INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
*(Volume 4, Issue 4)*
Available online at: www.ijariit.com

# Feature selection in network intrusion detection using metaheuristic algorithms

*Tahira Khorram*
*khorram_tahira@yahoo.com*
*Selçuk University, Konya, Turkey*

*Nurdan Akhan Baykan*
*nurdan@selcuk.edu.tr*
*Selçuk University, Konya, Turkey*

## ABSTRACT

***Network Intrusion Detection (IDS) mechanism is a primary requirement in the current fast growing network systems. Data Mining and Machine Learning (DM-ML) approaches are widely used for network anomaly detection during the past few years. Machine learning based intrusive activity detector is getting popular. However, they produce a high volume of false alarms. One of the main reasons for generating false signals is redundancy in the datasets. To resolve this problem, an efficient feature selection is necessary to improve the intrusion detection system performance. For this purpose, here we use Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), K-Nearest Neighbors (KNN) and Support Vector Machine (SVM). The three abovementioned algorithms are used to select the most relevant feature set for identifying network attacks, KNN and SVM algorithms are used as classifiers to evaluate the performance of these feature selection algorithms. The standard NSL-KDD dataset is used for training and testing in this study. We used different metrics to determine which of these algorithms provide a better overall performance when they are used for feature selection in intrusion detection. Our experiments show that PSO, ACO and ABC algorithms perform better than other approaches in feature selection. Feature selection based on ABC provides 98.9% of accuracy rate and 0.78% false alarm with KNN algorithm as the classifier, which is the best result among the examined algorithms.***

***Keywords***— *Feature selection, Intrusion detection, Network security, Machine learning, Metaheuristic algorithms*

## 1.INTRODUCTION

Recently the internet and computer networks have become the inseparable part of everyday life. A statistic until 2017 shows that there are 20.35 billion devices connected to the internet all over the world and this number will be increased up to 31.75 billion devices through 2020 [1]. By connecting more devices to the computer systems, the risk of unauthorized activities such as data destruction, data modification, and data theft from both internal intruders and external intruders will be increased [2].

Several types of security appliances and protocols are designed to protect our distributed systems from a variety of internet attacks. Firewalls, Intrusion Prevention System (IPS), and Intrusion Detection System (IDS) are the most widely used appliances. In this study, our focus is on IDS. Intrusions are a set of actions that try to overrule the security aspect of a system and violate the confidentiality, integrity, and availability of that computer network [3]. Intruders always try to find a vulnerability in the system to launch an attack; it is intrusion detection system that monitors and analyzes all events happening on the computer system, identifies intrusive activities and searches for a sign of security problems [3, 4]. In case of abnormal behavior, or an attack, IDS sends an alarm to the system administrator to react immediately before the intrusion affects the network. IDS can be deployed as network-based to monitor all network events or can be set up on a PC as host-based to audit all incidents happening on that PC [5]. Anomaly-based detection and misuse-based detection are two standard approaches to network anomaly detector. Misuse-based works based on signature, it generates an alarm when an intrusive activity matches the signature. Unlike misuse detection, anomaly-based warns system admins when there is an event deviating from the normal behavior of the system, and thus it is capable of detecting unknown attacks [5, 6]. IDS is used to protect a computer system. A protected network system is defined as a barrier, which prevents violations of availability, confidentiality, and integrity of information and resources [7, 8].

Researchers presume that anomaly traffic behavior differs from regular traffic and unknown network traffic patterns are similar to known traffic instances. Based on the mentioned hypothesis intrusion detection can be considered as data analysis problem [7]. To specify features for network traffic records, Data Mining and Machine Learning (DM-ML) algorithms are widely used. DM-ML algorithms help to define samples for intrusive traffic and regular traffic. Due to the massive amount of network data produced daily, determining related and useful patterns of data is a difficult task. Generated datasets are usually noisy and contain unnecessary and correlated features that confuse the intrusion classifier engine and reduce the overall performance of the system [9].

 

Therefore, feature selection is a significant issue in detecting network anomalies. Feature space that is fed to an intrusion classifier as training examples have a significant impact on the system performance [10]. Thus, plenty of works had been done on the selection of the right, and related features for network traffic records to boost the performance of the IDS and reduce the computational cost. A large number of that literature focus on heuristic searches for selecting the right features. In the recent approaches, metaheuristic algorithms such as Particle Swarm, Ant Colony, and Artificial Bee Colony algorithms are used for feature selection in the classification of intrusions.

In this paper, we aim to use these three metaheuristic algorithms for feature selection, KNN, and SVM as an evaluator for the selecting the right features by metaheuristic algorithms. We seek to discover the most efficient algorithm for feature space selection of intrusive detector system. In this paper, we will show how feature selection improves the overall performance of the intrusion detection system. This paper is structured as follows. In section 2 we provide brief information on methods and materials used in this research. Part 3 includes the result analysis and discussion, and in section 4 we conclude the paper.

## 2. METHODS AND MATERIALS

### 2.1 Related Works

In detecting network anomalies, an accurate dataset leads to optimal performance. Feature space of an intrusion classifier needs to be preprocessed. To remove redundant and unnecessary features, several methods have been proposed. In [11], a different intrusion detector is proposed by applying ant colony optimization for feature selection, and Support Vector Machine (SVM) algorithm for classification. In this study, they mapped the feature into a connected graph, in which each ant can select one feature. The selection of the next feature depends on pheromone value and heuristic information. The proposed method applied to a KDD99 dataset which includes 10,000 data records. The overall performance of the proposed technique is 97.7%. In [12], an Ant Colony Optimization (ACO) algorithm is applied to the KDD99 dataset as feature selector, it selects 14 features among 41 features, and SVM is used as a detection method. 5,823 records of the KDD99 dataset is used for training and 77,287 used for testing the intrusion detection model. The experimental result shows 98.5% of detection accuracy with ACO-SVM and 98.2% with SVM. Aghdam et al. [13], presented ACO as feature selection and select 19 features from the datasets. They used two datasets for their experiments; the NSLKDD dataset and the KDDCUP99 dataset. The experimental result using KNN classifier shows 98.9% detection accuracy.

A hybrid method using multi-objective particle swarm optimization and the random forest is recommended to detect Probe attack. Their objective is to improve the detection rate and decrease the false alarm discovery rate while identifying Probe attacks. Particle Swarm Optimization (PSO) eliminates the unnecessary features, and Random Forest (RF) detects the Probe attacks. The detection rate of this proposed method is 90.7% [14]. Ahmad [9], used Principle Component Analysis (PCA) and PSO for feature reduction. He used PCA to reduce features. The features are selected based on their eigenvalues. It is not guaranteed that attribute with higher eigenvalue provides optimal sensitivity for the classifier. To make sure that optimal features are chosen for the intrusion detector the author proposes optimization methods. He used PSO for optimization to improve the performance of PCA and Artificial Neural Network (ANN) for detection. The proposed method detection rate upgraded from 94.50 to 99.40 and false alarm reduced from 5.5 to 0.6. Srinoy in [15], proposed PSO for feature reduction and SVM for intrusion detection. The detection rate is 96.1%, and the false alarm rate is 3.89%.

In [16], the authors introduced the Artificial Bee Colony (ABC) algorithm for intrusion classification. They used the Classification And Regression Trees (CART) and Bayesian Network and Markov Blanket (BNMB) algorithms to select the most relevant features for classifier from the KDD99 dataset. By the proposed method, the accuracy rate of 97% is achieved for the known attack, and for unknown attacks, 93.25% accuracy has been measured. Ghanem et al. [17] use a new approach to design an efficient intrusive detector tool. This method uses a multiobjective ABC algorithm to minimize the number features of IDS dataset. Feed-Forward Neural Network (FFNN) is used to determine intrusive packets and standard traffic packets. In [18], ABC and SVM are used to design an IDS. In this work, the ABC algorithm is used for two purposes; First it is used to select the necessary features from the KDD99 dataset, and secondly, it is used to optimize the SVM parameters. For anomaly detection, ABC-SVM is proposed. The overall performance of this method is higher than PSO and GA-SVM at the same time.

### 2.2 Particle Swarm Optimization (PSO)

PSO is a population-based optimization method developed by Dr. Eberhard and Dr. Kennedy in 1995 [19]. PSO inspired by a scenario where a group of birds is searching for a piece of food in a specific area, none of the birds know where the food is but in each iteration the birds know how far the food is. The optimal way to find the food is to follow the nearest bird to the food [20].

In PSO, the birds are called as particles and population is called as a swarm. Each particle of the group keeps track of their attributes. These attributes include the particle's current position, the current velocity that keeps track of speed and direction in which the particle is flying. Each particle has fitness value that is obtained by calculating the fitness function at the particle's current position [20, 21].

PSO algorithms work as follows:

***Step1:*** Initialize a population with P particle. For each particle, with a d dimension of the problem, set the particle's position and velocity randomly.

***Step 2:*** Evaluate each particles fitness value

If *i*th particle position is better than previous best (*pbest*):

If fitness *i*th > *pbest*

then *pbest* = *X*

If *i*th particle position is better than previous global best in the swarm:

if fitness *i*th> *gbest*

then *gbest* = *X*

***Step 3:*** Update the velocity of particle *i*th

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{1i} * (p_{id} - x_{id}^{t}) + c_2 * \ r_{2i} * (p_{gd} - x_{id}^{t}) \quad (1)$$

Update the position of particle *i*th.

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

In the equations, $t$ shows the $t$th iteration of PSO, $d$ indicates search space dimension, w is initial weight, $c_1$ and $c_2$ are acceleration factors, $r_1$ and $r_2$ are random numbers between (0,1). And $p_{id}$ and $p_{gd}$ are pbest and gbest respectively.

 

***Step 4:*** Return to step 2 if stopping criterion (exp. max iteration) has not been met.
***Step 5:*** Return global best value (here is the selected features).

For the experiment, each particle represents features in the swarm search space. Random 1 or 0 values initialize each particle. The feature with value 1 is selected, and feature with 0 value will be eliminated. Thus, every particle illustrates a different subset of features. The particles are randomly initialized and then start moving in the search space to search for the best subset of features by updating its position and velocity. For example, out of 41 features, 11 attributes will be selected. this selected feature set might include attributes: x1, x4, x5, x6, x12, x25, x26, x29, x30, x33, x37. So after any generation, a particle might look like (1,1,1,0,0,1,0,1,1,0,1). As we mentioned before, value 1 indicates selected attributes while 0 shows ignored attributes. Now we can say that attributes selected by this particle are x1, x4, x5, x25, x29, x30, x37. In the next generation, because of the pbest and gbest of the other particles, this particle's position will be changed, and this time it will select a different set of attributes among these 11 attributes. The dimension of the particles will be updated according to the equation (1) and (2).

The KNN and SVM classification algorithms will be used for classification of attacks. The PSO selected features will be used for detecting attacks.

**2.3 Ant Colony Optimization (ACO)**

ACO, which is developed by Dorigo in 1992, has been inspired by the social behavior of ants searching for a food source all together in a group. The ants in the swarm communicate along with a chemical matter called pheromone. Ants spray pheromone on their route to food origin. Each ant can follow the path marked by pheromone. The pheromone concentration differs from the primary random direction. Ants always follow the roads with higher pheromone concentration as the number of ants' increases on that specific route the number of pheromones also will be expanded, and that route will be selected as an efficient direction to the food source [22, 23].

ACO algorithm works as follows:
***Step 1:*** Initialize ant colony population randomly.
***Step 2:*** Search for the solutions. The probability of selecting the next solution by ant is calculated by the equation 3.

$$p_{ij}^{k} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{m\epsilon N_i^k}((\tau_{im})^{\alpha}(\eta_{im})^{\beta}} & , j\epsilon N_i^k \\ 0 & , j \notin N_i^k \end{cases} \quad (3)$$

Where $\eta$ is a symbol for heuristic information (number of time a feature has been visited), $j$ is the set of neighbor nodes that have not been visited by ant k yet. $\alpha$ And $\beta$ parameters determine the amount pheromone concerning the heuristic information.
***Step 3:*** Update the local pheromone by using equation 4

$$\tau_j = (1-\rho) * \tau_j + \Delta_{j*\sigma} \quad (4)$$

Where,

$$\Delta_j = \begin{cases} \rho & if\ j\varepsilon N \\ 0 & otherwise \end{cases}$$

$N$ is the set of visited neighbors' nodes for that process. $\sigma\epsilon(0,1)$ controls $\Delta_j$.
*Step 3:* Update the global pheromone using the following equation.

$$\tau_j = (1-\rho) * \tau_j + \phi_j * \sigma \quad (5)$$

Where

$$\phi_j = \begin{cases} \rho\ if\ \varepsilon\ global\ best\ tour \\ 0 \qquad otherwise \end{cases}$$

All these steps repeated until the termination condition (max iteration or until the accuracy cannot improve more) is met.

In this experiment, ants are placed randomly on a graph node. Each ant selects a feature. For ant $k$ that is placed randomly on the node (feature) $i$, the probability of selecting next feature $j$ is calculated by equation 3. After every generation, the amount of local pheromone value is updated using equation 4. After completing the tour, the ants pass their selected features to the classifier. The dataset that is produced by ants gets global pheromone update by the equation 5. For example, out of 41 features, 7 attributes is selected by ACO, this feature set might include attributes: x2, x5, x6, x12, x17, x21, x39. After each generation, the route is changed, and a new set of features will be selected, until the max iteration is reached. The algorithm returns a set of features that are selected to be used by the KNN and the SVM classifiers to detect network attacks.

**2.4 Artificial Bee Colony (ABC)**

ABC is a swarm-based optimization method that was developed by Karaboga in 2005. The algorithm is inspired by the social behavior of honeybee [24]. The algorithm consists of two components:

**Employee bees:** they find a food source, store information about the quality of the food and share the information with others.
**Unemployed bees:** They are two types:
Onlooker bees receive information about food source and choose the food source with higher quality. The other type is scout bees when the existing food source is over; scout bees try to find new food origin [25].

For our experiment, we use ABC to select the best features of NSLKDD dataset for the KNN and SVM classifiers. The ABC algorithm works as follows:
***Step 1:*** Initialize the bee swarm by a feature vector, where the elements are placed in different positions of the vector. If the value of the vector at that position is 1 that attribute will be selected otherwise the attribute will be dropped. Equation 6 shows the bee swarm initialization.

$$x_m = l_i + rand(0,1) * (u_i - l_i) \quad (6)$$

Where $x_m$ is the food source, $u_i$ and $l_i$ are solution space the upper level and lower level. $u_i$ , $l_i$ and *rand (0, 1)* are a random number in range [0, 1].
***Step 2:*** The employee bee search for food sources in the neighborhood. This exploration is defined in equation 7:

$$v_{mi} = x_{mi} + \varphi_{mi}(x_{mi} - x_{ki}) \quad (7)$$

Where $i$ is a randomly selected parameter index, $x_k$ is a randomly selected food source, and $\varphi_{mi}$ is a random number in the range [-1, 1]. After $v_{mi}$ is generated we can obtain the fitness value for the feature or for the food origin according to equation 8.

$$fit_i = \begin{cases} \dfrac{1}{f_i + 1}, & f_i \geq 0 \\ 1 + |f_i|, & f_i < 0 \end{cases} \quad (8)$$

Where $f_i$ shows the objective value of the $i$th solution.
***Step 3:*** After employee bee has found the food source they will share the information about the food source and its quality with the onlooker bees, the probability of selecting that food source (feature) by onlooker bees is represented in equation 9.

 

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_i} \qquad (9)$$

Where $fit_i$ indicates the fitness solution represented by food source *i* and *SN* indicate the total number of food sources.

***Step 4:*** If the effectiveness of food sources cannot be improved, then the scout bee remove the existing solution and start searching for a new solution randomly using equation 6.
After the ABC algorithm is reached its max iteration, it will return a set of features that are used by KNN and SVM classifiers to detect network attacks.

### 2.5 K-Nearest Neighbors (KNN)

KNN is one of the supervised machine learning algorithms that are very easy to understand and very simple to implement. It works based on the minimum distance of the new instance from the training sample that determines as nearest neighbors [26]. After nearest neighbors are gathered, the majority vote of the neighbors decides what class the new instance belongs to. For example, if k=5, this algorithm will look to five nearest neighbors of the new instance to find a class for the unknown sample. KNN classifier usually works based on the Euclidean Distance (ED), which is the distance between the test sample and a specific training sample [27].

The equation 10 shows how the Euclidean Distance (ED) calculates the distance between a new instance and its neighbors, which is what the KNN is commonly based on. There is also some other distance calculator such as Hamming Distance (HD), Manhattan Distance (MD).

$$dist((x,y),(a,b)) = \sqrt{(x-a)^2 + (y-b)^2} \qquad (10)$$

Where (x, y) is the coordinate for the new instance and (a, b) is the coordinate of the existing example. The distance between the new instance and its neighbors is calculated using the above equation.

### 2.6 Support Vector Machine (SVM)

Support Vector Machine is a supervised ML algorithm which can be used for classification and regression problems. It is widely used in security software such as network anomaly detection. In this algorithm, each data item is plotted as a point in n-dimensional space. With the value of a particular coordinate, then classification is performed by finding the hyperplane that differentiates the classes very well. This algorithm is simple to apply and provides a good result [18]. In SVM there are two parameters that play a significant role in classification: Cost and Gamma parameters. Gamma determines the influence of training examples on the model that will be created. A low value shows that each training example does not have a high effect on the model and a higher value indicates that every training example has a high impact on the classifier model that is being created. Moreover, Cost parameter determines the cost of misclassification on the training examples. Cost with a high value make a rigorous classification, and the margin of error will be small, in this case, the classifier supposed to classify every training example correctly. Also Cost with lower values makes the margin of error a little loose; it might cause more misclassification. The optimal Cost values is a value that leaves some space for errors while it is intended to classify correctly [28].

## 3. RESULT ANALYSIS AND DISCUSSION

### 3.1 Approach Structure

Our experiment consists of two steps:

1. Feature Selection
2. Classification

In this study, we used Weka [29] for feature selection and classification experiments. PSO, ACO, and ABC algorithms select the best features for the KNN and SVM classifiers with standard parameters of Weka. We classify the network traffic into normal class and attack class. Accuracy Rate (AR) is considered as a fitness function of the classifier. The process of feature selection is performed using the abovementioned algorithms, and classification is done using the KNN and the SVM algorithms. The attribute selection algorithms loop will be terminated when the termination criteria (max iteration or highest accuracy) is reached. Our experiment flow is presented in Figure 1.

### 3.2 Dataset

The dataset that we used for the experiment is 20% of NSL-KDD dataset consists of 25192 records with 41 attributes [28]. The dataset class separation is binary and has 13449 normal data records which are labeled as 1 in the dataset and 11743 anomaly records which are labeled as 0 in the dataset [30]. Table 1 list all features of NSLKDD dataset.

The dataset feature contains different types of values as categorical and integer. We convert the categorical values to numeric values. The categorical features are protocol, flag, and service. Table 2 shows the conversion of text features to integers.

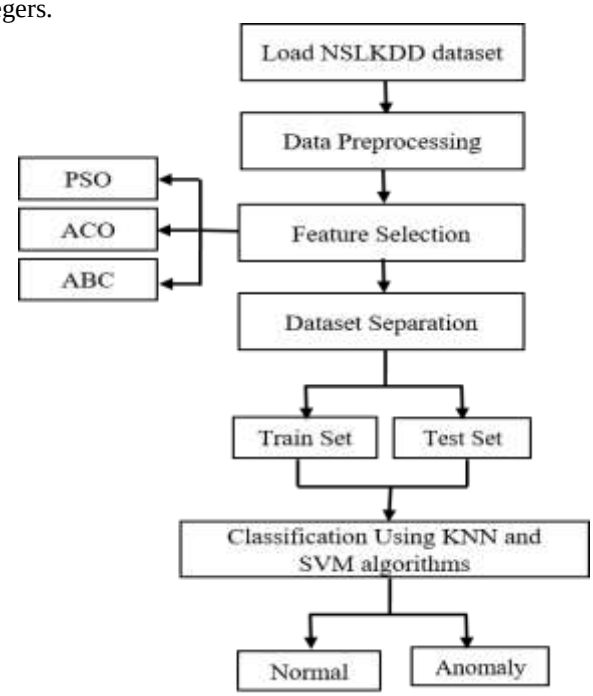


**Fig. 1: Approach structure for the study**

**Table 1: Dataset Attributes**

| No | Feature | No | Feature |
|---|---|---|---|
| 1 | Duration | 2 | Protocol_type |
| 3 | Service | 4 | Flag |
| 5 | Src_bytes | 6 | Dst_bytes |
| 7 | Land | 8 | Wrong_fragment |
| 9 | Urgent | 10 | Hot |
| 11 | Num_failed_logins | 12 | Logged_in |
| 13 | Num_compromised | 14 | Root_shell |
| 15 | Su_attempted | 16 | Num_root |
| 17 | Num_file_creations | 18 | Num_shells |
| 19 | Num_access_files | 20 | Num_outband_cmds |
| 21 | İs_hot_login | 22 | Is_guest_login |
| 23 | Count | 24 | Srv_count |

 

| | | | |
|---|---|---|---|
| 25 | Serror_rate | 26 | Srv_serror_rate |
| 27 | Rerror_rate | 28 | Srv_rerror_rate |
| 29 | Same_srv_rate | 30 | Diff_srv_rate |
| 31 | Srv_dif_host_rate | 32 | Dst_host_count |
| 33 | Dst_host_srv_count | 34 | Dst_host_same_srv_rate |
| 35 | Dst_host_diff_srv_rate | 36 | Dst_host_same_src_port_rate |
| 37 | Dst_host_srv_dif_host_rate | 38 | Dst_host_serror_rate |
| 39 | Dst_host_srv_serror_rate | 40 | Dst_host_rerror_rate |
| 41 | Dst_host_srv_rerror_rate | | |

### 3.3 Preprocessing

**Table 2: Converting Text Attributes to Numeric Values**

| Attribute | Attribute value with their numeric values |
|---|---|
| Protocol type | tcp=1, udp=2, icmp=3 |
| Service value | private=1, ftp_data=2, eco_i=3, telnet=4, http=5, smtp=6, ftp=7, ldap=8, pop_3=9, courier=10, discard=11, ecr_i=12, imap4=13, domain_u=14, mtp=15, systat=16, iso_tsap=17, other=18, csnet_ns=19, finger=20, uucp=21, whois=22, netbios_ns=23, link=24, Z39_50=25, sunrpc=26, auth=27, netbios_dgm=28, uucp_path=29, vmnet=30, domain=31, name=32, pop_2=33, http_443=34, urp_i=35, login=36, gopher=37, exec=38, time=39, remote_job=40, ssh=41, kshell=42, sql_net=43, shell=44, hostnames=45, echo=46, daytime=47, pm_dump=48, IRC=49, netstat=50, ctf=51, nntp=52, netbios_ssn=53, tim_i=54, supdup=55, bgp=56, nnsp=57, rje=58, printer=59, efs=60, X11=61, ntp_u=62, klogin=63, tftp_u=64, red_i=65, urh_i=66, http_8001=67, aol=68, http_2784=69, harvest=70 |
| Flag value | REJ=1 SF=2 RSTO=3 S0=4 RSTR=5 SH=6 S3=7 S2=8 S1=9 RSTOS0=10 OTH=11 |

The dataset also contains values with different scales; these types of values need to be normalized. We normalized the dataset feature values in the range [0, 1]. As we know the ACO algorithm is designed to work with the discrete dataset, we discretized the dataset by converting all feature values to 0 or 1 by the mean value of each feature values. If the value is smaller than the mean value of its feature values, these values are converted to 0; if not, converted to 1.

### 3.4 Feature Selection (FS)

Attribute selection means selecting the minimum number of features that are essential for the classifier to define the normal and intrusive activity effectively and efficiently. We apply three algorithms to choose the appropriate attributes for the KNN and SVM classifiers. Table 3 shows all the selected features in the feature selection algorithms.

### 3.5 Classification

After the feature selection process, the dataset is passed to KNN and SVM classifiers to be split using 10-fold cross-validation technique [31]. 10-fold means the dataset is divided into ten parts, e.g., D1….D10. Each time one subset is used for testing and the rest of the dataset is used for training. We will gain ten classification accuracies, the average of these accuracies are the main classification accuracy. The process of classification is to classify the dataset into normal and attack traffic. KNN classification algorithm obtains different results based on various features selection techniques.

**Table 3: Features selected by each algorithm**

| FS Method | Number of Features | Selected Features |
|---|---|---|
| PSO | 11 | 4, 5, 6, 12, 25, 26, 29, 30, 31, 33, 37 |
| ACO | 7 | 3, 5, 6, 12, 26, 29, 39 |
| ABC | 7 | 5, 6, 12, 23, 26, 31 39 |

In Table 4 the experimental results are presented. For comparison of results, Accuracy (AR), Detection Rate (DR), False Alarm Rate (FA) and Operation Time are used. First, we tested the performance of KNN on the full feature set. The KNN complete its training time in four minutes and fifty seconds. The accuracy rate is 93.9%, with 3.4% false alarms. Then the KNN classifier was trained and test on PSO based dataset. Here, the training and testing time is 52 seconds with an accuracy of 96.04%, and 2.7% false alarms. After that, the classifier is evaluated on ACO dataset with 67 seconds' time utilization, 98.1% accuracy and 0.82% false alarms. In the end, the KNN method is fed by ABC based dataset to check the performance of ABC based attribute selection. The KNN time consumption is 53 seconds. The ABC algorithm performs better than PSO and ACO. The accuracy of the KNN classifier on ABC based dataset is 98.9% with 0.78% false alarms.

**Table 4: KNN classification results based on PSO, ACO, and ABC**

| FS method | AR (100) | DR (100) | Best K Value | Num Features | False Alarm | Time (Sec) |
|---|---|---|---|---|---|---|
| None | 93.9 | 91.9 | 4 | 41 | 3.4 | 291 |
| PSO | 96.04 | 94.9 | 4 | 11 | 2.7 | 52 |
| ACO | 98.13 | 97.2 | 1 | 7 | 0.82 | 67 |
| ABC | 98.9 | 98.7 | 1 | 7 | 0.78 | 53 |

Table 5 shows the SVM classifier result on NSLKDD and reduced NSLKDD datasets. The SVM algorithm performs better on ABC reduced dataset. The accuracy rate is improved from 91.4% to 97.1%, and the false alarm decreased from 6.4% to 4.5%.

**Table 5: SVM classification results based on PSO, ACO, and ABC**

| FS method | AR (100) | DR (100) | Num. Features | FA | Time (Sec) | Best params |
|---|---|---|---|---|---|---|
| None | 91.4 | 89.9 | 41 | 6.4 | 722 | C=1, G=0.5 |
| PSO | 96.02 | 92.3 | 11 | 5.4 | 309 | C=1, G=2 |
| ACO | 95.6 | 93 | 7 | 4.9 | 142 | C=1, G=0.25 |
| ABC | 97.1 | 93.9 | 7 | 4.5 | 341 | C=2, G=0.14 |

Figure 2, 3 and 4 compare the feature selection techniques based on accuracy rate, False Alarm Rate, and Detection Rate, respectively. The ABC method provides the overall better performance.

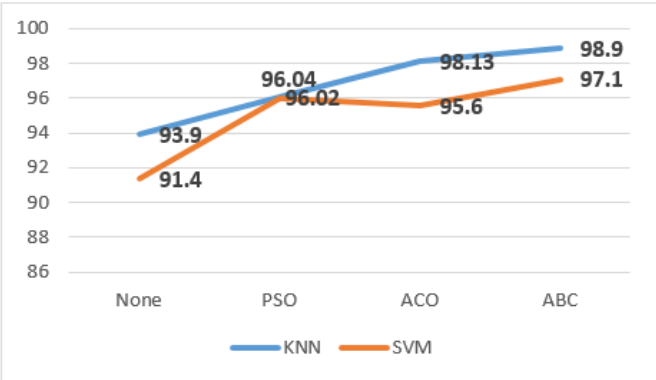


**Fig. 2: Accuracy Rates**

 

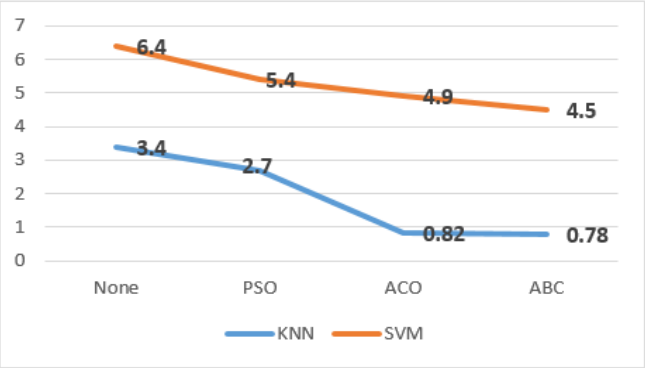


**Fig. 3**: **False Alarms Rate**

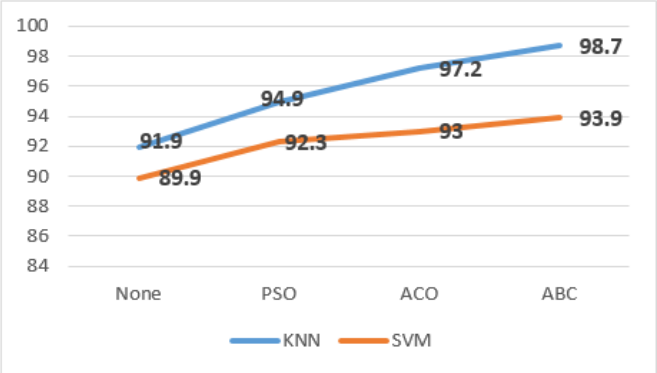


**Fig. 4**: **Detection Rate**

## 4. CONCLUSION

Network systems are vulnerable to various types of attacks. Therefore, an accurate and compelling attack detection mechanism is required to overcome such threats. Several intrusion detection methods are used to detect network anomalies, but their performance is an issue. The performance is degraded due to the noisy datasets that consist of unnecessary and correlated features. In this paper, we evaluated some feature reduction methods to improve the performance of IDS. We used PSO, ACO, and ABC to find the most appropriate features from the NSLKDD dataset. KNN and SVM methods are used for classification. It is discovered that the ABC method with a detection rate of 98.7% and an accuracy rate of 98.9% outperforms the existing methods in this experiment.

## 5. REFERENCES

[1] Internet of Things connected devices installed based worldwide from 2015 to 2025. Retrieved from Statista

[2] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, [Accessed in Feb 2017]

[3] S. Mukherjee, and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," Procedia Technology, pp 119-128, 2012.

[4] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, P. Yogesh, and A. Kannan, "Intelligent feature selection and classification techniques for intrusion detection in networks," EURASIP Journal on Wireless Communications and Networking, pp 913-921, 2013.

[5] M. Panda, & M. R. Patra, "Network Intrusion Detection Using Naive Bayes." Internation Journal of Computer Science and Network Security, pp. 258-263, December, 2007.

A. Med, A. Lisitsa, and C. Dixon, "A misuse-based network intrusion detection system using temporal logic and stream processing," IEEE Network and System Security (NSS), 5th International Conference on. Milan, 2011.

[6] Ismail Butun, Salvatore D Morgera, and Ravi Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks", IEEE Communications Surveys & Tutorials 266-282, 2013

[7] C. Darigue, I. H. Jang, and W. Zeng, "A new data-mining based approach for network intrusion detection" 7th annual Communication Networks and Services Research Conferences, pp. 372-377, 2009.

[8] M.A. Aydin, A. H. Zaim, & C. K. Gokhan, "A hybried detection system design for computer network security", Computer and Electrical Engineering, pp. 517-526, May 2009

[9] Ahmad, "Feature selection using particle swarm optimization in intrusion detection", International Journal of distributed sensor networks, 2015.

[10] G. Stein, B. Chen, A. S. Wu, & K. A. Hua, "Decision tree classifier for network intrusion detection with GA-based feature selection", Proceedings of the 43rd annual Southeast regional conference, Kennesaw, 2005.

[11] H. H. Goa, H. Uang, & X. Y. wang, "Ant colony optimization based network intrusion feature selection and detection", Proceeding of the fourth international conference on machine learning and cybernetics (pp. 3871-3875). Guangzhou: IEEE, 2005.

[12] T. Mehmod, & H. B.Rais, "Ant colony optimization and feature selection for intrusion detection", Springer international publication Switzerland, pp. 305-312, 2016

[13] M. Hosseinzadeh Aghdam, & P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization", Internation Journal of network security, pp. 420-432, 2016

[14] A. J. Malik, & F. Aslam Khan, "A hybrid technique using multi-objective particle swarm optimization and random forest for PROBE attacks detection in a network", IEEE international conference on systems, man, and Cybernetics, pp. 2473-2478, 2013

[15] S. Srinoy, "Intrusion detection model based on particle swarm optimization and support vector machine", IEEE Symposium on computational intelligence in security and defense applications, 2007.

[16] M. Aldwairi, Y. Khamayseh, and M. Al-Masri, "Application of artificial bee colony for intrusion detection systems", Security and communication networks, 2012

[17] W. Ghanem, A. & Jantan, "Novel multi-objective artificial bee colony optimization for wrapper-based feature selection in intrusion detection", International Journal of advanced soft computing, 2016

[18] Wang, T. Li, & R. Ren, (n.d.). "A real-time IDS based on artificial bee colony-support vector machine", a Third international conference on advanced computational intelligence. Suzhou, Jiangsu, 2010

[19] R.C. Eberhart, & J. Kennedy, "A new optimizer using particle swarm theory", In Proceedings of the 6th international symposium on micro machine and human science, Nagoya, Japan, pp. 39–43, 1999.

[20] Xin-She Yan, "Metaheuristic Optimization Algorithms", http://www.scholarpedia.org/article/Metaheuristic_Optimization [Accessed in Dec 2017]

[21] M. Karakoyun, N.A. Baykan, M. Hacibeyoglu, "Multi-Level Thresholding for Image Segmentation with Swarm Optimization Algorithms", International Research Journal of Electronics & Computer Engineering, Vol:30, 2017

[22] Xin-She Yan., "Metaheuristic Optimization: Nature-Inspired Algorithms and Applications", Cambridge: Luniver Press, 2010

 

[23] Benhala, Bachir, Ali Ahaitouf and Mechaqrane Abdellah., "Multiobjective Optimization of an Operational Amplifier by the Ant Colony Optimisation Algorithm", Electric and Electronic Engineering, 2012

[24] D. Karaboga, "An idea on honey bee swarm for numerical optimization", Kayseri: Erciyes University, 2005

[25] D. Karaboga, "Artificial Bee Colony Algorithm", Retrieved from Scholarpedia: http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm [Accessed in Jun 2010]

[26] E. Leif Peterson, "K-Nearest Neighbors", http://www.scholarpedia.org/article/K-nearest_neighbor [Accessed in July 2018]

[27] Yihua. Liao, and V Rao Vemuri., "Use of K-Nearest Neighbor classifier for intrusion detection", Computers and Security pp. 439-448, 2002.

[28] H. Voldan, "Anomaly detection using Machine learning techniques." Oslo, Norway: University of Oslo, 2016

[29] I.H. Witten, E. Frank, and M.A. Hall. "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, San Francisco, 2010.

[30] Dhanabal, & S. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms", International Journal of Advanced Research in Computer and Communication Engineering, pp. 446-451, 2015.

[31] N.A. Baykan, & N. Yılmaz, "A mineral classification system with multiple artificial neural network using k-fold cross validation", Mathematical and Computational Applications, Vol: 16, No: 1, pp.22-30, 2011.

 