ISSN: 0067-2904
GIF: 0.851

Hybrid Fuzzy Logic and Artificial Bee Colony Algorithm for Intrusion Detection and Classification

Mahmood S. Mahmood*
College of Science, Mosul University, Mosul, Iraq

Abstract

In recent years, with the growing size and the importance of computer networks, it is very necessary to provide adequate protection for users data from snooping through the use of one of the protection techniques: encryption, firewall and intrusion detection systems etc. Intrusion detection systems is considered one of the most important components in the computer networks that deal with Network security problems. In this research, we suggested the intrusion detection and classification system through merging Fuzzy logic and Artificial Bee Colony Algorithm. Fuzzy logic has been used to build a classifier which has the ability to distinguish between the behavior of the normal user and behavior of the intruder. The artificial bee colony algorithm has been used to build the classifier which was used to classify the intrusion into one of the main types (DoS, R2L , U2R, Prob). The proposed system has the ability to detect and classify intrusion at high speed with a small percentage of false alarms as well as to detect the new attacks. The NSL-KDD dataset used in the training and testing the proposed system.The results of experiments showed that the efficiency of the proposed system performance were (97.59%) for the intrusion detection, and (0.12%) for the false alarms. Also, the Classification rates for classes (DoS, R2L,U2R,Prob) were (97.19, 77.09, 98.43, 93.23) Respectively, which is considered a superior performance comparing with other methods in the literature.

Keywords: Intrusion Detection System (IDS), Artificial Bee Colony (ABC) algorithm, Fuzzy Logic, NSL-KDD

تهجين المنطق المضبب و خوارزمية مستعمرة النحل الذكي لكشف وتصنيف التطفل

محمود صبحي محمود*
كلية العلوم ، جامعة الموصل ، الموصل ، العراق

الخلاصة

في السنوات الاخيرة ، مع النمو في حجم الشبكات الحاسوبية واهميتها ، فمن الضروري جدا توفير حماية كافية لبيانات المستخدمين من التطفل من خلال استخدام احدى تقنيات الحماية : التشفير ، الجدار الناري ، انظمة كشف التطفل وغيرها . تعتبر انظمة كشف التطفل احدى اهم المكونات الرئيسية في الشبكات الحاسوبية التي تتعامل مع مشاكل أمنية الشبكات الحاسوبية . في هذا البحث ، اقترح نظام كشف وتصنيف التطفل من خلال الدمج بين المنطق المضبب و خوارزمية سرب النحل الذكي . استخدم المنطق المضبب لبناء المصنف الذي له القدرة على التميز بين سلوكيات المستخدم الاعتيادي وسلوكيات المتطفل . كذلك استخدمت خوارزمية سرب النحل الذكي لبناء مصنف يعمل على تصنيف التطفل الى احد اصنافه الرئيسية (DoS, R2L,U2R,Prob) . النظام المقترح له القدرة على كشف وتصنيف التطفل بسرعة عالية مع نسبة قليلة من الانذارات الكاذبة بالإضافة الى كشف الهجمات الجديدة . استخدمت مجموعة بيانات (NSL KDD) في تدريب واختبار النظام المقترح . أوضحت نتائج التجارب الى كفاءة أداء النظام المقترح مع نسبة كشف التطفل

*Email: mahmoodsubhy1981@gmail.com

(97.59%) و نسبة الانذارات الكاذبة (0.12%) . كذلك نسب التصنيف للأصناف (DoS, R2L,U2R,Prob) هي (97.19 ، 77.09 ، 98.43، 93.23) على التوالي . والذي يعتبر اداء مميز مقارنة مع اداء الطرائق المستخدمة في بحوث سابقة .

1. Introduction

The widespread use of computer networks in our daily life increased the possibilities of intrusion and viruses that may cause the loss of important information [1,2], Thus, the security of computer networks and operating systems that works in devices associated with the computer networks are very important issue. It was also noticed recently with the growth in the broad Internet, as it is becoming the use of the Internet not only through a PC, but on mobile devices as well. The intrusion is used by the intruder to attack by finding the weaknesses in computer systems or computer networks devices, then get spam on some allowances such as reading data in a secure and confidential or inflict mass in the system or the user files [3].

The intrusion detection systems (IDS) is a software or a hardware or both [4] to detect the behaviors that may be harmful through the use of complex tools monitor the mobile data through the system or the computer network continuously and give alarm signs to the user or the network administrator in case of security breach. [5].

Sometime, the intrusion detection problem is considered one of the pattern recognition problems which is used to make a distinction between TCP / IP connections (normal and abnormal), where training machine algorithms play a vital role in this area. For instance, Artificial Neural Networks (ANN) [6] , Support Vector Machine (SVM) [7] , Genetic Algorithm (GA) [8], Fuzzy Logic (FL) [9] and Data Mining (DM) [10] which are used in building the intrusion detection and classification systems , they work to detect and classify the intrusion whether known or unknown within the large amount of complex and dynamic data.

IDS have three functional components: Information sources, an analysis engine and a decision maker. Information sources are considered as an event generator. It monitors different sources that can be divided into three categories: data sources which are related to the operating system (e.g. System calls, System logs and transmitted the data over the network, etc.). Secondly, the network traffic monitors which generate raw network packets. Thirdly, the data collectors. The analysis engine detect the signs of intrusion. The outcome of analysis engines is forwarded to the decision maker. The decision maker takes the outcomes of the analysis and applies some rules, and gives them addresses concerning the reaction to be done (to stop a particular program or a particular treatment or shut down a port (port) within the hardware that is associated with the computer network, or give the alarm to the network manager) [11].

According to the strategies of the detection used, intrusion detection systems can be classified into two main categories [11,12] which are (1) Misuse detection which is considered as a complementary approach to anomaly detection. In misuse detection, already known attacks can be detected effectively and efficiently by using well defined patterns and explicit knowledge of them and (2) Anomaly detection typically involves the creation of knowledge based that contains the profiles of the monitored activities. Anomaly detection is based on the normal behavior of the user or a system. In anomaly detection both known and unknown attacks are detected by analyzing changes in the normal behavior of the computer system.

2. NSL-KDD Dataset

NSL-KDD dataset provided by Tavallaee et al [14]. Which is considered updated version of original KDD 99 dataset, it includes the same features of the KDD 99 dataset (41 features) with the name of the attack, which are located within one attack classes (Prob, User to Route (U2R), Remote to local (R2L), Denial of Service (DoS))) or normal connection (Normal). NSL-KDD contains 125973 records of training samples and 22544 records of testing samples which are used to achieve researchers experiments and compare different intrusion detection methods in this field [15]. The difference between NSL-KDD dataset and 99 KDD dataset can be summarized as follows [15-17].

- It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records.
- There is no duplicate records in the proposed test sets; therefore, the performance of the learners cannot be biased by the methods which have better detection rates on the frequent records.

- The number of the selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set.
- The number of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion.

3. Related Works

In 2010, Panda et al [18] proposed an efficient model for intrusion detection system that combines Naïve Bayes with three different algorithms principal component analysis (PCA), random projection (RP) and nominal to binary (N2B) to reduce features. A selected samples of NSL-KDD dataset was used in the experiments, and the results showed a high detection accuracy (96.5%) when combines Naïve Bayes with N2B algorithm and (3%) false alarm rate, while the accuracy detection of combining Naïve Bayes with PCA and RP were(81.4% and 94.8%), respectively, and the false alarm rate were (12.8% and 4.4%), respectively.

In 2010, Sini Joseph et al [19] presented intrusion detection system using anomaly detection strategy by combines the Principal component analysis (PCA) with neural networks (PCANNA). the proposed algorithm reduced training and testing time to (40%) and (70%), respectively, the selected of records in NSL KDD dataset has been used in training and testing .

In 2013, Ibrahim et al [20] used self-organizing maps (SOM) algorithm and anomaly detection strategy to build network intrusion detection system. KDD 99 dataset and NSL-KDD used in training and testing the system . The experiments obtained (92.37%) detection ratio when dealing with KDD 99 dataset and (75.49%) when dealing with NSL-KDD data set.

In 2013, Rowayda et al [21] presented intrusion detection system based on anomaly detection strategy by combining Neural Networks with Rough Set theory which is worked to reduce the number of features in the system while Neural Networks used as intrusion detection, NSL-KDD dataset used in all experiments. Testing result showed that the system works to reduce the time spent by the processor, as well as reduce the space used in memory were obtained (96.7%) detection rate with the error rate (3%).

In 2013, Deeman et al [22] used K-Star algorithm in building intrusion detection system using ready-made software (WEKA) which includes a range of training machine algorithms were used 60% of the NSL-KDD dataset in the training system and the rest of the data is used in the testing system, the experiences have given good results with minimizing the time of detection.

In 2015, mahmod et al [23] built a hybrid system to detect intrusion by integrating the artificial bee colony algorithm with MLP algorithm, artificial bee colony algorithm used to train the MLP algorithm by finding the optimal weights and Bias values. NSL-KDD dataset is used in training and testing the system . experiments have shown high intrusion detection rate (87.54%) with small false alarms (0.124%).

4. Using Fuzzy logic in building intrusion detection systems

Recently , due to the increasing number of computers which are connected to the Internet and the lack of intrusion detection, which prevents the occurrence of intrusion, it has become very important to detect intrusion at the first moment of occurrence and taking the necessary actions to reduce the potential damage [24]. The variety of the techniques that have been applied to the intrusion detection include data mining techniques, artificial intelligence techniques including the decision trees, neural networks and artificial Fuzzy logic. Since building the intrusion detection system by using Fuzzy logic has a higher features than intrusion detection system built by using of other artificial intelligent technology, the Fuzzy logic act to reduce the number of false alarms through the separation of overlap between normal behavior and abnormal behavior accurately [25] .

In recent years, many researchers have focused on generating Fuzzy rules for building intrusion detection system using one of data mining techniques [4,25]. In this research, the proposed method used to generate Fuzzy rules that operate to detect intrusion through dealing with data mining effectively. The proposed method gave high detection rate with few false alarms ratio in addition to detecting the unknown attacks. The proposed method involves the following subsections:

4.1. Classification training data

In the first step, the input to the system is classified into two classes which are normal data Communications (Normal Class) and abnormal data (Attack). Training NSL-KDD dataset was used in the training system, which includes four basic categories of intrusion in addition to the normal data communication. Each communication has (41 features). The quality of the data in the feature may be a

continuous or a discrete value. The proposed system works with the continuous data only because there exist an overlap between the behaviors of attacks, especially between the R2L class and U2R class and similarities between the behavior of the R2L class and the normal behavior [11]. For these reasons.The proposed system will deal with the 34 features in the training data and ignore the rest of features.

4.2. Strategy for generation of fuzzy rules

In this Section, we will follow the automatic strategy for generating fuzzy rules, so the training process make it better. In the traditional use of Fuzzy logic, Fuzzy rules given to fuzzy system is done manually or by taking from experts [11], but it is difficult to generate fuzzy rules manually because of the large number of records in the training data, in addition to the large number of features in each record. For this reasons we will use the methods of data mining in order to get the best set of fuzzy rules. The result fuzzy rules hold features that have many occurrences that make the training process is applied in the system correctly. Generating fuzzy rules steps are as follows:

4.2.1. Find occurrence value in features

At first, find the recurring values in features of both Classes (Normal, Attack) and through these values are determined important features in the training data. Because the training NSL-KDD dataset contains continuous values, the traditional data mining algorithms such as Apriori [26] and FP-Growth [27] is not suitable because they are dealing with binary data only. So, we will find the occurrences of each value within each feature. Finding occurrences by inputting the minimum support, these occurrences are identified for both classes, normal and attack.

4.2.2. Select of suitable attributes for rule generation

In this step, we will choose the appropriate features that operate on the correct classification between the two classes. The goal of this step is to make the input data to the proposed system which includes 34 features, because not all of these features are useful in the detecting intrusion process, determining the appropriate features by using the deviation method that relies on occurrences of values within the feature. Initially, occurrences values are stored within the feature vector, so that 34 vectors are obtained for each class and as follows:

$$\mathbf{C_i=[v_1,v_2,v_3,\ldots.,v_j\ldots v_{34}]}$$

Where

- i=0 (Normal class) , i=1 (Attack class) .
- v_J = vector hold the occurrences values within feature j .

After that, we find the deviation range {min, max}.Where the max represents the highest value and min represents the lowest value within the occurrences of the feature, then the comparison between the deviation range of the feature within the Normal class and Attack Class. If there is a difference that indicates the possibility of adding this feature to the Rule, this will get the highest percentage of detection from the use all the features in the Rule and selected features are represented as follows:

$$\mathbf{C_i=[v_1,v_2\ldots\ldots v_k]} \qquad \mathbf{k<=34}$$

4.2.3. Rule Generation

Selected features in the previous step is used to generate the derived rules from the range [max, min] through the comparison between the deviation range of the feature selected for the classes normal and attack and identify the points of intersection between them and through those points can generate the specific and non-specific rules, as shown in the following example. If we have the deviation range of the feature No. 1 in normal class is {1.5} and the deviation range of the feature No. 1 in attack class is {2.8}, then the resulting rules take the following formula:

If attribute 1 > 5 then Attack
If attribute 1 between 2,5 then Normal or Attack
If attribute 1 < 2 then Normal

In addition, some of the data contains only one intersection point, which provides two rules only.

4.2.4. Rule filtering

To get an efficient Intrusion detection system, we should focus on the following criteria [28], the first must reduce the number of rules and the second makes the if-part in the generated Rule have the fewest features. The generated rules in the previous step contain specific and non-specific Rules. The specific rules contains one class in the then-part while the non-specific rules contains two classes in the then-part, so we will take specified rules only in the proposed system.

4.2.5. Generating Fuzzy Rule

Fuzzy rules can be generated in the traditional way by manually or by experts, but the proposed system is obtained automatically based on the occurrences of values within the features in the specific rules resulting from the previous step. The specific rules contain numerical values in the if-part as well as in the then-part, but the Fuzzy rules must contain the specific linguistic variable that corresponds to each value in features. So we should fuzzify the numerical values of specific rules by using the five symbols (S: small, MS: medium small, M: medium, ML: medium large, L: large). Figure-1 shows the relationship between the values of the features and membership function μ(x).

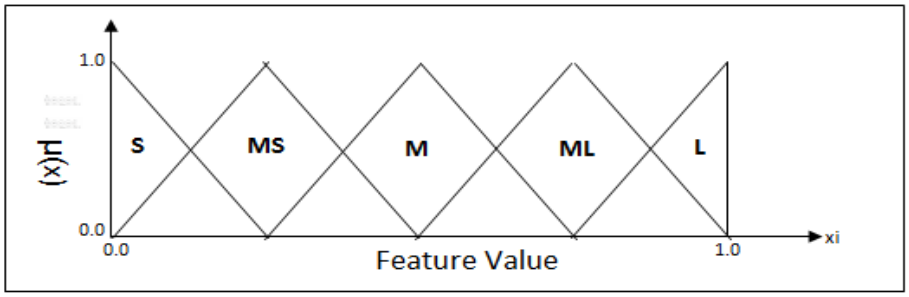


Figure 1- Relationship between the values of the features and membership function μ(x)

The membership function value μ (x) computed for each feature by applying the value of feature on Fuzzy group scheme as shown in Figure-2. According to the equation (1).

$$\mu(x) = Max\{0,1 - \left(\frac{|x-x0|}{b}\right)\} \quad \textbf{(1)}$$

Where

- b : Base of triangle .
- X0 : take value {0,0.25,0.5,0.75,1} correspond linguistic variable {S,MS,M,ML,L} .
- X : Value of Feature after Normalization operation.

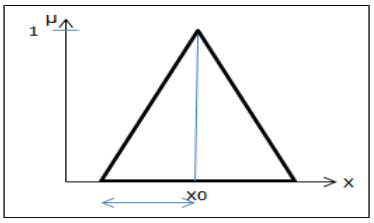


Figure 2- Fuzzy group scheme

Resulting Fuzzy rules from this stage is used in testing the proposed system, which gives a high detection rate in addition to reducing false alarms.

5. Swarm intelligence

Since the past two decades, Swarm intelligence used in many researches because of the unique behavior that mimics the insects behavior that live within the community [29-32]. Bonabeau [33] defined Swarm intelligence as any attempt to design algorithm inspired by the social behavior in social insects and other animal societies. It's mainly focused on insects that live within the community, such as ants and bees of different types. Also Swarm intelligence is defined as a group of individuals that interact with each other to solve complex issues.

5.1. Artificial Bee Colony (ABC)

ABC algorithm was proposed to solve optimization and classification and training neural networks based on the intelligent behavior of a swarm of bees [31, 34-36]. So, ABC algorithm became the most successful and powerful algorithm in resolving the issues that have more than one job, compared with PSO and GA algorithms, because the bees algorithm working on the division of accountability to different tasks and give everyone his duties entrusted to him.

5.2. Enhanced Artificial Bee Colony (ABC) Algorithm

The researcher [37] developed the basic components of the ABC algorithm and makes it perform the classification functions, as follows:

5.2.1. Rule Format

Classification rules consist of a set of features (N-1) and N represents the number of features found in the training dataset and each feature consists of two parts, the first part contains the minimum value and the second part contains the maximum value of this feature. Classifications rules contain in addition to the previous features, another three values, which are the Class that rule, belong to. The fitness function value and the percentage of the number of records are covered by that rule in the training dataset.

5.2.2. Fitness Function

The fitness function used to calculate the fitness value of the classification rules according to the equation 2.

$$\text{Fitness value}=(TP/(TP+FN))*(TN(TN+FP)) \quad \textbf{(2)}$$

Where

True positives (TP): the number of records covered by the rule that have the class predicted by the rule.

False negatives (FN): the number of records not covered by the rule but they have the class predicted by the rule.

False positives (FP): the number of records covered by the rule but their class does not predicted by the rule.

True negatives (TN): the number of records not covered by the rule and that do not have the class predicted by the rule.

We should consider the following notes when calculating the fitness function value for the classification rules:

a- When the algorithm examines the type of the record, it will measure every feature in the record. If the value of a feature is between the min value and max value for this feature, it means the feature can be covered by the rule. If all features for a record can be covered by the rule, it means the record can be covered by the rule.

b- If the class of the evaluated record is equal to the predictive class by the rule, this denotes that the record has the class predicted by the rule.

5.2.3. Exchanged Local Search Strategy

In the research step, in the case of failure to obtain the best solution or to reach the specified number of cycles, in this case, the bee needs to move to a new location by applying a local search, in the standard algorithm using equation 3 to apply a local search and it will take additional time if there are a large number of records in the training dataset and is not suitable for the classification applications. The researchers in [37] proposed a new method to achieve a local search and they called it (Exchange) to replace the original local search strategy in the standard algorithm according to equation 4.

$$V_{ij}=X_{ij}+\Phi_{ij}(X_{ij}-X_{kj}) \quad \textbf{(3)}$$

$$\mathbf{V_{ij}=X_{kj}} \quad \textbf{(4)}$$

Where V_{ij} represents a new location for the bee and X_{kj} represents the neighbor's location to the previous location, the value of each I, k must fall within the range (1 - SN) and K must take value different from the value of i and j which represents the number of Features in the records of training dataset.

5.2.4. Rule Discovery

The objective of classification rules exploration is to create a set of rules that act on identify specific class from a collection of classes, so the rule discovery stage is the most important stage for classification algorithm. Classification algorithms can discover classification rules for each class automatically, according to pre-defined class, classification rules discovery process will continue repeated until find a set of classification rules which cover all existing records in the training dataset of that class.

5.2.5. Rule pruning

After all classes have been processed and all classification rules have been generated, each rules will enter into a prune classification rules procedure. This procedure starts to calculate the fitness function value of the rule, then temporary deletes features one after the other form the current classification rules, after each delete operation the procedure calculates the fitness function value of the current rule. if delete operation of the current feature improve fitness function value then delete

feature from current rule Permanently else the current feature remain in classification rules. these process will continue repeated until all features are evaluated . The main goal of pruning is to delete the redundant features from the classification rules that makes it easy to be read. In addition to that it reduces the comparison between the values of the features and values of the test record. It leads to increase the speed of the of classification process and accomplishes high accuracy.

5.2.6. Predication Strategy

The classification rules resulting from the prune classification rules procedure will be used to predict all the new data in the test dataset that the classification has not already known. But in some cases, one record of the test dataset may cover more than one classification rule belonging to different classes. When that takes place, the prediction strategy will determine the classes by the following steps:

a. Calculate the prediction value for each classification rules that cover the test record by equation5.
Prediction Value =(α * rule fitness value) + (β * rule Cover Percentage) **(5)**
Where α and β are two weighted parameters associated with rule fitness value and rule cover percentage respectively, and α € [0,1] and β = [1- α] and rule cover percentage = (TP / N).

b. Accumulate the prediction values based on of different Classes (assembly prediction values of classification rules which fall within one class).

c. Select class which has the highest prediction value, test record is classify to that class.

6. Proposed System Design

The proposed system design passes through two basic Stages which are the training stage and the testing stage.

6.1. Training stage

The training stage is considered the most important stage in the design of any intrusion detection and classification system. The input to this stage is the training dataset as well as other factors according of training algorithm which is used by the system. When the training stage is completed successfully, we get the output that may be weighted or the classification rules. The proposed system has been training twice in parallel, the first system training is used to detect the intrusion by using Fuzzy logic and second system training is utilized for classifying the intrusion through the use of ABC algorithm. system training using the Fuzzy logic includes several steps as shown in Figure (3-A), it starts to enter the training NSL-KDD dataset from the text file in addition to the special values of the training variables, then it classifies the data into two classes (Normal or intrusion) and take 34 features which contain the continued values. After that, the process of generating the Fuzzy rules begin by finding the duplicated values within the features of both classes and then choose the appropriate features to generate classification rules that reduce the number of features in the classification rule through the use of the deviation range {min, max}. Based on finding the intersection points between the deviation range of the Normal class and the Attack class the classification rule will be generated which are either specific classification rules or non-specific classification rules. Before generating Fuzzy rules, the data in classification rules must be normalized which make the data within the range [0,1], in proposed system we used min-max method which is represented in the equation (6) to normalize the data.

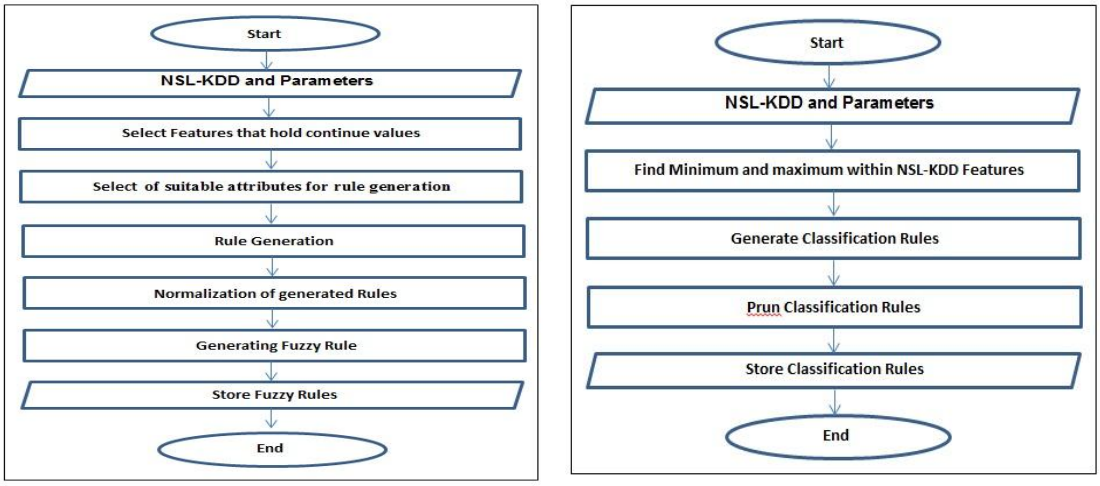
$$\text{Normalized (x)} = \frac{x - \min(X)}{\max(X) - \min(X)} \quad \textbf{(6)}$$

Where x represents the value in current feature (X), min(X) represents the lowest value in feature X and max(X) represents the highest value in feature X.

The normalized data in classification rules will be fuzzified through applying it on the Fuzzy group scheme as shown in Figure-2 and it calculates the value of a membership function μ (x) according to equation (1) to get the values in the if-part of classification rules that correspond to the Linguistic variables (S, SM, M, ML, L) and corresponds to the values in then-part to Linguistic variable (S, L) .

Training the proposed system using a ABC algorithm includes several steps as illustrated in Figure (3 - B),(1) enter the training NSL-KDD dataset from the text file and generate classification rules based on the style of the proposed research by the researchers [37], (2) calculates the highest and the lowest value for each feature in the training NSL-KDD dataset,(3) generate classification rules which consists of a set of features (41 features) and each feature includes two parts (the lowest value and the highest value of that feature). In addition to the fitness function values according to the equation (2), the percentage of the number of records covered by rule, class title of classification rule. (4) After

generating classification rule, Rule prune begin to pruning the classification rule by deleting some of the redundant features that makes the classification rules easy to read as well as to reduce the number of comparisons leading to speed up the process of classification.



A – Flow Chart for training Proposed system Using Fuzzy Logic

B – Flow Chart for training Proposed system Using ABC algorithm

Figure 3- Flow Chart for Training Proposed System

6.2. Testing stage

As we mentioned earlier, the proposed system consists of two parts, the first one will acts as intrusion detection, the NSL-KDD testing dataset input into the decision-making unit of Fuzzy system, the output from this unit either Normal connection or intrusion, if the output of the first part is intrusion, the same data of test record will enter the second part of the proposed system, which is turn classifies the test record into one of the classes (Normal, Dos, R2l, Prob, U2R) as shown in Figure-4, which represents the general outline of the proposed system.

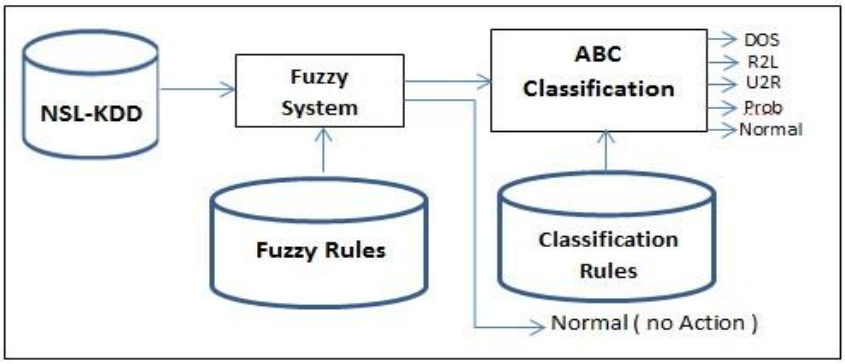


Figure 4- General diagram for Proposed System

In the system test stage, NSL-KDD testing data input from the text file, Features of continuous values are taken from the test data only (34 features) and applied to Fuzzifier, which converts the numerical values to fuzzy symbols (S, SM, M, ML, L) using organic triangular relationship as described in paragraph 4-2-5. Input the fuzzy data into the inference machine that operates relying on the comparison of the fuzzy data to the knowledge bases that contain a set of Fuzzy rules which resulted from the system training stage. The output from the inference machine will be one of the Linguistic variables (S or L), which turned into a (0 for Normal connection and 1 for intrusion) by applying difuzzifier. In the normal connection (output equal to 0) the test data will pass over computer network without any additional actions and the proposed system moves to the next data record of the NSL-KDD testing dataset. In the case of intrusion (output equal to 1) the test record will pass to part Two of the proposed system. It compares the values of features in the test record (41 features) with the classification rules resulting from the system training stage using a ABC algorithm. When the test record match with a classification rules the output will be equal to the class that rule belong to it .

7. Experiments and Results

The proposed system has been built using Microsoft Visual C # 2010 and was evaluating the performance of it by calculating each of the detection ratio and classification ratio in addition to the proportion of false alarms and the time spent for training and testing processes. NSL-KDD dataset was used in training and testing the proposed system, it is the most frequently used in the Evaluation of intrusion detection and classification systems. We used two computers to achieve experiments for training proposed system that works to detect intrusion and find Fuzzy rules by doing eight different experiences, which depends on change the number of training records and the number of features in addition to the minimum support values within the feature as shown in Table-1.

Table 1- Experiments of training the proposed system using fuzzy logic

Experiment	Records count	Features Count	Minimum	Time	Rules
1	125973	34	10	04:21:30	37
2	125973	17	10	02:40:42	45
3	75000	34	10	02:10:17	29
4	75000	17	10	01:10:02	35
5	125973	34	20	04:02:13	35
6	125973	34	25	03:56:01	34
7	125973	17	20	02:03:33	24
8	125973	17	25	01:03:48	32

In classification of intrusion, we used two computers to perform the experiments for training the proposed system to find classification rules by doing six experiences that depend on changing the number of bees members and the minimum number of the records covered by the classification rules as shown in Table-2.

Table 2- Experiments of training the proposed system using ABC algorithm

Experiment	Records	Bees Count	Minimum covered	Time	Rules
1	125973	41	35	13:15:13	102
2	125973	41	25	14:37:10	143
3	100000	41	35	09:47:37	93
4	100000	41	25	10:52:22	132
5	125973	34	35	10:12:21	89
6	125973	34	25	12:14:44	117

The testing process of the proposed system depends on the records of the NSL-KDD testing dataset (22544 records) and the Fuzzy rules and the classification rules resulting from the system training stage. Table 3 shows the test results of the proposed system to detect the intrusion by using Fuzzy logic, equation (7) adopted in the calculation of the intrusion detection rate.

$$DR = \frac{(TP+TN)}{(TP+TN+FN+FP)} \quad \textbf{(7)}$$

Table 3- Results of testing the proposed system using fuzzy logic

Experiment No.	Testing Records	Detection Rate	False Alarm	Time (H:M:S)
1	22544	94.30	0.23	00:00:52
2	22544	92.70	0.35	00:00:47
3	22544	87.12	0.63	00:00:36
4	22544	82.73	0.71	00:00:42
5	22544	94.67	0.27	00:00:49
6	22544	94.62	0.33	00:00:46
7	22544	86.87	0.54	00:00:32
8	22544	86.79	0.59	00:00:39

Table-4 shows the test results of the proposed system for the classification of Intrusion using ABC algorithm, the equation (8) adopted in the calculation of classification rate.

$$C.R. = \frac{Number\ of\ classified\ patterns}{total\ number\ of\ patterns} * 100 \quad \textbf{(8)}$$

Table 4- Results of testing the proposed system using ABC algorithm

Experiment No.	Testing Records Count	DOS	R2L	U2R	Prob	Norm	Time
1	22544	96.27	67	97.0	87.3	93.06	00:00:56
2	22544	97.02	89.09	96.2	89.0	95.03	00:01:13
3	22544	77.24	52.01	60.5	52.0	63.96	00:00:53
4	22544	78.82	58.77	63.0	58.7	70.61	00:01:37
5	22544	72.07	61.96	53.1	61.9	74.22	00:00:41
6	22544	73.66	67.81	58.4	67.8	79.92	00:01:08

Depending on the test results of the proposed system as described in Tables -3 and 4, we tested the proposed system fully by taking the Fuzzy rules resulting from the system training using Fuzzy logic in the experiment (5), which gave the highest percentage of intrusion detection as shown in Table-3 and the classification rules of intrusion resulting from the training system using ABC algorithm in the experiments (1 and 2) which gave the highest rating ratios as shown in Table 4. The final results of testing the complete proposed system are shown in Table-5.

Table 5- Final results of testing the proposed system

Experiment	Detection	False	DOS	R2L	U2R	Prob	Normal	Time
5+1	96.67	0.09	96.39	73.02	97.13	89.04	03.29	00:01:55
5+2	97.59	0.12	97.19	77.09	98.43	93.23	04.18	00:02:09

To investigate the performance efficiency of the proposed system, the obtained results from the proposed system compared to the results of a number of researchers whom work in the same field, depending on different factors including the number of records used in the testing process, the intrusion detection rate and Classification rate, the percentage of false alarms and time spent in the testing process. Table-6 shows that the integration of the fuzzy logic and ABC algorithm in the building intrusion detection and classification system gives good results for intrusion detection and classification. In addition to that, the researchers [38] obtained high detection and classification because they used a small number of records (7637 record) only compared to the number of records that were used in this proposed system (22544 records).

Table 6- The comparative results for the proposed system with the results of a number of researchers whom work in the same field

Methods	Testing	Detection	False	DO	R2L	U2	Prob	Nor	Time
ABC-MLP	22544	87.27	0.126						00:04:30
SOM [20]	22544	75.49	1.40						00:02:00
NN-IV [21]	9000	96.07	3						
LVQ_ERBP	7637	97.06	0.09	98.4	96.4	70.3	99.59	91	
LVQ_kNN	7637	89		99	39	81	96	94	
FLS [4]				80	85	95	80	10	
Proposed	22544	97.59	0.12	97.1	77.09	98.4	93.23	04.18	00:02:09

8. Conclusions

The detection and classification Intrusion system is considered one of the most important components in the computer networking environment and it plays a vital role in protecting the network from intrusion, since there are many artificial intelligence techniques used to build the intrusion detection systems by researchers in this field. In this research, intrusion detection and classification system was built using a combination of Fuzzy logic and Artificial Bee Colony algorithm. Moreover, the NSL-KDD dataset is used in the training and testing the proposed system. from the findings and observations, the proposed system gave a high rates of detection and classification with a small percentage of false alarms due to the use of Fuzzy logic that chooses the features that hold a continuing data which helps distinguish between the user's normal behaviors and Intrusion behaviors and uses the prediction strategy in artificial Bee Colony algorithm that runs on the distinction between the test records in the case of a test record covered by more than one classification rule that belong to the different classes by giving the test record to the class, which has the largest rate of prediction. The proposed system also runs at high speed to detect the intrusion and classification due to the use of the small number of Fuzzy rules which selected specified rules only and non- specified Fuzzy rules are not useful in the detection process, but useful in the process of classifying intrusion but further steps are needed to perform the task correctly, as well as changing the behavior of the Local Search in ABC

algorithm and the use of a prune classification rules procedure, which works to delete the redundant features from the classification rules. That, in turn reduces the comparison and the matching process. There are a future idea to apply the proposed system within the mobile network environment that contains many attacks to make some changes in the system in order to be compatible with the mobile network environment in terms of the system size and the system speed.

References

1. Yu, J. Tsai and Weigert, T. **2007**. An automatically tuning intrusion detection system. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 37(2):373-384.
2. Hu, W. and Maybank, S. **2008**. Ada Boost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 38(2):577-583.
3. Nivedita Naidu. And Dharaskar, R.V. **2010**. An Effective Approach to Network Intrusion Detection System using Genetic Algorithm. *International Journal of Computer Applications*, 1(3):26–32.
4. Dalila Boughaci, Samia Bouhali, and Selma Ordeche. **2011**. A Fuzzy Local Search Classifier for Intrusion Detection . Department of Computer Science, University of Sciences and Technology USTHB BP 32 El-Alia, Beb-Ezzouar, Algiers.
5. Mark Crosbie, and Gene Spa Ord. **1995** . Defending a Computer System using Autonomous Agents. Technical report.
6. Cannady J. **1998**. Artificial Neural Networks for Misuse Detection. In Proceedings of the '98 National Information System Security Conference (NISSC'98), pp:443-456.
7. Shon T, Seo J, and Moon J. **2005**. SVM Approach with A Genetic Algorithm for Network Intrusion Detection. *Lecture Notes in Computer Science, Springer Berlin / Heidelberg*, 3733: 224-233.
8. Yu, Y., and Huang, Hao. **2007**. An Ensemble Approach to Intrusion Detection Based on Improved Multi-Objective Genetic Algorithm. *Journal of Software*, 18(6):1369-1378.
9. Luo, J. and Bridges, S. M. **2000** . Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. *International Journal of Intelligent Systems*,15(8):687-704.
10. Lee, W., Stolfo, S. and Mok, K. **1999**. A Data Mining Framework for Building Intrusion Detection Model. In Proceedings of the IEEE Symposium on Security and Privacy, pp:120-132, Oakland, CA.
11. Falke A.D, Fulsoundar V.S, Pawase R.S., Wale S.B., Ghule S.J. **2014**. Network Intrusion Detection System using Fuzzy Logic . *International Journal Of Scientific Research And Education*, 2:626-635 ISSN (e): 2321-7545.
12. Honig, A., Howard, A., Eskin, E., and Stolfo, S. J. **2002**. *Adaptive Model Generation: An Architecture for the Deployment of Data Mining-Based Intrusion Detection Systems*. Applications of Data Mining in Computer Security, Kluwer Academic Publishers, Boston, MA, pp:154-191.
13. Karaboga, D. **2005** . An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey.
14. Tavallaee, M., Bagheri, E., Wei Lu, and Ghorbani, A . **2009**. A Detailed Analysis of the KDD CUP 99 Data Set . proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).
15. Shaheen, A. **2010** . A comparative Analysis of Intelligent Techniques for detecting Anomalous Internet Traffic . MSc. Thesis, Computer Eng. Dep. , King Fahd University(SAUDI ARABIA) .
16. Eid, H.F., Darwish, A., Hassanien, A.E., and Abraham, A. **2010** . Principle components analysis and support vector machine based intrusion detection system. In the Proceedings of 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010), Cairo, Egypt .
17. Datti, R., and Verma, B. **2010**. Feature reduction for intrusion detection using linear discriminant analysis. *(IJCSE) International Journal on Computer Science and Engineering*, 2(4):1072-1078.
18. Mrutyunjaya Panda, Ajith Abraham, and Manas RanjanPatra . **2010** . Discriminative Multinomial Naïve Bayes for Network Intrusion Detection . http://www.softcomputing.net/ias10_panda.pdf

19. Shilpalakhina, Sini Joseph and BhupendraVerma . **2010**. Feature Reduction Using Principal Component Analysis for Effective Anomaly–Based Intrusion Detection on NSL-KDD. *International Journal of Engineering Science and Technology*, 2(6):1790-1799.

20. L.M.Ibrahim,D.T.Basheer,M.S.Mahmod . **2013**. A comparison study for intrusion database (kdd99, nsl-kdd) based on self-organization map (som) artificial neural network. *Journal of Engineering Science and Technology***,** 8(1): 107 – 119.

21. Rowayda A. Sadek , M. Sami Soliman, Hagar S. Elsayed . **2013**. Effective Anomaly Intrusion Detection System based on Neural Network with Indicator Variable and Rough set Reduction. *IJCSI International Journal of Computer Science Issues*, 10(6), 2.

22. Deeman Y. Mahmood, Dr. Mohammed A. Hussein .**2013**. Intrusion Detection System Based on K-Star Classifier and Feature Set Reduction. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 15(5):107-112.

23. Mahmod S. Mahmod, Alnaish, Z. A. H., and Ismail Al-Hadi A. A. **2015**. Hybrid Intrusion Detection System Using Artificial Bee Colony Algorithm and Multi-Layer Perceptron. *(IJCSIS) International Journal of Computer Science and Information Security*, 13(2).

24. Qiang Wang and Vasileios Megalooikonomou. **2005**. A clustering algorithm for intrusion detection . in Proceedings of the conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, 5812:31-38.

25. Muna Mhammad T. Jawhar , Monica Mehrotra. **2010**. Design Network Intrusion Detection System using hybrid Fuzzy-Neural Network. *International Journal of Computer Science and Security*, 4(3).

26. Agrawal, R., Imielinski, T., and Swami, A. **1993**. Mining association rules between sets of items in large databases. In Proceedings of 1993 ACM SIGMOD Intl. Conf. on Management of Data, Washington, DC, pp: 207–216.

27. Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao. **2004**. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8(1):53 - 87.

28. Allen, J., Christie, A. and Fithen W. **2000**. State of the Practice of Intrusion Detection Technologies. Technical Report CMU/SEI-99-TR-028.

29. Van der Bergh, F., Engelbrecht, A. **2000**. Cooperative learning in neural networks using particle swarm optimizers. *South African Computer Journal***,** 26, pp: 84–90.

30. Ilonen, J., Kamarainen, J.I., Lampinen, J. **2003**. *Differential Evolution Training Algorithm for Feed-Forward Neural Networks*. Kluwer Academic Publishers, Neural Processing Letters 17: 93–105.

31. Eberhart, R.C., Shi, Y., and Kennedy, J. **2001**. *Swarm Intelligence*. USA, Morgan Kaufmann publishers ISBN 1-55860-595-9.

32. Darvis. Karaboga. **2005**. An idea based on honey Bee Swarm for Numerical Optimization Technique. Report-06, Erciyes University Engineering Faculty, Computer Engineering Department.

33. Bonabeau, E., Dorigo, M., and Theraulaz, G. **1999**. *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, New York.

34. Sandhya Samarasinghe. **2007**. *Neural networks for applied sciences and engineering*. September 12, 2006 by Auerbach Publications. ISBN 9780849333750 - CAT# AU3375.

35. Karaboga, D. and Bahriye Akay. **2009** . A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, 214:108–132.

36. Eiben, A.E. and Smith, J.E. **2003**. Introduction to Evolutionary Computing. *Springer*, ISBN 978-3-540-40184-1.

37. Mohd Afizi Mohd Shukran, and Yuk Ying Chung. **2011**. Artificial Bee Colony based Data Mining Algorithms for Classification Tasks. *Canadian Center of Science and Education*, 5(4).

38. Naoum, R., and Al-Sultani, Z. **2013**. Hybrid System of Learning Vector Quantization And Enhanced Resilient Back propagation Artificial Neural Network For Intrusion Classification. *International Journal of Academic Research*, 14(2):333.

39. Naoum, R. Abid, N. and Al-Sultani, Z. **2012**. A Hybrid Intrusion Detection System Based on Enhanced Resilient Backpropagation Artificial Neural Network and K-Nearest Neighbor Classifier. *International Journal of Academic Research*, 4(2):227.