Scientific Research Publishing

Communications and Network, 2019, 11, 11-20
http://www.scirp.org/journal/cn
ISSN Online: 1947-3826
ISSN Print: 1949-2421

A New FLAME Selection Method for Intrusion Detection (FLAME-ID)

Wafa Alsharafat

Al al-Bayt University, Mafraq, Jordan
Email: walsharafat@gmail.com

How to cite this paper: Alsharafat, W. (2019) A New FLAME Selection Method for Intrusion Detection (FLAME-ID). *Communications and Network*, **11**, 11-20.
https://doi.org/10.4236/cn.2019.111002

Received: December 15, 2018
Accepted: January 25, 2019
Published: January 28, 2019

Abstract

Due to the ever growing number of cyber attacks, especially of the online systems, development and operation of adaptive Intrusion Detection Systems (IDSs) is badly needed so as to protect these systems. It remains as a goal of paramount importance to achieve and a serious challenge to address. Different selection methods have been developed and implemented in Genetic Algorithms (GAs) to enhance the rate of detection of the IDSs. In this respect, the present study employed the eXtended Classifier System (XCS) for detection of intrusions by matching the incoming environmental message (packet) with a classifiers pool to determine whether the incoming message is a normal request or an intrusion. Fuzzy Clustering by Local Approximation Membership (FLAME) represents the new selection method used in GAs. In this study, Genetic Algorithm with FLAME selection (FGA) was used as a production engine for the XCS. For comparison purposes, different selection methods were compared with FLAME selection and all experiments and evaluations were performed by using the KDD'99 dataset.

Keywords

FLAME, Intrusion Detection, XCS, Genetic Algorithm

1. Introduction

Internet users increased greatly during the last decade. This increase gives the attackers motivation for initiating harm actions and making attacks. The Intrusion Detection Systems (IDSs), however, act as a defense line against the attack attempts. The weakness analysis tools, as vulnerability assessment tools, should be carefully evaluated and frequently developed to establish a solution package as is the case in other computing security applications. Such tools are concerned with keeping systems confidential, available, reliable, integral, authenticated, and

impregnable against any attempts of attack. Thus, intrusion detection (ID) can be defined as "the process of monitoring the events occurring in a computer system or network and noticeably different from normal system activities and thus detectable" [1]. An IDS is classified as a defense mechanism that is intended to keep a system secure while the prevention system prevents the attacks from accessing the system.

There is a necessity for developing adaptive methods and techniques that work well entire systems to identify attacks trials. Currently, different optimizing algorithms, artificial intelligence (AI), and hybrid algorithms are employed to improve the intrusion detection possibilities. For example, Genetic Algorithms [2], the Artificial Neural Network [3], Swarm Intelligence [4], Fuzzy Logic [5], Learning Classifiers, Support Vector Machine [6], Machine learning [7], and Data Mining [8]. In this study, Genetic Algorithm (GA) has been adapted because of its promised results in IDSs. Since GA has the ability to handle large state space [9].

In FLAME-ID, the XCS applied a modification on the classifier generator, GA, by applying the Fuzzy Clustering by Local Approximation Membership (FLAME) as a new selection method; FGA selection. FGA differs than previous selection methods that have been implemented in [2] [3] [10] [11].

In sum, this study applied a new approach, that is, FGA, as a classifier generator in the eXtended Classifier System (XCS) for ID. Afterwards, the study assessed ability of FALME selection to enhance the attack detection rate (DR) relative to the existing and commonly-used selection methods.

The remainder of this paper is organized as follows. Section 2 reviews related previous works. Section 3 describes the dataset that is commonly used in training the IDS and testing its performance. The research background is presented in Section 4. Thereafter, Section 5 discusses the proposed method for improving ID and Section 6 discusses evaluation of performance of the proposed method. Lastly, the experimental results are given in Section 7.

2. Related Works

Detecting intrusion is a tempting area to conduct a set of researches since 1982 [12]. Since that time, plentiful research accomplished and enhanced in term of achieving high DR amongst others. For example, in 2016, Sultana and Jabbar [8] proposed an intelligent IDS using the average intrusion detection method, which depends on estimators that represent enhancements of the Naive Bayes algorithm. The DRs of DoS, Probe and U2R were 98.63%, 98.48%, 98.65%, respectively. In 2017, Yin *et al.* [13] applied a novel computer network intrusion detection (NCNID) algorithm by combing the Support Vector Machine (SVM) as a statistical learning model with context validation as a preliminary analysis to extract information from the intrusion in order to reduce the probability of false alerts. As a result, the precision of alerts for the processed data was 94.79%.

Researchers in [14] used data-mining techniques for detecting intrusions. The data-mining IDS they developed consisted of sensors, detectors, a data warehouse, and an Adaptive Model Generator. The data warehouse serves as a centralized store for the data received by the sensors and models that are used by detector to determine if the incoming data is an attack or a normal request. For enhancements, an adaptive model generator facilitates the development and distribution of new intrusion detection models. However, this proposed IDS needs enhancements because it has low accuracy of detection of different attack types like R2L and DOS.

In matter of feature filtration, Alsharafat in [10] has developed a model by combing the Artificial Neural Network (ANN) and XCS Where ANN has been applied in the first stage. This stage concerns about filtering features according to the attack type. Then, XCS, with modifications, plays the role of intrusion detection. The overall DR was 98.01% and the false alarm rate was 0.9%.

A method for attack sequence detection in the Cloud environment using Hidden Markov Model was developed by researchers in [15]. The proposed approach was designed to determine and analyze the multiple logs for an attacked machine or a machine that is under attack to identify whether an attack sequence does, or does not, exist.

In [16], Shrivastava and his colleagues proposed a model consisting of Rough Set and SVM. The Rough Set was used to reduce the number of network features whereas the SVM was employed to train and test the proposed model. The final DR of the proposed system was 95.98% and the false positive rate was 7.52%.

Danane and Parvat in [11] proposed a combined Fuzzy algorithm and GA. The Fuzzy algorithm was used to reduce the number of network features to six features while the GA was utilized to detect the intrusion. The GA parameters; crossover probability (Pc) and mutation probability, were set to 0.8 and 0.088, respectively. The final accuracy of the proposed hybrid algorithm was 98%.

3. The KDD'99 Dataset

The KDD'99 dataset is a benchmark dataset that is frequently used in evaluating different IDSs like in [8] [10] [11] [13] [14] [15] [16]. This dataset consists of 41 network features. A list of these features can be found in [17]. These features are classified into two forms; continuous and discrete, with variable ranges. On the other hand, these 41 features are categorized into four categories:

- Basic Features

  These features are derived from network packet headers, regardless of the payload.

- Content Features

  These features are used to assess the payload.

- Time-based Traffic Features

Table 1. Distribution of the records of the training data sub-set.

Attack Type	Number of Records	% of the KDD'99 Dataset
Normal	97,227	19.69
DoS	391,458	79.24
Probe	4107	0.83
R2L	1126	0.23
U2R	52	0.01

These are features that are used to capture second temporal window properties.

- Host-based Traffic Features

These are features that utilize a historical window estimated over the number of connections.

Both of the time-based and host-based traffics are considered as traffic features in the KDD'99 dataset.

Likewise, attacks fall into four main categories [17]:

- DoS attacks

These attacks focus on making all resources of the attacked system too busy and unable to accept legal request of resources.

- R2L attacks

This sort of attacks focuses on exploiting the vulnerabilities of the attacked system by unauthorized access from a remote machine.

- U2R attacks

These attacks entail access to local user accounts and their related privileges using the weakness traps of the attacked system.

- Probe attacks

This sort of attacks concentrates on monitoring networks and their related information to discover vulnerabilities and related trapdoors.

In the present study, the KDD'99 dataset was partitioned into two sub-sets: a training sub-set and a testing sub-set. The training sub-set consists of 10% of five million records that were distributed as shown in **Table 1**.

4. Research Background

The Learning Classifier System (LCS) is classified as a machine-learning paradigm that uses simple string rules, *i.e.*, classifiers, to guide its performance in unknown environments. The LCSs have gained interest of researchers since 1995 [18] in their efforts to find solutions for different problems. The main part of the LCS is a group of rules called the population of classifiers. One of such rules is the XCS, which is one of the classifier systems that are categorized as rule-based systems that are employed for detecting intrusions.

The LCSs are often utilized to raise the DRs in the unknown environments. They employ two operations; evolution and learning, and the different problems require different learning styles to identify useful classifiers and find better rules.

Learning is concerned with the reply from the environment so as to decide on whether or not the chosen rule leads to correct detection. Therefore, any learning process achieved by taking feedback from environment in positive or negative way. In both cases, new rules must be created and evaluated to optimize DR. This step describe as an evolutionary process is realized via evolutionary algorithms as the GA [12].

The LCSs are of three types [9]:

- Zeroth level classifier systems (ZCSs);
- Anticipatory learning classifier systems (ALCSs);
- eXtended Classifier Systems (XCSs).

4.1. The eXtended Classifier System (XCS)

The XCS is a rule-based classifier system that was introduced by Wilson in 1995 [19]. Each rule, or classifier, consists of two parts:

- The condition part that can be represented using real code, integer code, or binary code, as is the case in the current study, for example, {0, 1, #}, where # denotes a non-significant value.
- The action part, which represents the result of the chosen rule to be applied in an environment. In general, the action can be either an attack or a normal action.

The XCS is based on two elements: reinforcement learning (RL) and GA. Reinforcement learning is concerned with gaining feedback from an environment which can be a reward or a penalty. Thus, the RL is designed to determine how the rule, that is, the classifier, will be useful in other situations [20] [21]. The GA, on the other hand, works as a discovering mechanism which feeds the system with new rules. The main components of the XCS are shown in **Figure 1**. Briefly, they are [19] [20] [21] [22]:

- Detector

The detector receives input from the environment that represents a network traffic record and converts it into a binary code.

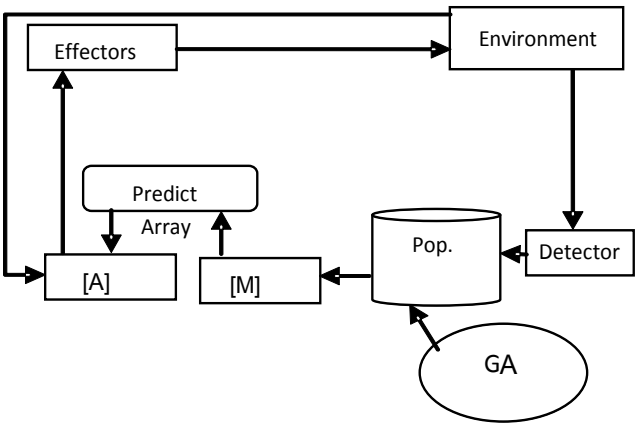

Figure 1. Components of the XCS [19].

- Match set [M]

The match set [M] is a pool of classifiers that are matched with a condition part of the incoming message from the detector.

- Prediction Array

This array is produced from [M] according to the average fitness-weighted prediction rules in [A].

- Action set [A]

This set represents a pool of classifiers derived from [M] that support the action which will be chosen.

- Effectors

The effectors are an output unit that will fire the action of the classifier selected from [A] to the environment where the result can be either a normal action or an attack.

4.2. Functioning of the XCS

Functioning of the XCS can be summarized in the following steps [19] [20] [21] [22]:

Step 1: Environment input is received via a detector that is concerned with the packet coming from the network.

Step 2: The received input is matched with a set of the existing system population (Pop) classifiers. The matching process is involved in matching the received input with the Pop. So, the matched populations will be placed in a separate set called the matching set [M]. The contents of [M] represent the entire knowledge of the XCS that is concerned with deciding what to do with the incoming input.

Step 3: A prediction array is generated. This array is used to select the appropriate action to be executed. The predictive ability of the action of every classifier in [M] is evaluated by taking a weighted average of the predictions of all the classifiers in [M], where the weight of every value of classifier represents the fitness, F.

Step 4: Producing an action set [A]. Next, a subset of [M] which supports the selected action is produced. This set is called the action set [A].

Step 5: Action execution and action set updating. The most appropriate classifier picked up from [A] will be fired to the environment by an effector. Then, feedback will be received from the environment in the form of reward if the taken action is correct. Otherwise, the system will get penalty. In each cycle, the XCS updates only the classifiers in [A] based on the feedback received, whether reward or penalty. Therefore, every classifier maintains a prediction about the feedback according to its response. Afterwards, the predictions are adjusted accordingly.

5. The Proposed Detection System

The related previous studies provided different AI and hybrid techniques for

developing IDSs. Every technique has its own pros and cons. Hence, these techniques have different detection powers in terms of the DR, FAR, or Accuracy measures of performance. The IDS proposed in the current study combines machine learning (ML), represented by the XCS, and an evolutionary algorithm, represented by the FLAME_GA (FGA), which applies a new selection method called FLAME selection. In the traditional GAs, a set of operations is implemented in every generation. These operations are selection, crossover, mutation, and replacement. The method of FLAME selection is a new selection method that was adopted in the present study to provide an enhancement for the GA so as to improve the DR of the proposed IDS.

Fuzzy clustering by Local Approximation of Memberships (FLAME) determines clusters according to fitness of the classifier. This selection method depends on the neighborhood relationships between parents that were applied to the neighboring memberships in fuzzy membership of populations [23]. Consequently, the FLAME is made up of several processes to perform clustering for the GA parents. It performs clustering via the following steps [23] [24]:

1) Extracting the information structure. This is usually realized by

a) Creating a graph to connect each parent to the best neighbor according to the k-Nearest Neighbors (KNN) algorithm.

b) Calculating the density for each parent according to its closeness to its k nearest neighbor.

c) Categorizing each parent into either of three classes; the inner, outer, or rest classes.

As such, the FLAME identifies the parents having archetypical features, which known as a Cluster-Supporting Objects (CSOs). Also, CSO represent an inner class.

2) Assigning a fuzzy membership vector to each parent. Achievement of this step entails

a) Provide initial values of fuzzy membership function:

i) Each parent in the inner CSO is fully assigned to one class to be associated with one class.

ii) All parents in the outlier class have fixed and full memberships to this class.

iii) The rest class contains parents with equal membership.

b) Updating the fuzzy membership value for all parents in all three classes by applying a linear aggregation of the fuzzy memberships of the nearest neighbors.

3) Assignment of every parent to the cluster that has the highest membership value on a one-to-one object-cluster basis according to the following:

a) Passing on the highest membership value, based on a one-to-one parent-class basis, to all parent in the class.

b) Every parent will be assigned to specific class if that class has higher membership value based on a one-to-multiple basis.

After implementing FLAME, all parents in the GA belong to one of the following classes:

- Inner class: where all parents have superior density between neighbors.

- Outer class: where parents have less density value compared with neighbors and a predetermined threshold.
- Rest class: where any parents not belong to inner or outer will be assigned to this class.

6. Performance Evaluation

Researchers in different studies like [8] [10] [11] [13] [14] [15] [16] have applied varying performance evaluation measures to develop a judgment on the performance of different IDSs. The DR and FAR are the two most popular measures that were used in previous studies to assess IDS performance. As well, they are used in the present study to evaluate performance of proposed IDS. A briefing on each follows.

- Detection Rate (DR)

The DR can be defined as "the ratio between the number of correctly detected attacks and the total number of attacks" [25] which can be expressed as in Equation (1) [26]:

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1}$$

- False Alarm Rate (FAR)

The FAR can be defined as "the number of 'normal' patterns classified as attacks (False Positive) divided by the total number of 'normal' patterns" [25] which can be expressed as in Equation (2) [26]:

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{2}$$

The parameters appearing in Equation (1) and Equation (2) are explained in Table 2.

7. Experimental Results and Conclusions

This study proposed a new selection method called FLAME selection that has been implemented within a GA, which served as the classifier generator in the XCS. The initial results show that the FLAME selection method produced classifiers with higher fitness than traditional selection methods such as the Ranking, Roulette Wheel, and Tournament methods.

As regards ID, the experimental results showed that the XCS with FLAME selection is an effective method that can be used to improve the DR and reduce the

Table 2. Definitions of the parameters of Equation (1) and Equation (2) [26].

Parameter	Definition
True Positive Rate (TP)	Attack occurs and an alarm is raised
False Positive Rate (FP)	No attack occurs but alarm is raised
True Negative Rate (TN)	No attack occurs and no alarm is raised
False Negative Rate (FN)	Attack occurs but no alarm is raised

FAR. Having a low FAR and high DR is an advantage for any IDS. Furthermore, parent (classifier) selection in the GA is a critical decision that must be soundly taken in the early stages of classifier generation for detecting the network attacks. Quite often, several modifications and trials will need to be conducted to gain high DR and several comparisons should be made to investigate and evidence gain of superior DR by the inner class than the other classes. In this regard, the FLAME selection method aims at selecting the best set of parents to breed a new generation with high potentiality to increase the DR and reduce the FAR.

In the future, further enhancements will be needed to improve the DR for all types of network attacks.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Surat, S., Werasak, K., Witcha, Ch. and Siriporn, Ch. (2005) Network Anomaly Detection Using Soft Computing. *Proceedings of World Academy of Science, Engineering and Technology*, **9**, 140-144.

[2] Khan, M.S.A. (2011) Rule Based Network Intrusion Detection Using Genetic Algorithm. *International Journal of Computer Applications*, **18**, 26-29.

[3] Srinivasu, P. and Avadhani, P.S. (2012) Genetic Algorithm Based Weight Extraction Algorithm for Artificial Neural Network Classifier in Intrusion Detection. *Procedia Engineering*, **38**, 144-153. https://doi.org/10.1016/j.proeng.2012.06.021

[4] Li, W.S., Bai, X.M., Duan, L.Z. and Zhang, X. (2011) Intrusion Detection Based on Ant Colony Algorithm of Fuzzy Clustering. *International Conference on Computer Science and Network Technology*, IEEE, Piscataway, 1642-1645.

[5] Geramiraz, F., Memaripour, A.S. and Abbaspour, M. (2012) Adaptive Anomaly-Based Intrusion Detection System Using Fuzzy Controller. *International Journal of Network Security*, **14**, 352-361.

[6] Ganapathy, S., Yogesh, P. and Kannan, A. (2012) Intelligent Agent Based Intrusion Detection Using Enhanced Multiclass SVM. *Computational Intelligence and Neuroscience*, **10**.

[7] Chan, P.K., Mahoney, M.V. and Arshad, M.H. (2003) A Machine Learning Approach to Anomaly Detection. Florida Institute of Technology, Tech. Rep. CS-2003-06.

[8] Amreen Sultana, A. and Jabbar, M.A. (2016) Intelligent Network Intrusion Detection System Using Data Mining Techniques. 2*nd International Conference on Applied and Theoretical Computing and Communication Technology* (*iCATccT*), 21 July 2016.

[9] Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Boston.

[10] Wafa, A. (2013) Applying Artificial Neural Network and Extended Classifier System for Network Intrusion Detection. *International Arab Journal of Information Technology* (*IAJIT*).

[11] Danane, Y. and Parvat, T. (2015) Intrusion Detection System Using Fuzzy Genetic

Algorithm. *International Conference on Pervasive Computing* (*ICPC*). https://doi.org/10.1109/PERVASIVE.2015.7086963

[12] Kemmerer, R.A. and Vigna, G. (2002) Intrusion Detection: A Brief History and Overview. *Computer*, **35**.

[13] Yin, G., Zhang, Y. and Zhao, Z. (2017) A Novel Computer Network Intrusion Detection Algorithm Based on OSVM and Context Validation. *International Conference on Progress in Informatics and Computing* (*PIC*).

[14] Kamble Jayshree, R. and Rangdale, S.P. (2014) Intrusion Detection Using Data Mining Approach. *International Journal of Science and Research* (*IJSR*), **3**, 1142-1145.

[15] Chen, Ch., Guan, D.J., Huang, Y. and Ou, Y. (2012) Attack Sequence Detection in Cloud Using Hidden Markov Model. *The Seventh Asia Joint Conference on Information Security* (*Asia JCIS*), 100-103. https://doi.org/10.1109/AsiaJCIS.2012.24

[16] Shrivastava, S.K. and Jain, P. (2011) Effective Anomaly Based Intrusion Detection Using Rough Set Theory and Support Vector Machine. *International Journal of Computer Applications*, **18**, 35-41.

[17] https://kdd.ics.uci.edu/

[18] Urbanowicz, R.J. and Moore, J.H. (2009) Learning Classifier Systems: A Complete Introduction, Review, and Roadmap. *Journal of Artificial Evolution and Applications*, **2009**, Article ID: 736398.

[19] Wilson, S.W. (1995) Classifier Fitness Based on Accuracy. *Evolutionary Computation*, **3**, 149-175. https://doi.org/10.1162/evco.1995.3.2.149

[20] Luca, L.P. (2008) Learning Classifier Systems: Then and Now. *Evolutionary Intelligence*, **1**, 63-82.

[21] Holmes, J.H., *et al*. (2002) Learning Classifier Systems: New Models, Successful Applications. *Information Processing Letters*, **82**, 23-30. https://doi.org/10.1016/S0020-0190(01)00283-6

[22] Bull, L. and Kovacs, T. (2005) Foundations of Learning Classifier Systems. Springer Science & Business Media, Berlin, Vol. 183. https://doi.org/10.1007/b100387

[23] Sampath, P. and Prabhavathy, M. (2015) Web Page Access Prediction Using Fuzzy Clustering by Local Approximation Memberships (FLAME) Algorithm. *ARPN Journal of Engineering and Applied Sciences*, **10**, 3217-3220.

[24] Fu, L. and Medico, E. (2007) FLAME, a Novel Fuzzy Clustering Method for the Analysis of DNA Microarray Data. *BMC Bioinformatics*, **8**, 3.

[25] Elhamahmy, M.E., Elmahdy, H.N. and Saroit, I.A. (2010) A New Approach for Evaluating Intrusion Detection System. *International Journal of Artificial Intelligent Systems and Machine Learning*, **2**.

[26] Gulshan, K. (2014) Evaluation Metrics for Intrusion Detection Systems—A Study. *International Journal of Computer Science and Mobile Applications*, **2**, 11.