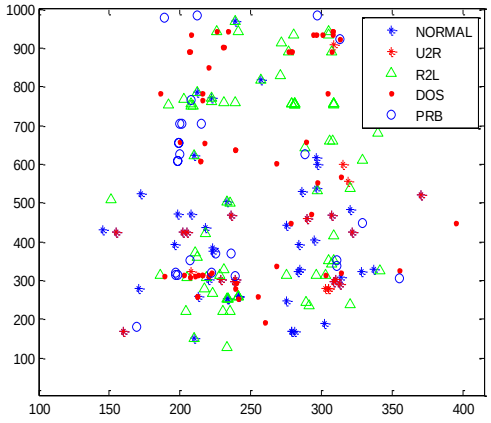

Fig 6. Results obtained by using various classification techniques

Table: 2 Input Attributes for Kdd99 Dataset

NO.	Attributes name	Type	NO.	Attributes name	Type
1	Duration	Con.	22	is_guest_login	Dis.
2	protocol_type	Dis.	23	Coun	Con.
3	Service	Dis.	24	srv_count	Con.
4	Flag	Dis.	25	serror_rate	Con.
5	src_bytes	Con.	26	srv_serror_rate	Con.
6	dst_bytes	Con.	27	rerror_rate	Con.
7	Land	Dis.	28	srv_rerror_rate	Con.
8	wrong_fragment	Con.	29	same_srv_rate	Con.
9	Urgent	Con.	30	diff_srv_rate	Con.
10	Hot	Con.	31	srv_diff_host_rate	Con.
11	num_failed_logins	Con.	32	dst_host_count	Con.
12	logged_in	Con.	33	dst_host_srv_count	Con.
13	num_compromised	Con.	34	dst_host_same_srv_rate	Con.
14	root_shell	Con.	35	dst_host_diff_srv_rate	Con.
15	su_attempted	Con.	36	dst_host_same_src_port_rate	Con.
16	num_root	Con.	37	dst_host_srv_diff_host_rate	Con.
17	num_file_creation	Con.	38	dst_host_serror_rate	Con.
18	num_shells	Con.	39	dst_host_srv_serror_rate	Con.
19	num_access_files	Con.	40	dst_host_rerror_rate	Con.
20	num_outbound_cmds	Con.	41	dst_host_srv_rerror_rate	Con.
21	is_host_login	Dis.			

7. Conclusion and Future Work

In this paper, we compare the performances of different methods used for Intrusion detection on the basis of DR and FAR values. The results obtained as, the value of rates obtained from Naïve Bayes is quite lesser than SVM. And increased efficiency of Naïve Bayes classifier with ACO is increased upto (97% approx.) much better than the two methods which is upto (91% approx.).

One more point to be noticed is that with the increase in the train values, there shows slight increase in the DR values in each method which indicates better performance of our work. We conclude that with such a great improvement in percentages of DR and FAR values, our work seems will be useful for more research in this field. Here shows that using optimization technique the results be improve for better detection. The future works focus on applying and ensemble some other techniques to improving the performances of these methods for intrusion detection upto (99.99%), and also improve support vector machine technique by applying some other methods with it.

Table 3.Comparison of Performances of Various Methods Used

Train value	SVM DR	NB DR	NB-ACO DR	SVM FAR	NB FAR	NB-ACO FAR
0.20	91.44	90.97	95.95	5.9149	4.3543	1.6505
0.45	91.52	91.17	96.03	5.9909	4.5567	1.7265
0.50	91.56	91.19	96.17	6.0322	4.5766	1.7678
0.68	91.59	91.25	96.23	5.9909	4.6324	1.7265
0.75	91.65	91.31	97.07	6.0322	4.6907	1.7678
0.86	91.70	91.36	97.21	6.1746	4.7498	1.9102

References

[1] T. Aulakh, (2009) "Intrusion detection and prevention system: CGI attacks" thesis for the degree master of science San José state university, pp 19-22.
[2] S.Panigrahi, S.Sural, (2009) " Detection of Database Intrusion Using a Two-Stage Fuzzy System", LNCS Volume 5735/2009, 107-120, DOI: 10.1007/978-3-642-04474-8_9
[3] J. Casillas, O. Cordon, F. Herrera, (2000) "Learning fuzzy rules using ant colony optimization algorithms" ,dept. of computer science and artificial intelligence, university of granada, Spain.
[4] F. Herrera, (2008)"Genetic fuzzy systems taxonomy, current research trends and prospects "Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain Springer-Verlag.
[5] J. Han and M. Kamber, (2011) "Data Mining:Concepts and Techniques Slides for Textbook — Chapter 7 " Intelligent Database Systems Research Lab School of Computing Science Simon Fraser University, Canada ,October 15.
[6] Su-Yun Wu, and E. Yen, (2009) "Data mining-based intrusion detectors," Expert Systems withApplications, Vol. 36, Issue 3, Part 1, April ,pp. 5605-5612.
[7] M. Dorigo, "Ant colony optimization web page" , http://iridia.ulb.ac.be/mdorigo/ACO/ACO.html
[8] M.Dorigo, M. Birattari, and T. St¨utzle, (2006) "Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique "Universite Libre de Bruxelles, Belgium, IEEE computational intelligence magazine, November .
[9] Anderson, P. James, (1980) "Computer security threat monitoring and surveillance," Technical Report 98-17, Washington, Pennsylvania, USA.
[10] Dorthy, E. Denning, (1987) "An intrusion detection model, "IEEE Transaction on Software Engineering, SE-13(2), pp. 222-232.
[11] A. Valdes, K. Skinner, (2000) "Adaptive model-based monitoring for cyber attack detection," in Recent advances in Intrusion Detection Toulouse, France, pp. 80-92.
[12] C. Kruegel, (et.al.), (2003) "Bayesian event classification for intrusion detection," in Proc. of the 19th Annual Computer Security Applications Conference, Las Vegas, NV.
[13] J.E. Dickerson, J.A. Dickerson, (2000) "Fuzzy network profiling for intrusion detection," In Proc. of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS), Atlanta, GA, pp. 301-306.
[14] C. Tsang and S. Kwong, (2005) "Multi-Agent Intrusion Detection System in Industrial Network using Ant Colony Clustering Approach and Unsupervised Feature Extraction", Proceedings of the IEEE, pp. 51-56.
[15] N. B. Amor, S. Benferhat, and Z. Elouedi, (2004) "Naïve Bayes vs. decision trees in intrusion detection systems," In Proc. of the 2004 ACM Symposium on Applied Computing, New York, pp. 420-424.
[16] Md. S. Abadeh, J. Habibi, (2010) "A Hybridization of Evolutionary Fuzzy Systems and Ant Colony Optimization for Intrusion Detection", Volume 2, Number 1 (pp. 33-46), Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.
[17] Y. Shi,;(et.al.), (2011) "Optimization Based Data Mining:Theory and Applications" Chengdu, China, Springer- Verlag London Limited, pp 18-134.
[18] J. Christopher, C. Burges, (2004) "A Tutorial on Support Vector Machines for Pattern Recognition" Bell Laboratories, Lucent Technologies, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
[19] M. Glick, A. Klon, P. Acklin, and J. Davies, (2011) "Enrichment of Extremely Noisy High-Throughput Screening Data Using a Naïve Bayes Classifier", Journal of biomolecular Screening, Published by:http://www.sagepublications.com, in August 6.
[20] L.Bertino and G. Evensen, "The Bayes Theorem" ,Nansen Environmental and Remote Sensing Center, Bergen, Norway, Hydro Research Centre, Bergen, Norway.
[21] A.J.M .Abu Afza, Dewan Md. Farid, and C.M.Rahman, (2011) "A Hybrid Classifier using Boosting, Clustering, and Naïve Bayesian Classifier" United International University Dhaka-1209, Bangladesh, World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741Vol. 1, No. 3, 105-109.
[22] LK. Behera and A. Sasidharan, (2011) "Ant Colony Optimization for Co-operation in Robotic Swarms", Pelagia Research Library Advances in Applied Science Research, 2 (3): 476-482
[23] HA. Zurba , T. Landolsi, Md. Hassan and F. Abdelaziz, (2011), "On The Suitability of Using Ant Colony Optimization for Routing Multimedia Content Over Wireless Sensor Networks", International journal on applications of graph theory in wireless ad hoc networks and sensor networks Vol.3, No.2, June.
[24] M. Galea, (2003) "Fuzzy Rules from ANT-Inspired Computation" PhD Proposal School of Informatics University of Edinburgh ,July.
[25] M. Gilli, (2004) "An Introduction to Optimization Heuristics", Department of Econometrics, University of Geneva and FAME www.unige.ch/ses/metri/gilli, Seminar University of Cyprus.
[26] T. Singh, (2010) "Thesis work on: Ant Colony Optimization (ACO) based Intrusion Detection System" CSE dept Thapar University, patiala,India.
[27] Aghdam MH, Tanha J, Naghsh-Nilchi AR, Basiri ME (2009) Combination of Ant Colony Optimization and Bayesian Classification for Feature Selection in a Bioinformatics Dataset. J Comput Sci Syst Biol 2: 186-199. doi:10.4172/jcsb.1000031.
[28] The KDD Archive. KDD99 cup dataset, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[29] M. Tavallaee, (et.al.), (2009) "A detailed analysis of the KDD CUP 99 data set", in Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications, pp. 53-58, Ottawa, Ontario, Canada.

Author's Profile

NAMITA SHRIVASTAVA

Completed Bachelor of engineering in Computer Science and Engineering from Technocrats Institute of Technology Bhopal, Rajiv Gandhi University, Bhopal (M.P.), India in 2008. Final Year Student Master of technology in Computer Science and Engineering (Dec'11) from Lakshmi Naraian college of Technology Bhopal, Rajiv Gandhi University, Bhopal(M.P.), India.

VINEET RICHARIYA

Vineet Richariya is Head of Department of Computer Science And Engineering, Lakhmi Narain College of Technology, Bhopal, India. He did his MTech (Computer Science & Engineering) from BITS Pillali in year 2001. He did his B.E (Computer Science & Engineering) from Jiwaji University, Gwalior, India in year 1990.