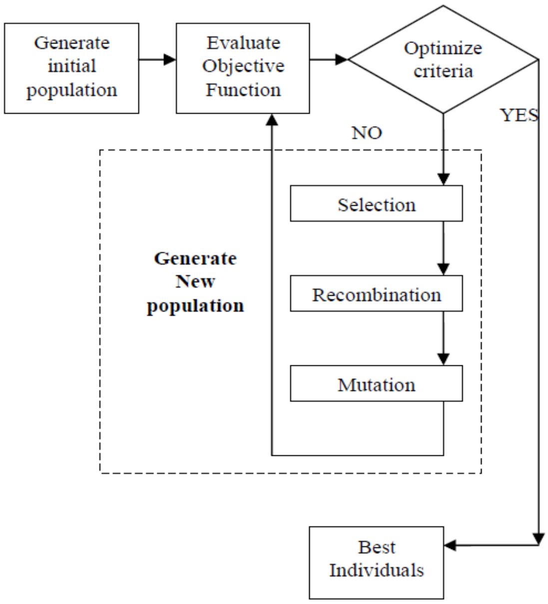

Figure 3.1 Genetic Algorithm Flow

Next, a number of individuals are selected and paired with each other. Each individual pair produces one offspring by partially exchanging their genes around one or more randomly selected crossing points. That is the selection of chromosomes for survival and combination is biased towards the fittest chromosomes [8].

At the end, a certain number of individuals are selected and the mutation operations are applied, i.e., a randomly selected gene of an individual abruptly changes its value.

When a GA is used for problem-solving, three factors will have impact on the effectiveness of the algorithm[3], also the determination of these factors often depends on applications. They are: the selection of fitness function, the representation of individuals and the values of the GA parameters.

4. Intrusion Detection System using Genetic Algorithm

4.1Encoding

Genetic algorithms can be used to evolve simple rules for traffic in networks. These rules are used to distinguish normal network connections against anomalous connections. These anomalous connections refer to events with probability of intrusions. The rules stored in the rule base are usually in the following syntax:

If {condition} then {act}

Here the condition usually refers to a match between current network connection and the rules in IDS, such as source and destination IP addresses and port numbers (used in TCP/IP network protocols), duration of the connection, protocol used, etc., indicating the probability of an intrusion.

The features in the condition part are connected using logical AND operator. The act field usually refers to an action defined by the security policies within an organization, such as reporting an alert to the system administrator, stopping the connection, logging a message into system audit files, or all of the above. There are some networks features have higher possibilities to be involved in network

Classification rule. Table 4.1 shows the feature name, number of genes and their formats in first, second and third respective columns.

Table 4.1: Selected Network Feature

Feature Name	*Number of genes*	*Format*
Source IP address	4	a.b.c.d
Destination IP address	4	a.b.c.d
Source Port Number	1	Int
Destination Port Number	1	Int
Duration	3	h:m:s
State	1	Int
Protocol	1	Int
Number of Bytes Sent by Originator	1	Int
Number of Bytes sent by Responder	1	Int

Different genes can be represented in different data types such as byte, integer, and float. This is necessary because of different formats and data ranges for different features, as shown in table 4.2.

For example, the feature "Duration" has three components (hours, minutes, and seconds), each of which is represented by one gene of type byte. Similarly, each of the features "Protocol", "Source port", "Destination port" and "Attack name" is encoded using one gene of type integer, and each of the features "Source IP" and "Destination IP" has four components (a, b, c, and d), each of which is represented by one gene of type byte.

4.2Data Representation

In order to fully exploit the suspicious level, we need to examine all fields related with a specific network connection. For simplicity, we only consider some obvious attributes for each connection.

(d, 1, 0,b, -1, -1, -1, -1, 8, 2, 1, 2,b, -1, -1, -1, 4,2, 3, 3, 5, 0,0, 0, 8,0, 0, 0,0, 0, 0,4, 8, 2,1, 1, 2,0, 0, 0, 0, 0,0, 7, 3,2, 0, 0,0, 0, 0,0, 3, 8,8, 9, 1)

Figure 4.2 Chromosome structure for example

The definition of rules (for TCP/IP protocols) is shown in Table 4. 2. The corresponding rule for the "Example Value" attribute in Table 4.2 could be translated as:

if {the connection has following information: source IP address 209.11.??.??; destination IP address: 130.18.176+?.??; source port number: 42335; destination port number: 80; connection time: 482 seconds; the connection is stopped by them originator; the protocol used is TCP; the originator sent 7320 bytes of data; and the responder sent 38891 bytes of data } then {stop the connection}

Table 4.2: Rule definition for connection and range of values of each field

Attribute	Range of Values	Example Values	Descriptions
Source IP address	0.0.0.0~255.255.255.255	d1.0b.**.** (209.11.??.??)	A subnet with IP address 209.11.0.0 to 209.11.255.255
Destination IP address	0.0.0.0~255.255.255.255	82.12.b*.** (130.18.176+?.??)	A subnet with IP address 130.18.176.0 to 130.18.255.255
Source Port Number	0~65535	42335	Source port number of the connection
Destination Port Number	0~65535	00080	Destination port number, indicates this is a http service
Duration	0~99999999	00000482	Duration of the connection is 482 seconds
State	1~20	11	The connection is terminated by the originator, for internal use
Protocol	1~9	2	The protocol for this connection is TCP
Number of Bytes Sent by Originator	0~9999999999	0000007320	The originator sends 7320 bytes of data

In the example shown in Table 4.2, some wild cards (the '*' character and the '?' character) are used and the corresponding genes within the chromosome are shown as –1. These wild cards are used to represent an appropriate range of specific values. It is useful when representing a network block (a range of IP addresses or port numbers) in a rule.

4.3 System Architecture

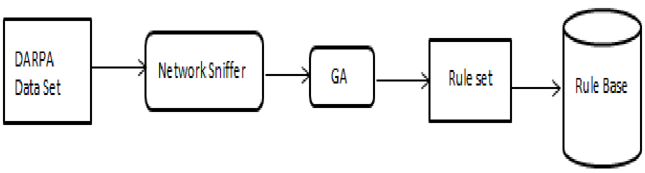

Figure 4.2: Architecture of applying GA into intrusion detection

The network traffic used for GA is a pre-classified **data set** that differentiates normal network connections from anomalous behavior. First we have enough historical data which contains normal as well as anomalous network connections. This data set is gathered using **network sniffers**. It is used for the fitness evaluation during the execution of GA. By starting **GA** with only a small set of randomly generated rules, we can generate a larger data set that contains rules for IDS. These rules are good solutions for GA and can be used for filtering new network traffic.GA can evolve randomly selected population of rules, further with crossover and mutation operators. The fitness function is biased towards the rule that matches anomalous connection. When algorithm satisfied some predefined criteria of stopping, then rules are selected and added to IDS rule base.

5. Parameters in Genetic Algorithm

There are many parameters like number of populations, number of generations, selection criteria, fitness function, crossover rate, mutation rate etc. to consider for the application of GA. Each of these parameters has heavily influences on the effectiveness of the genetic algorithm. These parameters should be adjusted according to the application environment of the system and the organization's security policy.

Genetic Algorithm contains of a sequence of operations, which are: Selection, Crossover Mutation and sometimes Replacement, but the first operation is depending on the fitness value that obtained by Fitness Function.

The main problem of GA is to find a suitable Fitness Function for a chromosome evaluation to get a solution for Intrusion Detection. So we concentrate on fitness function and analyses different fitness function formulae as follow:

1. Genetic Algorithm to identify the attack connection, the algorithm used different features in network connections to generate a classification rule set [2], they used the fitness function given by the formula,

$$F = \frac{a}{A} - \frac{b}{B} \quad (1)$$

where,'A' is total number of attack connections, 'a' is number of attack connections the individual correctly classified,'B' is total number Normal connections in the population and 'b' is number of normal connections a network correctly classified.The fitness function value would lie in the region [-1.1]. A positive fitness value will denote that the individual classifies more number of attacks correctly than it does the normal ones.To select the fit individuals, they have set a threshold value of 0.95. Thus, all individuals that have a fitness value > 0.95 are selected to produce subsequent generations and are deemed fit.Fitness Scale=[-1,1].

2. Genetic Algorithm used for Intrusion Detection System. The following steps are used to calculate the fitness function[8],as follow:

Outcome=$\sum_{i=1}^{57} Matched * Weight$ (i)

Δ=*outcome–suspicioulevel* (ii)

penalty=(Δ*ranking)/100 (iii)

fitness=1-penalty (iv)

First, outcome is calculated based on whether field of connection matched the pre-classified data set & the multiply the weight of that field the value of matched is 0 or 1.Secondly, the absolute difference between the outcome of the chromosome and the actual suspicious level is then calculated. The suspicious level is a threshold that indicates the extent to which two network connections are considered a "match." The actual value of suspicious level reflects observations from historical data. Third step, penalty value is computed if mismatch happens, the ranking in the equation indicates whether or not an intrusion is easy to identify.

Finally in fourth step the value of fitness value is computed. Fitness scale=[0,1].

3. Determination of the fitness of a rule[9], they use Support-Confidence Framework which identifies network intrusions or precisely classifies types of intrusion.

Support = |A and B| / N

Confidence=|AandB|/|A|

Fitness=w1*support+w2 * confidence

Here',N' is the total number of network connections in the audit data, '|A|' stands for the number of network connections matching the condition A, '|A and B|' is the number of network connections that matches the rule if A then B.The weights w1 and w2 are used to control the balance between the two terms and have default values of w1=0.2 and w2=0.8.They set threshold value 0.Fitness scale=[0,1].

4)Reward Penalty based fitness function presented[10].The basic idea behind it is that chromosomes vary in their strength and weakness . Hence fitness function must takes two points to consideration; first, the reward must be as more as the chromosome strength, and the penalty must be as more as the chromosome weakness.

If *c&a* of selected record= *c&a* of compared record, then AB+1.Else if *c* of selected record= *c* of compared record but not *a*, then A+1.('*c*' is condition and '*a*' is action) The reward-penalty fitness function is as the following:

$$\text{Fitness} = 2 + \frac{AB-A}{AB+A} + \frac{AB}{X} - \frac{A}{Y}$$

$$\text{Fitness} = 2 + \frac{AB}{AB+A} - \frac{A}{AB+A} + \frac{AB}{X} - \frac{A}{Y}$$

Where= the maximum value AB in the population= the maximum value A in the population. (AB/(AB+A)) will reflect the strength of the record (A/(AB+A)) will reflect the weakness of the record. We take strength minus weakness as in the function above. For example, as shown in table 6 as follow,

Table 6: Example of record of fitness function

Record	A	AB	Fitness=((AB-A)/(AB+A))
Record1	0	1	1
Record2	0	5	1

AB/X: gives the rate which reflects the strength of the record depending on the most strongest record in the population.The resulted value= 0 in the worst case (if AB value = 0) and 1 in best case (if the AB has highest value in the population), so it is logically should be added to the function to reward the record.

A/Y: gives the rate which reflects the weakness of the record depending on the most weakness record in the population.The resulted value=0 in the best case (If A value =0) and 1 in the worst case (if the A has highest value in the population), so the value of A/Y must be subtracted from the function to give the penalty on the record.

Now, assume that the record with Best case, i.e. AB has highest value & A=0 this means that Fitness = 2, On other hand, the record with Worst case, i.e. A has highest value & AB=0 this means that Fitness = -2. But the fitness value provided by the fitness function must assign a non-negative cost to each candidate, so the constant value of 2 will be added to the function to get fitness value equal to 0 in the worst case and fitness value equal to 4 in the best case.Fitness scale=[0,4].

Selection

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a *fitness-based* process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Two chromosomes are selected from the population to be parents to crossover.

Crossover (Recombination):

Crossover creates one or more new offspring from parent chromosomes to get better chromosomes.

Table 7: Crossover example in binary format (| is the crossover point)

Chromosome1	11011 | 00100110110
Chromosome2	11011 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11011 | 00100110110

Mutation

Mutation changes randomly the new offspring. This is to prevent falling all solutions in population into a local optimum of solved problem.

Table 8: Mutation example in binary format

Original offspring	1101111000011110
Mutated offspring	1100111000011110

6. Conclusion

Applying genetic algorithm to network intrusion detection techniques is presented.GA is used to derive a set of classification rules from network audit data. Some network features including both categorical and quantitative data fields were used when encoding and deriving the rules.GA used as an appealing tool in the search for intrusions in audit trail files. The main goal of GA is to create rules that match only the anomalous connections. These rules are tested on historical connections and are used to filter new connections to find suspicious network traffic. Also parameters and evolution processes for GA is presented.

We suggest reward penalty based fitness function which evaluates population of chromosomes efficiently. This fitness function able to get good results in using GA for misused intrusion detection systems. If crossover rate is greater, then record with highest fitness value get selected more probably during selection step of GA and we can get better result.

Future work includes generating a standard test data set for the genetic algorithm and applying it to a test environment. Detailed specification of parameters to consider for genetic algorithm should be determined during the experiments. Combining knowledge from different security sensors into a standard rule base is one field of research. Also additional work needed to experiment with use of different types of crossover and mutation.

References

[1] ChSatyaKeerthi N.V.L., Prasanna P.I., Priscilla B.M., ”Instuction Detection system Using Genetic Algorithm”,Int. Journal of P2P Network rends and Technology, vol.1.no. 2.pp 1-7, 2011.
[2] Goyal A., Kumar C., “GA-NIDS: A Genetic Algorithm based Network Instruction Detection System”, 2008
[3] Mohammad S. H., MukitMd.A.,BikasMd.A. N.,” An Implementation of Intrusion Detection System Using Genetic Algorithm”, Int.Journal of Network Security and Its Applications, vol.4 no.2. pp 109-119, 2
[4] Jiang M., Munavar M., Reidemeister T., Ward P.,”Efficient Fault Detection and Diagnosis in Complex Software Systems with Information- Theoretic Monitoring”IEEE Trans.On Dependable and Secure Computing.Issue 99, 2011.
[5] Chittur A.,”“Model Generation for an Intrusion Detection System Using Genetic Algorithms”, 2011.
[6] Lu W., Traore I.,“ Detecting New Forms of Network Instruction Using Genetic Programming”, Computational Intelligence, vol. 20, pp. 3, Blackwell Publishing, Malden, pp. 475-494, 2004.
[7] Pedro A. Diaz-Gomez and Dean F. Hougen ”Three Approaches to Intrusion Detection Analysis And Enhancements”, National Computer And Information Security Conference Acis2006 .
[8] Li W. “Using Genetic Algorithm for Network Intrusion Detection”, Proceedings of the United States Department of Energy Cyber Security Group, 2004.
[9] Gong R. H., Zulkernine M., Abolmaesumi P.,”A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection”, 2005.
[10] Alabsi,F., Naoum,R., "Fitness Function for Genetic Algorithm used in Intrusion Detection System",International Journal of Applied Science And Technology.Vol. 2.no 4, 2012.
[11] Kandeeban,S.S., Rajesh R.S.,”A Mutual Construction For IDS Using GA”, Int. Journal of Advance Science And Technology,vol.29, 2011.
[12] Uppalaiah B., Anand K., Narsimha B., warajS., BharatT., ”Genetic Algorithm Approach to Intrusion Detection System”, IJCST vol.3.1,2012.
[13] Owais S.S.J., Kromer P., Snasel V., "Implementing GP onoptimizing Boolean and Extended Boolean Queries in IRs withRespectto Users Profiles", Proc. IEEE ICCES'06 Egypt. pp412-417.2006
[14] Owais S.,Kromer P.,Snasel V., "Query Optimization By Genetic algorithms”, DATESO ISBN: 80-01- 03204-3,pp125-137,2005.