International Journal of Computer Science Trends and Technology (IJCST) – Volume 4 Issue 5, Sep - Oct 2016

RESEARCH ARTICLE OPEN ACCESS

An Enhanced Genetic Algorithm based Intrusion Detection System for detection of Denial –of-Service Attacks

B.Koustubha Madhavi [1]

Assoc Prof., Dept. of CSE, NMREC, Hyderabad, India.

V.Mohan [2]

Sr. Astt. Prof., Dept. of CSE, NMREC, Hyderabad, India.

Dr. Vilas M Ghodki [3]

Assoc. Prof, Dept. of Computer Science, J.B. College of Science, Wardha, Maharashtra, India.

ABSTRACT

The continuously evolving technology and increased scalability of humongous number of users in today's era demands for the highest level of security. To ensure a good level of security, Intrusion Detection Systems have been widely deployed and several offline or online IDS techniques to detect, identify and classify attacks have been proposed. This paper discusses about an improved and modified version of Genetic Algorithm for network anomaly detection. Also, we have applied an attribute subset reduction technique. We have adopted a soft computing approach for rule generation to make it more efficient as compared to the hard computing rule generation method used in the existing genetic algorithm. The precision of our method is tried on various subsets from KDD99 dataset. Rapid Miner 5.1 tool used to simulate the algorithm. The Empirical results of our proposed method have indicated higher detection rates and low false positive rates.

Keywords: - *Intrusion Detection System (IDS), Neural network, Intrusion detection system (NNIDS), Genetic algorithm (GA), Detection rate (DR), False Positive (FP).*

I. INTRODUCTION

A. Background

At the point, when an outsider tries or attempts to break-in one's system, that action is referred as intrusion, which is detected or caught with the help of an intrusion detection system (IDS). IDS essentially breaks down the movement of suspicious traffic in the system and screens the system inside and remotely for any sort of noxious action. At whatever point it finds any unapproved action, it consequently sends a caution or alert to the individual in charge of making a move. IDS are turning into a key procedure for security of frameworks over systems from dangers and abuse. Intrusion detection systems are an asset to effective security implemented on networked systems. It is fundamentally an arrangement of systems that are utilized to recognize suspicious movement at system and host level. It is by and large classified into two essential classifications: Signature based and Anomaly identification frameworks. By and large, an IDS catches information from system and identify peculiarities in it by applying its predefined rules. An IDS can have diverse capacities which relies on how intricate and complex the segments are. NIDS (Network-based IDS) intrusion detection systems that obtain information parcels wandering on system media and contrast them with the database of signatures.

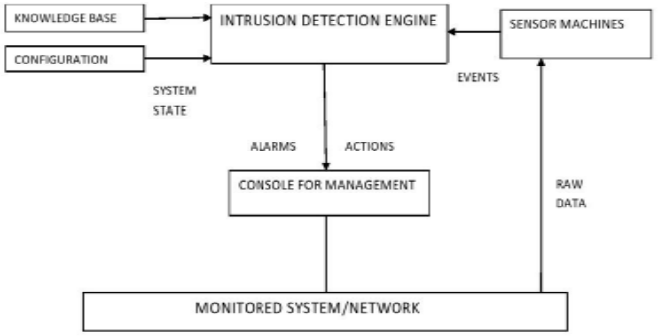

Figure 1 : Architecture of an Intrusion Detection System

In Fig 1, Intrusion detection engine gathers crude information from system framework screen through sensor machine for various events that happen in network and on any suspicion, triggers alarm and cautions and suitable moves are made by console administration to shield framework from pernicious assaults. In our methodology intrusion detection can be considered as information examination process. evolve over multiple generations for finding better solutions.

B. Basic Genetic Algorithm

Genetic Algorithms (GAs) are robust and stochastic search procedures based on the principles of natural selection and genetics. They follow the evolutionary process as stated by Charles Darwin. Since their introduction by John Holland in 1960 and their popularization through [9], these algorithms have been widely used in various engineering and scientific areas such as image processing, pattern recognition, and more recently intrusion detection field [10-12]. The basic Genetic algorithm process commence with a set of potential solutions (chromosomes) which comprises a population, are generated or selected on random basis. These chromosomes evolve during various generations producing new offsprings by using techniques like crossover and mutation. Selection, on the basis of fitness, determines which chromosome will be chosen from the population for recombination. Crossover splits two chromosomes and then combine the first part with the second whereas mutation flips one or more bits of a chromosomes. The population is generally a compilation of candidate solutions that are

Step1 : Initialize population.
Step 2. Compute population fitness.
Step 3 Implement operators like selection, crossover and mutation.
Step 4. If stopping criteria is not met, go back to step 2.
Step 5. Exit

Figure 2 : Basic Genetic Algorithm

considered during the course of the algorithm. A single solution in the population is described as an individual. The fitness of an individual measures the 'goodness' of the solution represented by the individual that is higher the fitness function, better the solution is. Basic algorithm for genetic algorithm illustration can be defined in Figure 2. Functioning of genetic algorithms initially choose a random population of chromosomes. Each chromosome, which represents the problem, is formed of finite number of genes, which are pre-defined in each implementation [15]. This initial population is enhanced to a high quality population of chromosomes, where each chromosome satisfies a predefined fitness function. As per the need of the solution, different positions of genes in a chromosome are encoded as numbers, bits, or characters. Each population is enhanced by applying mutation, crossover, inversion, and selection processes. The basic genetic algorithm is given in Figure 2 for better understanding of the process:

II. PROPOSED SYSTEM

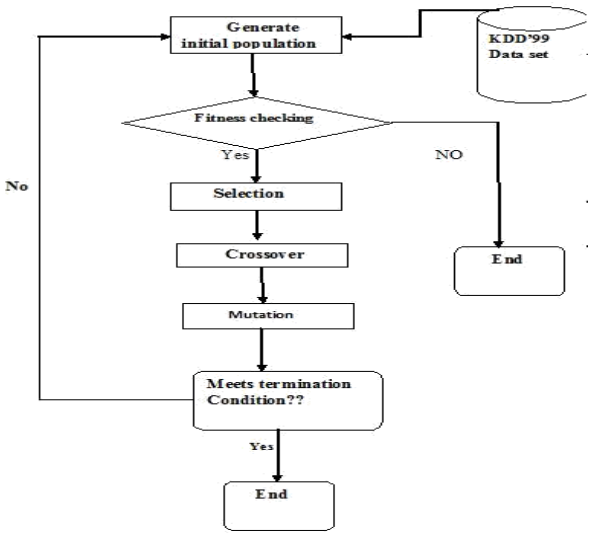

Figure 3 : Proposed Genetic Algorithm Flowchart

Algorithm: Rule set generation using genetic algorithm.

Input: Network audit data, number of generations, and population size.

Output: A set of classification rules.

Step 1->Pre-process data by converting the symbolic feature into numeric data
Step 2->Select 15 features based on information gain
Step 3->For each extracted features
Step 4->Normalize and Fuzzify each selected attribute and divide into fuzzy classes
Step 5->Initialize the population
W1 = 0.2, *W2* = 0.8, *T* = 0.5
N = total number of records in the training set
Step 6->For each chromosome in the population
A = 0, *AB* = 0
Step 7-> For each record in the training set
I f the record matches the chromosome
AB = *AB* + 1
End if
Step 8->If the record matches only the "condition" part
A = *A* + 1
End if
End for
Step 9->Fitness = W1 * AB / N + W2 * AB / A
Step 10-> If Fitness > T
Select the chromosome into new population
End if

III. EXPERIMENTAL SETUP

We used a windows operating system with a Pentium 4 processor, 1GB of hard disk space and 512 MB of RAM to execute the computer program. We also used a tool known as RapidMiner, formerly YALE (Yet Another Learning Environment), is an environment for machine learning, data mining, text mining.

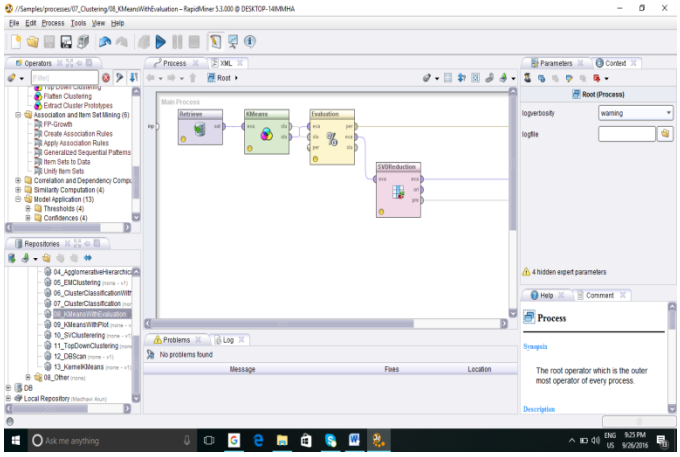

Figure 4 : RapidMiner 5.1 Tool

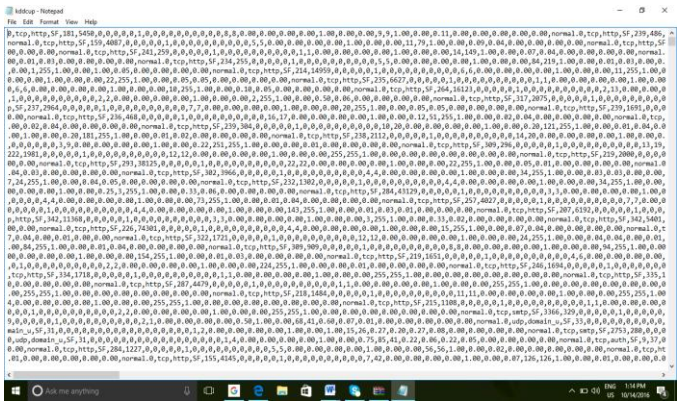

Figure 5 : KDD Cup'99 Dataset 10%

1) Preprocess Data Set: It consists of 41 features, 38 of them are numeric and 3 are symbolic. Convert these symbolic features into numeric before applying to genetic algorithm.. We use the RapidMiner tool to convert the symbolic features into numeric one. Preprocess operator can be used to access the repositories introduced in RapidMiner
5.1 It should replace all file access, since it provides full file, it will provide the complete meta data of the data, so that all meta data transformations are possible.

2) Feature Extraction: Due to large amount of network traffic captured in terms of number of features and number of record, it is very difficult to process all the network traffic before making any

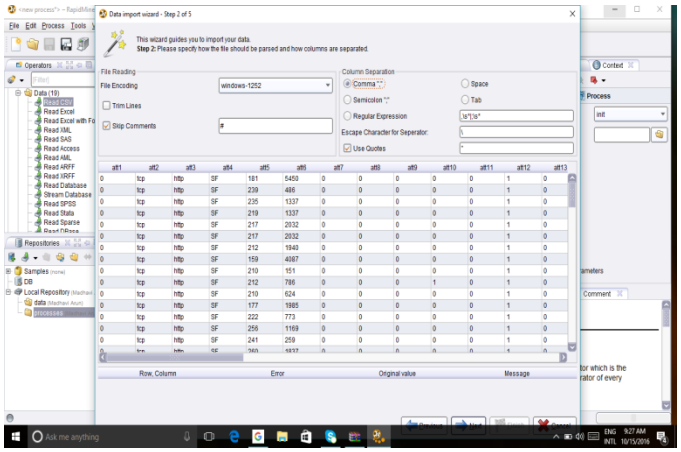

Figure 6 : Importing KDD Cup'99 Dataset into RapidMiner

decision about normal or abnormal. We used this pre-processed data as a input to the and generate
output as only the top 15 features based on information gain using 3.2.3. Table 1 shows the top 15 attribute based on information gain.

3) Normalization and Fuzzification of the Preprocessed Data Set

We first reduced the data set based on selected 15 attribute in section . Now we normalize the reduced data set so that every value lies in the range of (0, 1). Besides, we fuzzify it by using the triangular function as shown below.

$$triangle(x,a,b,c) = max\left(min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right)$$

Attribute	Information Gain
Att 23	1
Att 34	0.885
Att 2	0.850
Att 38	0.795
Att 25	0.770
Att 39	0.762
Att 26	0.723
Att 1	0.711
Att 37	0.652
Att 10	0.622
Att 8	0.613
Att 22	0.585
Att 5	0.570
Att 22	0.511
Att 4	0.501

Table 1 : List of features extracted from the KDD

5) Selection Operator

Selection Operator always ensures the best individual must be chosen. Selection of any rule can be takes place if its value is greater than the threshold value of the fitness. Fitness of the rule can be determined by line 18 of the Proposed algorithm mentioned in section 3. We select only those rules whose fitness value is greater than the threshold value i.e. Fitness >0.5.

6) Crossover

Crossover Operator randomly chooses a pairs individuals among those previously elected to breed and exchange substrings (L, LM, M, MH, and H) between them. The exchange occurs around randomly selected crossing points.

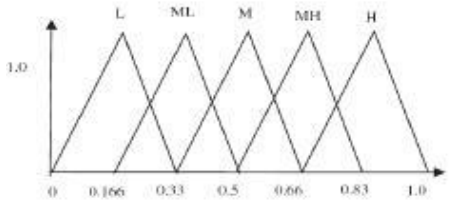

Figure 7 : Class Divisions

7) Mutation

Some of the substrings of rule generated is flipped randomly. After the 500th generation, the best chromosome was selected.

III. EXPERIMENTAL RESULTS

A. Dataset Used

For the purpose of experiments, KDD'99 benchmark dataset is used. The Massachusetts Institute of Technology- MIT developed this dataset during the international competition on data mining in 1999 [14]. It contains a large volume of network TCP connections, the results of 5 weeks of capture in the Air force network. Each connection consists
of 41 attributes plus a label of either normal or a type of attack. Simulated attacks fall into one of the four following categories:

• *DOS (Denial Of Service)* attack aim to make a service or a resource unavailable.
• *U2R (User to Root)* attack witch a simple user tries to exploit a vulnerability in order
• *R2L (Remote to Local)* here, the attacker attempts to gain access (account) locally on a machine accessible via the network.
• *PROBE* represents any attempt to collect information about the network, the users or the security policy in order to outsmart it.
The size of the full dataset is very large (about 4 million registers), a shorter version is usually employed for the purpose of training. It represents 10% of the original dataset (see details in Table.2).

Record Type	Training Dataset	Testing Dataset
NORMAL	7500	6000
ATTACK	1000	869
TOTAL	8500	6869

Table 2: Distribution of KDD Dataset

B. Feature Selection

KDD'99 is a dataset containing connections described by 41 attributes among which 8 are categorical and the remaining 33 are numeric. The importance and the relevance of each attribute has been the main interest of several researchers [17, 18]. In this paper, we considered the 15 attributes listed in Table 1. According to the authors of [19],
these attributes are most suited to the use of evolutionary algorithms. It is noteworthy that nominal attributes were replaced by their probability of occurrence in the dataset to simplify the computation of the similarity distance. algorithms. It is noteworthy that nominal attributes were replaced by their probability of occurrence in the dataset to simplify the computation of the similarity distance.

C **.** ***Results***

The accuracy of any intrusion detection system is determined by the detection rate or true positive rate (TPR), the false positive rate (FPR) and the false negative rate (FNR). These measures are calculated based on the confusion matrix Table 3.

		Classified Class		*Total*	
		Normal	**Attack**		
Actual Class	**Normal**	TN	FP	TN+FP	FPR=FP/(FP+TN)
	Attack	FN	TP	FN+TP	TPR=TP/(TP+FN)
Total		TN+FN	FP+TP	Datset	

Table 3 : Confusion Matrix

The detection rate value is expected to be as large as possible, while the false positive rate value is expected to be as small as possible

	Proposed System							
(%)	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**
TPR	75,81	77,51	83,1	96,34	97,85	98,47	98,67	98,96
FPR	0,12	0,14	0,17	0,22	0,24	0,33	0,35	0,38
FNR	24,16	22,48	16,87	3,65	2,13	1,51	1,3	1,02
Existing System								
TPR	75,76	62,89	77,57	80,93	64,56	80,45	77,78	71,4
FPR	0,42	0,27	0,36	0,41	0,26	0,37	0,45	0,32
FNR	24,23	37,1	22,42	19,06	35,43	19,54	22,21	28,59

Table 4 : Experimental Results

Table 4 shows how the detection rate and false positive rate vary when different data subsets are used for both proposed and existing algorithms. Detection rate (DR) is computed as the ratio between the number of correctly detected intrusions and the total number of intrusions.

$$DR = \frac{True\ Positive\ Rate}{False\ Negative\ Rate + True\ Positive\ Rate}$$

	Proposed Algorithm		*Existing Algorithm*	
Type	***Detection rate***	***False Positive***	***Detection Rate***	***False Positive***
Normal	95	3.8	92.64	5.13
Attack	96.66	2.2	93.88	5.85

Table 5 : Performance Measure of Proposed GA vs Existing GA

The experimental results show that the proposed method yielded good detection rates when using the generated rules to classify the training data itself. That is what we expected. When the resulting rules were used to classify the testing dataset, the detection rates of network attacks were decreased by great extent. The results have indicated that the generated rules were biased to the training data.

IV. CONCLUSION AND FUTURE WORK

The above test results clearly suggest that Genetic-based Intrusion detection system combined with feature selection, enables the system to produce optimal subset of attribute in the midst of huge network information.. We evaluate the performance of our scheme using different subsets of KDD99. The simulation results show that our approach gives high detection rates between 75% and 98%, and low false positive rates between 1.3% and 0.12%. This proves that the proposed Genetic Algorithm with soft computing method for rule generation is more efficient compared to existing GA. However, we noticed that the number of rejected instances increases as the dataset size increases. Analyzing and reducing rejected instances are among future improvements to this work. Once this number reaches an acceptable level, online tests will be performed.

REFERENCES

[1] Aickelin, U., J. Greensmith, and J. Twycross. "Immune System Approaches to Intrusion Detection-A Review", Natural Computing, Springer, in print, Vol. 6 No. 4, pp 413-466, 2007.

[2] Bobor, V. "Efficient Intrusion Detection System Architecture Based on Neural Networks and Genetic Algorithms.", Department of Computer and Systems Sciences, Stockholm University / Royal Institute of Technology, KTH/DSV, 2006.

[3] Faraoun, K M., and A. Boukelif. "Genetic Programming Approach for Multi-Category Pattern Classification Applied to Network Intrusions Detection."INTERNATIONAL JOURNAL OF COMPUTATIONAL INTELLIGENCE, Vol. 3, No. 1, 2006 pp. 79-90

[4] Subhajit Pal, Debnath Bhattacharya, G.S. Tomar& Tai-hoon Kim, "Wireless Sensor Networks and its Routing Protocols: AComparative Study", IEEE International Conference on Computational Intelligence and Communication Networks CICN 2010, pp 314-319, Nov 2010.

[5] [5] A. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur, and J. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," in *SDM*, 2003, pp. 25-36.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, p. 15, 2009.

[7] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *GI/ITG Workshop MMBnet*, 2007.

[8] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis* vol. 344. New York: John Wiley & Sons, 1990.

[9] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," *Addion wesley*, vol. 1989, 1989.

[10] W. Li, "Using genetic algorithm for network intrusion detection,"*Proceedings of the United States Department of Energy Cyber Security Group*, pp. 1-8, 2004.

[11] D. E. Denning, "An intrusion-detection model," *Software Engineering, IEEE Transactions on*, pp. 222-232, 1987.

[12] C. Kruegel and T. Toth, "A survey on intrusion detection systems," in*TU Vienna, Austria*, 2000.

[13] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo, "Anomaly Detection Methods in Wired Networks: A Survey and Taxonomy," *Computer Communications*, vol. 27, pp. 1569-1584, 2004.

[14] C. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*: Springer Science & Business Media, 2007.

[15] S. Sivanandam and S. Deepa, *Introduction to genetic algorithms*: Springer Science & Business Media, 2008.

[16] [Z. Michalewicz, *Genetic algorithms+ data structures= evolution programs*: Springer Science & Business Media, 1996.

[17] Y. Chen, Y. Li, X.-Q. Cheng, and L. Guo, "Survey and taxonomy of feature selection algorithms in intrusion detection system," in *Information Security and Cryptology*, 2006, pp. 153-167.

[18] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets," in *Proceedings of the third annual conference on privacy, security and trust*, 2005.

[19] A. I. Madbouly, A. M. Gody, and T. M. Barakat, "Relevant Feature Selection Model Using Data Mining for Intrusion Detection System," *arXiv preprint arXiv:1403.7726*, 2014.