Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 2 (2017) pp. 235-246

http://www.ripublication.com

Intrusion Detection System for NSL-KDD Data Set using Vectorised Fitness Function in Genetic Algorithm

Partha Sarathi Bhattacharjee
Assam University, Silchar
Cachar, Assam, India.

Abul Kashim Md Fujail
M H C M Sc College
Hailakandi, Assam, India

Shahin Ara Begum
Assam University, Silchar
Cachar, Assam, India.

Abstract

With rapid increase in the use of network computers over last few decades, there has been increase in many different types of network attacks by intruders. To detect different network attacks, Genetic Algorithm (GA) based Intrusion Detection System (IDS) is employed in this paper. The objective is to find a suitable Vectorised Fitness function for chromosome evaluations to get a solution for IDS. To achieve this objective, GA based IDS with weighted Vectorised Fitness function is proposed and evaluated over the NSL-KDD data set. In the present work, Fuzzy membership function is used with Vectorised Fitness function in GA for efficient intrusion detections. The experimental results show that the proposed Fuzzy Vectorised GA performs better than the Vectorised GA and Weighted Vectorised GA in detecting network attacks for the considered NSL-KDD data set.

Keywords: IDS, Genetic Algorithm, Vectorised Fitness Function, Fuzzy Membership function, NSL-KDD Data Set.

I. INTRODUCTION

Intrusion Detection System (IDS) is used to monitor network traffic and suspicious activity and alerts the system or network administrator. In some cases, the IDS may

also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network [1].

There are network based and host based intrusion detection systems. Some IDS detect by looking for specific signatures of the known threats similar to the way antivirus software typically detects and protects against malware. There are also a different type of IDS that detect attacks based on comparing traffic patterns against a baseline and looking for anomalies. There are IDS that simply monitor and alert whereas some other IDS perform an action or actions in response to a detected threat [2]. In this paper an attempt has been made to develop an Vectorised Fitness Function based Genetic Algorithm for Intrusion Detection System for NSL-KDD data set.

II. TYPES OF INTRUSION DETECTION SYSTEM

IDS can be classified into the following two categories [3]:

a) Signature Based IDS

A signature based IDS monitors packets on the network and compares them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a gap between a new threat being discovered and the signature for detecting that threat being applied to IDS [4].

b) Anomaly Based IDS

An IDS which is anomaly based, monitors network traffic and compares it against an established baseline. The baseline will identify what is "normal" for that network, what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other and alert the administrator or user when traffic is detected which is anomalous, or significantly different, than the baseline. IDS come in a variety of approach for detecting suspicious traffic in different ways [5].

III. GENETIC ALGORITHM

It is adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. GAs represents an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space [6].

Vectorised Fitness function is a mathematical optimization function introduced by Howard H. Rosenbrock in 1960 [7]. The GA usually runs faster and produces optimized value if vectorized fitness function is used. This means that the GA only calls the fitness function once, but expects the fitness function to compute the fitness for all individuals in the current population at once [8][9].

IV. FUZZY MEMBERSHIP FUNCTION

The Fuzzy membership function is a graphical representation of the magnitude of participation of each input. It associates weighting with each of the inputs that are processed. Figure 1 shows the graphical representation of membership functions for short, medium and tall [10].

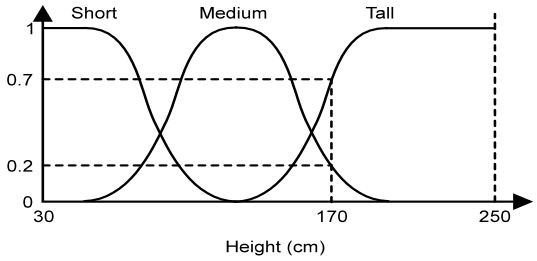

Figure 1: Fuzzy Membership functions

V. GENETIC ALGORITHM WITH RESPECT TO VECTORISED FITNESS FUNCTION

Genetic Algorithm with respect to Vectorised Fitness function (abbreviated as VGA) is implemented as follows [11]:

Step 1: The GA starts with population of individuals generated randomly.

Step 2: The vectorised fitness function is adopted which can be expressed as follows:

$$y = a \times (x_{ij}^2 - x_{ik})^2 + (b - x_{ij})^2$$

where, $i = 1, 2, 3,..., n$ is the number of rows and $j = 1$, $k = 2$ and x is a matrix, (here it is NSL-KDD data set). The optimization function is defined by Rosenbrock which is known as Rosenbrock's Banana function, where, the values of a and b is usually defined as a=1 and b=100.

Step 3: A matrix x (here, it is NSL-KDD data set) with an arbitrary number of points is taken, the rows of x and returns a column vector y, with the same number of rows as x. Here, the final fitness value (fz_out) of the vectorised fitness function and the points f (f1 & f2) at which the final fitness values are attained as given below by calling a MATLAB function ga( ) by passing two arguments - the vectorised fitness formula and number of variables which is defined as two because first two columns from the matrix/data set is taken here.

[f, fz_out] = ga(Vectorised Fitness Function, Number of Variables)

Step 4: The number of different types of attack is identified based on the points where the final value of fitness is attained.

Step 5: After evaluating the fitness of the individuals of the population, a new population may be created. This can be accomplished by selection, crossover and mutation operations until a termination criteria is satisfied.

Experimental results of GAs with respect to Vectorised Fitness Function for NSL-KDD data set [12] is presented in Tables 1-2. The tables represent the number of different types of attack detected with Vectorised GA over 20 percent and full dataset using 42 features. Figure 2 represents the points (f1 and f2) at which the final fitness value is attained and the fitness value (fz_out) of the vectorised fitness function.

Table 1: Attack detection results using GA with Vectorised Fitness for NSL-KDD 20 percent data set using 42 features

Genetic Algorithm	Fitness Points	Normal	Dos	Probe	R2l	U2R
VGA	0.9652 (f1)	210	35	63	3	0
VGA	0.9340 (f2)	124	20	64	4	0

Table 2: Attack detection results using GA for Vectorised Fitness for NSL-KDD full data set using 42 features

Genetic Algorithm	Fitness Points	Normal	Dos	Probe	R2l	U2R
VGA	0.9652 (f1)	928	143	329	8	0
VGA	0.9340 (f2)	639	98	380	8	0

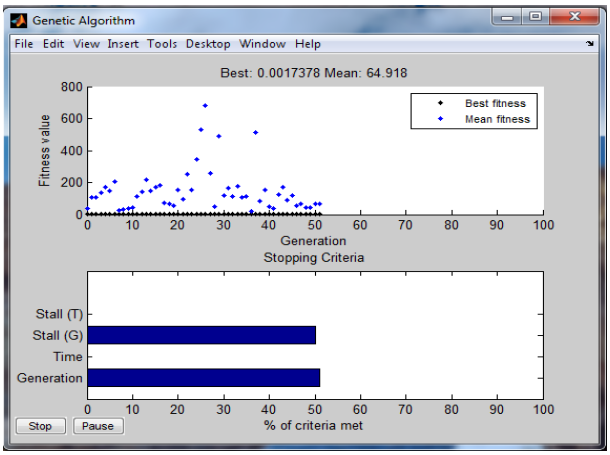

Figure 2: f1 = 0.9652 and f2 = 0.9340 fz_out = 0.0017 with VGA for NSL-KDD full data set using 42 features

From Table 1 and Table 2, it is observed that for the fitness point value 0.9652, the maximum number of attack is identified. The number of normal attack identified for NSL-KDD 20 percent data set is 210 and for NSL-KDD full data set, it is 928.

VI. GENETIC ALGORITHM WITH RESPECT TO WEIGHTED VECTORISED FITNESS FUNCTION

Genetic Algorithm with respect to Weighted Vectorised Fitness function (abbreviated as WV-GA) is implemented as follows:

Step 1: A population of individuals are generated randomly.

Step 2: The weighted vectorised fitness value (y) is calculated as follows :

$$y = w_1 \times a \times (x_{ij}^2 - x_{ik})^2 + w_2 \times (b - x_{ij})^2$$

where, i = 1, 2, 3,.., n is the number of rows, j = 1, k = 2 and x is a matrix, (here it is NSL-KDD data set). $w1$ = 0.2 and $w2$ = 0.8 are the weights. The values of $w1$ and $w1$ are assigned in such a way that the summation equals to one, so that the fitness points where the optimum fitness value is reached can be acquired and a=1 and b=100.

Step 3: A matrix x (here, it is NSL-KDD data set) with an arbitrary number of points is taken, the rows of x and returns a column vector y, with the same number of rows as x. Here, the final fitness value (fz_out) of the vectorised fitness function and the points f (f1 & f2) at which the final fitness values are attained as given below by calling a MATLAB function ga( ) by passing two arguments - the vectorised fitness formula and number of variables which is defined as two because first two columns from the matrix/data set is taken here.

[f, fz_out] = ga(Vectorised Fitness Function, Number of Variables)

Step 4: The number of different types of attack is identified based on the points where the final value of fitness is attained.

Step 5: After evaluating the fitness of the individuals of the population, a new population may be created. This can be accomplished by selection, crossover and mutation operations until a termination criteria is satisfied.

The results obtained with GA with respect to Weighted Vectorised Fitness Function for NSL-KDD data set is presented in Tables 3-4. The number of different types of attack identified in NSL-KDD 20 percent and full data set using 42 features are

tabulated in Table 3 and Table 4 respectively. Figure 3 shows the points (f1 and f2) at which the final fitness value is attained and the fitness value (fz_out) of the weighted vectorised fitness function.

Table 3: Attack detection results using GA with Weighted Vectorised Fitness for NSL-KDD 20 percent data set using 42 features

Genetic Algorithm	Fitness Points	Normal	Dos	Probe	R2l	U2R
WV-GA	0.9547 (f1)	196	25	88	2	0
WV-GA	0.9186 (f2)	94	18	93	1	0

Table 4: Attack detection results using GA for Weighted Vectorised Fitness for NSL-KDD Full data set using 42 features.

Genetic Algorithm	Fitness Points	Normal	Dos	Probe	R2l	U2R
WV-GA	0.9547 (f1)	849	115	372	13	0
WV-GA	0.9186 (f2)	493	91	512	7	0

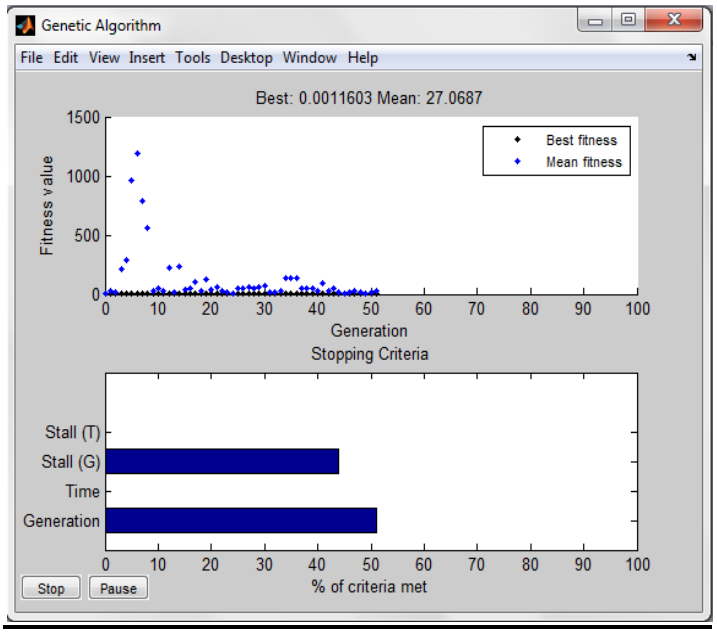

Figure 3: f1 = 0.9547 and f2 = 0.9186
fz_out = 0.0012 with WV-GA for NSL-KDD full data set using 42 features

It is seen from Table 3 and Table 4 that the maximum number of attack is identified at the fitness point 0.9547. The number of normal attack for NSL-KDD 20 percent data set and NSL-KDD full data set are 196 and 849 respectively.

VII. GENETIC ALGORITHM WITH RESPECT TO FUZZY VECTORISED FITNESS FUNCTION

Genetic Algorithm with respect to Fuzzy Vectorised Fitness function (abbreviated as FV-GA) is implemented as follows:

Step 1: A population of individuals is generated randomly.

Step 2: Considering NSL-KDD data set as X every gene/cell of the data set as defined x and the fuzzy membership function as μ(x) for a fuzzy set A. Then

$$A = \{(x, \mu(x)) \mid x \in X\}$$

Here, the following membership function for Fuzzy Vectorised Fitness function is considered :

$$\mu(x_{im}) = 1/[1 + (x_{im} - 1)^{-2}]$$

where, i = 1,2, 3, ..., n is the number of rows and m = 1 or 2

So, the Vectorised Fitness function for Fuzzy Genetic Algorithm may be represented as:

$$y = \mu(x_{ij}) \times a \times (x_{ij}^2 - x_{ik})^2 + \mu(x_{ik}) \times (b - x_{ij})^2$$

where, i = 1, 2, 3, ..., n is the number of rows, j = 1, k = 2, x is a matrix (NSL-KDD data set) and a = 1 and b = 100.

Step 3: A matrix x (here, it is NSL-KDD data set) with an arbitrary number of points is taken, the rows of x and returns a column vector y, with the same number of rows as x. Here, the final fitness value (fz_out) of the vectorised fitness function and the points f (f1 & f2) at which the final fitness values are attained as given below by calling a MATLAB function ga( ) by passing two arguments - the vectorised fitness formula and number of variables which is defined as two because first two columns from the matrix/data set is taken here.

[f, fz_out] = ga(Vectorised Fitness Function, Number of Variables)

Step 4: The number of different types of attack is identified based on the points where the final value of fitness is attained.

Step 5: After evaluating the fitness of the individuals of the population, a new population may be created. This can be accomplished by selection, crossover and mutation operations until a termination criteria is satisfied.

Experimental results of GAs with respect to Fuzzy Vectorised Fitness function for NSL-KDD data set is presented in Tables 5-6. The number of different types of network attack found in NSL-KDD 20 percent and full data set using 42 features are tabulated in Table 5 and Table 6. Figure 4 represents the points (f1 and f2) at which the final fitness value is attained and the fitness value (fz_out) of the fuzzy vectorised fitness function.

Table 5: Attack detection results using Fuzzy GA for Vectorised Fitness for NSL-KDD 20 percent data set using 42 features

Genetic Algorithm	Fitness Points	Normal	Dos	Probe	R2l	U2R
FV-GA	0.9902 (f1)	316	98	20	0	0
FV-GA	0.9918 (f2)	316	98	20	0	0

Table 6: Attack detection results using Fuzzy GA for Vectorised Fitness for NSL-KDD full data set using 42 features

Genetic Algorithm	Fitness Points	Normal	Dos	Probe	R2l	U2R
FV-GA	0.9902 (f1)	1733	577	111	0	0
FV-GA	0.9918 (f2)	1733	577	111	0	0

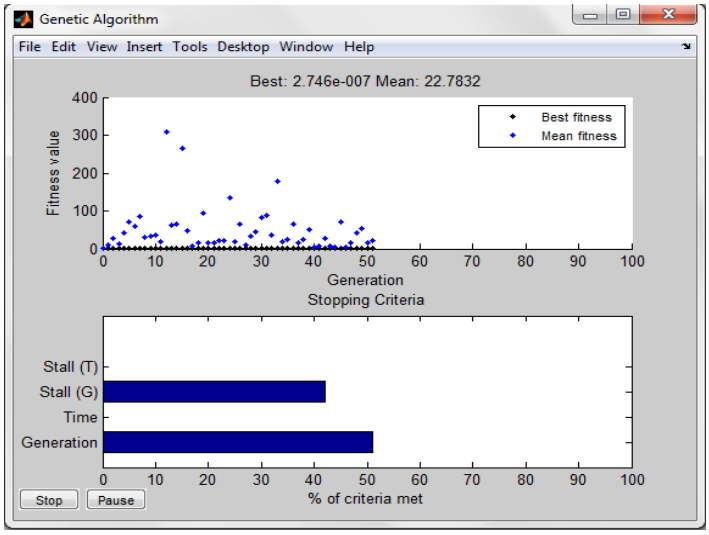

Figure 4: f1 = 0.9902 f2 = 0.9918
fz_out = 2.746e-007 with FV-GA for NSL-KDD full data set using 42 features

From Table 5 and Table 6, it is observed that for the fitness point values 0.9902 and 0.9918, the maximum number of attack is identified. The number of normal attack identified in the considered 20 percent and full data set are 316 and 1733 respectively.

VIII. COMPARISON BETWEEN VGA, WV-GA AND FV-GA WITH RESPECT TO ATTACK IDENTIFICATION

To find the best model, the results obtained with different models i.e. Vectorised GA, Weighted Vectorised GA and Fuzzy Vectorised GA are compared in this section. Table 7 summarizes the results obtained with Vectorised GA, Weighted Vectorised GA and Fuzzy Vectorised GA over the considered NSL-KDD data set.

Table 7: Attack detection results with respect to VGA, WV-GA and FV-GA for NSL-KDD 20 percent data set using 42 features

Genetic Algorithm	Fitness Points	Normal	Dos	Probe	R2l	U2R
VGA	0.9652 (f1)	210	35	63	3	0
VGA	0.9340 (f2)	124	20	64	4	0
WV- GA	0.9547 (f1)	196	25	88	2	0
WV-GA	0.9186 (f2)	94	18	93	1	0
FV-GA	0.9902 (f1)	316	98	20	0	0
FV-GA	0.9918 (f2)	316	98	20	0	0

From Table 7, it is observed that for Fuzzy Vectorised GA, the maximum number of attack is identified in comparison with Weighted Vectorised GA and Vectorised GA for NSL-KDD 20 percent data set. The number of normal attack detected by Fuzzy Vectorised GA for 20 percent data set is 316 which is much higher than the attack identified by VGA and WV-GA i.e. 210 and 196 respectively.

The graphical representation of the attacks identified with Vectorised GA, Weighted Vectorised GA and Fuzzy Vectorised GA are presented in Figure 5.

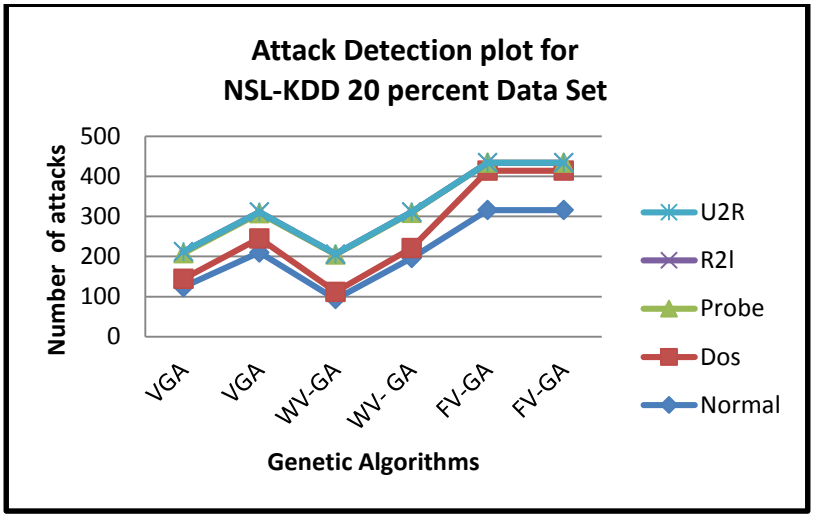

Figure 5: Attack Detection Plot with respect to Gas for NSL-KDD 20 percent data set using 42 features

To identify the best model with respect to full data set, the network attack identification results obtained with different models i.e. Vectorised GA, Weighted Vectorised GA and Fuzzy Vectorised GA are compared and the results summarized in Table 8.

Table 8: Attack detection results with respect to VGA, WV-GA and FV-GA for NSL-KDD full data set using 42 features

Genetic Algorithm	Fitness Points	Normal	Dos	Probe	R2l	U2R
VGA	0.9652 (f1)	928	143	329	8	0
VGA	0.9340 (f2)	639	98	380	8	0
WV- GA	0.9547 (f1)	849	115	372	13	0
WV-GA	0.9186 (f2)	493	91	512	7	0
FV-GA	0.9902 (f1)	1733	577	111	0	0
FV-GA	0.9918 (f2)	1733	577	111	0	0

Table 8 shows that the maximum number of attack identified with Vectorised GA, Weighted Vectorised GA and Fuzzy Vectorised GA for full data set. The highest number of normal attack is detected by Fuzzy Vectorised GA for NSL-KDD full data set i.e. 1733, for NSL-KDD full data set using 42 features.

Figure 6 shows the graphical representation of the attacks identified with Vectorised GA, Weighted Vectorised GA and Fuzzy Vectorised GA. From Figure 5 and Figure 6, it is seen that Fuzzy Vectorised GA detects higher number of attacks than VGA and WV-GA.

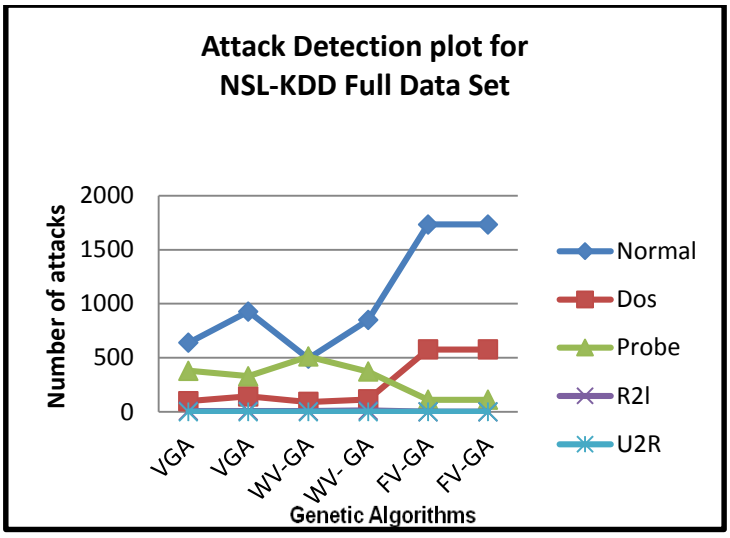

Figure 6: Attack Detection Plot with respect to GAs for NSL-KDD full data set using 42 features

IX. CONCLUSION

In this paper an Vectorised Fitness Function based Genetic Algorithm for Intrusion Detection System for NSL-KDD data set is developed. Genetic Algorithm with two variables Vectorised Fitness function is employed on NSL-KDD data set consisting of 42 features to detect different types of network attack. The experimental results obtained demonstrates that the total number of attacks detected with Vectorised GA, Weighted Vectorised GA and Fuzzy Vectorised GA varies and more number of attacks are detected with Fuzzy Vectorised GA, in comparison with Vectorised GA and Weighted Vectorised GA. The proposed Fuzzy Vectorised GA out performs the Vectorised GA and Weighted Vectorised GA in detecting network attacks over the considered NSL-KDD data set.

As a future work, from a data-centric perspective, intrusion detection may be considered as a data analysis process and data mining techniques like efficient feature selection may be applied to intrusion detection for building an efficient model for intrusion detection.

REFERENCES

[1] A.A. Ojugo, A.O. Eboka, O.E. Okonta, R.E Yoro and F.O. Aghware, "Genetic Algorithm Rule-Based Intrusion Detection System", 2012, CIS, Vol. 3, No. 8. ISSN 2079- 8407

[2] Ren Hui Gong, Mohammad Zulkernine, Purang Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection", 2005, Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence and Networking, IEEE

[3] Santosh Kumar Sahu, "A Detail Analysis on Intrusion Detection Datasets" IEEE International Advance Computing Conference (IACC) 2014 pg-1348-1353

[4] Neha Rai, Khushbu Rai, "Genetic Algorithm Based Intrusion Detection System", 2014, IJCSIT, Vol. 5 (4), pp.4952-4957

[5] Aaron M. Cramer,"Vectorizing Fitness Functions and Calling External Functions from Matlab", IEEE, CCECE-2016

[6] Jahnavi, .S. Vithalpura, H. M. Diwanji, "Analysis of Fitness Function in Designing Genetic Algorithm Based Intrusion Detection System", 2015, IJSRD, Vol. 3, Issue 1, ISSN (online): 2321-0613

[7] Rosenbrock function: http://www.utdallas.edu/~jwz120030/Teaching/GradSciCtg/MatlabDemos/PlottingSurfaces/ PlottingSurfaces.pdf

[8] Firas A labsi , Reyadh Naoum, "Fitness Function for Genetic Algorithm used in Intrusion Detection System", 2012, IJAST, Vol. 2 No. 4

[9] Mohammad Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas, "An implementation of Intrusion Detection System using Genetic Algorithm", 2012, International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2

[10] Y Bai, D Wang, "Fundamentals of Fuzzy Logic Control – Fuzzy Sets, Fuzzy Rules and Defuzzifications", Springer , 2006,pp.17-36

[11] Subhadip Samanta, "Genetic Algorithm: An Approach for optimization (Using MATLAB)", 2014, IJLTET, Vol. 3 Issue 3

[12] NSL-KDD data set: http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html