Development of Genetic-based Machine Learning for Network Intrusion Detection (GBML-NID)

Wafa' S.Al-Sharafat, and Reyadh Naoum

***Abstract*—**Society has grown to rely on Internet services, and the number of Internet users increases every day. As more and more users become connected to the network, the window of opportunity for malicious users to do their damage becomes very great and lucrative. The objective of this paper is to incorporate different techniques into classier system to detect and classify intrusion from normal network packet. Among several techniques, Steady State Genetic-based Machine Leaning Algorithm (SSGBML) will be used to detect intrusions. Where Steady State Genetic Algorithm (SSGA), Simple Genetic Algorithm (SGA), Modified Genetic Algorithm and Zeroth Level Classifier system are investigated in this research. SSGA is used as a discovery mechanism instead of SGA. SGA replaces all old rules with new produced rule preventing old good rules from participating in the next rule generation. Zeroth Level Classifier System is used to play the role of detector by matching incoming environment message with classifiers to determine whether the current message is normal or intrusion and receiving feedback from environment. Finally, in order to attain the best results, Modified SSGA will enhance our discovery engine by using Fuzzy Logic to optimize crossover and mutation probability. The experiments and evaluations of the proposed method were performed with the KDD 99 intrusion detection dataset.

***Keywords*—**MSSGBML, Network Intrusion Detection, SGA, SSGA.

I. Introduction

WITH the ever-increasing growth of computer networks and emergence of electronic commerce in recent years, computer security has become a priority. Since intrusions take advantage of vulnerabilities in computer systems or use socially engineered penetration techniques, intrusion detection is often used as another wall of protection. In addition, intrusion prevention techniques such as user authentication and Intrusion Detection system (IDS) is not an easy task due to the vastness of the network activity data and the need to regularly update the IDS to cope with new, unknown attack methods or upgraded computing environments. Thus, Intrusion Detection (ID) is becoming one of main technologies that can be used to monitor network traffics and identify network intrusions such as Denial of Service (DoS), Probe, User to Root (U2R) and Remote to User attacks (R2L). An IDS is a system used for detecting intrusions and reporting them accurately to the proper authority [1]. There are different taxonomies for IDSs which contain different categories. One of these categories classified IDS according to the source of audit data that will be used to detect possible intrusions which divided into two groups; Network intrusion detection (NID) and Host-based intrusion detection (HID) [2]. HID identifies the intruders by monitoring host-based traffic. While, NID refers to identify the intruders by monitoring network traffics (packets) and this can be accomplished by identify the attack by their known pattern (signature). Fig. 1 shows the computer network with IDSs [1].

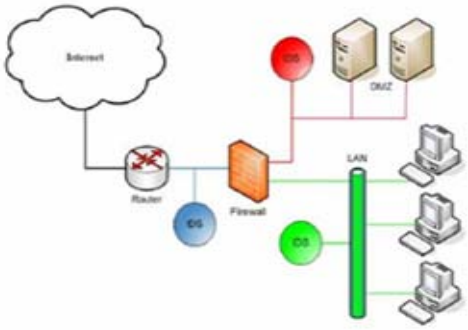

Fig. 1 Computer Network with Intrusion Detection System

A number of approaches have been proposed for detecting network intrusions. One of these approaches that has been used in [3] focuses on using different computing paradigm; neuro-fuzzy network, fuzzy inferences, and Genetic algorithm (GA) to detect intrusion activities to detect Network attacks. In [4][5] researchers tried build an application to enhance the knowledge domain to detect vast range of intrusion by using Machine Learning (ML) techniques to create rules of Expert system (ES) that can learn from dynamic environment to acquire expert knowledge to be adapted with new attacker behaviors. Knowledge is represented as a set of if-then rules. In this paper, we present a SSGBML, its modification, (MSSGBML) for network intrusion detection. SSGA is chosen because of its useful properties, e.g., robust to noise, no gradient information is required to find the global optimal or sub-optimal solution, self learning capabilities etc. The software is experimented by using KDD cup 99 data sets on

Wafa' S. Al-Sharafat is with Al Al-Bayt University, IT College, Jordan (e-mail: wafa@aabu.edu.jo).

Reyadh Naoum is with Arab Academy for Financial and Banking Science, IT Collage, Jordan.

intrusions, which has become the Benchmark for testing intrusion detection systems [6]. The experiments results will show the effectiveness of proposed Algorithm to detect network intrusion types.

II. Steady State Genetic-based Machine Learning Algorithm

Machine Learning (ML) is the study of computational methods for improving the performance of acquisition of knowledge from experience. Expert performance requires much domain specific knowledge and tries to produce ES that can be used in different domains such as industry. Thus, ML will reduce human time-consuming. In addition, this will increase the level of automation to improve the accuracy and the efficiency of detection systems; by discovering and exploiting regularities through training data [7]. SSGBML takes into account ability to learn from environment and adapted to the changes to be not restricted to static inputs to learn from. SSGBML merges ES and SSGA that enables learning from incomplete information. In addition, in the early stages for developing our algorithm SSGA show better detection rate rather than using SGA. Different combinations of inputs will be produced to perform rules in the form of: {condition} --> action. SSGA will play the role of discovery engine in SSGBML. SSGA is used to give a chance for previous rules from previous generation to participate in detecting intrusions in next generations. [11] has indicated that SSGA achieved better and faster solution than SGA. This lead us to use SSGA to generate rules from previous rules (act as parents), taking into account how the output will be closed to the problem solution. By using MSSGBML, False Positive Rate (FPR) will be reduced while Detection Rate will be improved. Fig. 2 provides an overview for MSSGBML. MSSGBML has a set of components which they are:

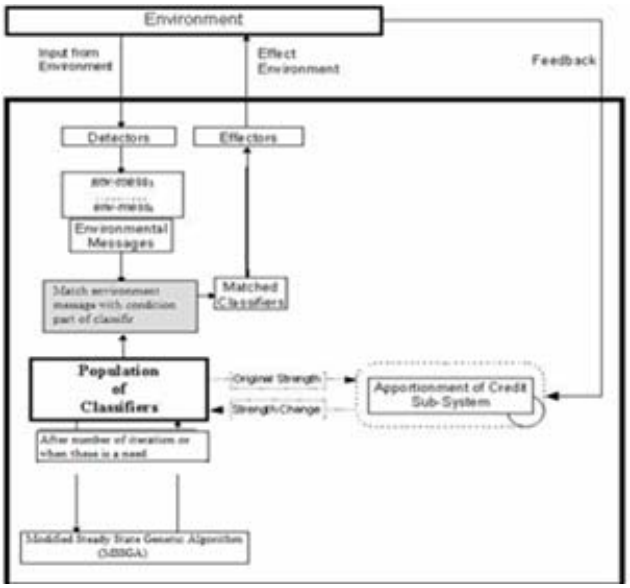

Fig. 2 MSSGBML

TABLE I
Network Attack Feature Classes

Class #	Attack type: Features
Class 1	DoS: 5,10,24,29,33,34,38,40
	Probe: 2,3,23,34,36,40
	U2R: 3,4,6,14,17,22
	R2L: 3,4,10,23,33,36
Class 2	DoS: 1,2,3,4,5,6, 12,23,24,31,32,37
	Probe: 1,2,3,4,12,16,25,27,28,29,30,40
	U2R:1,2,3,10,16
	R2L: 1,2,3,4,5,10,22
Class 3	DoS:1,5,6,23,24,25,26,32,36,38,39
	Probe: 1,2,3,4,5,6,23,24,29,32,33
	U2R: 1,2,3,4,5,6,12,23,24,32,33
	R2L:1,3,5,6,32,33
Class 4	DoS: 7,8,12,13,23
	Probe: 3,12,27,31,35
	U2R: 14,17,25,38,36
	R2L: 6,11,12,19,22

A. Detector

Detector takes the input variables that perform network traffic features from network environment. Real encoding for inputs will be used our proposed algorithm. Real encoding was the most appropriate encoding method. Features have either continues values or discrete values. Furthermore, features are classified into: significant features; where they have a significant role to detect attacks. In contrast, insignificant features they don't have a significant role in detecting attacks, in this case, they have been replaced by the least value for number in Java. These features composed the condition part of an environment message. Determining related features for each type of network attack is exhaustive stage. Different techniques have been used to find a relation and to find correlation between network features and type of attack that these features can be used to predicate it. In [8][9] researchers tried to find a correlation between features and network attacks. In [10] the researcher has used features for each type of attack according previous studies also. As a result, we can perform different classes taking the advantages of other results. We use classes contains main features can be used to detect specific type of network attacks. Table I shows these classes.

B. Effector

Effector responsible for firing the action of the winning rule to the environment. The result can be either normal, Probe, U2R, R2L or DoS.

C. Feedback

Feedback will influence the rule that has been selected to fire its action by adding positive value (for reward) to the selected rule strength if it gave right prediction for the type of attack or otherwise adding negative value (for penalty).

D. Classifier System

The Zeroth Level Classifier System (ZCS) rule base consists of a population of P condition-action rules or "classifiers". The rule condition is a string of characters from the significant network features with real valued where insignificant features represented by least value for number in Java acts as a wildcard allowing generalization. The action is represented by one character and both conditions and actions are initialized randomly. Also associated with each classifier is a fitness scalar which acts as an indication of the perceived utility of that rule within the system. The fitness of each rule is initialized to a predetermined value. On receipt of an input message, the rule-base is scanned and any rules whose condition matches the message at each position is placed in a match set [M]. An action is selected from the match set. This is done by different methods such as a simple roulette wheel selection policy based on the fitness of the classifiers. When an action has been selected, all rules in the [M] that advocate this action are placed in action set [A] and the system executes the action. Reinforcement in ZCS consists of redistributing payoff between subsequent action sets. A fixed fraction (β) of the fitness of each member [A] at each time step is placed in a common bucket. A record is kept of the previous action set [A]-1 and if this is not empty then the members of this set receive an equal share of the content of the current bucket, once this has been reduced by a pre-determined discount factor γ. If a reward is received from the environment then a fixed fraction (β) of this value is distributed evenly amongst the members of [A]. Finally, a tax is imposed on the members of [M] that do not belong to the [A]. ZCS employs SSGA as a discovery mechanism for constructing new rules from previous rules. When SSGA called, the SGA may use roulette wheel method to determine the parent rules based on fitness. Two offspring are produced via mutation. Settings the parameter used in ZCS are tuned to gain better results.

E. Genetic Algorithm

GA will be used as a classifiers producer to produce classifiers from existing classifiers. GA within there is a set of operation performed to produce new good classifiers used to detect network intrusion with minimum effort. These operations are:1) selection, 2) crossover, 3) mutation and 4) replacement to reproduce new rules from existing one and arising the performance detection intrusions. There are number of decisions must be taken into account to tune up the performance for the proposed method. First, type of GA we are going to use. There were a set of studies that can be used to determine which type of GA will be used. The causes for selecting SSGA mentioned early. In [11] had came up with results that SSGA will be suitable to be used instead of SGA after conducting set of experiments on both and get better and faster results than SGA. By SSGA current best solutions are automatically maintained in the population and only the poorest individuals are being replaced. In contrast, in SGA, every individual is replaced in every generation, thus there is a much greater pressure to produce Childs that are not degraded by crossover and mutation. Second, selection method for selecting rules as Roulette Wheel, Ranking and other methods mentioned in [12]. Third, parameterized crossover and mutation (Pc, Pm respectively) will be used to determine if we can accomplish the crossover and mutation or not on rules. In the our proposed algorithm, Pc will be adaptive according to the performance of MSSGBML in detecting intrusions. For crossover, the position will be selected randomly and can be either single, or multiple positions. As for Pc, Pm will be used to determine if we can mutate rules component or not. Even more, Pc was adapted to be aware from undesired behavior such as premature convergence. The Pc and Pm will be enhanced in our future work. The enhancement will be accomplished by using fuzzy logic. Adaptation can be accomplished by using counter for each rule to be indicator on rules age. Rule counter value increased automatically with each generation. To formalize the situation, the average and standard deviation will be calculated for rules age to determine if the rule Young, Mid-age, or Old. By applying fuzzy logic on parent's rules age, Pc can be Low, Medium or High. Also, we will try to take the advantage of rules age by applying also fuzzy logic on rules age to determine Pm. So Pm can be Low, Medium or High. Table II summarizes the idea of adaptive Pc. Fourth; fitness function is one of the MSSGBML keys that will be used to give judgment rules. Fitness function in our proposed Algorithm takes into account the performance of the rules to detect network intrusion.

TABLE II
CROSSOVER PROBABILITY

		Parent II		
		Young	Mid-age	Old
Parent I	Young	Low	Medium	Low
	Mid-age	Medium	High	Medium
	Old	Low	Medium	Low

III. RELATED WORK

This section briefly summarizes some of the techniques for intrusion detection. The early effort of using GAs for intrusion detection can be dated back to 1995, when [13] applied the multiple agent technology and Genetic programming (GP) to detect network attempts. Each agent monitors one parameter of the network packet and GP is used to find the set of agents that collectively determine anomalous network behaviors. This method has the advantage of using many small autonomous agents, but the communication among them is still a problem. Also the training process can be time-consuming if the agents are not appropriately initialized. Researchers in [3] have proposed paradigm consist from; neuro-fuzzy network, fuzzy inferences, and GA to detect intrusion activities in networks. This method firstly used a set of parallel nero-fuzzy classifiers (five layers 4- for type of attack, and one for normal). Then fuzzy inference used the output from classifiers to take a decision whether the current

action is normal or not. The role of GA was used to optimize the classifier engine to give the right decision. This Method also used the same data KDD CUP 99 for training and for testing the system. As a result this technique will be effectively used to detect intrusion. To enhance our proposed algorithm, we reduce the number of necessary features that have been used in this method.

IV. KDD 99 Dataset

The KDD 99 dataset includes a set of 41 features derived from each connection and a label which specifies the status of connection records as either normal or specific attack type. The list of these features can be found in [6]. These features had all forms of continuous, discrete with significantly varying ranges falling in four categories [6]: 1) Basic Features: Basic features can be derived from packet headers without inspecting the payload. 2) Content Features: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts. 3) Time-based Traffic Features: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval. 4) Host-based Traffic Features: Utilize a historical window estimated over the number of connections.

Time-based and Host-based traffic described as a Traffic features in KDD 99. Likewise, attacks fall into four main categories: 1) DoS (Denial of Service): making some computing or memory resources too busy to accept legitimate users access these resources. 2) R2L (Remote to Local): unauthorized access from a remote machine in order to exploit machine's vulnerabilities. 3) U2R (User to Root): unauthorized access to local superuser (root) privileges using system's susceptibility. 4) Probe: host and port scans as precursors to other attacks. An attacker scans a network to gather information or find known vulnerabilities.

KDD dataset is divided into training and testing record sets. Total number of connection records in the training dataset is about 5 million records. This is too large for our purpose; as such, only concise training dataset of KDD, known as 10% training dataset, was employed here and this sample distributed as shown in Table III.

TABLE III
10% of KDD Dataset

Type	Number of samples	%Training Dataset
Normal	97227	19.69
DoS	391458	79.24
Probe	4107	0.83
R2L	1126	0.23
U2R	52	0.01

V. Performance Evaluation

One of main issues involved in solving problems or trying to find optimal solution is how we can test these proposed systems. As for NIDS, testing proposed algorithm can provide a good indicator for either the proposed algorithm can give high performance compared with others or not. If our proposed algorithm has performance less than others algorithms, this encourage us to tune up our algorithm to improve our research. In addition, It is natural to assume that the difficulty of any problem relevant to the scale these problems. In general evaluating security system is a complex task to be accomplished. The main thing we interest, is measuring the performance of IDS effectively. Evaluate IDS can be expressed as how far it can correctly classify intrusions and avoid false detection. The difficulty to evaluate IDS came from the fact that it has to work properly in unknown situations and deal with new type of attackers in network environment. In previous works there was variety ways used to evaluate IDS which they are: False Positive Rate (FPR) and Detection Rate (DR).In [15] False positive rate is "the ratio of incorrectly classified normal examples (false alarms) to the total number of normal examples". FPR was calculated by using (1).

$$FPR = \sum false\,alarm \;/\; \sum Normal \;\; \text{Re}\, cord \tag{1}$$

In addition, Detection Rate (DR) was defined as "the ratio of correctly classified intrusive examples to the total number of intrusive examples". The DR is computed by using (2).

$$DR = \sum \det ectedattac\;\; k \;/\; \sum TotalAttac\;\; k \tag{2}$$

The experimental result shows that the proposed algorithm yielded good DR compared with other algorithms.

VI. Results

All samples of correctly labeled dataset of KDD 99 dataset, and after training our system, we will test it by using test dataset. After performing a set of experiments on proposed different classes it shows that 4 type of classes provide us with different DR, where class 4 produced better detection rate especially for Normal and for DoS. Even for other type of attacks there was an improvement on detecting them.

The performance of the MSSGBML has been compared with some other ML Algorithms tested on the KDD dataset. The proposed method demonstrates better performances in a number of attacks categories as is shown in Table IV.

TABLE IV
DR for several algorithms

Model	Normal	DoS	Probe	U2R	R2L	DR%
MSSGBML	96.32	97.6	37.72	28.85	83.93	97.2%
ESC-IDS[3]	98.2	99.5	84.1	14.1	31.5	95.3%
Multi-Classifier [16]	n/r	97.3	88.7	29.8	9.6	n/r
PNrule [17]	99.5	96.9	73.2	6.6	10.7	91.1

Detection rate (DTR) for the different algorithms performances on the KDD 99 with corrected labels of KDD Cup 99 dataset (n/r stands for not reported)

It can be stated that all the ML algorithms tested on the KDD 99 dataset offered an acceptable level of detection performance for DoS, Probe or both. Also, noticed that there is poor performance on the U2R and R2L categories.

VII. Conclusion

In this paper, a Modified algorithm was introduced to detect network intrusions and was successfully demonstrated its usefulness on KDD 99 Dataset, training and test data, MSSGBML was used as intrusion detection mechanism by using rules without the need for human experts. Also, matching between rules became adaptive according DR values.

Discover engine has been improved by using SSGA instead of SGA taking into account the suitable method for selection. Fuzzy logic was used to adapt the probability of crossover and mutation within discovery engine.

Also our research focuses on reducing the number of features to be used in classifying and detecting various attacks. Our research will be continued to use different fitness function since it plays important role in intrusion detection. In addition, we will use another type of GA that takes both advantages of SGA and SSGA, called CHC algorithm.

References

[1] S.Selvakani, R.S. Rajesh, "Genetic Algorithm for framing rules for intrusion Detection", IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.11, November 2007

[2] A.Christie, W. Fithen, J.McHugh, J.Pickel, E. Stoner, "State of the Practice of Intrusion Detection Technologies", Technical Report, Carnegie Mellon University, 2000.

[3] N.Toosi, M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers", Computer Communications 30(2007) 2201–2212, 2007

[4] M. Sabhnani, G. Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context", Proceeding of International Conference on Machine Learning: Models, Technology and Application, Las Vegas, Nevada, USA, June 2003.

[5] Ch. Sinclair, L. Pierce, S. Matzner, "An Application of Machine Learning to Network Intrusion Detection", 15th Annual Computer Security Applications Conference Phoenix, Arizona, December 6-10, 1999

[6] KDD-CUP 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[7] A.Osareh, Bita Shadgar, "Intrusion Detection in Computer Networks based on Machine Learning Algorithms", International Journal of Computer Science and Network Security, VOL.8 No.11, November 2008

[8] I.Guayan, A.Elisseeff, "An Introduction to Variable and Selection", Journal of Machine Leaning Research 3, March 2003

[9] L.Yu, H.Lin, "Feature Selection for High-Dimensional Data: A Fast Correlation-based Filter Solution", Proceeding of 20th International Conference on Machine Learning (ICML-2003), Washington D.C., August 2003.

[10] T. S. Chou, K. K. Yen, and J. Luo, "Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms", International Journal of Computational Intelligence 4;3 © www.waset.org Summer 2008.

[11] J.Jones, T.Soule, "Comparing Genetic Robustness in Generational vs. Steady State Evolutionary Algorithms", GECCO'06, Seattle, Washington, USA. July 8–12, 2006, ©Copyright 2006 ACM

[12] M.Mitchell, "An Introduction to Genetic Algorithm", MIT Press, 1996.

[13] Crosbie M and Spafford E, "Applying genetic Programming to Intrusion Detection", Proceedings of the AAAI Fall Symposium, 1995.

[14] S.M. Bridges and R.B. Vaugha, " Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection", Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp.109-122, 2000.

[15] L. Kuang, "DNIDS: A Dependable Network Intrusion Detection System Using the CSI-KNN Algorithm", Master thesis, Queen's University , Canada, September 2007.

[16] M.R. Sabhnani, G. Serpen, Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context, in: Proceedings of International Conference on Machine Learning: Models, Technologies, and Applications, 23–26 June 2003, Las Vegas, Nevada, USA, 2003, pp. 209–215.

[17] R. Agarwal, M.V. Joshi, PNrule: A New Framework for Learning Classifier Models in Data Mining, Department of Computer Science, University of Minnesota, Report No. RC-21719, 2000.

Wafa' Al-Sharafat was born in Jordan. She received the B.S. in Computer Science from Hashemite University, Jordan in 2001 and M.Sc. degree in Computer Information system from Arab Academy for Banking and Financial Science, IT collage, Jordan in 2004. She is currently pursuing her PhD Degree in Computer information system, presently she is working as an Instructor in Computer Information System department in Al Al- Bayt University, IT Collage. Her main interest are Genetic Algorithm, Machine Learning, Security Issues and E-Commerce.

Reyadh Sh.Naoum was born in Basrah, Iraq. He received the B.S. in Mathematics from Basrah University, Iraq in 1969 and M.Sc. degree in Computing from Manchester University, England in 1973 and PhD in Computing from Manchester university, England in 1976. His main interests are Neural Computing, Genetic Computing and Numerical Software.