ISSN: 0067-2904

Genetic Algorithm based Clustering for Intrusion Detection

Noor Fouad*, Sarab M. Hameed
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq.

Abstract

Clustering algorithms have recently gained attention in the related literature since they can help current intrusion detection systems in several aspects. This paper proposes genetic algorithm (GA) based clustering, serving to distinguish patterns incoming from network traffic packets into normal and attack. Two GA based clustering models for solving intrusion detection problem are introduced. The first model coined as $GA\ \#1$ handles numeric features of the network packet, whereas the second one coined as $GA\ \#2$ concerns all features of the network packet. Moreover, a new mutation operator directed for binary and symbolic features is proposed. The basic concept of proposed mutation operator depends on the most frequent value of the features using mode operator. The proposed GA-based clustering models are evaluated using Network Security Laboratory-Knowledge Discovery and Data mining (NSL-KDD) benchmark dataset. Also, it is compared with two baseline methods namely k-means and k-prototype to judge their performance and to confirm the value of the obtained clustering structures. The experiments demonstrate the effectiveness of the proposed models for intrusion detection problem in which $GA\ \#1$ and $GA\ \#2$ models outperform the two baseline methods in accuracy (Acc), detection rate (DR) and true negative rate (TNR). Moreover, the results prove the positive impact of the proposed mutation operator to enhance the strength of $GA\#2$ model in all evaluation metrics. It successfully attains 6.4, 5.463 and 3.279 percentage of relative improvement in Acc over $GA\ \#1$ and baseline models respectively.

Keywords: Clustering, Genetic Algorithms, Intrusion Detection, K-Means

العنقدة على أساس الخوارزميات الجينية لكشف التسلل

نور فؤاد*، سراب مجيد حميد
قسم علوم الحاسبات ، كلية العلوم، جامعة بغداد، بغداد، العراق.

الخلاصة

مؤخراً حصلت خوارزميات التجميع على اهتمام من قبل البحوث ذات العلاقة حيث تساعد أنظمة الكشف الحالية في نواحي عدة . هذا البحث يقترح الخوارزمية الجنية باعتماد على تقنية التجميع ، حيث تساعد لتمييز الأنماط القادمة الى الشبكة فيما اذا كانت نمط طبيعي او نمط هجومي. تم تقديم نموذجين لمشكلة كشف التسلل النموذج الأول أطلق عليه أسم $GA\ \#1$ حيث يتعامل مع ميزات حزمة شبكة الرقمية ، بينما اطلق على النموذج الثاني GA #2 التي تتعامل مع كل ميزات حزمة الشبكة. علاوة على ذلك ، تم اقتراح معامل طفرة جديد لميزات الثنائية والرمزية لحزمة الشبكة . حيث ان المفهوم الرئيسي للمعامل الطفرة المقترح يعتمد على القيمة الاكثر تكرار للميزات حزمة الشبكة باستخدام معامل mode . ولغرض تقييم الخوارزمية الجينية

*Email: noor.fauadh@gmail.com

باعتماد على تقنية التجميع المقترحة لكشف التسلل يتم باستخدام مجموعة بيانات NSL−KDD ومقارنتها مع طريقتين هما k−means, k−prototype للحكم على أدائها وأثبات القيم التي تم الحصول عليها من التجميع .اثثبت التجارب العملية فعالية النماذج المقترحة لمشكلة كشف التسلل . أن نماذج المقترحة GA # 1 و GA # 2 تمتاز بأداء متفوق على الأساليب التقليدية في كافة المقاييس من حيث مقياس(ACC)، كشف معدل الكشف (DR) ومعدل سلبي صحيح (TNR). وعلاوة على ذلك، فإن النتائج ثبتت الأثر الإيجابي للعامل الطفرة المقترح لمضاعفة قوة نموذج الثاني $GA\ \#2$ في كل المقاييس التقييم. حيث حصلت $GA\ \#2$على اعلى تحسن نسبي مئوي في معيار الدقة 6.4، 5.463 و 3.279 بالنسبة الى $GA\ \#1$ و الطرق التقليدية.

1. Introduction

In the last few decades, the computers network security has become one of the most important issues in the world due to the rapidly increased use of computers and Internet [1]. Although intrusion prevention techniques such as user authentication, firewalls, intrusion prevention and data encryption are used to protect the network and computer systems; but those techniques alone is insufficient since computer networks became more complicated and intrusion methods became more intelligent [2, 3]. Accordingly, it became necessary to monitor and protect network security infrastructure to detect intruders or any kinds of intrusions with intrusion detection system (IDS).The components of IDS are event generator, analyzer and response module [4]. Event generator generates data that are well formatted and suitable for analyzer and response module. An incoming network packet is analyzed to identify whether it is a normal activity or abnormal activity by analyzer using misuse detection and/or anomaly detection. Misuse detection depends on a signature database of previously known attacks to match with, while anomaly detection depends on normal behavior to detect abnormal behaviors in patterns. Then, an alarm is generated to the network administrator. Finally, set of actions should be taken by the response module when an intrusion is detected. The common problem of IDS regardless if it is signature based or anomaly based is the high number of false alerts [5].

Several methods dealt with intrusion detection problem namely supervised and unsupervised classification. In supervised classification, the class labels should be known in prior. While in unsupervised classification or clustering, the class labels are not required. Clustering analysis can be used for intrusion detection in which the network packets are classified into a set of clusters in such a way that the similarity of the network packets in the same cluster is as large as possible while in the different clusters is as small as possible [1].

However, trying to find a set of clusters for partition network packets is one of the biggest challenges. This challenge will be stated in this paper while considering *How the clustering structures that should be capable enough for detecting an intrusion can be found*? One of the evolutionary algorithms namely genetic algorithm is adopted to find an answer for how to generate the clusters for network packets.

The organization of the paper is as follows. Section 2 describes the related work. Section 3 presents a brief description of the basic concepts of genetic algorithm and clustering analysis. Section 4 introduces the proposed models for intrusion detection based on genetic algorithm. Section 5 evaluates the proposed models. Finally, section 6 concludes the work and suggests some research future line.

2. Related Work

Clustering or unsupervised learning methods are used for network intrusion detection to identify attack without known labels. The current trend in the paper is the use of evolutionary algorithms based clustering for intrusion detection. In what follows, some of the previous researchers are presented:

1. Liu and Geo in 2008 [1] proposed a classification algorithm based on rival penalized competitive learning (RPCL) coined as conscientious rival penalized competitive learning (CRPCL). CRPCL modifies RPCL neural network through making each neural node win the competition with the same probability. Results of comparison with unsupervised principal component classifier (UNPCC), k-means, Expectation Maximization (EM) and density based clustering with k-nearest neighbor (KNNDB) showed that CRPCL algorithm outperformed other methods on KDD-Cup99 dataset in terms of accuracy (Acc), detection rate (DR) and false alarm rate (FAR) metrics.

2. Wang et al. in 2010 [6] proposed an approach for intrusion detection based on artificial neural network (ANN) and fuzzy clustering coined as FC-ANN. FC-ANN approach consists of three stages. The first stage is to partition the training set into subsets using fuzzy c-means (FCM). Then, ANN learns the pattern of each subset. Finally, the results of ANN are aggregated. Testing of the algorithm was performed on KDD-Cup99 benchmark dataset. The results showed that FC-ANN provides better DR in comparison with BPNN, decision tree and Naïve Bayes.

3. Li and LexXu in 2011 [7] proposed an intrusion detection approach using k-means and particle swarm optimization (PSO) coined as (KM-PSO). PSO is used to guide k-means to choose initial centroids. KDD-Cup99 dataset was used as an evaluation data for testing the performance of KM-PSO, and the experimental results showed that the result of KM-PSO is better than k-means algorithm regarding DR and FAR.

4. Horng et al. in 2011 [8] proposed an intrusion detection approach based on support vector machine (SVM) and balanced iterative reducing and clustering using hierarchies (BIRCH) algorithm. BIRCH algorithm is used to reduce the KDD-Cup99 training dataset to a smaller one that contains the qualified data. The results of the proposed approach provide better performance than SVM.

5. Casas et al. in 2012 [5] introduced an unsupervised network intrusion detection system (UNIDS) to detect a different type of attacks without seen their labels. It uses density-based clustering technique and does clustering in low dimension space. KDD-Cup99 dataset and real traffic were used for testing the performance of UNIDS. The results showed that the performance of UNIDS is better than misuse detection.

6. Karami and Zapata in 2014 [9] presented a fuzzy anomaly detection system that consists of two stages. Firstly, PSO and k-means are combined for finding the best number of clusters. Secondly, a fuzzy decision is applied for detecting anomalies. The results of the proposed model proved its effectiveness against existing state-of-the-art anomaly detection systems in terms of accuracy, DR and FAR when applied on KDD-Cup99 dataset.

7. Eslamnezhad and Varjani in 2014 [10] proposed a new intrusion detection based on min-max k-means clustering algorithm that assigns weight to the cluster depending on internal variance. The proposed algorithm when compared to k-means on Network Security Laboratory-Knowledge Discovery and Data mining (NSL-KDD) dataset, it was found that it is better than k-means regarding DR and FAR.

8. Lin et al. in 2015 [3] introduced an intrusion detection approach based on cluster centroids and nearest neighbor coined as (CANN). In the first step of CANN, a k-means clustering algorithm is used to determine the cluster centroids. Then, k-nearest neighbor (k-NN) is adopted to identify the nearest neighbor for each data instance in the identical cluster. Afterward, the data is represented by distance feature, and k-NN classifier is used for classifying data. KDD-Cup99 dataset was used as an evaluation data and accuracy, DR and FAR were used as an evaluation metrics. Experimental results showed that the CANN outperforms the k-NN and SVM.

9. Duque and Omar in 2015 [11] proposed an intrusion detection model based on k-means clustering algorithm and signature-based approach. Which used to evaluate the proposed model, NSL-KDD dataset was used. The results showed that the proposed model could produce high DR and low FAR if the number of clusters is properly identified.

10. Aissa and Guerroumi in 2015 [12] proposed algorithm based clustering for anomaly detection coined as (GC-AD). In GC-AD, a dissimilarity measure is used to form clusters. KDD-Cup99 dataset was used as an evaluation data, and k-means was considered as a baseline classifier. Experimental results indicated that performance of GC-AD is better than k-means.

3. The Proposed Intrusion Detection Models

Intrusion detection problem has been viewed as a clustering problem. In such away, we need to cluster an incoming network packet into either normal or attack. Searching for the the cluster structures that has the high discriminatory power to distinguish between normal and attack network packet is a Non-deterministic Polynomial-time hard (NP-hard) problem which requires algorithm with specified properties. Therefore, searching capability of GA can be used to provide proper cluster centroids.

The components of the proposed intrusion detection model can be recapitulated into three phases. These are dataset preprocessing phase, GA-based clustering phase and detection phase. The role of

GA-based clustering phase is for identifying the centroids $\mathbb{C}$ of normal and attack clusters such that the detection rate and false alarm rate of the resultant clusters are optimized. Whereas, the detection phase role is to detect whether an incoming network packet is an attack or normal based on the centroids produced from the GA-based clustering phase. The following sections handle each phase in details.

3.1 Dataset Preprocessing Phase

The first stage of the proposed intrusion detection model is preprocessing the incoming NSL-KDD dataset [13], which is used as an evaluation data. The purpose of preprocessing is to make the dataset appropriate to be trained and tested.

Each network packet in NSL-KDD dataset is characterized by 41 features of mixed type having 32 numeric features of different scales, 6 boolean features and 3 symbolic features. NSL-KDD dataset can be formally described as $\mathbb{S} = \{S_1, S_2, \dots, S_t\}$

Where t is the number of network packets.

Moreover, each network packet, $S_j \in \mathbb{S},\ j \in \{1, \dots, t\}$, can be stated as:

$S_j = \{s_{j1}, s_{j2}, \dots, s_{jF}\}$, where

F is the number of features in each network packet.

The numeric features are normalized to avoid some features' domination over others as introduced in the following [14]:

1. Feature number two and three coined as src_bytes and dest_bytes respectively that have a scale in the range [1, 1.3 billion] are normalized by applying the logarithmic scaling (with base 10) to reduce their range.
2. The remaining 30 features are scaled linearly to the range [0, 1] using minimum maximum normalization as in equation (1)

$$\mathcal{F}_i' = \frac{\mathcal{F}_i - \mathcal{F}_i^{min}}{\mathcal{F}_i^{max} - \mathcal{F}_i^{min}} \quad (1)$$

Where:

$\mathcal{F}_i$: is the value of a numeric feature i excluding src_byt.es and dest_bytes

$\mathcal{F}_i^{min}$: is the minimum value feature $\mathcal{F}_i$ can get, and

$\mathcal{F}_i^{max}$: is the maximum value feature $\mathcal{F}_i$can get.

Whereas the symbolic features including protocol type, service and flag are mapped to integer values. The protocol type feature takes three values TCP, UDP and ICMP accordingly; these values are mapped to 1, 2 and 3 respectively. Service feature has 67 nominal values so these values are mapped to integer values in range $[1, 67]$ and flag feature has 11 nominal values so these values are mapped to integer values in range [1 ,11]

Moreover, after analyzing NSL KDD dataset, feature number 20 named num_outbound_cmds which is a boolean feature has only zero value for all network packets. Accordingly, this feature is removed from the dataset.

3.2 GA-based Clustering Phase

In this section, optimization models based on GA to identify the centroids for clustering the network packets into normal and attack are proposed. Two models are introduced for intrusion detection problem that take into consideration to satisfy maximum detection rate and maximum true negative rate. The first model coined as ($GA\ \#1$) handles numeric features of the network packet, whereas the second one coined as ($GA\ \#2$) concerns all features. The detailed explanation of the main characteristic components of the proposed models is provided in what follow.

3.2.1 Individual Representation

An individual denotes a solution in the search space and its representation depends on the problem to be solved. In the proposed models, cluster centroids are encoded in the individual. Each individual I is represented as a vector of n genes that corresponds to K cluster centroids where n equals to $F \times K$. This means, the first F genes represent the first cluster, the next F genes represent the second cluster and so on.

In $GA\ \#1$ model, each gene of an individual is represented as a real value. The mathematical formulation of the individual for $GA\ \#1$ model is as follows:

$I = \{i_1, i_2, \dots, i_n\}$,

$\forall k \in \{1, \dots, K\}$,

$\forall j \in \{1, \dots, F\}, i_h \in [min_j, max_j]$
Where
$h = (k-1) \times F + j$,
min_j : is the minimum value feature j can get, and
max_j: is the maximum value feature j can get.

Whereas, $GA\ \#2$ model adopts mixed representation of real valued, binary and integer corresponding to numeric features, boolean features and symbolic features of a network packet, respectively. The mathematical formulation of the individual for $GA\ \#2$ model is as follows:
$I = \{i_1, i_2, \dots, i_n\}$,
$\forall k \in \{1, \dots, K\}$,
$\forall j \in \{1, \dots, F\}$,

$$i_h \in \begin{cases} [min_j, max_j] \text{ if the type of feature } j \text{ is a numeric} \\ \{min_j, \dots, max_j\} \text{ if the type of feature } j \text{ the is a boolean or a symbolic} \end{cases}$$

Where
$h = (k-1) \times F + j$,
min_j : is the minimum value feature j can get, and
max_j: is the maximum value feature j can get.

3.2.2 Population Initialization

GA is a population-based optimization algorithm and starts with a population $\mathbb{P}$ of N solutions (individuals) generated by selecting randomly K network packets from the training NSL-KDD dataset for each individual. Formally speaking, $\mathbb{P}$ can be formulated as follows**:**
$\mathbb{P} = \{I_1, I_2, \dots, I_N\}$
Where N is the population size.

3.2.3 Objective Function

The objective function measures the quality of the individuals. According to the intrusion detection problem, the formulation of the objective function requires maximizing the number of network packets that are correctly detected as an attack (TP) and maximizing the number of network packets that are correctly classified as normal(TN). Objective function of the proposed models is formulated as in Equation 2.

$$Maximize\ \ ObjFun(I) = \beta \times \frac{TP \times TN}{TP + TN} \quad (2)$$

Where
β is scalar parameter > 1.
The computation of the objective function for each individual is as follows. First, the centroids, $\mathbb{C} = \{C_1, C_2, \dots, C_K\}$, encoded in the individual are extracted and the clusters are formed. Then, the cluster is attained by assigning the network packets $S_j = \{s_{j1}, s_{j2}, \dots, s_{jF}\}$, $\forall j \in \{1, \dots, t\}$, to a cluster corresponding to the closest centroid.
In $GA\ \#1$ model, Euclidean distance metric is adopted for the computation of the distance between a network packet and the centroid including F_n features of numeric type as in Equation 3.
$\forall k \in \{1, .., K\} \wedge \forall j \in \{1, \dots, t\}$

$$d(S_j, C_k) = \sqrt{\sum_{i=1}^{F_n} (s_{ij} - c_{ik})^2} \quad (3)$$

Whereas, in $GA\#2$ model, the computation of distance between a network packet and the centroid including F_n numeric features and F_d boolean and symbolic features is formulated as in Equation 4.
$\forall k \in \{1, .., K\} \wedge \forall j \in \{1, \dots, t\}$

$$d(S_j, C_k) = \sqrt{\sum_{i=1}^{F_n} (s_{ij} - c_{ik})^2} + \sum_{i=1}^{F_d} x_i \quad (4)$$

$$x_i = \begin{cases} 0 \ \ if\ s_{ji} = c_{ki} \\ 1 \ \ Otherwise \end{cases}$$

3.2.4 Selection Operator

The selection process selects individuals based on their quality to allow them to be parents in next generations. The most popular selection operator utilized in GA literature is tournament selection. In tournament selection, two individuals are selected randomly out of the population. The individual

with higher fitness value is chosen to be a parent. Also, elitism is used in the proposed work to avoid losing the best individual. In elitism, a small portion of the fittest individuals is copied into the next generation.

3.2.5 Recombination Operator

Recombination operator is used to explore the search space and it depends on the individual representation. In recombination, two parents ($I^1 and\ I^2$) are combined to generate one child (I^c) with recombination probability $p_c \in [0,1]$.

In $GA\ \#1$ model, the intermediate recombination is adopted in which gene values of I^1 and their corresponding gene values of I^2 are averaged to yield I^c as stated in Equation 5.

$$\forall j \in \{1, \dots, n_r\},\ i_j^c = \frac{i_j^1 + i_j^2}{2} \tag{5}$$

Where

n_r: is the number of real coded genes.

On the other hand, $GA\ \#2$ model adopts intermediate recombination for real coded genes as stated in Equation 5 in addition to discrete recombination for binary and integer coded genes. In discrete recombination as stated in equation 6, each gene of the child chromosome is inherited from one of the two parents depending on uniform random number, r, over range $[0, 1]$. The gene of the child chromosome is inherited from the first parent, I^1, if $r \leq 0.5$. Otherwise, the gene of the child chromosome is inherited from the second parent, I^2.

$$\forall h \in \{1, \dots, n_h\}, i_h^c = \begin{cases} i_h^1, & r \leq 0.5 \\ i_h^2, & r > 0.5 \end{cases} \tag{6}$$

Where

n_h: is the number of binary and integer coded genes**.**

3.2.6 Mutation Operator

In mutation operator, the genetic information of the child chromosome (I^c) is modified. For each gene in individual (I^c), a random number, r over range $[0,1]$ is generated and the gene is mutated with mutation probability $p_m \in [0,1]$. The mutation probability is set to $p_m = \frac{2}{n}$

The mutation operator of $GA\ \#1$ model that includes only real gene values in individual is Gaussian mutation [15] that modifies real coded genes through adding to the gene value, a number generated randomly from a Gaussian distribution with mean and variance equivalent to the mean and variance of the corresponding feature in $\mathbb{S}$ as stated in Equation 8.

$\forall k \in \{1, \dots, K\},$

$\forall j \in \{1, \dots, F\}$

$$G_M(i_h^c) = \begin{cases} i_h^c & , r > p_m \\ i_h^c + \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{\frac{-\left(i_h^c - \mu_j\right)^2}{2\sigma_j^2}} & , r \leq p_m \end{cases} \tag{8}$$

Where

μ_j and σ_j^2 are the mean and variance respectively of jth feature in $\mathbb{S}$.

$h = (k-1) \times F + j,$

On the other hand, the mutation operator of $GA\ \#2$ model that includes real, boolean and integer gene values in the individual is performed as follow: A gene j value is checked if it is real, binary or integer. If its value is real, then the value of the j^{th}mutated gene is changed using Gaussian mutation as stated in Equation 8. Otherwise, a new mutation operator is proposed to handle the existence of binary and integer value in the individual that exploits the mode value of binary and symbolic features in $\mathbb{S}$. The j^{th} gene value is modified to value Mo, that corresponds the mode of j^{th}features in dataset $\mathbb{S}$ as stated in Equation 9.

$$G'_M(i_h^c) = \begin{cases} i_j^c\ , & r > p_m \\ Mo\ , & r \leq p_m \end{cases} \tag{9}$$

Where

Mo: is the mode of jth feature in $\mathbb{S}$.

3.2.7 Termination condition

The process of producing a new mating pool via selection, recombination and mutation operators is continued up to a maximum number of generations (max_{gen}) is reached.

3.3 Detection Phase

After GA-based clustering phase constructs the clusters, the detection phase is ready to detect intrusions. The purpose of detection phase is to categorize the network packets in NSL-KDD testing dataset, $\mathbb{S}' = \{S'_1, S'_2, \dots, S'_{n_t}\}$, as normal or attack. In detection phase, the centroids $\mathbb{C} = \{C_1, C_2, \dots, C_K\}$ resulted from the GA based clustering phase are used as the input to the detection phase**.** Then, the distance between a network packet and the generated centroids is computed using either Equation 3 or Equation 4 depending on which GA model is adopted (i.e. $GA\#1$ model or $GA\ \#2$ model).

Finally, after computing the distance, the network packet is assigned to the closest cluster. It is recognized as an anomaly if the cluster label is attack, otherwise, it is recognized as normal.

4. Experimental Results

As mentioned previously, NSL-KDD dataset is considered as a source of information for the proposed genetic algorithm based clustering for intrusion detection. The NSL-KDD dataset is divided into two sets. One set is a training set, which is used for tuning the centroids of the clusters. Whereas, the second one is a testing set which is used for evaluating the performance of the proposed genetic algorithm based clustering for intrusion detection. A five-fold cross-validation is adopted to produce five distinct folds. One fold is used as a testing set and the other four folds are used as training set. Then, this procedure is repeated five times such that each fold is used as a testing set only one time. Table- 1 quantifies the number of instances in training and testing folds.

Table 1- Number of normal and attack for each fold in training set and testing set

Fold #	Training set		Testing set	
	Normal	**Attack**	**Normal**	**Attack**
1	10781	9373	2668	2370
2	10788	9366	2661	2377
3	10727	9427	2722	2316
4	10727	9426	2722	2317
5	10773	9380	2676	2363

The experiments run under Windows 7 professional service pack 1 operating system, Intel(R) Core(TM)2 i5-337U CPU @ 1.80GHz, 4 GB random access memory and 64-bit system type and the proposed GA-based clustering for intrusion detection are coded in MATLAB R2010b. Ten experiments for each fold are conducted for evaluating the capability of the proposed GA-based clustering for intrusion detection in partitioning NSL-KDD dataset. Moreover, an average of ten different runs for each fold is reported as average detection rate DR_{avg}, average true negative rate TNR_{avg} and average accuracy Acc_{avg}. All the experiments are conducted with the same GA parameters setting as clarified in Table- 2. The performance evaluation of $GA\#\ 1$ model and $GA\#\ 2$ model for each fold is reported in Table- 3.

Table 2- GA parameters setting

Parameter	**Value**
Population size, N	100
Maximum number of generations, max_{gen}	100
Crossover probability, p_c	0.8
p_m	$\frac{2}{n}$
Elitism	10

Table- 3 clearly points out that $GA\#2$ model significantly outperform $GA\#1$ model. This belongs to the positive impact of inclusion of symbolic and binary features in the individual representation of $GA\#2$ model. Furthermore, the proposed mutation for handling symbolic and binary features has a positive impact on the performance of $GA\#2$ model.

Table 3- Acc_{avg}, TNR_{avg} and DR_{avg} of GA #1 and GA #2 models for five folds

Fold #	Acc_{avg}%		TNR_{avg}%		DR_{avg}%	
	GA #1	GA #2	GA #1	GA #2	GA #1	GA #2
1	92.6	95.8	91.3	96.6	94.8	95.9
2	92.8	95.9	90.9	95.9	95.7	95.8
3	93.1	95.90	91.9	96.6	94.6	95.50
4	92.3	95.9	90.3	96.3	94.5	95.00
5	92.8	95.3	91.6	95	94.3	95.1

To show the effectiveness of the proposed GA-based clustering for intrusion detection, its performance is compared against two baseline models namely k-means and k-prototypes. The performance comparison of the proposed models is reported in terms of an average accuracy(Acc') over ten runs, average of true negative rate (TNR') average of detection rate (DR') and computation time. Table-4 clarifies the performance comparison regarding $Acc'\%$, $TNR'\%$ and $DR'\%$ of the proposed GA based clustering for intrusion detection over k-means and k-prototypes. Furthermore, the percentage of relative improvement of the proposed models at all evaluation metrics over the two baseline models is reported in Tables- 5, 6.

Table 4- Performance comparison regarding $Acc'\%$, $TNR'\%$ and $DR'\%$ of the proposed models against k-means and k-prototypes

Model	$Acc'\%$	$TNR'\%$	$DR'\%$
k-means	90	89	90.8
k-prototypes	90.8	90.4	91
GA **#1**	92.72	91.2	94.78
GA **#2**	95.76	96.08	95.46

Table 5- Percentage of relative improvement of the proposed $GA\#1$ over baseline models k-means and k-prototypes)

Model	Relative Improvement% of GA #1		
	Acc'	TNR'	DR'
k-means	2.311	1.663	3.458
k-prototypes	1.41	0.088	3.231

Table 6- Percentage of relative improvement of the proposed $GA\#2$ over the proposed $GA\#1$ and baseline models

Model	Relative Improvement% of GA #2		
	Acc'	TNR'	DR'
k-means	6.4	7.955	5.132
k-prototypes	5.463	6.283	4.901
$GA\#1$	3.279	5.351	0.717

Comparison result reported in Table-4 in addition to the percentage of relative improvement results observed in Tables- 5 and 6 show that the proposed $GA\#1$ model outperforms k-means and k-prototypes baseline models in all evaluation metrics. Results observed in the above tables ensure that the proposed $GA\#2$ model outperforms the proposed $GA\#1$ model and the baseline in all evaluation metrics. This proves the positive impact of adopting binary and symbolic features for intrusion detection with the assistance of the new mutation.

Table- 7 shows the results obtained by comparing the run time of the preprocessing phase and clustering phase (i.e., training time) and detection phase (i.e., testing time) of the proposed models and baseline models.

Table 7- Comparison regarding runtime of the proposed models against baseline models.

Model	Training time (sec)	Testing time (sec)
k-means	329.709	0.228
k-prototypes	668.949	0.284
$\boldsymbol{GA\#1}$	1924.672	0.252
$\boldsymbol{GA\#2}$	2464.112	0.2606

From Table- 7, it is obvious that the elapsed time to build the required centroid by the proposed GA-based clustering models is greater than baseline models. However, the detection time of the proposed models is comparable to k-means and k-prototypes. The time of detection phase of the proposed $GA\#2$ models is larger than the detection phase time of $GA\#1$ models. Due to extra computation required for the additional binary and symbolic features.

5.Conclusions

The proposed work is concerned with the problem of intrusion detection and how discrimination the normal from attack network traffics can be achieved. The problem of constructing a set of clusters for intrusion detection is modeled using GA. The proposed two models adopt two representations for a solution. In the first one, a solution is represented as real values that correspond to cluster centroid. While, in the second one, the solution is represented as a mixed type of real values, binary and integer. Furthermore, the proposed objective function of the two models depends upon how many normal network traffics are classified correctly as normal and how many attack traffics are correctly detected as attacks. Moreover, a new mutation operator was proposed to handle the mixed type of values. The overall results reveal the following points:

1. The positive impact of normalizing NSL-KDD dataset in the preprocessing step that avoids some features' domination over others.
2. The proposed $GA\#1$ model provides better accuracy, detection rate and true negative rate than the baseline models k-means and k-prototypes.
3. The proposed $GA\#2$ model ensures their ability to partition NSL-KDD dataset with higher accuracy, detection rate and true negative rate over the proposed $GA\#1$ models and baselines models.
4. The results indicate that adding binary and symbolic features in the individual representation of $GA\#2$ models has a positive impact on its final performance.

This work can be developed via working on preprocessing phase through adding a feature selection technique for reducing the number of features and selecting the relevant features. Furthermore, one of the important future directions for the proposed work is to adopt ISCX 2012 dataset for intrusion detection.

References

Liu, J. and Gao, M. **2008**. Unsupervised Classification Algorithm for Intrusion Detection based on Competitive Learning Network. *International Symposium on Information Science and Engineeringpp* (ISISE '08), pp: 519-523, IEEE.

2. Kruegel, C., Valeur, F. and Vigna, G. **2005**. *Intrusion Detection and Correlation Challenges and Solutions*. Springer Science + Business Media, USA.

3. Lin, W. C., Ke, S. W., and Tsai C. F. **2015**. CANN: An intrusion detection system based on

combining cluster centers and nearest neighbors. *Knowledge-Based System*, **78**(1): 13–21 , Elsevier.

4. Lundin, E. and Jonsson, E. **2002**. Survey of research in the intrusion detection area.Technical Report 02-04 , Chalmers University of Technology.

5. Casas, P., Mazel, J., and Owezarski, P. **2012**. Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge. *Computer Communication.*, **35**(7): 772–783, Elsevier.

6. Wang, G., Hao, J., Ma, J. and Huang, L. **2010**. A new approach to intrusion detection using Artificial Neural Networks and Fuzzy Clustering. *Expert Systystem Application.*, **37**(9): 6225–6232, Elsevier.

7. Li, Z., Li, Y., and Xu, L. **2011**. Anomaly Intrusion Detection Method Based on K-Means Clustering Algorithm with Particle Swarm Optimization. *International Conference onInformation Technology, Computer Engineering and Management Sciences (ICM),* **2**: 157–161, , IEEE.

8. Horng, S. J., Chen, M. Su , Kao, T. , Chen, R., Lai, J. , and Perkasa, C. D. **2011**. A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert System Application*, **38**(1): 306–313, Elsevier.

9. Karami, A. and Guerrero-Zapata, M. **2015** A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks. *Neurocomputing*, 149(Part C): 1253–1269, Elsevier.

10. Eslamnezhad, M. and Varjani, A. Y. **2014**. Intrusion Detection Based on MinMax k-means Clustering. *International Symposium on Telecommunications*, pp. 804–808, IEEE.

11. Duque, S. and Bin, Omar, M. N. **2015**. Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS). *Procedia Computer Science*, **61**: 46–51, Elsevier.

Aissa, N. B. and Guerroumi, M. **2015.** A genetic clustering technique for Anomaly-based ntrusion Detection Systems, 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, IEEE.

13. *The NSL-KDD Dataset.* Available at http://iscx.ca/NSL-KDD/.

14. Saad, S. **2012**. A Fuzzy Based Clustering for Intrusion Detection, MSc. Thesis, Department of Computer Science, College of Science, University of Baghdad.

15. Simon, D. **2013**. *Evolutionary Optimization Algorithms*. John Wiley & Sons.