Uludağ University Journal of The Faculty of Engineering, Vol. 25, No. 1, 2020 RESEARCH

DOI: 10.17482/uumfd.649003

A MODIFIED FIREFLY ALGORITHM-BASED FEATURE SELECTION METHOD AND ARTIFICIAL IMMUNE SYSTEM FOR INTRUSION DETECTION

Melike GÜNAY [*]
Zeynep ORMAN [*]

Received: 20.11.2019; revised: 05.02.2020; accepted: 27.03.2020

Abstract: Intrusion detection systems generally produce high dimensional data in network-based computer systems. It is required to analyze this data effectively and create a successful model by selecting the important features to save only the meaningful data and protect the system against suspicious behaviors and attacks that can occur in a system. Firefly Algorithm (FFA) is one of the most promising meta-heuristic methods which can be used to select important features from big data. In this paper, a modified Firefly Algorithm-based feature selection method is proposed. The traditional Firefly Algorithm is improved by using the K-Nearest Neighborhood (K-NN) classifier and an additional feature selection step. The proposed method is tested on 4 different datasets of various types of attacks. Three different sub-feature sets are obtained for each dataset and the classification performances are compared. Artificial Immune System (AIS) method is also implemented to generate artificial data for the datasets that have an insufficient number of data. This study shows that the proposed Firefly Algorithm performs successfully to decrease the dimension of data by selecting the features according to the obtained accuracy rates of the K-NN method. Memory usage is dramatically decreased over 50% by reducing the dimension with the proposed FFA. The obtained results indicate that this method both saves time and memory usage.

Keywords: Firefly Algorithm, Artificial Immune System, K-NN, Feature Selection

Saldırı Tespiti için Ateş Böceği Algoritması Tabanlı Özellik Seçim Yöntemi ve Yapay Bağışıklık Sistemi

Öz: Saldırı tespit sistemleri, genel olarak, ağ-tabanlı bilgisayar sistemlerinde yüksek boyutlu veri üretmektedir. Sistemi meydana gelebilecek ataklardan ve ağdaki şüpheli hareketlerden korumak ve sadece anlamlı veriyi saklamak için bu yüksek boyutlu verinin etkili bir şekilde analiz edilmesi ve başarılı bir model oluşturulması gerekmektedir. Ateş Böceği Algoritması, büyük veriden önemli özelliklerin seçilmesi için kullanılan en önemli üst-sezgisel algoritmalardan biridir. Bu çalışmada, Ateş Böceği Algoritmasına dayalı yeni bir özellik seçme yöntemi önerilmiştir. Önerdiğimiz bu yöntemde Ateş Böceği Algoritması, K-en yakın komşuluk algoritması ve ek bir özellik seçimi adımı ile iyileştirilmiştir. Önerilen yöntem, çeşitli saldırı türlerini içeren dört farklı veri kümesi ile test edilmiştir. Her veri kümesi için 3 farklı alt özellik kümesi elde edilmiştir ve her birinin sınıflandırmadaki başarısı ölçülerek karşılaştırılmıştır. Ayrıca, Yapay Bağışıklık Sistemi yöntemi ile veri sayısı yetersiz veri kümeleri için yapay veri üretildikten sonra Ateş Böceği Algoritması uygulanmıştır. Bu çalışma, önerilen Ateş Böceği Algoritması'nın, K-en yakın komşuluk yöntemi ile elde edilen sınıflandırma sonuçlarına göre özellikleri seçerek verilerin boyutunu azaltmak için başarılı bir şekilde çalıştığını göstermektedir. Veri boyutunun azaltılması ile hafıza kullanımı da %50'den fazla bir oranda azalmıştır. Elde edilen sonuçlar, önerilen yöntem sayesinde hem zamandan ve hem de hafıza kullanımından tasarruf edildiğini göstermektedir.

Anahtar Kelimeler: Ateş Böceği Algoritması, Yapay Bağışıklık Sistemi, K-NN, Özellik Seçimi

[*] İstanbul University-Cerrahpasa, Department of Computer Engineering, 34320, Avcilar/İstanbul
Corresponding Author: Zeynep Orman (ormanz@istanbul.edu.tr)

1. INTRODUCTION

Network-based computer systems are commonly used in different technological areas. The data used in such systems must be protected against the illegal attacks on the network. Intrusion detection systems (IDS) are developed to prevent computer systems from these attacks and provide high-level security.

IDS are mostly developed using data mining techniques and rule-based classifiers to determine the patterns that can be used to analyze user behaviors (Lee & Stolfo, 1998). Traditional machine learning techniques can also be applied to big amounts of data that are produced by network systems. The recursive support vector machine (R-SVM) method produces a high accuracy rate to detect abnormal activities and reduces the processing time by extracting the main features (Shang-fu & Zhao, 2012).

In recent studies, heuristic algorithms and well-known classification methods are used together to increase accuracy and develop more reliable systems. One of the popular heuristic algorithms that are commonly used in this area is the Genetic Algorithm. It is mentioned that the combination of the Genetic Algorithm and the K-means Algorithm performs better when it is compared with the K-means++ Algorithm (Sukumar, Pranav, Neetish, & Narayanan, 2018).

The Firefly Algorithm has been used for feature selection in several studies in the literature. Return-based Binary Firefly Algorithm (Rc-BBFA) was one of the methods that were implemented for feature selection by using FFA (Zhang, Song, & Gong, 2017). In (Li, Kamlesh, Lim, & Neoh, 2017), the Firefly optimization was implemented for feature selection with the combination of classification and regression models. Moreover, FFA obtained some successful results in fingerprint feature extraction (Tariq, Al-Ta'i, & Abdulhameed, 2013).

FFA was also used to detect intrusions on the networks in many studies in the literature. An FFA based feature selection method was developed to protect the network from the attacks by using KDD CUP 99 dataset (Selvakumar B, 2018). The Firefly Algorithm is generally used in optimization problems. There are several striking studies that use FFA in recent years for solving optimization problems. One of them was about EEG signals that were needed to recover true brain signals from noises (Majdouli, Bougrine, Rbouh, & Imrani, 2017). A hybrid approach was implemented by using FFA and PSO to solve optimization problems (Aydilek, 2018). Another firefly-based hybrid method was developed for churn prediction (Ahmed & D., 2017). A Firefly-based Algorithm was developed in three phases which were the feature selection phase, the model construction phase and the prediction phase in the study (Mashhour, Houby, & Khaled Tawfik Wassif, 2018). The model was tested with 7 datasets and compared with Ant-miner Algorithm. The most successful approach was found to be the FFA based on distance method for all datasets where accuracy rates vary between 83% and 90%.

The Basic Firefly Algorithm is simple but needed to be improved because its accuracy is not enough and has the local optimum problem. To solve these problems, the Firefly Algorithm based on the gender difference algorithm (GDFA) was implemented (Wang & Song, 2019). In this study, different equations were used to determine the movement of fireflies for two subgroups of their genders. As a result, the proposed method presented that the GDFA's performance was higher than other Firefly-based Algorithms for 30-dimensional problems.

In another study, FFA was modified with neighborhood attraction to reduce computational time complexity (Hui, et al., 2017). The fireflies were attracted by only neighbors, not the entire population. The proposed method is tested with well-known benchmark functions and was compared with traditional FFA, variable step size FFA(VSFFA), wise step strategy FFA (WSSFA), memetic FFA (MFA), and FFA with chaos (CFA). The algorithms were ranked by fitness values, and neighborhood attraction FFA (NaFA) is found to be the most successful one.

Artificial ants and fireflies were used in a color quantization study (Pérez-Delgado & María-Luisa, 2018). The Artificial Ant Algorithm was supported by the Firefly Algorithm to find the best parameters for image quantization. Ant-Tree for color quantization (ATCQ) algorithm was

combined with the Firefly Algorithm in this study. The result of the experiments showed that the proposed algorithm is better than ATCQ and FA independently.

Another experiment was achieved to solve the flexible job-shop scheduling problem (FJSP) with the Discrete Firefly Algorithm and multi-objective Genetic Algorithm (Lunardi & Voos, 2018). It was discussed that the proposed FFA was effective only for small instances of the problem. According to the experiments, the proposed FFA was faster than the proposed GA. Another comparative study was conducted with the algorithms PSO, Artificial Bee Colonies (ABC), Cuckoo Search (CS) and FFA on graph coloring problem (Aranha, Junior, & Kanoh, 2018).

Firefly Algorithm was modified and used to find opinion leaders in social networks (Jain & Katarya, 2019). The fireflies in the algorithm presented people in the social network. The attractiveness of the firefly was represented as user prominence and the distance between fireflies was calculated as centrality in the algorithm. The proposed algorithm showed the best result with 94% accuracy, 96% f1-score measurements in the real data set.

In our proposed approach, we modify the traditional FFA (TFFA) with the K-nearest neighborhood (KNN) classification algorithm, which is called the proposed Firefly Algorithm as PFFA. PFFA is developed by using the collaboration of classification (KNN) and probability theory (eq 4) to obtain the best sub-feature set from original features. Features that are common in the feature sets were created by the PFFA (Sknn) and the other features that are obtained by the TFFA (Sr) are used to obtain a new feature set (Scommon). The three feature sets and original features are compared by using the K-NN classification method. The success of feature sets in classifications is measured by using four different datasets that are related to different types of attacks in computer systems. In addition to this, the Artificial Immune System Algorithm is applied to the dataset of the user to root attacks due to the insufficient number of data before FFA is implemented. The sufficient number of data is one of the most crucial topics for the systems that are developed by using machine learning methods.

The remainder of this paper is organized as follows. In Section 2, the original FFA and the proposed FFA methods are explained. Detailed information about the datasets and analysis of the four experiments are given in Section 3. Finally, a summarization of the study and the obtained results are discussed in Section 4.

2. METHODS

2.1. Firefly Algorithm

Firefly Algorithm is one of the new optimization techniques that is developed by analyzing the flashing behavior of fireflies. It is a metaheuristic and nature-inspired algorithm, which was proposed by Xing-She Yang in 2008 (Eren, B.Küçükdemiral, & Üstoğlu, 2017). The algorithm relies on three important characteristics of fireflies. One of these characteristics is the fireflies' gender which is known to be unisex. Thus, each firefly can be attracted by any other fireflies. Secondly, the attractiveness and distance between the fireflies have an inverse ratio. If the distance between two fireflies is small, the attractiveness of the fireflies will be high. When a firefly is close to other fireflies, it looks brighter than normal because of this attractiveness. Moreover, less bright fireflies move towards more brighter ones. The third point is that the brightness of a firefly is determined by the objective function. The objective function can be different according to each problem.

$$B(r) = B_0 e^{-\gamma r^2} \quad (1)$$

The attractiveness of fireflies can be calculated by using equation (1). $B(r)$ is the attractiveness of a firefly at distance r (Marie-Sainte & Alalyani, 2018). B_0 is the attractiveness when the distance (r) is zero. γ is the fixed light absorption coefficient and generally taken as 1.

$$x_i = x_i + B(r) * (x_k - x_i) + \alpha(rand - \frac{1}{2}) \quad (2)$$

By using the attractiveness formula, the new position of less bright firefly to move to the brighter one is calculated as in equation (2). α and $rand$ are the random numbers in the equation that are uniformly generated between [0,1]. x_i and x_k are i^{th} and k^{th} fireflies in the population. Distance between two fireflies can be calculated with the Euclidean distance formula as given in equation (3).

$$r(i,k) = |\overrightarrow{x_i} - \overrightarrow{x_k}| = \sqrt{\sum_{j=1}^{d}(x_{ij} - x_{kj})} \quad (3)$$

In equation (3), d is the dimension of fireflies, x_{ij} is the jth dimension of i^{th} firefly and x_{kj} is the j^{th} dimension of k^{th} firefly.

2.2. Proposed Firefly Algorithm Based Feature Selection

In this study, our objective function is the accuracy of a classifier by using the dataset with selected feature sets. The population is generated by selecting different feature subsets whose size is smaller than the original dataset. Each dataset with different features is a firefly which means a candidate solution. Our aim is to find the best feature subset for the dataset to perform classification successfully. Distances between fireflies and new positions of subsets are calculated by using equations (2) and (3). The traditional Firefly Algorithm (TFFA) is given in Figure 1. N is the number of features to be selected in the algorithm. We choose 20 as N for this study which means the dataset length will be reduced to 20 from 41.

```
1   Randomly choose N-feature subset and generate M firefiles(ff)
2   Compute fitness function Q for each firefly
3   for i=1 to M-1 do
4       for j=1 to M
5           if Q(i) < Q(j)
6               move ff_i to ff_j using eq 2 : new position ff_i
7               Compute fitness function Q for new ff_i
8           end
9       end
10  end
```

Figure 1:
Traditional Firefly Algorithm (TFFA)

In our proposed FFA (PFFA), we use the K-nearest neighborhood classifier as the fitness function. The fitness function was used the same as the attractiveness function in the literature. In addition to this, we apply the rule given in equation (4) (Marie-Sainte & Alalyani, 2018).

If $P(x_i)$> rand
    Select the feature
Else
    Do not select the feature

where, $$P(x_i) = \frac{1}{1+e^{x_i}} \quad (4)$$

According to the results of the classifier, fitness values are ranked, and the best feature set (Sknn) for K-NN is obtained. In addition to this, the second feature set is taken from the rule in equation (4) as Sr. In equation 4, x_i represents the current value of each feature and $P(x_i)$ is the probability of x_i taking 1. Moreover, we assume that if a firefly is not affected by any other firefly, it should continue its random fly as stated in (Saim, 2017) using equation (5) where $rand$ is random number generator distributed in [0 1] and α is another randomization parameter between [0,1]. The pseudocode of our proposed FFA algorithm is given in Figure 2 in detail.

$$x_i = x_i + \alpha(rand - \frac{1}{2}) \quad (5)$$

In the PFFA, the accuracy result of the K-NN algorithm is used as a fitness value. Accuracies of fireflies are compared, and the firefly, %, however, which has lower accuracy moves towards the firefly that has higher accuracy. Thus, lower accuracy firefly is updated according to equation (2). After each update, the fitness value is recalculated. The feature set that is selected by K-NN (Sknn) and selected by equation (4) (Sr) is analyzed. Sr is generated by choosing the feature that is selected more than the threshold value which is determined as 3 for this study. The threshold is decided after several trials. As a result, the features that occur in both feature sets are selected as the most effective features for classification or dimension reduction. In addition to two subsets Sknn and Sr, Scommon is created by choosing the common features in both Sknn and Sr.

Figure 2:

```
Initialize trainset and testset
Randomly choose N-feature subset and generate M firefiles(ff) from trainset
Classify testset using each firefly and calculate accuracy : fitness value Q
for i=1 to M-1 do
    for j=1 to M
        if Q(i) < Q(j)
            calculate attractiveness of ff_i using eq(1)
            move ff_i to ff_j using eq 2 : new position ff_i using eq (2)
            classify testset using new ff_i:update Q for new ff_i
            Sr <- calculate P using eq (4)
    else
        new position for ff_i using eq (5): random fly
        end
    end
end
rank the fitness values for each firefly
take the best firefly's feature set : Sknn
list the features that are selected more than threshold : Sr
find the features common in both Sknn and Sr
```

Proposed Firefly Algorithm (PFFA)

3. EXPERIMENTS AND RESULTS

3.1 Dataset

The dataset is taken from KDD CUP 99 (JR, 1993) that consists of normal flow and attacks to the network. There are 22 different types of attacks in the dataset. Each data is recorded with 41 features. 22 different types of attacks are divided into four categories as the denial of service, remote to local, the user to root and probe (Selvakumar B, 2018). In this study, four attack types in different subsections are analyzed because each attack has different characteristics and different sizes of data. In the experiments, MATLAB R2013b software on Intel Core i7-6700HQ CPU @2.60 GHZ with a 16GB RAM computer is used for the implementation of the FFA algorithm.

The first three experiments have the same steps as shown in Figure 3. Proposed Firefly and traditional Firefly Algorithms are implemented to obtain feature subsets Sr and Sknn. By using Sknn and Sr, feature set Scommon is generated as mentioned in the previous section. K-NN classifier is used as the fitness function and accuracy result is the fitness value for Firefly Algorithms. To be able to make a comparison, K-NN is implemented on the original data with 41 features. In addition to the feature set Scommon, other feature sets Sr and Sknn are separately used to classify data. As a result, we get four accuracy and time measurement results for original data, and three selected feature sets that are Sknn, Sr and Scommon.

In the fourth experiment, we apply an additional algorithm called Artificial Immune System (AIS) to generate artificial data due to the unsatisfying number of data.

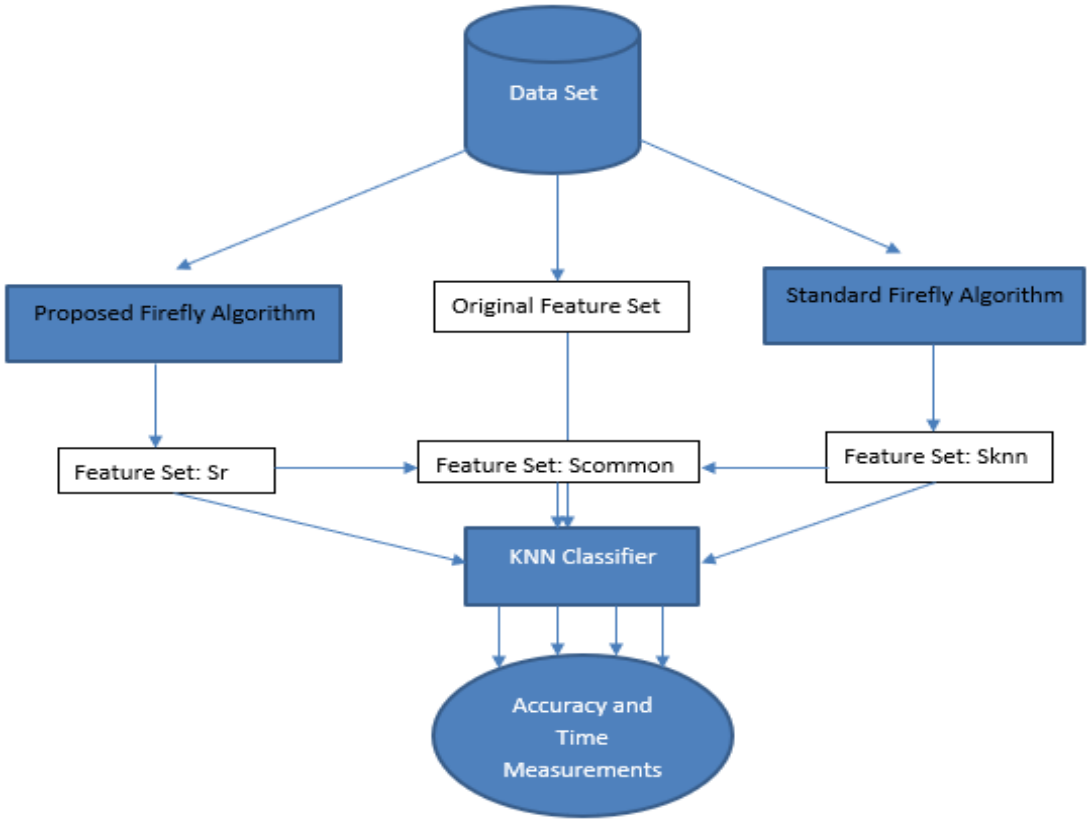

Figure 3:
General Process Diagram for Experiments

3.1.1. Experiment-1: Remote to Local (R2L) Attack

The first experiment is performed for the R2L attack type. We create our training and testing datasets from KDD CUP 99 dataset by taking 6684 normal data and 1114 data that are tagged as R2L attack. We choose randomly 2339 data for the testing set and 5459 data for the training set. %70 of data is divided as the training set and 30% of data is divided for the testing set. 768 of 5459 training data is tagged as an attack. 346 of 2339 testing data is tagged as an attack.

We complete our experiment in three steps. First of all, the dataset is classified with all 41 features by using K-NN. Then, we implement traditional FFA (TFFA). TFFA finds the most successful feature set that consists of 20 features. We classify our dataset with selected 20 features by TFFA. In the last step, our proposed FFA (PFFA) is implemented. As a result of PFFA, we obtain 18 features that are common in two approaches in PFFA. The feature set Sknn that is obtained by TFFA and Sr that is obtained by the second technique in PFFA are given in Table 1. The final feature set Scommon is also given in Table 1.

Table 1. Feature Sets selected by TFFA(Sknn) and PFFA (Sr and Scommon) for R2L Attack Type

	Feature Indexes
Sknn	6-**17**-**32**-41-**22**-**39**-**23**-31-30-3-16-**2**-7-14-29-5-**12**-**27**-33-**25**
Sr	**2**-9-11-**12**-15-**17**-19-**22**-**23**-24-**25**-**27**-**32**-34-37-38-**39**-40
Scommon	2-12-17-22-23-25-27-32-39

We measure the accuracy results of TFFA and PFFA. K-NN Algorithm classifies the testing data with 99.79% accuracy using 41 features. TFFA with 20 features that are given as Sknn in Table 1 classifies the testing data with 99.83% accuracy. Our proposed method PFFA reduces the dimension of data from 41x7798 to 9x7798 in the total training set and testing set. In addition to this, the correct prediction rate is 90.51% with 9 features in the feature set Scommon. We also classify the data by using the feature set Sr with 18 features whose accuracy rate 93.67% as seen in Figure 4.

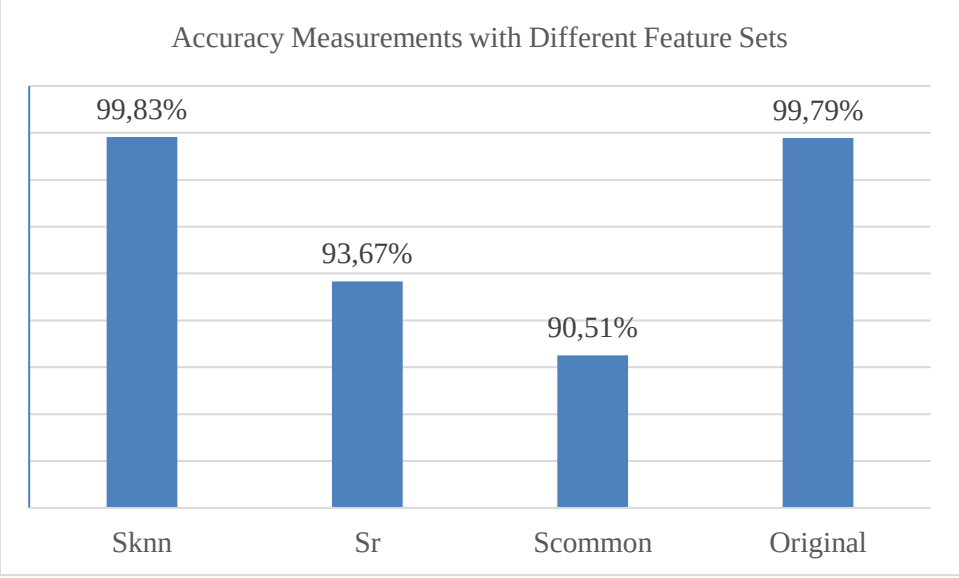

Figure 4:
Accuracy measurements of different feature sets for R2L attack type

Changes in the number of features in different feature sets that are obtained by the algorithm are given in Figure 5 to compare the size of reduced datasets with the original. The usage of memory is dramatically increased with the FFA by comparing the accuracy results.

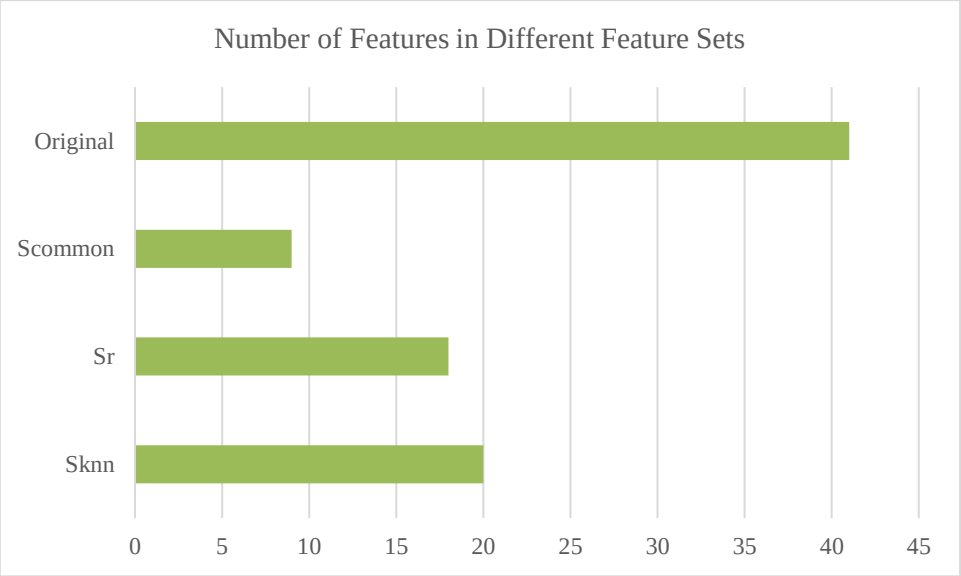

Figure 5:
Number of Features in Different Feature Sets

Processing time that is required for the classification is as important as the memory requirements. The time is measured in seconds as stated in Figure 6.

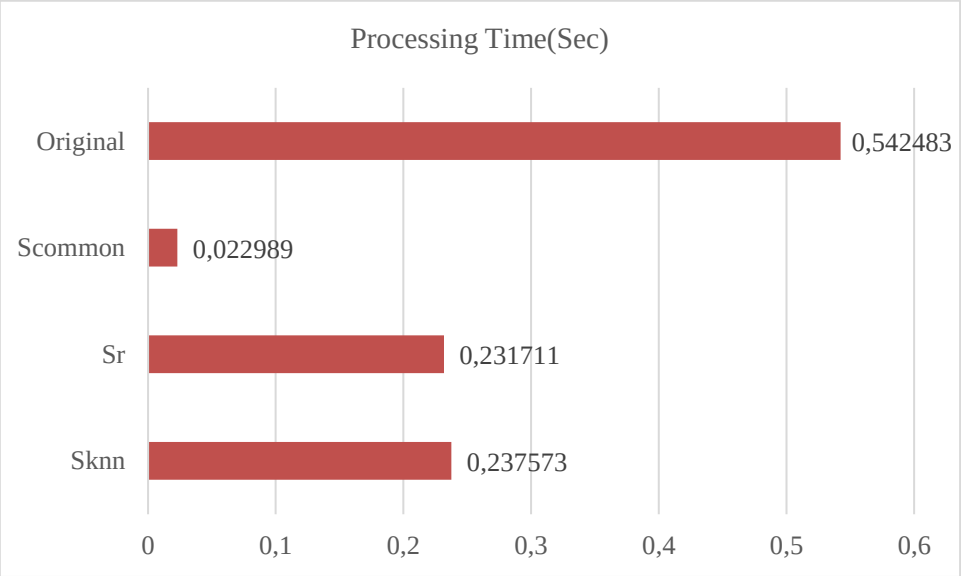

Figure 6:
Processing Time to classify data using feature sets

3.1.2. Experiment-2: Probe Attack

We choose random 8624 data for the testing data set and 20125 data for the training set. Generally, 30% of data is taken as the testing set and %70 of data is taken as the training set. 1224 data in the testing set and 2883 data in the training set are tagged as a probing attack. 7400 data from the testing set and 17242 data from the training set are tagged as normal in total. Similar to experiment-1, we classify our data first originally with K-NN, second with TFFA and lastly PFFA using K-NN. TFFA reduces the number of features from 41 to 20 called feature set Sknn. PFFA reduces the number of features to 5 called Scommon. The second method in PFFA has also produced a feature set called Sr that consists of 11 features. The index of selected features is given in Table 2.

Table 2. Feature Sets selected by TFFA(Sknn) and PFFA (Sr and Scommon) for Probe Attack Type

	Feature Indexes
Sknn	**27**-25-17-4-31-5-15-35-19-**10**-8-38-**29**-23-30-**36**-**2**-16-13-14
Sr	1-**2**-7-**10**-20-24-26-**27**-**29**-**36**-37
Scommon	2-10-27-29-36

K-NN algorithm classifies the original testing data with 41 features with 99.83% accuracy. TFFA generates a feature set Sknn with 20 features and its correct classification rate is 99.95%. PFFA reaches very few numbers of features that is 5 in the feature set Scommon. Using PFFA, the size of data is reduced by 88%. However, the correct classification rate is decreased by only 2.03%. The accuracy of PFFA using Scommon is 97.92%. In addition to this, the feature set that is obtained in PFFA; Sr is used for classification, and it classifies 98.33% of data correctly as shown in Figure 7.

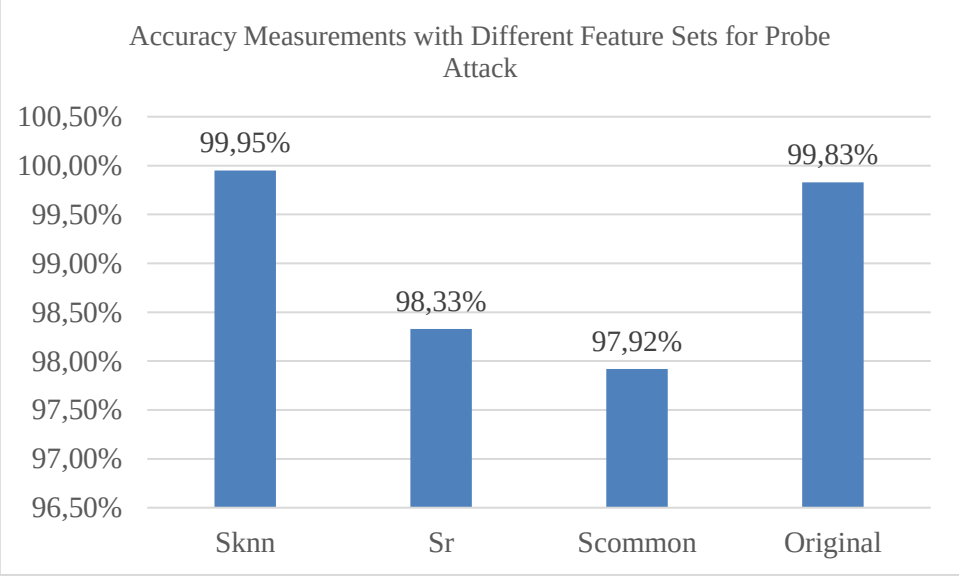

Figure 7:
Accuracy Measurements with Different Feature Sets for Probe Attack

In addition to accuracy measurements, changes in the number of features are shown in Figure 8.

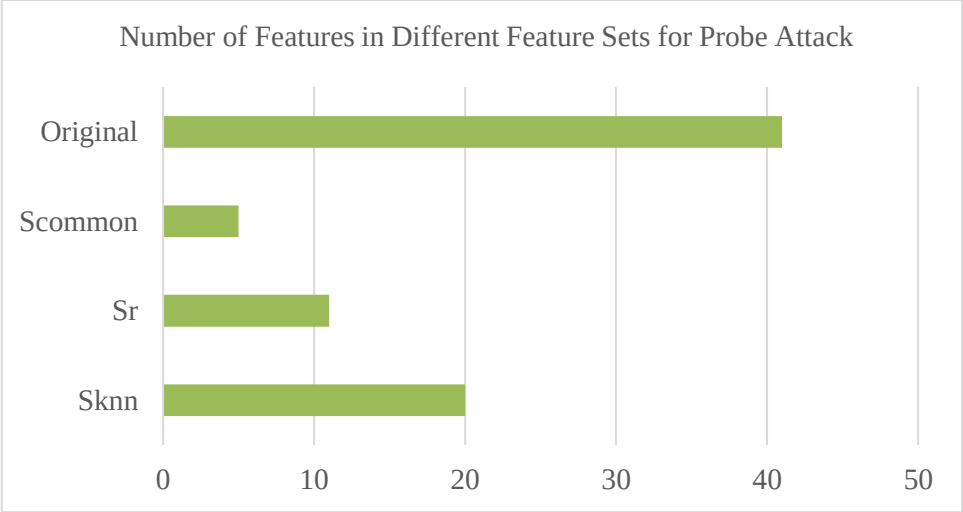

Figure 8:
Number of Features in Different Feature Sets for Probe Attack

When we measure the time that is necessary for classification with the produced feature sets by FFA, the following chart is obtained as given in Figure 9. In addition to memory constraints, time is another critical constraint for processing big data.

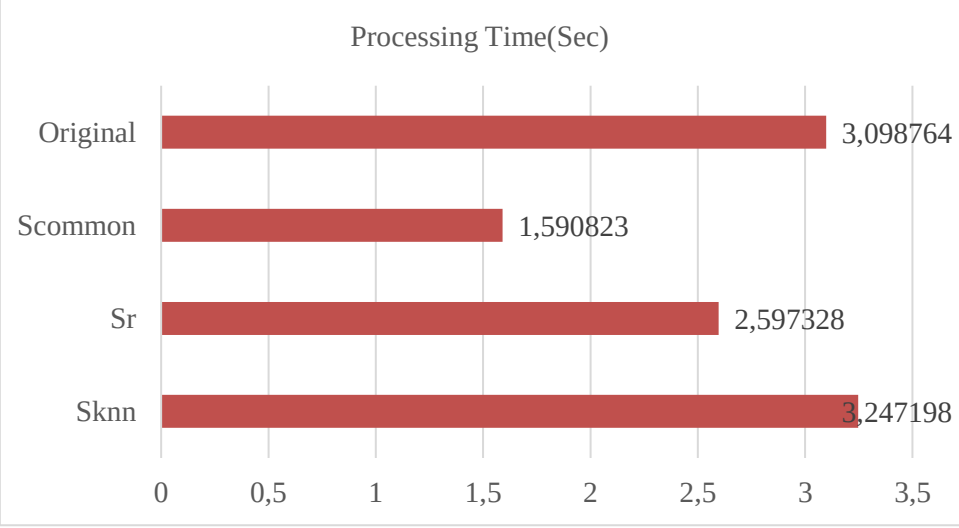

Figure 9:
Processing Time of Feature Sets for Probe Attacks

3.1.3. Experiment-3: DOS Attack

For the DOS attack experiment, we choose 9000 data for the testing set and 21000 data for the training set like the previous experiments. 50% of data is tagged as normal and 50% of data is tagged as DOS attack. In addition to this, 10476 of training data tagged as DOS attack and 4524 of the testing set is tagged as a DOS attack. K-NN algorithm classifies the original data with a 99.70 % accuracy rate. The feature sets that are obtained by TFFA and PFFA are given in Table 3.

Table 3. Feature Sets selected by TFFA(Sknn) and PFFA (Sr and Scommon) for DOS Attack Type

	Feature Indexes
Sknn	**4**-**26**-**5**-39-**3**-18-**12**-38-**13**-9-**21**-35-10-6-14-**17**-33-41-34-25
Sr	1-**3**-**4**-**5**-8-11-**12**-**13**-15-16-**17**-19-20-**21**-22-23-24-**26**-29-30-32-36
Scommon	3-4-5-12-13-17-21-26

TFFA with 20 features and PFFA with 8 features classify the testing data with 99.88% and 99.93% accuracy rates as shown in Figure 10. The other feature set Sr that is generated by PFFA is used and give 98.22% accuracy.

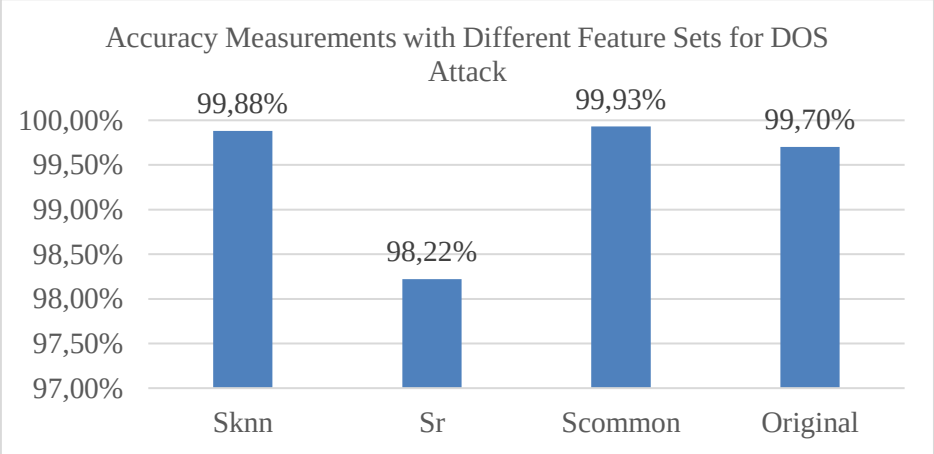

Figure 10:
Processing Time of Feature Sets for DOS Attacks

The changes in the number of features after TFFA and PFFA and the processing time of the algorithms are plotted in Figure 11 and Figure 12.

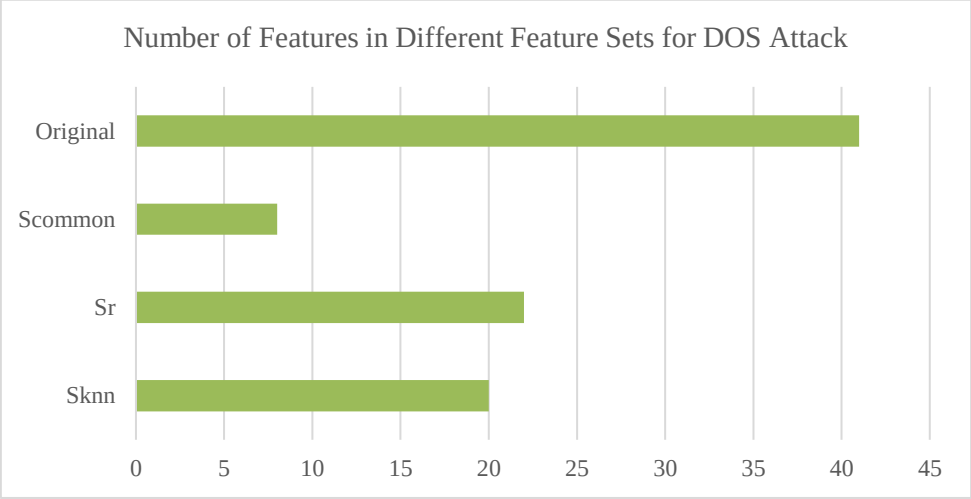

Figure 11:
Number of Features in Different Feature Sets for DOS Attack

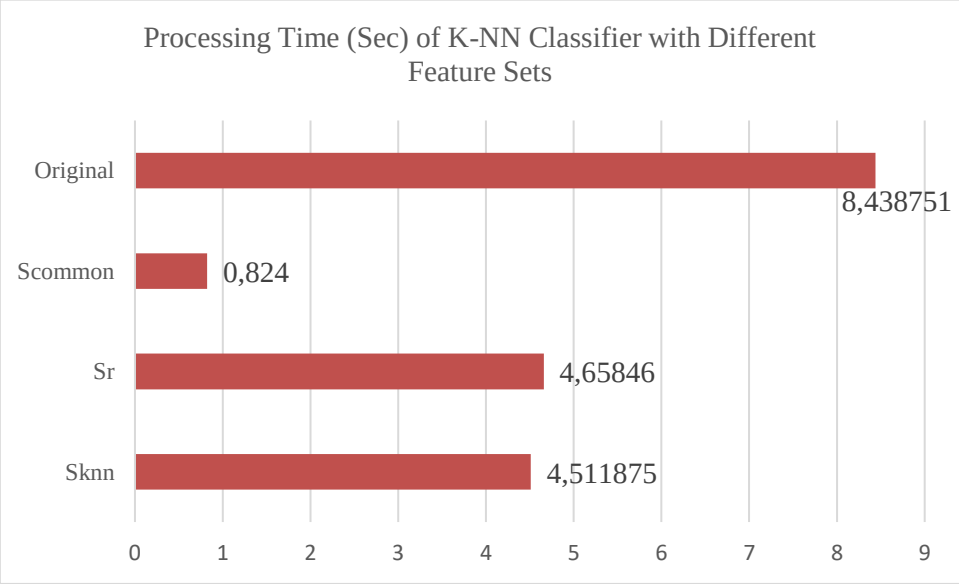

Figure 12:
Processing Time of K-NN Classifier with different feature sets

3.1.4. Experiment-4: User to Root(U2R) Attack and Artificial Immune System (AIS)

There are only 52 data that are tagged as U2R in the dataset. The number of U2R attacks is not enough to complete our experiments. Thus, we need to generate artificial data for U2R attack types. We use the Artificial Immune System Algorithm to generate artificial data. It is a human-inspired algorithm that uses the general properties of the natural immune system. The immune system has some important properties that are used to implement the algorithm. These properties are uniqueness, distributed detection, and self-regulation, approximate detection and pattern matching, diversification, anomaly detection, self-protection and learning, and memorization. By

using these characteristics properties, the immune system algorithms can be improved. There are several types of immune-based algorithms in the literature as negative selection algorithm, clonal selection algorithm, artificial immune network algorithm, danger theory algorithm (Fernandes, Freire, Fazendeiro, & Inácio, 2017). We prefer to use the clonal selection algorithm in this study. Our goal is to produce a population using 52 data tagged as U2R attack. The immune system recognized the antigens that enter the body and produce new cells to protect the body. The system can remember the same antigens or similar antigens even after many years later and protect itself by producing and generating clones faster than the first time. Using this behavior of the system, the clonal selection algorithm will produce new data and check whether the new data is suitable for the system with affinity measurements in the algorithm. A new clone can be generated by using two existing data (Er, Yumusak, & Temurtas, 2012). We generate new clones by taking two data's average from U2R. we need a threshold to calculate the affinity of new clones. Average of distances between each couple from U2R data for each feature identify the thresholds E for the features. The affinity of a clone is increased one by one for each feature if the feature value greater than the threshold. We have 41 features, and if 20 of them pass the selection, the clone is added to the population. In the last step, the population is classified using K-NN and the final population is created if the clone's class is from U2R. In general, we implement the basic steps of AIS that are given in Figure 13.

```
Step 1: Generate clones
Step 2: Determine the threshold E for each feature
Step 3: Calculate affinity for clones using E
Step 4: Classify the data
Step 5: If affinity and class is sufficient, add clone to population
```

Figure 13:
Pseudocode of AIS

With the AIS algorithm, we produce 922 artificial data addition to original data in the dataset that consist of 52 data tagged as U2R attack and 974 data tagged as normal. We choose random 584 data for the testing set and 1364 data for the training set. TFFA selects 20 features (Sknn) where PFFA selects 8 features (Scommon) as shown in Table 4.

Table 4. Feature Sets selected by TFFA(Sknn) and PFFA (Sr and Scommon) for U2R Attack Type

	Feature Indexes
Sknn	31-41-35-1-12-**13**-40-10-19-38-18-**30**-**17**-**2**-**7**-**21**-**5**-8-29-**15**
Sr	**2**-**5**-**7**-**13**-**15**-16-**17**-**21**-**30**-37-39
Scommon	2-5-7-13-15-17-21-30

Accuracy results show that TFFA and PFFA increase the correct prediction rate. TFFA classifies the data with a 99.83% accuracy rate by selecting features to feature set Sknn. Other feature sets that are produced by PFFA, Sr and Scommon also give better results than the original feature set. The accuracy rate of feature sets is given in Figure 14.

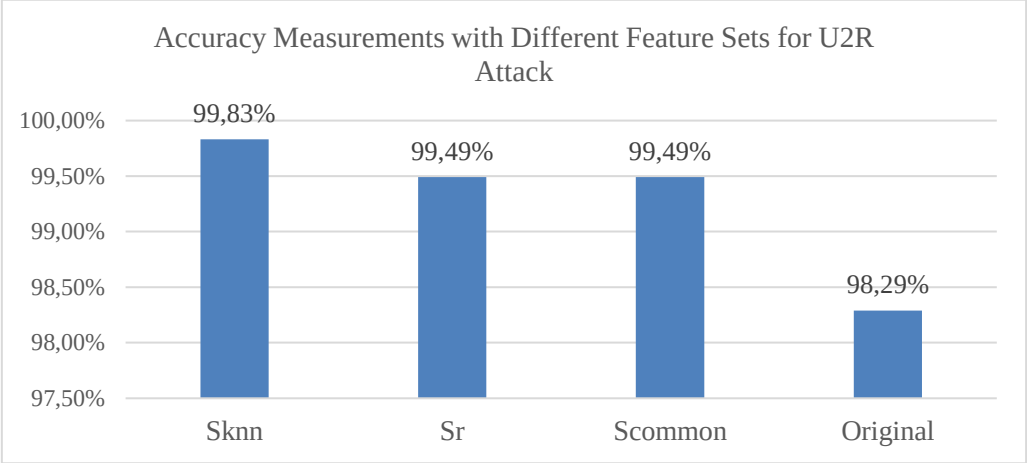

Figure 14:
Accuracy Measurements with Different Feature Sets for U2R Attack

The processing time of K-NN differs in different feature sets as similar to previous experiments. There is a correlation between the number of features in the dataset and the processing time of K-NN. The increase in time and number of features can be seen in Figure 15 and Figure 16.

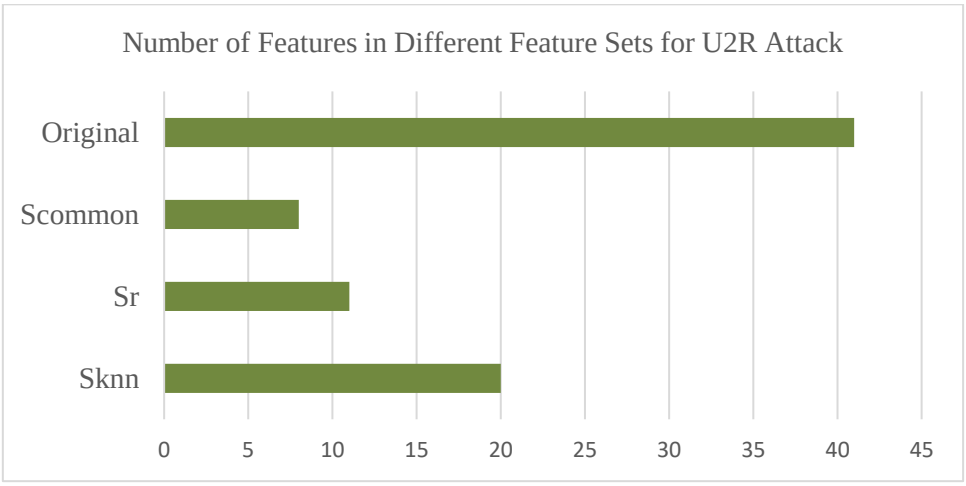

Figure 15:
Number of Features in Different Feature Sets for U2R Attack

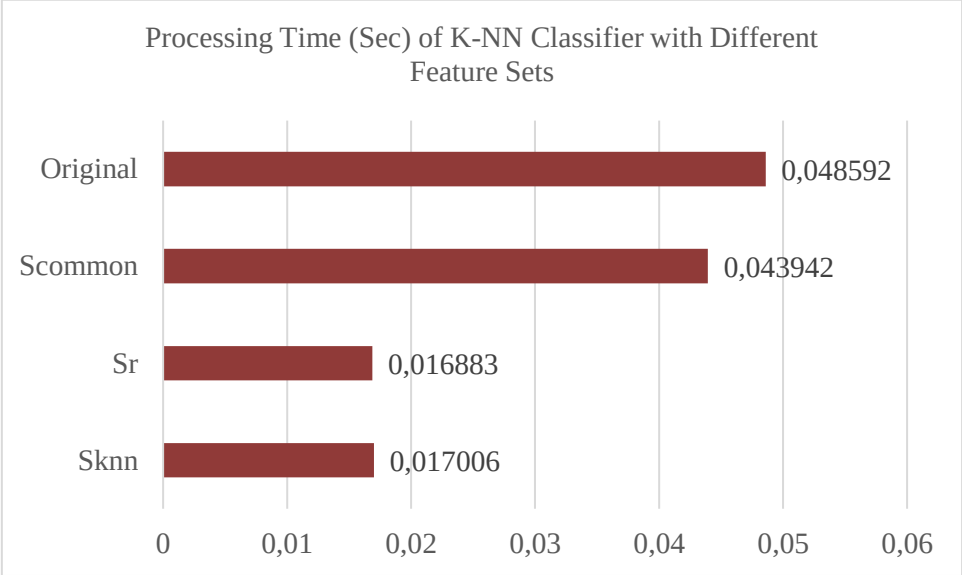

Figure 16:
Processing Time of K-NN Classifier with different feature sets

3.2. Results

Experiments show that dimension reduction with feature selection using TFFA increases the accuracy rate. In experiment 2, probe attack type classification shows the highest accuracy rate as 99.95% from 99.83% comparing to the original dataset and Sknn. In other experiments, TFFA gives better results than K-NN classification with the original dataset. Accuracy results of experiment-1 for R2L attack and experiment-3 for DOS attack are also increased from 99.79% to 99.83% and from 99.70% to 99.88% after TFFA is implemented. In addition to the accuracy results, the dimension of the data is decreased to a 50% ratio.

Our proposed method PFFA generates two different feature sets that are Sr and Scommon. The accuracy results with the feature sets are not greater than the original dataset but a little different in the first two experiments. The results are decreased by 9% for experiment-1, 2% for experiment-2 when Scommon is used. In experiments 3 and 4, we analyze that the accuracy rate is increased by 2% and 5% with Scommon. The results with the feature set Sr are close to Scommon. In Table 5, we compared our proposed data with the most similar studies. The proposed method shows higher result than (B & K, 2019) that same dataset used for DOS, probe and U2R attack types. For other studies in the table except (Tariq, Al-Ta'i, & Abdulhameed, 2013), accuracy results are not better than proposed method.

Sr and Scommon feature sets have a slight difference when they are compared with the original dataset but the difference in the memory that the dataset is required, and the processing time have very big changes.

Processing times decreased from 0.048 sec to 0.043 sec for U2R attack, from 8.43 sec to 0.824 sec for DOS attack, from 3.098 sec to 1.59 sec for Probe attack and from 0.52sec to 0.0229 sec for R2L attack types with Scommon feature set. In addition to this, the dimension of the data set is decreased by 80% for U2R and DOS, %88 for Probe and 78% for R2L attack type with common feature sets.

Table 5. Comparison of PFFA with Other Methods in the Literature

	Method	**Dataset**	**Accuracy**
Our Proposed Method	TFFA	KDD CUP 99-Original Data Probe Attack R2L DOS U2R	 %99.83 %99.79 %99.70 %98.29
	PFFA	KDD CUP 99-Scommon Probe Attack R2L DOS U2R	 %97.92 %90.51 %99.93 %99.49
(Selvakumar B, 2018)	FFA with Bayesian Network Algorithm	KDD CUP 99 Probe Attack R2L DOS U2R	 %93.42 %97.83 %99.95 %68.97
(Anbu & Mala, 2017)	SVM with FFA KNN with FFA NB with FFA	PROMISE Software Dataset	%91 %88 %87
(Tariq, Al-Ta'i, & Abdulhameed, 2013)	Features Extraction of Fingerprints using Firefly Algorithm	Fingerprint Dataset	%100
(Mashhour, Houby, & Khaled Tawfik Wassif, 2018)	A Novel Classifier based on Firefly Algorithm	Lung Dataset Hepatitis Dataset Dermatology Dataset Prostate Dataset Leukemia1 Dataset DLBCL Dataset SRBCT Dataset	%80 %82 %90 %90 %83 %90 %90

In this study, we also implement the AIS algorithm to generate artificial data with a clonal selection mechanism due to the insufficient number of data in experiment 4. As we obtained high accuracy results with the dataset, including artificial data, the proposed method is found to be successful for the unbalanced datasets.

As a result, PFFA can be preferable when we analyze the accuracy rates of the feature sets that are closed to original feature sets and memory utilization.

4. CONCLUSION

In this paper, the traditional Firefly Algorithm was modified to obtain a subset from the features that give the best classification accuracy. We obtained new feature sets from the traditional Firefly Algorithm and the modified Firefly Algorithm. We also created a feature set that consisted of features from the modified and original Firefly Algorithms in common. Classification accuracies for the feature sets and the original feature set were calculated and compared in 4 datasets of intrusion detection. One of the datasets that were about user to

root (U2R) attacks had only 52 data tagged as attacked and 974 data tagged as normal. When we compared it with other datasets, the number of attacked and normal data in this dataset was recognized to be unbalanced so as to apply the FFA. Thus, before the feature selection step, the Artificial Immune System (AIS) Algorithm was applied to generate artificial data to support this dataset. By using AIS, 922 additional data was generated.

When the results were compared, it was determined that the feature set obtained from TFFA (Sknn) gave better accuracy rates than the original feature set. TFFA provided to decrease the number of features from 40 to 20. Although the selected feature sets obtained from PFFA (Scommon) gave lower classification accuracies than Sknn for all datasets, there was a slight difference between them. On the other hand, we could decrease the dimension of data with PFFA more than 65% on average. For that reason, PFFA was found to be successful in both accuracy results and memory saving. In addition to this, according to time measurements to classify data after PFFA was implemented, the method could also be used to save time.

Although PFFA was found to be successful, it should be enhanced to reach the success of TFFA for classification accuracy. For this purpose, equations (4) and (5) could be improved, and some other different classification methods could be used to calculate the attractiveness of fireflies and to test the overall system in the future. Moreover, equation (3) is used to calculate the distance but a big difference between one feature may decrease the effect of another feature difference. Therefore, there is a gap in the distance formula for large dimensional data. Equation (3) may be modified to make more strength calculation. In addition to improved equations, the firefly algorithm can be modified and used with other methods, including standard machine learning algorithms and heuristic approaches. This study and the studies we mention applied firefly algorithm with continues variables due to algorithm's computing architecture, but the algorithm can be modified for the binary datasets in the future to be able to expand the implementation area of the algorithm.

ACKNOWLEDGMENTS

This work was supported by Council of Higher Education of Turkey. Project number: MEV-2018-863

REFERENCES

1. Aranha C., Junior J. P., & Kanoh, H. (2018). Comparative study on discrete SI approaches to the graph coloring problem, *Genetic and Evolutionary Computation Conference,* Kyoto, Japan, 15-19. doi:10.1145/3205651.3205664

2. Anbu M., & Mala G. S. (2019). Feature selection using firefly algorithm in software defect prediction. *Cluster Computing*, 22, 10925–10934. doi:10.1007/s10586-017-1235-3

3. Aydilek İ. B. (2018). A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems, *Applied Soft Computing*, 66, 232-249. doi:10.1016/j.asoc.2018.02.025

4. B Selvakumar., & K Muneeswaran. (2018). Firefly algorithm based feature selection for network intrusion detection. *Computers & Security*, 81, 148-155. doi:10.1016/j.cose.2018.11.005

5. Er O., Yumusak N., & Temurtas, F. (2012). Diagnosis of chest diseases using artificial immune system, *Expert Systems with Applications*, 39(2), 1862-1868. doi:10.1016/j.eswa.2011.08.064

6. Eren Y., Küçükdemiral İ., & Üstoğlu İ. (2017). Introduction to Optimization, *In Optimization in Renewable Energy Systems*, 27-74, Elsevier Butterworth-Heinemann. ISBN:9780081010419, 0081010419

7. Fernandes, D. A., Freire, M. M., Fazendeiro, P., & Inácio, P. R. (2017). Applications of artificial immune systems to computer security: A survey, *Journal of Information Security and Applications*, 35, 138-159. doi:10.1016/j.jisa.2017.06.007

8. Hui W., Wenjun W., Xinyu Z., Hui S., Jia Z., Xiang Y., & Zhihua C. (2017). Firefly algorithm with neighborhood attraction. *Information Sciences*, 382-383, 374-387. doi:10.1016/j.ins.2016.12.024

9. Jain L., & Katarya R. (2019). Discover opinion leader in online social network using firefly algorithm, *Expert Systems With Applications*, 122, 1-15. doi: 10.1016/j.eswa.2018.12.043

10. JR, Q. (1993). *C4.5: Programs for Machine Learning.* Erişim Adresi: https://github.com/defcom17/NSL _ KDD (Erişim Tarihi: 12.11.2018)

11. Lee W., & Stolfo S. J. (1998). Data Mining Approaches for Intrusion Detection, *Proceedings of the 7th USENIX Security Symposium*, San Antonio, Texas: Usenix, 1-15. doi:10.5555/1267549.1267555

12. Li Z., Kamlesh M., Lim C. P., & Neoh S. C. (2017). Feature selection using firefly optimization for classification and regression models, *Decision Support Systems*, 106, 64-85. doi: 10.1016/j.dss.2017.12.001

13. Lunardi W. T., & Voos H. (2018). Comparative study of genetic and discrete firefly algorithm for combinatorial optimization, *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, Pau, France, 300-308. doi:10.1145/3167132.3167160

14. Majdouli M. A., Bougrine, S., Rbouh, I., & Imrani, A. A. (2017). A Comparative Study of the EEG Signals Big Optimization problem using evolutionary, swarm and memetic computation algorithms, *The Genetic and Evolutionary Computation Conference*, Berlin, Germany, 1357-1364. doi:10.1145/3067695.3082489

15. Marie-Sainte, S. L., & Alalyani, N. (2020). Firefly Algorithm based Feature Selection for Arabic Text Classification, *Journal of King Saud University- Computer and Information Sciences*, 32(3), 320-328. doi:10.1016/j.jksuci.2018.06.004

16. Mashhour E. M., Houby E. M., & Khaled Tawfik Wassif, A. I. (2018). A Novel Classifier based on Firefly Algorithm, *Journal of King Saud University – Computer and Information Sciences*, In Press, Corrected Proof. doi:10.1016/j.jksuci.2018.11.009

17. Pérez-Delgado, & María-Luisa. (2018). Artificial ants and fireflies can perform colour quantisation, *Applied Soft Computing Journal*, 73, 153-177. doi:10.1016/j.asoc.2018.08.018

18. Saim B. (2017). Retrieved from Bilal Saim Website: https://bilalsaim.com/ates-bocegi-algoritmasi-fafirefly-algorithm-h1635 (Erişim Tarihi: 06.11.2019)

19. Shang-fu, G., & Zhao, C.-I. (2012). Intrusion Detection System Based on Classification, *2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment*, Beijing, China, 78-83. doi:10.1109/ICADE.2012.6330103

20. Sukumar J. V., Pranav I., Neetish, M., & Narayanan, J. (2018). Network Intrusion Detection Using Improved Genetic k-means Algorithm, *International Conference on Advances in Computing, Communications and Informatics*, Bangalore, India, 2441-2446. doi:10.1109/ICACCI.2018.8554710

21. Tariq, Z., Al-Ta'i, M., & Abdulhameed, O. Y. (2013). Features extraction of fingerprints using firefly algorithm, *Proceedings of the 6th International Conference on Security of Information and Networks*, Aksaray, Turkey, 392-395. doi:10.1145/2523514.2527014

22. Wang, C.-F., & Song, W.-X. (2019). A novel firefly algorithm based on gender difference and its convergence, *Applied Soft Computing Journal*, 80, 107-124. doi:10.1016/j.asoc.2019.03.010

23. Zhang, Y., Song, X.-f., & Gong, D.-w. (2017). A return-cost-based binary firefly algorithm for feature selection, *Information Sciences*, 418, 561-574. doi:10.1016/j.ins.2017.08.047