*Academic Journal of Nawroz University (AJNU), Vol.9, No.2, Jan 2020*
*This is an open access article distributed under the Creative Commons Attribution License*
Copyright ©2017. e-ISSN: 2520-789X
https://doi.org/10.25007/ajnu.v9n2a816

# Rule Mining Using Particle Swarm Optimization for Intrusion Detection Systems

Adel Sabry Eesa
Computer Science Department, Faculty of Science, Zakho University, Duhok City, KRG, Iraq

## ABSTRACT

Traditional data mining techniques are commonly used to build the Intrusion Detection Systems IDSs. They are designed on the basis of some probabilistic methods that still do not take into account some of the important properties of each feature in the dataset. We believe that each feature in the dataset has its own crucial role for its characteristics, which should be taken into consideration. In this work, instead of using the traditional technique or applying feature selection methods we proposed max and min boundary mining approach to solve Anomaly Intrusion Detection System AIDS problem. The main idea of the proposed method is to handle each feature in the dataset independently extracting two important properties represented by max-boundary and min-boundary. First, Particle Swarm Optimization PSO is used to search for the optimal max and min boundary for each feature in each class from the train data set. Second, the generated max and min boundaries are used as detection rules in order to detect anomalies from normal behavior using test dataset. KDD Cup 99 and the new version of KDD Cup 99 called NSL-KDD datasets are used to test the proposed model and its performance is compared with four well-known techniques such as J48, Naïve Bayes, PART and SMO. In addition, performance is also compared with some recent work. Experiment results show that the proposed model is outperformed all other algorithms in all terms (true positive rate, false positive rate, f-measure, Recall, Precision, MCC and AUC).

**KEY WORDS:** Anomaly intrusion detection system, Data mining, Particle swarm optimization, NSL-KDD data, Feature extraction

## 1. Introduction

There are two main types of IDSs: signature-detection technique SDT and anomaly-detection technique ADT (Aljawarneh, Aldwairi, & Yassein, 2018), (A.S. Eesa, Orman, & Brifcani, 2015). SDT systems rely on pattern recognition techniques where they maintain the database of signatures of previously known attacks and compare them with analyzed data.

On the other hand, ADT systems center on the concept of a baseline for network behavior. This baseline is a description of accepted network behavior, which is learned or specified by the network administrators, or both. Events in an anomaly detection engine are caused by any behaviors that fall outside the predefined or accepted model of behavior.

Recently, many papers adopt of using bio-inspired optimization algorithm for solving intrusion detection problem such as Genetic Algorithm (Hamamoto, Carvalho, Sampaio, Abrão, & Proença, 2018), (Gauthama Raman, Somu, Kirthivasan, Liscano, & Shankar Sriram, 2017). Swarm (Kanaka Vardhini & Sitamahalakshmi, 2017), (Ali & Jantan, 2011), (Chung & Wahid, 2012). Cuttlefish optimization algorithm (Adel Sabry Eesa, Orman, & Brifcani, 2015) and Ant Colony (Varma, Kumari, & Kumar, 2016), (Aghdam & Kabiri, 2016).

The most of anomaly detection models are designed based on the traditional data mining techniques or using the enhanced version of these techniques. Traditional techniques have a bias towards classes which have a number of instances and output with a higher probability for an instance belonging to the majority class (Guo & Viktor, 2008).

However, we believe that each feature in the dataset has its own crucial role for its characteristics, which should be taken into consideration. In this work, instead of using the traditional technique or applying feature selection methods, a new data mining approach has been proposed based on PSO, called PSO-AIDS. The proposed method can tackle each feature in the dataset independently extracting some important information (rules) such as max-boundary and min-boundary. Where max and min boundaries for feature *j* in class *i* are the highest and the lowest boundaries for feature *j* belonging to class *i*, respectively. In this way, all values for the feature *j* in class *i* must be ranged between the max-boundary and the min-boundary. Therefore, PSO is used to find the optimal max and min boundaries for each feature in each class, then the extracted max and min boundaries are used as a classification rule in order to classify each instance in the test dataset into either normal behavior or anomaly.

The paper is organized as follows: Section 2 presents some related works. The general background of the PSO algorithm is presented in Section 3. Section 4 presents the detail of the proposed PSO-AIDS model. Experimental setup of the proposed technique is illustrated in section 5, while the experimental results are described and discussed in Section 6. Finally, the conclusion and feature plan are stated in section 7.

**2. RELATED WORK**

Some recent existing studies have proposed the use of different techniques to build AIDSs. (Aljawarneh, Aldwairi and Yassein, 2018) proposed a hybrid method to solve the obtained of the high false and the low false rate. In their work, vote method and information gain are used to filter the data, the filtering result was combined with some classifier such as: "J48, Meta Paging, RandomTree, REPTree, AdaBoostM1, DecisionStump, and NaiveBayes". Both, (Mazini, Shirazi and Mahdavi, 2018) and (Adel Sabry Eesa, Orman and Brifcani, 2015) proposed a combination of optimization and classification techniques, the optimization technique was used to select the optimal features, while the classification algorithm was used to evaluate the selected features. (Khraisat, Gondal and Vamplew, 2018) have proposed C5 classifier to build AIDS to reduce false alarm rate and increase detection accuracy. In their work, NSL-KDD dataset was used to test their proposed C5 and its performance was compared with C4.5, SVM, and Naïve Bayes. Hybrid methods based on feature discrete and cluster analysis was proposed by (Liao, Liu and Wang, 2018). The main idea was to split the training dataset into two subsets (normal and abnormal), then another level of classification was built to enhance the performance of the subgroup classification using decision tree and Bayesian network. (Hamamoto et al., 2018) proposed a combination method between Genetic Algorithm GA and Fuzzy Logic to build an anomaly detection system. GA was used to extract a digital signature from network flow data for the given time interval, then FL was used to detect anomalies. (Hajisalem and Babaie, 2018) proposed a combination approach using artificial bee colony and artificial fish swarm so that Fuzzy C_means clustering was used to split train dataset, then a correlation method was used to select important features and removing noisy ones. In addition, CART technique was also used in their work to classify selected features to normal or anomaly instances. (Benmessahel, Xie and Chellal, 2018) proposed a combination of natural evolutionary algorithm and artificial neural network ANN to solve intrusion detection problem.

**3. PARTICLE SWARM OPTIMIZATION ALGORITHM**

PSO algorithm was firstly produced in 1995 by (R. Eberhart & Kennedy, 1995), it is a population-based stochastic optimization technique. Each particle is

initialized randomly and flies in the search domain having its velocity and position. The velocity is updated dynamically based on its flying history and the history of the other particles in the swarm. Each particle keeps tracking its position following its best position (*pBest*) and the best particle position among all particles in the swarm called global best position (*gBest*). The formulation of updating the velocity and the position of each particle is described in (1) and (2).

$$V[] = w * V[] + c_1 * r_1 * (pBest[] - position[]) + c_2 * r_2 * (gBest[] - position[]) \quad (1)$$

$$position[] = position[] + V[] \quad (2)$$

Where $V[]$ is the particle velocity, $position[]$ is the current particle position, $pBest[]$ is the previews current particle best position, $gBest[]$ presents the global best position (best particle position among all particle in the swarm). The parameters $w$, $c_1$, $c_2$, $r_1$, and $r_2$ are used to control the behavior of the particle in the swarm. Where $w$ presents the inertia weight, $c_1$ and $c_2$ are social learning factors and their values are usually defined as constants. While $r_1$ and $r_2$ are two random values generated between the interval (0, 1). The main steps of PSO are illustrated in Algorithm 1.

**Algorithm 1:**

***Input***:

*N: is the number of particles in the swarm S.*

*w, $c_1$, $c_2$: PSO parameters.*

***Output***:

*Best solution gBest.*

***Method***:

- *Initialize S and keep the best solution in gBest.*
- *While the terminate condition does not meet do*
  - *a. For each particle in the swarm S do*
    - *i. Update the velocity using Eqs. 1.*
    - *ii. Update the position using Eqs. 2.*
    - *iii. Evaluate the new_solution,*
    - *iv. if new_solution is better than the previews local best solution (pBest) replace pBest with the new_solution.*
  - *b. Find the best solution of all particles and replace gBest with it.*
- *End while.*

## 4. PROPOSED PSO-AIDS

As it is mentioned before, PSO is proposed to search for the optimal maximum and minimum boundaries for each feature in each class in the training dataset. The extracted boundaries are then used as classification rules to detect anomaly instances using testing dataset. First of all, the population $S$ is initialized with $N$ random solutions. Each particle in the population $S$ is associated with fitness and six vectors of size $C$, where $C$ is the number of classes in the training dataset. The structure of each particle is described as follows,

```
Particle {
        Velocity_max_boundary[C];
        Velocity_min_boundary[C];
        Max_boundary[C];
        Min_boundary[C];
        pBest_Max_bundary[C];
        pBest_Min_boundary[C]
        Fitness;
}
```

The *maximum* and the *minimum* values of each feature in each class are extracted from the training dataset, then the two vectors: *Lower* Limit and *Upper* Limit are calculated using (3) and (4), respectively. The *Upper* limit and the *Lower* limit are used to initialize the velocities of particles in the PSO algorithm.

$$Lower[i] = -1 * (maximum[i] - minimum[i] \quad (3)$$

$$Upper[i] = |maximum[i] - minimum[i]| \quad (4)$$

where $i = 1, 2, \ldots, C$.

The six vectors of each particle are then initialized as follows,

*Velocity_max_boundary[i] = random() * (Upper[i] -*

*Lower[i]) + Lower[i].*

*Velocity_min_boundary[i] = random() * (Upper[i]-Lower[i]) + Lower[i].*

*Max_boundary[i] = random() * (maximum[i] – minimum[i]) + minimum[i].*

*Min_boundary[i] = random() * (maximum[i] – minimum[i]) + minimum[i].*

*pBest_Max_bundary[i]= Max_boundary[i].*

*pBest_Min_boundary[i] = Min_boundary[i].*

After the initialization process, the population *S* is evaluated based on the fitness function and the best particle is kept in *gBest*. Then the PSO algorithm starts searching for the optimal Max_boundary and Min_boundary of each feature in each class in the training dataset. The main steps of the proposed PSO-AIDS technique are described in Algorithm 2.

**Algorithm 2:**

- *While (stop condition does not meet)*
- *For each particle p in population S do*
  - *Update velocity as follows: // i = 1, 2, …, C*
    - *p.Velocity_max_boundary[i] = w*p.Velocity_max_boundary[i] + $c_1$*$r_1$* (p.pBest_Max_boundary[i]-Max_boundary[i])+ $c_2$*$r_2$*(gBest.Max_boundary[i] – Max_boundary[i])*
    - *p.Velocity_min_boundary[i] = w*p.Velocity_min_boundary[i] + $c_1$*$r_1$* (p.pBest_Min_boundary[i]-Max_boundary[i])+ $c_2$*$r_2$*(gBest.Min_boundary[i] – Min_boundary[i])*
    - *If (p.Velocity_max_boundary[i] > Upper[i] ) // Upper [i] is calculate using Eqs. 4.*
      *Then p.Velocity_max_boundary[i] = Upper[i]*
    - *If (p.Velocity_max_boundary[i] < Lower[i] ) // Lower [i] is calculate using Eqs. 3.*
      *Then p.Velocity_max_boundary[i] = Lower[i]*
    - *If (p.Velocity_min_boundary[i] > Upper[i] )*
      *Then p.Velocity_min_boundary[i] = Upper[i]*
    - *If (p.Velocity_min_boundary[i] < Lower[i] )*
      *Then p.Velocity_min_boundary[i] = Lower[i]*
  - *Update particle position as follows:*
    - *p.Max_boundary[i] = p.Max_boundary[i] + p.Velocity_max_boundary[i].*
    - *p.Min_boundary[i] = p.Min_boundary[i]+p.Velocity_min_boundary[i].*
  - *Evaluate the new position using fitness function and update pBest*
    - *If (p.Max_boundary and p. Min_boundary is better than p.pBest_Max_boundary and p.pBest_Min_boundary)_*
    - *Then*
      *p.pBest_Max_boundary = p.Max_boundary*
      *p.pBest_Min_boundary = p.Min_boundary*
  - *Update gBest*
    - *If(p.Max_boundary and p. Min_boundary is better than gBest_Max_boundary and gBest_Min_boundary)*
    - *Then*
      *gBest_Max_boundary = p.Max_boundary*
      *gBest_Min_boundary=p.Min_bounday*
- *End for*
- *End while*
- *Return gBest*

## 5. USING GENERATED RULES FOR CLASSIFICATION PURPOSE

After PSO algorithm finds the optimal max and min boundaries, these boundaries are then used as classification rules to classify each instance in the testing dataset into normal behavior or anomaly. For example, consider a dataset with four features and two classes. In this case, we will have two sets of rules for class 1 and two sets of rules for class 2. For instance,

Class1_Rules = {Max_boundary$_1$[4], Min_Boundary$_1$[4}, and

Class2_Rules = {Max_boundary$_2$ [4], Min_boundary$_2$[4]},

Where 4 is the number of features. Considering an instance $x = \{v_1, v_2, v_3, v_4\}$, the obtained rules are working as follows,

**If** ( x[i] > $Min\_boundary_1$[i] and x[i] < $Max\_boundary_1$[i])

**Then** the feature x[i] is belonging to class 1.

In this way, each value of the instance $x$ will assign to either class 1 or class 2. As a final decision, the majority vote is used to decide whether the instance $x$ belongs to class 1 or class 2. For example, if $v_1$, $v_2$, and $v_3$ are assigned to class 1 and $v_4$ is assigned to class 2, then the instance $x$ is classified as class 1.

## 6. EXPERIMENTAL SETUP

### 6.1. Data preprocessing

The KDD Cup 1999 dataset is used for benchmarking intrusion detection problems. The dataset is a collection of a period of nine weeks on a local area network (UCI Machine Learning Repository, 2015). The attacks types are grouped into five categories (Normal, Probing, DoS, U2R, and R2L) (Levin, 2000). Another set of data was extracted from the KDD Cup 1999 called NSL-KDD by (Tavallaee, Bagheri, Lu, & Ghorbani, 2009), which consists of the same features without any redundant and duplicates record and this dataset is widely used in the literature, and it is available online at ("NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB," 2017). In our work, all symbolic values of KSL-KDD are converted to continuous values. For example, the protocol_type attribute consists of three symbolic values (tcp, udp, icmp), and these symbolic values will be converted to (10, 20, and 30), respectively. In other words, If an attribute consists of 100 symbolic values, these values will be converted to (10, 20, 30, …, 1000), respectively. Thus, all symbolic values will be converted to continuous values.

### 6.2. Evaluation

In order to evaluate the performance of the proposed PSO-AIDS model, seven well-known metrics are suggested in our evaluation process, namely TPR, FPR, Precision, Recall, F-measure, Matthews Correlation Coefficient MCC, and Area Under the Curve AUC. All such metrics were produced from the confusion matrix (Jiao & Du, 2016) shown in Table 1.

TP and TN in Table 1 represent the number of instances that are correctly classified as positive and negative, respectively. FP and FN are the numbers of instances that are incorrectly classified as a positive and negative class, respectively. The formulas of the seven metrics are stated below:

**Table 1**
**The confusion matrix**

| Prediction / Actual | Positive | Negative |
|---|---|---|
| Positive | TP | FP |
| Negative | FN | TN |

$$TPR = TP/(TP + FN) \quad (5)$$

$$FPR = FP/(FP + TN) \quad (6)$$

$$Precision = TP/(TP + FP) \quad (7)$$

$$Recall = TP/(TP + FN) \quad (8)$$

$$F\_measure = (Precision * Recall * 2)/(Precision + Recall) \quad (9)$$

$$Mcc = ((TP * TN) - (FP * FN)) \Big/ \sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)} \quad (10)$$

$$AUC = (1 + TPR - FPR)/2 \quad (11)$$

### 6.3. Fitness Function

The fitness function is formulated based on the TPR and FPR as shown in (12). The values of TPR and FPR are extracted from the confusion matrix.

$$Fitness = x * TPR + y * (1 - FPR), \quad (12)$$

where $x$ and $y$ are two parameters, and their values determine the importance of the TPR and FPR, respectively. The value of $x$ is between (0, 1) and $y$ = 1 - $x$. In this work, both TPR and FPR have the same importance and their values are equally set to 0.5.

## 7. EXPERIMENTS AND RESULTS

The implementation of the proposed model is carried out using C# language within the Microsoft Visual Studio 2013 environment. The performance of the

proposed method is compared with the performances of the four well-known techniques in Weka (Hall et al., 2009), such as J48, PART, SMO and Naïve Bayes. The parameters of the PSO were set best on the work of (R. C. Eberhart & Shi, 2000) as follows: $c_1$ and $c_2$ were set equal to 1.49445, while the inertia factor $w$ is set to 0.729. Population size is set to 20, and the number of iteration is set to 100. In all experiments, the results obtained from the proposed method are the average of 10 independent runs.

**Experiment 1:** In this experiment, NSL-KDD training and testing dataset were used to evaluate the proposed model. Table 2, Fig. 1 and 2 illustrate the performance of the proposed PSO-AIDS model compared to the performance of the other four algorithms. It can be seen that the performance of the proposed model is much better than all other techniques in all terms. The next best result is obtained with the J48 technique; however, there is still a significant difference between the performance of our proposed PSO-AIDS model compared to the J48 technique. Furthermore, the confusion matrix shown in Table 3 and 4 describes that the J48 technique has incorrectly classified 3996 cases among 9698 anomaly instances as normal instances. However, with the proposed technique only 25 anomaly instances are incorrectly classified as normal class.

**Table 2**
**Result of the proposed PSO-AIDS compared to other algorithms using NSL-KDD.**

| algorithm | TPR | FPR | Precision | Recall | f-measure | MCC | AUC |
|---|---|---|---|---|---|---|---|
| J48 | 0.64 | 0.179 | 0.839 | 0.64 | 0.681 | 0.356 | 0.730 |
| PART | 0.604 | 0.352 | 0.774 | 0.604 | 0.649 | 0.195 | 0.626 |
| SMO | 0.527 | 0.362 | 0.754 | 0.527 | 0.579 | 0.127 | 0.583 |
| Naïve Bayes | 0.558 | 0.348 | 0.766 | 0.558 | 0.607 | 0.161 | 0.605 |
| PSO-AIDS | 0.988 | 0.002 | 0.993 | 0.988 | 0.99 | 0.988 | 0.988 |

**Table 3**
**Confusion matrix using PSO-AIDS model**

| | Predict | normal | anomaly |
|---|---|---|---|
| Actual | | | |
| Normal | | **2137** | 15 |
| Anomaly | | 25 | **9673** |

**Table 4**
**Confusion matrix using J48 technique**

| | Predict | normal | anomaly |
|---|---|---|---|
| Actual | | | |
| normal | | **1879** | 273 |
| anomaly | | 3996 | **5702** |

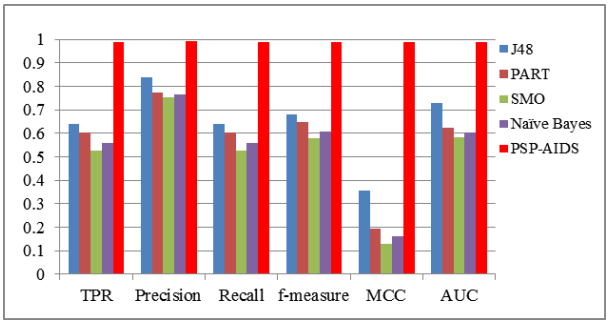


**Fig. 1. Chart diagram of the performance for the proposed model compared to other techniques**

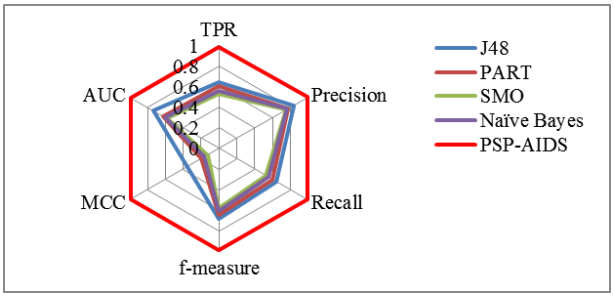


**Fig. 2. Radar chart of the results for the proposed model compared to other techniques**

**Experiment 2:** In this experiment, two new subsets were randomly selected from the original 10%KDD Cup 99 dataset. The 10%KDD training and testing datasets contained about 494020 and 311028 instances, respectively. To keep the proportion of each attack in both, train and test dataset, each tack is divided by 100. In this way, many attacks will be missed in the testing dataset which will make the process of classification very difficult. The number of instances in the newly generated training and testing data will be 4947 and

3117, respectively. The description of the generated data is shown in Table 5. The bolded text indicates the number of attacks in the training dataset that are not seen in the testing dataset. The left values represent the number of attack cases in the training dataset, while values on the right represent the number of cases of that attack in the testing dataset.

Both Table 6, and Fig. 3 and 4 show the results obtained from the proposed technique compared with the results of the other four techniques. Once again, the performance of the proposed PSO-AIDS model outperforms all other techniques in all terms,. These results clearly show the robustness and the capability of the proposed model to detect most of anomaly instances and have recognized them from normal behavior.

**Table 5**
**Number of each attack in the extracted train and test dataset**

| Dos attacks | U2R attacks | R2L attacks | Probing | Normal |
|---|---|---|---|---|
| (3915; 2299) | (5; 10) | (13; 160) | attacks | (973; |
| | | | (41; 42) | 606) |
| apache2 (0; | buffer_overflow | Guesspasswd | Ipsweep | |
| 8) | (3; 1) | (2;44) | (12;3) | |
| Back (22; 11) | Httptunnel ( 0; | Snmpgetattack | Mscan | |
| Mailbomb | 3) | (0;77) | (0;11) | |
| (0; 50) | Rootkit (2; 2) | Snmpguess | Nmap | |
| Neptune | Xterm (0; 2) | (0;24) | (2;1) | |
| (1072; 580) | Ps (0; 2) | Warezclient | Portsweep | |
| Processtable | | (10;0) | (11;4 ) | |
| (0; 8) | | Warezmaster | Saint (0;7) | |
| Pod(3; 1) | | (1;15) | Satan | |
| Smurf(2808; | | | (16;16) | |
| 1641) | | | | |
| teardrop(10; | | | | |
| 0) | | | | |

Tables 7, 8, 9, 10 and 11 illustrate the confusion matrix of all used techniques, while Table 12 describes the number of instances for each attack that is correctly classified by each technique. From these tables, we can clearly observe that the proposed technique is performed much better than any other techniques used. Despite the fact that the traditional classification techniques can recognize instances based on some probabilistic methods or enhance it by feature selection, they are still not taking into account some important properties of each feature in the dataset. In other words, in the traditional techniques, the probability or the frequency of instances direct the results to the majority class and ignore the minority classes. However, our new proposed technique adds this property into account. In other words, the strategy of the proposed technique is designed to deal with each feature for each class in the dataset independently extracting some other important properties represented by max and min boundaries of each feature for each class. Results of our conducted experiments supported the importance of the new mentioned strategy.

**Table 6**
**Results using the extracted train and test dataset**

| algorithm | TPR | FPR | Precision | Recall | f-measure | MCC | AUC |
|---|---|---|---|---|---|---|---|
| J48 | 0.918 | 0.028 | 0.931 | 0.918 | 0.901 | 0.864 | 0.945 |
| PART | 0.919 | 0.085 | 0.926 | 0.919 | 0.921 | 0.851 | 0.917 |
| SMO | 0.746 | 0.024 | 0.911 | 0.746 | 0.795 | 0.641 | 0.861 |
| Naïve Bayes | 0.703 | 0.186 | 0.855 | 0.703 | 0.753 | 0.480 | 0.756 |
| PSO | 0.95 | 0.014 | 0.955 | 0.95 | 0.952 | 0.947 | 0.968 |

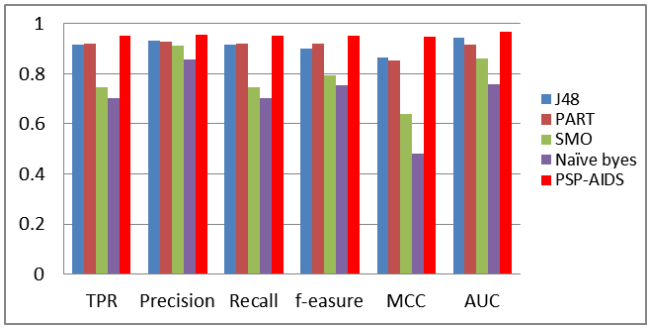


**Fig.3. Chart diagram using the extracted train and test dataset**

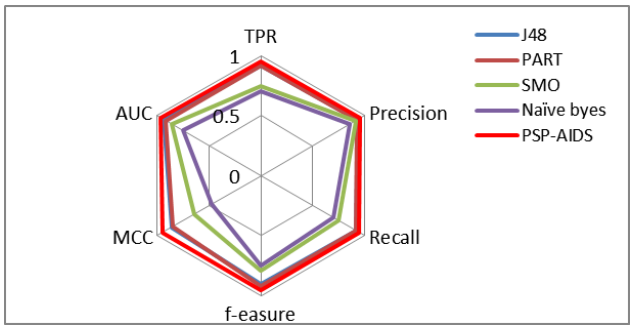


**Fig. 4. Radar chart using the extracted train and test dataset**

**Table 7**
**Confection matrix of PSO-AIDS**

| Predict / Actual | Normal | Dos | U2R | R2L | Probing |
|---|---|---|---|---|---|
| Normal | **597** | 7 | 2 | 0 | 0 |
| Dos | 6 | **2284** | 3 | 4 | 2 |
| U2R | 1 | 1 | **8** | 0 | 0 |
| R2L | 2 | 3 | 0 | **154** | 1 |
| Probing | 0 | 4 | 0 | 4 | **34** |

**Table 8**
**Confection matrix of J48**

| Predict / Actual | Normal | Dos | U2R | R2L | Probing |
|---|---|---|---|---|---|
| Normal | **590** | 9 | 0 | 1 | 6 |
| Dos | 55 | **2226** | 0 | 0 | 18 |
| U2R | 2 | 0 | **4** | 0 | 4 |
| R2L | 147 | 0 | 1 | **7** | 5 |
| Probing | 5 | 4 | 0 | 0 | **33** |

**Table 9**
**Confection matrix of PART**

| Predict / Actual | Normal | Dos | U2R | R2L | Probing |
|---|---|---|---|---|---|
| Normal | **493** | 28 | 4 | 77 | 4 |
| Dos | 20 | **2250** | 9 | 3 | 17 |
| U2R | 3 | 0 | **4** | 0 | 3 |
| R2L | 4 | 62 | 9 | **80** | 5 |
| Probing | 3 | 0 | 3 | 0 | **36** |

**Table 10**
**Confection matrix of SMO**

| Predict / Actual | Normal | Dos | U2R | R2L | Probing |
|---|---|---|---|---|---|
| Normal | 595 | 3 | 1 | 0 | 7 |
| Dos | 10 | 1693 | 16 | 0 | 580 |
| U2R | 4 | 0 | 4 | 1 | 1 |
| R2L | 150 | 5 | 0 | 1 | 4 |
| Probing | 10 | 1 | 0 | 0 | 31 |

**Table 11**
**Confection matrix of Naïve Bayes**

| Predict / Actual | Normal | Dos | U2R | R2L | Probing |
|---|---|---|---|---|---|
| Normal | 491 | 92 | 0 | 2 | 21 |
| Dos | 56 | 1654 | 0 | 0 | 589 |
| U2R | 5 | 0 | 2 | 0 | 3 |
| R2L | 40 | 102 | 4 | 5 | 9 |
| Probing | 1 | 0 | 0 | 1 | 40 |

**Table 12**
**Number of instances that are correctly classified to their correct class**

| Technique | #correctly classified as Normal | #correctly classified as Dos | #correctly classified as U2R | #correctly classified as R2L | correctly classified as Prob |
|---|---|---|---|---|---|
| #Actual instances | 606 | 2299 | 10 | 160 | 42 |
| J48 | 590 | 2226 | 4 | 7 | 33 |
| PART | 493 | 2250 | 4 | 80 | 36 |
| SMO | 595 | 1693 | 4 | 1 | 31 |
| Naïve Bayes | 491 | 1654 | 2 | 5 | 40 |
| PSO-AIDS | 597 | 2284 | 8 | 154 | 34 |

### 7.1. Comparing with Existing Literature

Table 13 shows the comparison results with some existing literature and it is clearly seen that the performance of the proposed methods is better than all other existing work except the work of (Hosseini Bamakan, Wang, & Shi, 2017) which gives better result with small deference about 0.1 in term accuracy, while the proposed PSO-AIDS is obtained better results in both FPR and f-measure.

**Table 13**
**Comparing PSO-AIDS with some existing work**

| Reference | Accuracy % | FPR % | f-measure |
|---|---|---|---|
| HG-GA SVM(Gauthama Raman et al., 2017) | 97.14 | 0.83 | N/A |

| | | | |
|---|---|---|---|
| Ramp-KSVCR(Hosseini Bamakan et al., 2017) | 98.68 | 0.86 | 98.74 |
| ABC- AdaBoos (Mazini, Shirazi, & Mahdavi, 2018) | 98.9 | 0.01 | N/A |
| ANN(MVO-ANN) (Benmessahel, Xie, & Chellal, 2018) | 98.21 | 0.032 | N/A |
| PSO-AIDS (This work) | 98.8 | 0.002 | 99.00 |

## 8. CONCLUSION AND FEATURE WORKS

In this paper, we proposed and investigated a new data mining technique, called PSO-AIDS. The motivation behind developing this new technique is to extract important rules represented by max and min boundaries from each feature in each class to solve anomaly detection problems. Unlike traditional techniques, our suggested method provides a new way to extract some important properties not only depending on the probabilistic methods but also takes into account the role of each feature for each class in the dataset. In order to test the accuracy of the new method suggested, we used two datasets, the original KDD Cup 99 and the preprocessed NSL-KDD dataset. The obtained results were promising and show the robustness and the ability of the proposed method to detect anomaly instances,. During the experiments, we observed that the proposed method is time-consuming to find the optimal rules. For example, when NSL-KDD data is used, the execution time for training and testing processes for each run takes about two minutes. Although this time is not a long time, this limitation is suggested for future works. In addition, the use of the proposed method to solve the classification problems in different domains is suggested.

## 9. REFERENCES


1. Aghdam, M. H., & Kabiri, P. (2016). *Feature Selection for Intrusion Detection System Using Ant Colony Optimization*. *International Journal of Network Security* (Vol. 18). Retrieved from https://pdfs.semanticscholar.org/022d/50ecb37eb6c78be9728ed7bc198a29cc6915.pdf
2. Ali, G. A., & Jantan, A. (2011). A New Approach Based on Honeybee to Improve Intrusion Detection System Using Neural Network and Bees Algorithm (pp. 777–792). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-22203-0_65
3. Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, *25*, 152–160. https://doi.org/10.1016/j.jocs.2017.03.006
4. Benmessahel, I., Xie, K., & Chellal, M. (2018). A new evolutionary neural networks based on intrusion detection systems using multiverse optimization. *Applied Intelligence*, *48*(8), 2315–2327. https://doi.org/10.1007/s10489-017-1085-y
5. Chung, Y. Y., & Wahid, N. (2012). A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Applied Soft Computing*, *12*(9), 3014–3022. https://doi.org/10.1016/J.ASOC.2012.04.020
6. Eberhart, R. C., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)* (Vol. 1, pp. 84–88). IEEE. https://doi.org/10.1109/CEC.2000.870279
7. Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39–43). IEEE. https://doi.org/10.1109/MHS.1995.494215
8. Eesa, A.S., Orman, Z., & Brifcani, A. M. A. (2015). A new feature selection model based on ID3 and bees algorithm for intrusion detection system. *Turkish Journal of Electrical Engineering and Computer Sciences*, *23*(2). https://doi.org/10.3906/elk-1302-53
9. Eesa, Adel Sabry, Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*,

*42*(5), 2670–2679. https://doi.org/10.1016/J.ESWA.2014.11.009

10. Gauthama Raman, M. R., Somu, N., Kirthivasan, K., Liscano, R., & Shankar Sriram, V. S. (2017). An efficient intrusion detection system based on hypergraph - Genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowledge-Based Systems*, *134*, 1–12. https://doi.org/10.1016/J.KNOSYS.2017.07.005
11. Guo, H., & Viktor, H. L. (2008). *Learning from Skewed Class Multi-relational Databases*. Retrieved from https://pdfs.semanticscholar.org/63f4/09c747a7a556701246cb3d69f669d3961690.pdf
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10. https://doi.org/10.1145/1656274.1656278
13. Hamamoto, A. H., Carvalho, L. F., Sampaio, L. D. H., Abrão, T., & Proença, M. L. (2018). Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic. *Expert Systems with Applications*, *92*, 390–402. https://doi.org/10.1016/J.ESWA.2017.09.013
14. Hosseini Bamakan, S. M., Wang, H., & Shi, Y. (2017). Ramp loss K-Support Vector Classification-Regression; a robust and sparse multi-class approach to the intrusion detection problem. *Knowledge-Based Systems*, *126*, 113–126. https://doi.org/10.1016/j.knosys.2017.03.012
15. Jiao, Y., & Du, P. (2016). Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quantitative Biology*. https://doi.org/10.1007/s40484-016-0081-2
16. Kanaka Vardhini, K., & Sitamahalakshmi, T. (2017). Implementation of Intrusion Detection System Using Artificial Bee Colony with Correlation-Based Feature Selection (pp. 107–115). Springer, Singapore. https://doi.org/10.1007/978-981-10-2471-9_11
17. Levin, I. (2000). KDD-99 classifier learning contest LLSoft's results overview. *ACM SIGKDD Explorations Newsletter*, *1*(2), 67. https://doi.org/10.1145/846183.846201
18. Mazini, M., Shirazi, B., & Mahdavi, I. (2018). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2018.03.011
19. NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2017). Retrieved January 2, 2019, from https://www.unb.ca/cic/datasets/nsl.html
20. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1–6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528
21. UCI Machine Learning Repository. (2015). KDD Cup 1999 Data. Retrieved from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
22. Varma, P. R. K., Kumari, V. V., & Kumar, S. S. (2016). Feature Selection Using Relative Fuzzy Entropy and Ant Colony Optimization Applied to Real-time Intrusion Detection System. *Procedia Computer Science*, *85*, 503–510. https://doi.org/10.1016/J.PROCS.2016.05.203