Journal of Business Data Science Research

Intrusion Detection in Computer Networks Through Combining Particle Swarm Optimization and Decision Tree Algorithms

Amin Rezaeipanah[a,*], Musa Mojarad[b], Samaneh Sechin Matoori[c]

[a] Department of Computer Engineering, University of Rahjuyan Danesh Borazjan, Bushehr, Iran.
[b] Department of Computer Engineering, Firoozabad Branch, Islamic Azad University, Firoozabad, Iran.
[c] Department of Managment, Najafabad Branch, Islamic Azad University, Najafabad, Iran.

ABSTRACT

Nowadays, network-based computer systems have an essential role in modern society and therefore can be targeted by enemies or intruders. To provide complete security in a computer system that is connected to the network, the use of firewalls and other intrusion prevention mechanisms is not always enough, and it is necessary to use other systems called intrusion detection systems. This type of system detects and notifies the user if an intruder passes through the firewall and antivirus and enters the system. Data mining techniques and methods are used to improve the function of these types of systems and to correctly detect intrusions. Due to a large number of features in the intrusion detection data, in this study, a subset of desired features was first selected by using a combination of graph-based clustering algorithm and Particle Swarm Optimization (PSO). Then, to classify the data and to detect intrusion, a model using the standard decision tree data mining technique is shown. The implementation of the proposed method is evaluated by using the NSL-KDD database, which has more realistic records than other intrusion detection data. The results of the experiments show a high functionality of the proposed method.

KEWWORDS: Intrusion Detection Systems, Data Mining, Feature Selection, Particle Swarm Optimization.

ORGINAL ARTICLE

Received 2 September 2020
Accepted 3 March 2021

CITATION

Rezaeipanah, A., Mojarad, M., Sechin Matoori, S. (2021). Intrusion Detection in Computer Networks Through Combining Particle Swarm Optimization and Decision Tree Algorithms. *Journal of Business Data Science Research*, 1(1), 14-22.

1. Introduction

Detection and prevention of intrusion is one of the main mechanisms in providing the security of computer networks and systems. The importance of network security is increasing day by day and the intrusion detection systems (IDS) have been developed at a high speed to help achieve this goal and increase the security. There are several intrusion detection systems to detect attacks, in which the main challenge is to increase efficiency (Pan et al., 2015). Most current intrusion detection systems use all the parameters in network packets to evaluate and detect attack patterns, while some of these parameters are irrelevant and redundant. The use of all parameters prolongs the detection process and degrades the efficiency of the intrusion detection system. Therefore, the main challenge in intrusion detection systems is the huge volume of data (Zuech et al., 2015). On the other hand, due to the high traffic, reducing the wrong alarm rate in the intrusion detection system is also of particular importance. All intrusion detection systems are capable of generating intrusion alarms on the network. However, due to the high volume of alarms generated by these systems and also the production of false alarms, these systems cannot manage and analyze the generated alarms. Most approaches today focus on the intrusion detections related to the problem of selecting or extracting important features.

* Corresponding author: amin.rezaeipanah@gmail.com

• http://www.jbdsr.com/article_680445.html

Intrusion detection methods include detecting abnormal behavior and abuse (based on signature) (Kenkre et al., 2015). There are several types of intrusion detection system architectures that can generally be divided into three categories: host-based, distributed, and network-based. Intrusion detection systems are responsible for detecting any unauthorized use of the system, abuse, or damage by both internal and external users. Intrusion detection systems are created in the form of software and hardware systems, and each has its own advantages and disadvantages. Speed and accuracy are considered as the advantages of hardware systems, and lack of security breaches by intruders is another feature of such systems. In general, the three main functions of IDS are monitoring, evaluation, detection, and response.

One of the problems in implementing intrusion detection systems is the high level of information and the high number of features of each attack. The presence of a large number of these unrelated and redundant features in the data set negatively affects the performance of the machine learning algorithm and also increases the computational complexity. The use of Evolutionary Computation grew in interest recently. Among various Evolutionary Computation approaches, the Genetic Algorithm (Rezaeipanah & Ahmadi, 2020) and PSO are used in optimization problems; they have much in common but also have some differences. In this research, we try to select features as basic characteristics that improve the accuracy of intrusion detection. Here we try to select the best subset of features by presenting a new approach based on combining clustering and PSO algorithms. The decision tree algorithm is used to detect data intrusion and classification. The NSL-KDD database, which has more realistic records than other intrusion detection data, is used to evaluate the performance of the proposed method.

The remainder of this paper is structured as follows. Section 2 summarizes the methods and results of previous works on the intrusion detection systems. The proposed method is highlighted in Section 3. Experimental results are discussed in Section 4. Finally, the conclusion is drawn in Section 5.

2. Literature review

Many studies have been done to reduce the characteristics of intrusion detection systems, most of which are based on artificial intelligence and data mining methods. An intrusion detection system with a machine learning approach and a genetic algorithm is presented in the KDDCUP99 database to obtain features (Goyal & Kumar, 2008; Rezaeipanah & Ahmadi, 2020). In this method, by producing a series of rules, based on each rule, a specific type of influence is identified. Another study proposed a combined learning approach of k-means clustering and simple Bayesian classification for intrusion detection (Muda et al., 2011). This clustering method assigns all data to the related groups before classification. An intrusion detection system based on a genetic algorithm and SVM has been developed to automatically determine the appropriate set of features (Saha et al., 2012). In this research, there is a predefined dictionary of the types of attacks, which provides the most desirable features for each type of attack. An intrusion detection system using a genetic algorithm has been developed to improve the initial population-making sections and the selected operator (Benaicha et al., 2014). A new method has been proposed to select the effective features in constructing the intrusion detection model, which works according to the average of the features of each class compared to the whole class(Chae et al., 2013).

The ant colony algorithm has been used to extract features in the intrusion detection system (Aghdam & Kabiri, 2016). Due to the use of simple subsets for classification, this method has fast execution feasibility and low computational complexity. Reducing the number of features by using graph displays an exploratory information to update pheromones which have resulted in higher accuracy in intrusion detection and lower error warning. Two methods, SVM and SOM neural networks have been investigated in intrusion detection systems (Mubarak, 2016). Analysis of these methods on two datasets, KDDCUP and DARPA shows that SVM has higher computational efficiency and speed than SOM. One of the reasons for the weakness of neural networks in intrusion detection systems is the difficulty of determining the appropriate size of the network and its weights. In another study, a hybrid classification model based on tree algorithms for network disturbance detection has been proposed (Kevric et al., 2017). In this research, a combined NB tree algorithm, consisting of decision tree classification and simple Bayesian is used. The orientation towards creating a classifier has been done through using the artificial security system (AIS) along with population-based incremental learning (PBIL) and participatory filtering to detect the intrusion into the network (Chen et al., 2016).

3. Proposed method

In this research, to select the features, first, the initial features are represented as a weighted graph. The nodes in this graph, properties, and edges show the similarity between properties. The graph properties are then divided by the number of clusters and the final properties are selected to detect intrusion by using the particle swarm optimization algorithm. The proposed method includes four steps: graph representation, feature clustering, optimal subset search, and final intrusion detection. Figure 1 shows the general flowchart of the proposed method. One of the main advantages of this method is the use of feature clustering in the optimal subset search process, which minimizes the redundancy between the selected features.

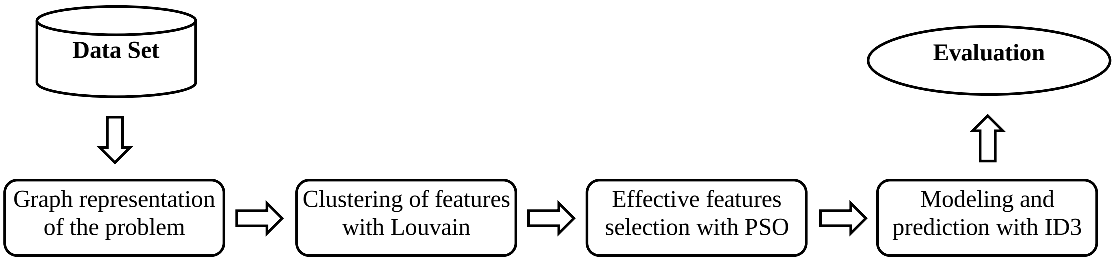

Data Set
Evaluation
Graph representation of the problem
Clustering of features with Louvain
Effective features selection with PSO
Modeling and prediction with ID3

Figure 1. Flowchart of the proposed method

3.1. Graph representation of the problem

The proposed method uses a graph-based community recognition algorithm to cluster the features. To do so, the property space must be represented as a graph. The present problem is represented as a complete weighted graph without direction $G = (F, E, w_F,$), where $F = \{F_1, F_2, \dots, F_n\}$ represents the primary properties n and $E = \{(F_i, F_j): F_i, F_j \in F\}$ represents the edges of the graph. Also $w_F: (F_i, F_j) \to \mathbb{R}$ is a function that shows the similarity between the two properties F_i and F_j. To calculate the similarity between the features, the absolute value of Pearson correlation coefficient is used (Benesty et al., 2009). Pearson correlation coefficient between two properties F_i and F_j is calculated according to equation 1.

$$w_{ij} = \left| \frac{\sum_p (x_i - \bar{x}_i)(x_j - \bar{x}_j)}{\sqrt{\sum_p (x_i - \bar{x}_i)^2}\sqrt{\sum_p (x_j - \bar{x}_j)^2}} \right| \tag{1}$$

Where, x_i and x_j represent the vector of the properties F_i and F_j, respectively. Also $\bar{x}_i$ and $\bar{x}_j$ represent the mean values for the vector x_i and x_j in sample p respectively. In order to normalize the calculated similarity values, the nonlinear scaling technique according to equation 2 is used (Theodoridis & Koutroumbas, 2009).

$$\hat{w}_{ij} = \frac{1}{1 + \exp(-\frac{w_{ij} - \bar{w}}{\sigma})} \tag{2}$$

Where, w_{ij} is the similarity between the properties F_i and F_j, $\bar{w}$ and σ show the mean and the standard deviation for the calculated similarities between all the properties, respectively.

3.2. Clustering of features

The main purpose of feature clustering is to place similar features based on the amount of similarity in the same clusters. Most of the methods presented for feature clustering have some shortcomings. Some of these shortcomings include not considering the scattering of the features of a cluster, the unknown number of clusters, considering the same effect for all features, and so on. In this research, a graph-based clustering algorithm called Louvain is used to cluster the properties (Blondel et al., 2008). This algorithm performs graph clustering automatically by maximizing the modulus function and finding the number of clusters. At the beginning of the algorithm, each node is considered as a cluster, and then clustering is done in three iterative steps.

Step 1: For each node i, the benefit of assigning that node to cluster C is calculated using equation 3.

$$\Delta\varphi = \left[\frac{\Sigma_{in} k_{i.in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{in}\cdot}{2m} - \left(\frac{\Sigma_{tot}\cdot}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \quad (3)$$

Where, Σ_{in} is the sum of the weights in cluster C, Σ_{tot} is the sum of the weights of the edges connected to the nodes of cluster C, k_i is the sum of the edges of node i and $k_{i,in}$ represents the total weight of the edges between node i and cluster C. Also, m is the sum of the weights of all the edges of the graph.

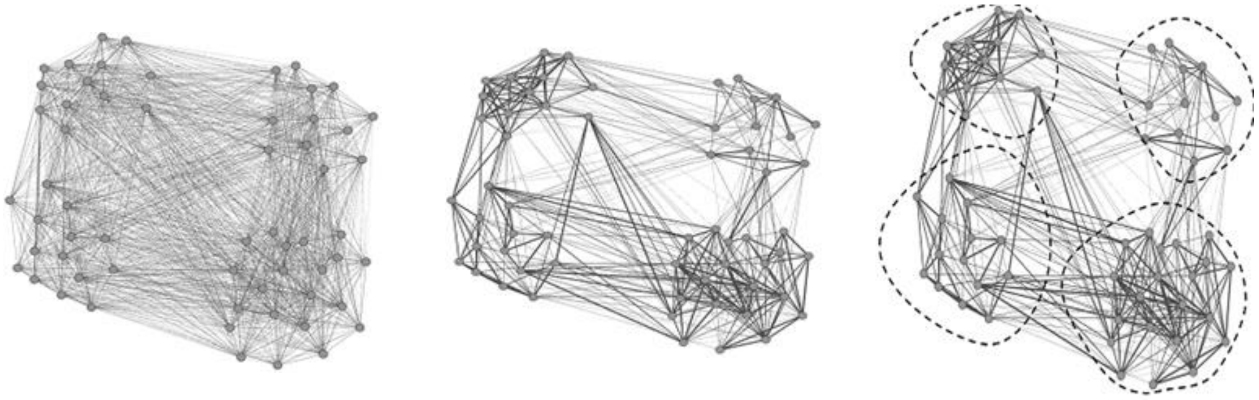
Step 2: Each node is assigned to a cluster that maximizes the module function. Clusters are then rebuilt based on this new structure.

Step 3: The previous two steps are repeated until there is no more change in the structure of the clusters or we reach a certain threshold of modularity. Modularity is one measure of the structure of networks or graphs. It was designed to measure the strength of division of a network into modules. Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. The modularity value is defined between 1- and 1, which measures the density of links within communities relative to the relationship between communities. This criterion is calculated as equation 4.

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(i,j) \quad (4)$$

Where, A_{ij} is the weight between nodes i and j, k_i and k_j is the sum of the weights of the edges connected to nodes i and j, respectively. m is the sum of all the weights of the edges in the graph and $\delta(i,j)$ shows the relationship of nodes i and j. δ is equal to 1, when nodes i and j are related and otherwise its value is 0.

Due to the completeness of the graph, a large number of edges participate in feature clustering, most of which are small in weight and have no effect on clustering. Therefore, before the implementation of the community detection algorithm, edges with weights less than threshold θ are omitted. Figure 2 shows the steps for clustering features.



A) Features representation in a complete weighted graph

B) Remove the edges of the graph by applying the threshold θ

C) Graph clustering with Louvain algorithm

Figure 2. The features clustering stages

3.3. Selecting a subset of features

In this section, using the particle swarm optimization algorithm and with the help of feature clustering, the optimal feature subset is searched. The steps of the algorithm are as follows:

Step 1: Create the initial population of the particles randomly, so that each particle represents a subset of properties. The length of each particle is equal to the size of the principal properties, n. Each particle gene has two values [0, 1], meaning the selection or non-selection of the corresponding trait. The number of selected properties per particle is the same for the entire initial population ($\omega \times k$). k is the number of clusters and ω is a parameter to control the number of selected properties.

Step 2: The evaluation function is calculated for each particle. Here, a combination of KNN classification accuracy (to $k = 3$) and the sum of similarities between the selected features are used. The appropriateness of the FS^k feature subset in the iteration t denoted by $J(FS^k(t))$ is calculated by using Eq. (5).

$$J\left(FS^k(t)\right) = \frac{CA\left(FS^k(t)\right)}{\frac{2}{|FS^k(t)| * (\ |FS^k(t)| - 1\ )} \quad \sum_{F_i, F_j \in FS^k} Sim(F_i, F_j)} \tag{5}$$

Where, $CA(FS^k(t))$ classification accuracy for the selected feature subset $FS^k(t)$, $|FS^k(t)|$, the subset size of the selected attribute and $Sim(F_i, F_j)$ indicate the similarity between the attribute F_i and F_j.

Step 3: The position of the particles is tested by using $x = x + v$ (Kennedy & Eberhart, 1995).

Step 4: Particle velocity is updated using Eq. (6) (Kennedy & Eberhart, 1995).

$$v = v \times \omega + C_1 \times r_1 \times (pbest - x) + \ C_2 \times r_2 \times (gbest - x), \ \ v \in [1, V_{max}] \tag{6}$$

Step 5: The repair operator is applied on each particle. The purpose of the restoration operator is to modify the particles so that at least ω properties are selected from each cluster. In fact, this will select the selected features from the entire search space and minimize the redundancy of the selected features. Figure 3 shows the steps performed by the repair operator for a particle with eight properties and $\omega = 1$. Therefore, the purpose of the restoration operator is to modify the particles so that a property must be selected from each cluster.

Step 6: The optimal positions of Pbest and Gbest are calculated according to the particle evaluation values.

Step 7: The previous steps are repeated until the termination condition of the algorithm is established. In this research, a certain number of iterations are used for the termination condition of the PSO algorithm.

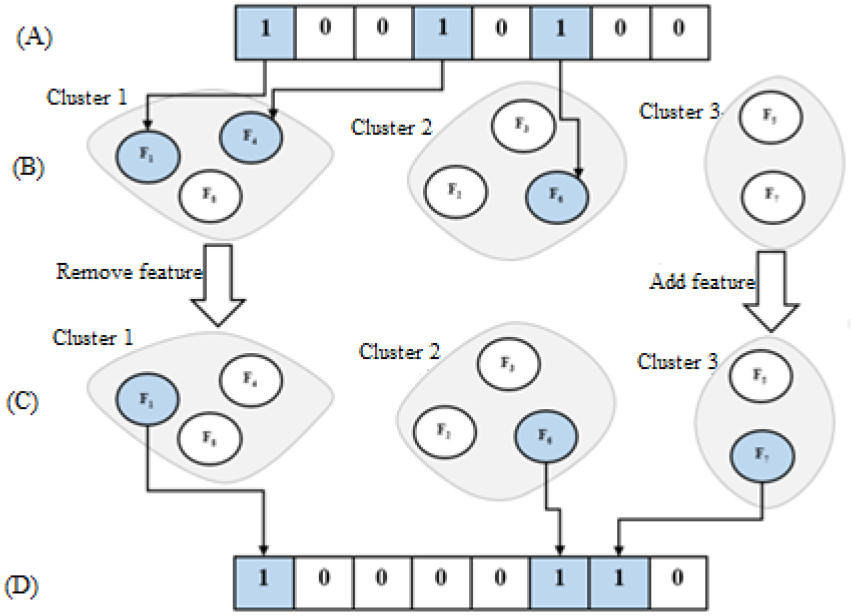


Figure 3. An example of the steps performed by the restoration operator

The details of this figure are as follows: A) a particle with three properties F_1, F_4 and F_6, B) Select properties F_1 and F_4 from the first cluster and feature F_6 from the second cluster, C) Random deletion of a feature from the first cluster (F_1 and F_4) and select a property from the third cluster and D) modified particle and the final solution.

3.4. Intrusion modeling and prediction using the decision tree

At this stage, the intrusion detection data set is reduced according to the selected features and the data classification decision is made by using the algorithm tree. In this research, ID3 algorithm, which is one of the efficient decision tree algorithms is used to detect intrusion (Sabharwal et al., 1992). In the decision tree, a statistical criterion called information interest is used. This criterion is used to determine the ability of a feature to classify training samples. The information gain of a feature is the amount of entropy reduction that results from the separation of samples through this feature. The information gain (S,A) for feature A relative to training samples S is defined as Eq. (7).

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{7}$$

Where, $Values(A)$ is a set of all values of the feature A and S_v is a subset of S in which A has A value of v.

4. Results and discussion

The NSL-KDD database was used to evaluate the proposed intrusion detection system, which includes 41 features and five classes (Normal, DOS, R2L, U2R and Probe)(Revathi & Malathi, 2013). This data set is presented in two parts: TrainSet and TestSet. The Dos class contains records that consume system resources, rejecting normal requests. The R2L class contains records that the intruder remotely connects to the victim system and uses the user's legal account. The U2R class contains records of an intruder successfully taking control of a victim system. The Probe class also contains records of intruders trying to obtain information about network services.

The simulation was performed with MATLAB software. The results of this study are an average of 30 iterations to ensure. In the implementation, the value of the parameter $\theta = 0.4$, $\omega = 3$, the number of particles is 35 and the number of iterations is 50. The results of clustering are shown in Table 1. The number of 41 features are automatically categorized into 7 clusters.

Table 1. Results from the clustering of the Louvain algorithm

Clusters Number	Features	Clusters Number	Features
Cluster 1	3, 4, 6, 10, 12, 20, 21, 34	Cluster 2	7
Cluster 3	1, 11, 13, 14, 15, 16, 19	Cluster 4	5, 9, 17, 18
Cluster 5	28, 35	Cluster 6	2, 8, 22, 23, 24, 25, 26, 29, 31, 32, 33, 36, 37, 38, 39
Cluster 7	27, 30, 40, 41	-	-

The detection of the number of effective features for classification is automatically determined by the PSO algorithm. Figure 4 shows the accuracy of intrusion detection by the proposed algorithm with a number of different features.

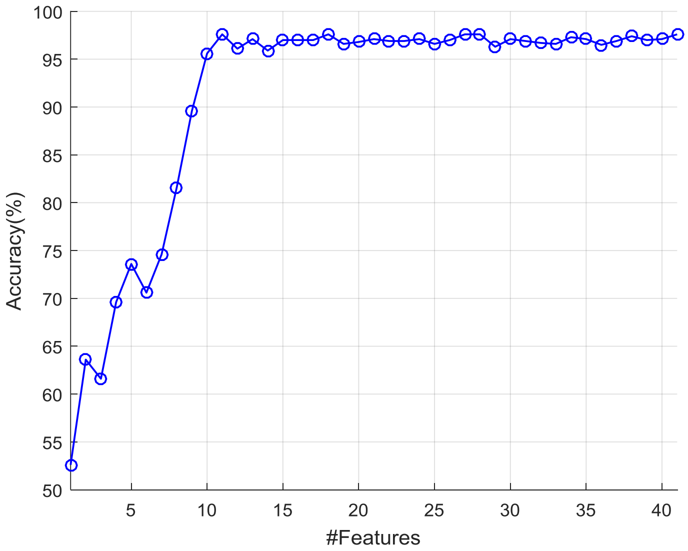


Figure 4. Accuracy of the proposed method with a number of different features

The results show the best classification accuracy with 11 features equal to 97.58% for the NSL-KDD dataset. Table 2 shows the intrusion matrix data of the intrusion detection system. This table lists the number of records for each type of attack along with the number of detections.

Table 2. Clutter matrix for NSL-KDD datasets based on the attack type

Actual Records			**Number of predicted**				
Records Type	Dataset	Number	Normal	DOS	U2R	R2L	Probe
Normal	Train	67343	62219	60	2	11	51
	Test	9710	9517	37	10	104	41
DOS	Train	45927	89	45834	0	0	4
	Test	7458	32	7401	2	3	20
U2R	Train	52	34	0	16	0	2
	Test	200	58	1	133	5	3
R2L	Train	995	27	0	0	958	2
	Test	2754	127	10	4	2606	4
Probe	Train	11656	88	5	0	3	11560
	Test	2421	47	18	9	6	2341

The results based on class show the accuracy of the proposed algorithm in each type of attack. Each of the surveyed attacks has different types. In the following, the performance of the proposed method in detecting sub-attacks is shown. Figure 5 shows the following results of intrusion data attacks for test samples.

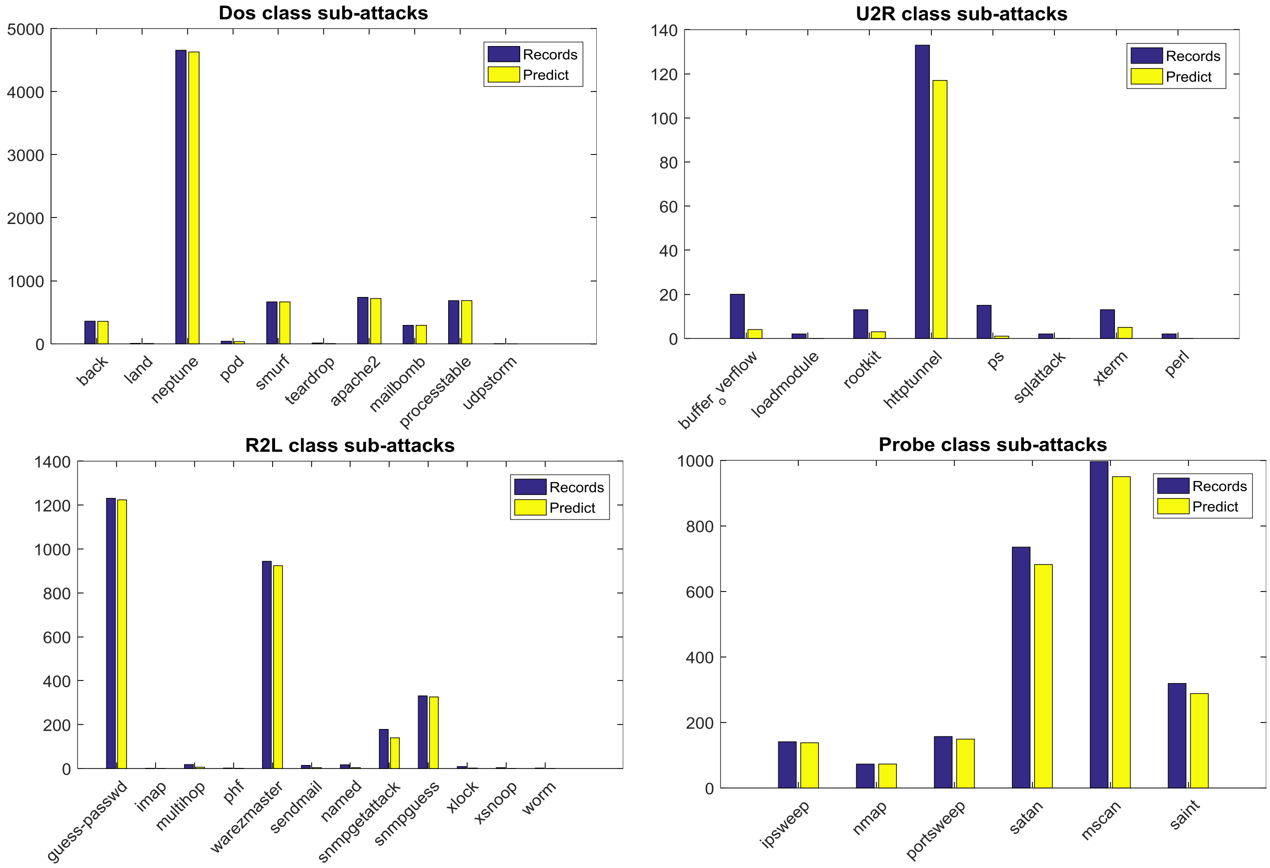


Figure 5. Accuracy of the proposed intrusion detection system by sub-attacks

The results show that the proposed intrusion detection system has high accuracy in detecting DOS type intrusion and its sub-attacks. U2R intrusion also has lower detection accuracy than other attacks. The reason for this is the small number of training samples compared to the test used in the data set.

In order to further evaluate the above approach, the performance of the proposed system is compared with other intrusion detection methods. The results are shown in Table 3. As it turns out, the proposed method is more accurate than other intrusion detection methods for some attacks and provides good accuracy in others. In the provided table, the values of each class is based on the values calculated in the relevant research, so some fields may not be presented in the research. The results show that the proposed method works uniformly on all classes and provides the desired accuracy. The reason for this is the selection of features according to the similarities of different clusters.

Table 3. Comparison of intrusion detection methods by attack type for NSL-KDD dataset

Methods	Normal	DOS	U2R	R2L	Probe	Accuracy
Adaptive IDS (Goyal & Kumar, 2008)	66.51	88.64	66.51	20.88	99.15	75.15
SIPSO (Warsi et al., 2015)	-	99.80	67.50	82.50	99.70	-
Ant Colony (Aghdam & Kabiri, 2016)	97.41	99.78	81.51	**99.17**	74.65	97.32
MARS (Mubarak, 2016)	**99.71**	**99.97**	76.00	98.75	**99.85**	92.75
Proposed Method	97.64	98.01	**82.18**	96.36	97.19	**97.58**

5. Conclusion

In this study, the selection of desirable features with maximum predictability of the target class and the minimum redundancy has been considered. Particles with these properties have a better chance of survival due to the simultaneous use of the two criteria of correlation and redundancy in the evaluation function. The selection of a number of specific properties from each cluster has led to the orientation of the particle swarm optimization algorithm. The proposed restoration operator increases the ability of the algorithm to find the appropriate answer and increases the convergence speed of the particle swarm optimization algorithm. The results of the experiments of the proposed method show an accuracy of 97.58%, which is superior to similar algorithms. Another requirement of intrusion detection systems is to know the optimal feature set for each type of attack. Because in this case, the intrusion detection system will be able to detect only one set of features appropriate to that attack to detect any type of attack. In future research, it is suggested to evaluate a model with this design capability and performance. Another suggestion for future work is to find plug-in features in intrusion detection datasets.

References

Aghdam, M. H., & Kabiri, P. (2016). Feature selection for intrusion detection system using ant colony optimization. *IJ Network Security*, *18*(3), 420-432.

Benaicha, S. E., Saoudi, L., Guermeche, S. E. B., & Lounis, O. (2014). Intrusion detection system using genetic algorithm. 2014 Science and Information Conference,

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1-4). Springer.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, *2008*(10), P10008.

Chae, H.-s., Jo, B.-o., Choi, S.-H., & Park, T.-k. (2013). Feature selection for intrusion detection using nsl-kdd. *Recent advances in computer science*, *20132*, 184-187.

Chen, M.-H., Chang, P.-C., & Wu, J.-L. (2016). A population-based incremental learning approach with artificial immune system for network intrusion detection. *Engineering Applications of Artificial Intelligence*, *51*, 171-181.

Goyal, A., & Kumar, C. (2008). GA-NIDS: a genetic algorithm based network intrusion detection system. *Northwestern university*.

Kenkre, P. S., Pai, A., & Colaco, L. (2015). Real time intrusion detection and prevention system. Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014,

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95-international conference on neural networks,

Kevric, J., Jukic, S., & Subasi, A. (2017). An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Computing and Applications*, *28*(1), 1051-1058.

Mubarak, S. L. (2016). Intrusion Detection System using SVM SOM & NN. *Journal of network and computer applications*, *30*(1), 114-132.

Muda, Z., Yassin, W., Sulaiman, M., & Udzir, N. (2011). Intrusion detection based on K-Means clustering and Naïve Bayes classification. 2011 7th international conference on information technology in Asia,

Pan, S., Morris, T., & Adhikari, U. (2015). Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, *6*(6), 3104-3113.

Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)*, *2*(12), 1848-1853.

Rezaeipanah, A., & Ahmadi, G. (2020). Breast Cancer Diagnosis Using Multi-Stage Weight Adjustment In The MLP Neural Network. *The Computer Journal*.

Sabharwal, C. L., Hacke, K. R., & St. Clair, D. C. (1992). Formation of clusters and resolution of ordinal attributes in ID3 classification trees. Proceedings of the 1992 ACM/SIGAPP Symposium on Applied computing: technological challenges of the 1990's,

Saha, S., Sairam, A. S., Yadav, A., & Ekbal, A. (2012). Genetic algorithm combined with support vector machine for building an intrusion detection system. Proceedings of the International Conference on Advances in Computing, Communications and Informatics,

Theodoridis, S., & Koutroumbas, K. (2009). Feature generation I: data transformation and dimensionality reduction. *Pattern Recognition*, 323-409.

Warsi, S., Rai, Y., & Kushwaha, S. (2015). Selective Iteration based Particle Swarm Optimization (SIPSO) for Intrusion Detection System. *International Journal of Computer Applications*, *124*(17).

Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). Intrusion detection and big heterogeneous data: a survey. *Journal of Big Data*, *2*(1), 1-41.