International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 6– No. 7, January 2014 – www.ijais.org

Selecting GA Parameters for Intrusion Detection

S.N.Pawar
Asso.Professor (E &TC),
Jawaharlal Nehru Engineering College,
Aurangabad, MS, India.

R.S.Bichkar
Professor (E &TC),
G.H.Raisoni College of Engineering & Management,
Pune, MS, India.

ABSTRACT

Genetic algorithms happen to be one of the preferred techniques for intrusion detection. It needs careful selection of its parameters like population size, number of generations, mutation rate, crossover rate, selection type etc. and also requires selecting appropriate percentage of attack samples in a data set to be able to find good solutions. Choosing unsuitable parameters and methods might result into longer program runs or even bad optimization results. In the proposed method, genetic algorithm is used for intrusion detection rule generation. It is implemented and run using different configurations and results are compared. Then the best GA parameters are selected for intrusion detection.

Keywords

Genetic algorithm, intrusion detection, parameter selection, crossover, mutation, selection.

1. INTRODUCTION

Any unauthorized access causing violation to the security policy of a system is called intrusion [1]. Intrusions are detected using intrusion detection system (IDS). The sooner we are able to detect an attack, the quicker we can act. Intrusion detection helps to collect more information about attacks, strengthening the intrusion prevention methods.
Various soft computing techniques such as Genetic Algorithm, Artificial Neural Network and Fuzzy Logic are used to design IDS [2].
IDSs are designed based on various detection techniques, namely Anomaly intrusion detection and Misuse intrusion detection [2].

Anomaly intrusion detection

In anomaly IDS the user’s behavior is compared with a known standard behavior to detect any significant deviation from normal behavior. This approach can be more effective in protection against unknown or novel attacks since no prior knowledge about specific intrusions is required. However, it may cause more false positives because abnormality can be due to a new normal behavior [3].

Misuse intrusion detection

This is the most widely used IDS. It uses patterns of known attacks or weak spot of the system to identify known intrusions. The signatures and patterns used to identify attacks consist of various options in the packet like source address, destination address, source and destination ports and even the keywords in the content area of a packet.

IDS can also be classified into two categories based on their location [4], as host-based and network-based IDS. A host-based IDS monitors activities associated with a particular host; whereas a network based IDS monitors activities associated with network.

GA is found to be the most efficient technique for intrusion detection in terms of detection accuracy [5]. Researchers have used GA for either generation of classification rules [2, 3, 6, 7] or for the selection of appropriate features of the chromosome [8, 9].

This paper uses an effective GA-based approach to generate the classification rules for network intrusion detection and to select the appropriate GA parameters for the same. Choosing appropriate parameters and methods in genetic algorithm might result in substantial improvement in results. A good configuration might cause the algorithm to converge to best results in a short time while a worse setting might cause the algorithm to run for a long time before finding a good solution or even it might never be able to find a good solution. This paper presents GA implementation with different parent selection techniques, crossover rate, mutation rate, population size, attack samples in a data set etc. It tries to identify the settings that work better in generating the fittest rules for network intrusion detection.

Thus far, the motivation of the presented work and a brief overview of the IDS are discussed. The remaining paper is organized as follows. Section 2 gives an overview of the genetic algorithm employed in this work. Section 3 presents the proposed GA based IDS. Section 4 presents results and discussion. Section 5 concludes the paper.

2. GENETIC ALGORITHM

Genetic Algorithm (GA) is a technique which works on the mechanics of natural selection. They are based on the Darwin’s theory of survival of the fittest. Simplicity of operation and power of effect are the two main attractions of the genetic algorithmic approach. It combines survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search [10]. The major application of GA is in the area of optimization.

The GA process begins with a set of potential solutions or chromosomes which are randomly generated or selected. These chromosomes are normally encoded in the binary form but other forms of encodings are also used. The entire set of these chromosomes comprises a population.

In every generation the fitness of these chromosomes is checked. Fitness function is used to determine the fitness of the chromosomes. Based on the fitness, fittest chromosomes are selected. The chromosomes with poor fitness value are discarded. The selected fit chromosomes undergo crossover and mutation to form a new population. This new population is used for the next generation. Normally, the algorithm terminates when either a set number of generations or a satisfactory fitness level has been achieved. Genetic algorithm is composed of three operators. They are reproduction or selection, crossover or recombination and mutation.

Selection is the process of choosing parents for reproduction. Selection is usually the first operator applied on population. Selection operator selects good strings in a population and forms a mating pool. This is one of the reasons for the selection operation to be sometimes known as reproduction operator. Thus, in selection operation the process of natural selection cause those individual that encode successful structures to produce copies more frequently. There are many techniques to select the chromosomes, e.g. Roulette wheel selection, tournament selection, rank selection, steady-state selection etc.

After two parents have been selected by the selection method, crossover takes place. Crossover is an operator that recombines the two parents (chromosomes) to produce two offspring. Two newborn chromosomes may be better than their parents. The crossover is carried out according to the crossover probability. There are various crossover techniques like one-point, two-point, uniform etc.

Mutation operator alters one or more gene values in a chromosome. With these new gene values, the genetic algorithm may arrive at a better solution. Mutation prevents the population from stagnating at any local optima. Mutation takes place according to a pre-defined mutation probability. Mutation probability is set fairly low. If it is high, the search becomes primitive random search. The various types of mutations are flip bit, boundary, non-uniform, uniform, Gaussian etc.

The chromosomes are then evaluated using a certain fitness criteria. When the GA terminates, the chromosome with best fitness is taken as the best solution of the problem.

3. THE PROPOSED GA-BASED IDS

In this work the classification rules are generated using GA-based approach. These rules are then used to classify or detect the infected connections.

3.1 Data set

MIT Lincoln Laboratory, under the Defence Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) sponsorship, has collected and distributed the first standard data for evaluation of computer network intrusion detection systems. This Data is DARPA 1998 data [11]. It consist of tcpdump and BSM list files. Each line in a list file corresponds to a separate session. Each session corresponds to an individual TCP/IP connection between two computers. The first nine columns in list files provide information which identifies the TCP/IP connection.

The proposed GA based IDS approach also used this DARPA 1998 data for the generation of rule set and then for the detection of infected connections. The data of Wednesday of first week was used in this experimentation which comprised 46,250 samples. Two subsets are generated from this data set. One is used for training the system and the other is used to testing.

3.2 Feature selection and representation

Seven most important features of the data set are selected for defining the intrusion rules. These are *duration* (h: m: s), *service* (int.), *source port* (int.), *destination port* (int.), *source IP* (a. b. c. d), *destination IP* (a. b. c. d), *attack name* (int.).

These features can be represented in different form in a GA chromosome. They can be encoded in binary, integer or floating-point representation. The three parts of *duration* (hours, minutes and seconds) are encoded in one gene of type integer. The features *protocol*, *source port*, *destination port* and *attack name* are encoded using one gene of type integer [4].The feature *source IP* and *destination IP* have four parts (a, b, c and d), each of which is encoded in one gene of type integer. All the classification rules are in *if-then* form as in if *A then B*. All the features except *attack name* form condition *A* whereas *attack name* is the outcome *B* of that condition [4].

Wild card values can be used in each of the fields of the rule. In the proposed implementation, wild card values are used for *source port* as well as in the third and fourth part of both *source IP* and *destination IP*. A (-1) is used in the field chosen for the wild card.

3.3 Fitness function

The fitness function of a rule is decided by checking the number of times it matches with the record connections [4]:

$$support = |A \text{ and } B| / N$$

$$confidence = |A \text{ and } B| / |A|$$

$$fitness = w_1 * support + w_2 * confidence$$

Where, N is the total number of connections, $|A|$ is the number of connections where the rule matches first six features of the connection. $|A \text{ and } B|$ represent the number of connections that match both condition part A and outcome part B of the rule. The weights w_1 and w_2 can be adjusted to fine tune the algorithm and have been set to values of $w_1 = 0.2$ and $w_2 = 0.8$.

4. RESULTS AND DISCUSSION

This section shows the effect of GA parameters on intrusion detection. The effect of various GA parameters namely population size, GA generations, selection scheme and crossover and mutation rate are presented. In addition, the effect of number of attack samples in data set is also presented.

The default parameter values used in GA runs are as follows. Two–point crossover with crossover rate = 0.5, uniform mutation with mutation rate = 0.01 and Roulette wheel selection mechanism.

The results were obtained using desktop pc having dual core processing running at 3.00 GHz at 4GB RAM on windows 7 using JAVA programming in NetBeans 7.0 environment.

4.1 Population size

The first step of any genetic algorithm is random population generation. In the presented approach, the initial population is generated using enumeration technique [12]. The GA is run for 2000 generations. The population size is varied in different GA runs and its effect on the detection accuracy is noted. Table 1 presents different population sizes and the corresponding detection accuracy. It is also represented in fig.1. It is observed that the detection accuracy increases with increase in population size and remains almost constant after population size of 300.

Table 1 Detection accuracy for different GA population sizes

Sr. No	Population size	Detection accuracy (%)
1	50	55.0
2	100	63.0
3	150	75.2
4	200	83.0
5	250	87.0
6	300	98.0
7	350	97.9
8	400	97.8
9	450	97.7
10	500	97.5
11	550	97.6
12	600	97.7
13	650	97.7
14	700	97.8
15	750	97.8

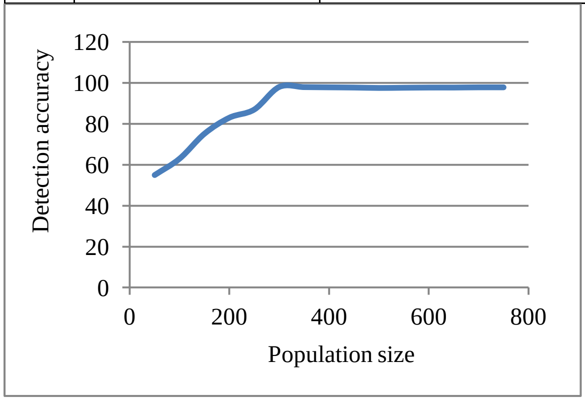

Fig.1 Effect of population on detection accuracy

The time taken by a genetic algorithm to reach to a required solution is an important aspect. This execution time increases linearly as the population size increases for the equal number of generations. Table 2 shows the population size and corresponding execution time taken by the GA. It is also represented in fig.2. The maximum number of generations is set to 2000.

Table 2 Execution time against population size

Sr. No	Population size	Execution time (Hours)
1	50	10
2	100	16
3	150	22
4	200	26
5	250	34
6	300	41
7	350	47
8	400	52
9	500	61
10	600	68
11	700	73
12	800	77
13	900	85
14	1000	92

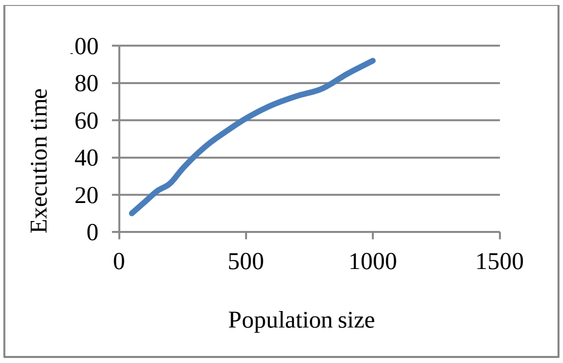

Fig.2 Effect of GA population size on execution time

4.2 Number of Generations

As the GA run progresses, the accuracy of intrusion detection generally improves until maximum accuracy is obtained. Often, GA may also land in local maxima unless GA parameters are properly set. Table 3 shows the percentage detection for different number of generations of GA for the population size of 300. It is also represented in fig.3. As the number of generations is increased, the detection rate is improved. The best results are obtained after 2000 generations.

Table 3 Number of evaluations against detection accuracy

Sr. No	Generations	Detection accuracy (%)
1	500	76.6
2	1000	82.0
3	1500	92.2
4	2000	98.0

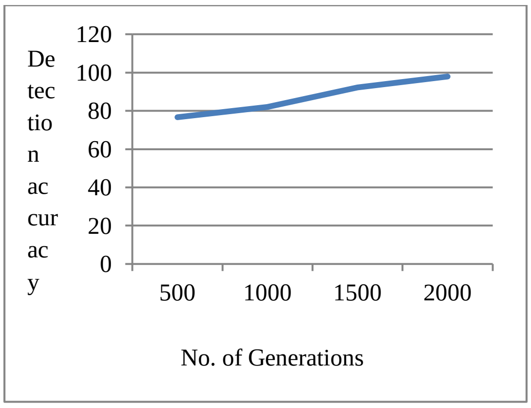

Fig.3 Effect of generations on detection accuracy

4.3 Selection scheme

Table 4 shows the percentage detection accuracy for three different types of selection techniques. The population size is 300 and GA generations are set to 2000. It is observed that the Roulette wheel selection gives higher detection accuracy of 98% closely followed by Rank selection.

Table 4 Maximum detection accuracy for various selection schemes

Sr. No	Selection type	Detection accuracy (%)
1	Roulette wheel selection	98
2	Rank selection	97
3	Steady-state selection	94

4.4 Crossover rate

In the proposed approach, two point crossover technique is used. Crossover probability or crossover rate is a ratio of number of chromosomes that undergo mating. The table 5 gives the crossover rate and the corresponding detection rate. The population size and number of generations are set to 300 and 2000 respectively. Fig.4 shows the effect of crossover rate on detection accuracy graphically. It is observed that highest detection accuracy is obtained for crossover rate of 0.6.

Table 5 Detection accuracy against crossover rate

Sr. No	Crossover rate	Detection accuracy (%)
1	0.90	94.0
2	0.85	95.0
3	0.80	95.5
4	0.75	96.0
5	0.70	96.2
6	0.65	97.0
7	0.60	98.0
8	0.55	97.0
9	0.50	96.3
10	0.45	95.0
11	0.40	94.2

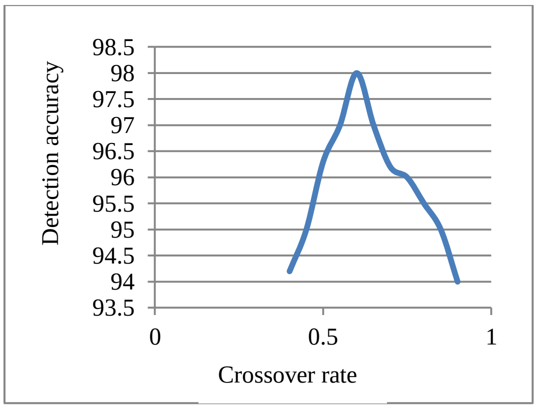

Fig.4 Effect of crossover rate on detection accuracy

4.5 Mutation rate

In the proposed approach, uniform mutation technique is used. Mutation rate or mutation probability is a measure of the likeness that random elements of the chromosome will be flipped into some other valid value. For example if the chromosome is encoded as a binary string of length 100 and if the mutation probability is 2%, then 2 out of 100 bits (on average) picked at random will be flipped.

Table 6 shows different mutation rate and corresponding detection accuracy. The population size is 300 and the maximum number of generation is set to 2000. Fig. 5 shows the effect of mutation rate on detection accuracy. It is observed that detection accuracy is maximum for mutation rate of 0.01.

Table 6 Detection accuracy against mutation rate

Sr. No	Mutation rate	Detection accuracy (%)
1	0.01	97.1
2	0.02	97.0
3	0.03	96.5
4	0.04	95.4
5	0.05	95.3
6	0.1	96.5
7	0.2	96.4
8	0.3	96.0
9	0.4	95.6
10	0.5	95.0

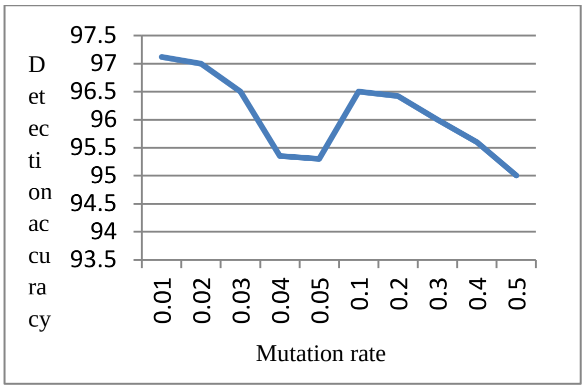

Fig.5 Effect of mutation rate on detection accuracy

4.7 Number of attack samples in a data set

The amount of attack samples present in a data set also plays an important role in intrusion detection. Table 7 shows the attack samples in a data set and the corresponding detection accuracy. It is also shown in fig.6. It is observed that, as the percentage of attack samples in a data set is increased up to 8%, the detection accuracy decreases. For further addition of attack samples, the detection accuracy increases linearly.

Table 7 Attack samples in a data set against Detection accuracy

Sr. No	Attack samples in a data set (%)	Detection accuracy (%)
1	2	95
2	4	93
3	6	90
4	8	89
5	10	92
6	12	93
7	14	94
8	16	95
9	18	96
10	20	97

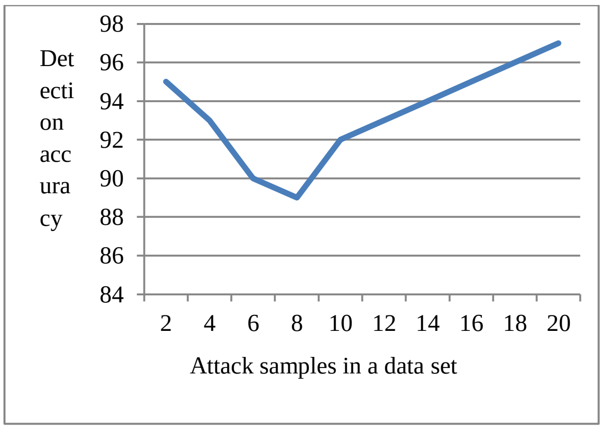

Fig.6 Effect of attack samples in a data set on detection accuracy

5. CONCLUSION

As the population size increases, the detection accuracy also increases. The maximum detection rate is obtained for the population size of 300. But after this population, there is no significant improvement in the detection rate. As the population size increases, the execution time of GA also increases. As the number of generations is increased, the detection rate is improved.

The highest detection accuracy in various GA runs is achieved while using Roulette wheel selection, two-point crossover with crossover rate of 0.6 and uniform mutation with mutation rate of 0.01. Further the detection accuracy was observed to be dependent on the number of attack samples in a data set.

6. REFERENCES

[1] Ko, Calvin, M. Ruschitzka and K. Levitt, "Execution monitoring of security-critical programs in distributed systems: A specification-based approach", Security and Privacy, Proceedings. IEEE symposium. IEEE, 1997.

[2] S. Owais, V. Snasel and P. Kromer, A. Abraham, “Survey Using Genetic Algorithm Approach in Intrusion Detection Systems Techniques”, 7th Computer Information Systems and Industrial Management Applications, 2008, IEEE press, June 2008, pp.300-307, DOI 10.1109/CISIM. 2008.49.

[3] Y. Wang, D. Gu, X. Tian and J. Li, “Genetic Algorithm Rule Definition for Denial of Services Network Intrusion Detection”, International Conference on Computational Intelligence and Natural Computing, IEEE, 2009, pp.99-102.

[4] R. H. Gong, M. Zulkernine, P. Abolmaesumi, “A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection”, SNPD/ SAWN’ 05, IEEE, 2005.

[5] S. Mukkamala and A. H. Sung, “A Comparative Study of Techniques for Intrusion Detection”, Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’03), IEEE, 2003.

[6] S. Ashfaq, M.U. Farooq and A. Karim, “Efficient Rule Generation for Cost-Sensitive Misuse Detection Using Genetic Algorithms,” IEEE, 2006.

[7] T. Xia, G. Qu, S.Hariri and M. Yousif, "An efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm", IEEE, 2005.

[8] M. Middlemiss and G. Dick, "Weighted Feature Extraction Using a Genetic Algorithm for Intrusion Detection", 2003 Congress on Evolutionary Computation (cec-03) 2003, pp.1669-1675.

[9] C. H. Lee, S.W. Shin and J. W. Chung, "Network Intrusion Detection through Genetic Feature Selection", SNPD, IEEE, 2006.

[10] B. Mukherjee, L.T. Herberlein and K. N. Levitt, "Network Intrusion Detection", IEEE Network, 8(3):26-41, May/June 1994.

[11] MIT Lincoln Laboratory, DARPA datasets, MIT, USA, http://www.ll.mit.edu/mission/communications /ist/ corpora/ideval/data/1998data.html

[12] S. N. Pawar and R. S. Bichkar, "Using Enumeration in a GA based Intrusion Detection", International Journal of Computer Applications (IJCA), October- 2012.