Received: June 23, 2022. Revised: July 20, 2022. 592

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Feature Selection of The Anomaly Network Intrusion Detection Based on Restoration Particle Swarm Optimization

Mohammad R. Aziz[1]* **Ali Saeed Alfoudi[1,2]**

[1]*College of Computer Science and Information Technology, University of Al-Qadisiyah, Iraq*
[2]*College of Computer Science, Liverpool John Moores University, United Kingdom*
* Corresponding author's Email: mohammad.aziz@qu.edu.iq

Abstract: Intrusion detection systems are vital for detecting networking attacks due to their ability to analyze network data and find different types of attacks. The high-dimensional internet data leads to feature selection becoming a fundamental process in network intrusion detection systems. The current approaches are insufficient to determine the most effective features in the network data due to the nature of intrusion attacks appearance compared to the normal data. Moreover, the wrapper feature selection methods suffer from the search time complexity such as the standard PSO algorithm for feature selection. The standard Particle Swarms Optimization (PSO) algorithm suffers from the stagnation effect in local optima. This paper proposes a new wrapper feature selection model called Restoration Particle Swarms Optimization (RPSO) to select highly relevant feature data taking into consideration the limitation of the premature convergence between the particles which results in a stagnation problem during the iteration for the optimal features. Moreover, we utilize the randomness value to overcome the stagnation problem, reduce data volume and decrease the processing time. The Random Forest algorithm uses to classify the feature selected with our solution. As a result, we consider the NSL-KDD benchmark dataset to evaluate the proposed solution. The experiments show that the performance evaluation achieves high results in general accuracy (85%) compared to standard PSO up to 83.86%. Additionally, the results show that the proposed solution increased the detecting rate of low distributing classes in the training data up to (521 classes) compared with standard PSO by (79 classes).

Keywords: Intrusion detection system, Feature selection, NSL-KDD, Particle swarm optimization, Machine learning.

1. Introduction

Internet services have developed dramatically in recent years, and access to them has become more manageable, reliable, and at low prices [1]. Because of this significant expansion of the internet, security risks, cyber threats and electronic attacks have increased [2].

Intrusion detection systems are the best solutions for cybersecurity threats[3]. Generally, intrusion detection systems(IDS) are categorized into two main types: signature and anomaly based IDS[4]. The first type is signature-based works as antivirus programs work [5]. The signature-based IDS relies on searching the payload of the packages for any malicious program's signature, but it fails to detect Zero-Day attacks despite it has high accuracy in detecting attacks that contain a signature in a database [6]. Anomaly-based intrusion detection systems (A-IDS) learn and build based on networking traffic profiles. This type of IDS distinguishes cyberattacks by analyzing users' behavior and how they interact via the networking. Moreover, A-IDS suffers from many problems, including a high rate of false alerts and takes a long time in training the model [7].

Researchers have utilized machine learning approaches to designing intrusion detection systems to obtain a high detection rate and reduce false alarms to reach a high degree of network security[8]. In machine learning, statistical and mathematical models are used to find and identify patterns in large datasets[9]. A network packet is high-dimensional

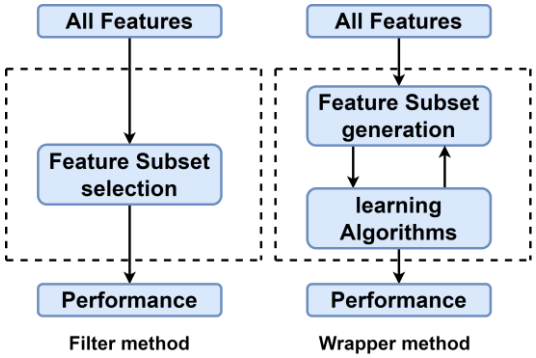

Figure. 1 Filter and wrapper methods feature selection

data and contains various features used in the classification process, but not all of these features are essential to identify whether a packet is normal or an attack [10]. Therefore, choosing highly relevant features increases the detection rate and reduces training data size and execution time complexity because of the large intrusion detection data volumes and their wide dimensions [11, 12].

The high relevant features determining are the main objective of feature selection algorithms, which describe the overall dataset without detracting from the accuracy of the classification model[13]. Moreover, there are two standard feature selection methods (the Filter and wrapper), as shown in Fig.1. The filter methods use fundamental data properties as a score to choose features, and it does not depend on machine learning techniques to decide whether it is selected or not[14]. This score is based on statistical measures such as entropy and Laplacian score. Moreover, these methods are faster in processing time; however, they are suitable only for independent features [15].

The wrapper consists of three feature selection components (Search strategy, prediction function, and fitness function)[16]. The search technique selects the subset of qualities to be analyzed. The prediction function evaluates the performance of the given features against the fitness function using any classification technique[17]. However, the wrapper methods suffer from processing time-consuming in the search strategy. Therefore, metaheuristics methods appear as a promising solution to overcome this limitation.

The Particle Swarm Optimization PSO algorithm is one of the most popular metaheuristic algorithms suitable for feature selection in this field[18]. That is because the PSO finds the optimal subset features faster than the other algorithms. Moreover, the standard PSO suffers from the swarm's premature convergence, which causes stagnation during the iteration for the optimal features[19].

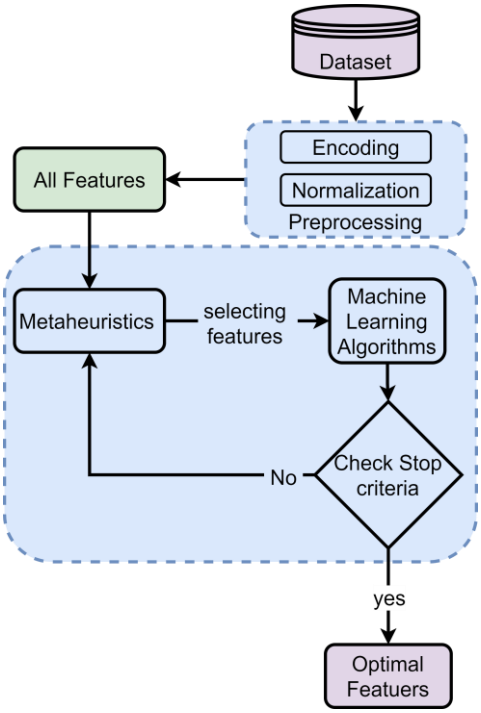

Figure. 2 Feature selection using metaheuristics search

This paper proposes a new wrapper feature selection model called Restoration Particle Swarms Optimization (RPSO) to select a highly relevant feature subset in A-IDS. Furthermore, we enhance the standard PSO by generating a new population randomly if the stagnation accrues in the algorithm iteration. Additionally, we utilize the Random Forest algorithm to classify the feature selected by the proposed RPSO algorithm.

As a result, we consider the NSL-KDD benchmark dataset to evaluate the proposed solution. The experiments show that the performance evaluation achieves high results in general accuracy (85%) compared to related works. Additionally, the results show that the proposed solution increased the detecting rate of low distributing classes in the training data up to (521 classes) compared with standard PSO by (79 classes).

The rest of the article is organized as follows: Section 2 presents the related works, Section 3 provides the proposed solution and section 4 shows the experiment result analysis. Finally, Section 5 discusses the conclusion.

2. Related work:

The previous literature studies discussed many anomaly-based IDS models focusing on feature selection. Additionally, in this section, we described only studies that used the NSL-KDD dataset to evaluate their solution.

Recently many proposed intrusion detection systems were unreliable because the training and testing process was performed on the same partitions of dataset. This approach might achieve biased results. For example, the accuracy of the performance might achieve 99.9% in some cases[20, 21].

Tama *et al.* [20] presented a hybrid intrusion detection system model utilizing a hybrid feature selection approach particle swarm optimization (PSO), ant colony optimization (ACO), and genetic algorithms (GA) to decrease the dimensionality of the dataset. A rotation Pruning Tree was used to choose subgroups of features. The classification accuracy is evaluated using subsampling (Monte-Carlo cross validation) technique. A two-stage meta classifier, incorporating rotating forest and bagging, was used as a classifier. In multi-classification, the suggested model obtained 72.52% performance accuracy, which was considered low detection accuracy.

Wei *et al*. [22] proposed an intrusion detection model based on the deep belief network (DBN-IDS). The GA optimization was utilized to reduce the dataset dimensionality. The GA algorithm's initial value was calculated using the fish swarm technique based on the learning factor and the adaptive weight to find the initial solution. The proposed model achieved 82.36% and 66.25% performance accuracy on the full and reduced versions of the NSL-KDD dataset, respectively. In this study, an optimal network structure is acquired by comparing the DBN structure obtained from five optimization algorithms. This model has high complexity in terms of processing time.

Yang *et al*. [23] proposed a hybrid model based on Deep Belief Networks (MDPCA-DBN). The Modified Density Peak Clustering Algorithm was utilized to find similar features in the dataset to reduce the dimensions of the dataset. the Euclidean distance calculation method of the density peak clustering algorithm (DPCA), and use the kernel function to project the original features into the high-dimensional kernel space to better cluster the complex nonlinear inseparable network traffic data. The proposed model divides the training dataset into small subsets according to the similarity of the features. All subsets are trained their classifier independently. According to the nearest neighbor criterion, the fuzzy membership weights of each test sample in each sub-DBNs classifier are calculated. The output of all subsets classifiers is aggregated based on fuzzy membership weights. This model achieves 82.08% and 66.18% performance accuracy on the full and reduced versions of the NSL-KDD dataset, respectively. The proposed model achieved good results but has high complexity in terms of processing time.

GAR-Forest is a new tree-based ensemble approach presented by Kanakarajan and Muniasamy [24]. The proposed model is combined with the symmetrical uncertainty feature selection approach. The prposed classifier utilized the greedy randomized adaptive search procedure metaheuristic GRASP with annealed randomness to increase the diversity of ensemble. By utilizing the full NSL-KDD dataset, the proposed model achieves a detection accuracy of 77.6038 percent when 23 features are selected. The model, however, has a high false alert rate of 12.2 percent. This model achieved low detection accuracy with high false alert rates.

A two-layer dimension reduction and two-tier classification (TDTC) model for IDS is presented by Pajouh *et al.* [25]. Linear discriminate analysis and component analysis reduce the data dimension and choose highly relevant features. After calculating eigenvectors by Covariance matrix, PCA was utilized to sort the eigenvectors in descending order. A common approach is to rank the eigenvectors from the highest to the lowest and choose the top k eigenvectors based on eigenvalues. A two-tier classification module involving the NB and k-NN classifier is applied to discover potential attacks. The proposed model achieved 84.86 percent performance accuracy when applied on the NSL-KDD dataset. This model achieved good overall accuracy; furthermore, there were no false alarm rates in the results and no detection rates on the attack classes.

Riyadh *et al.* [26] proposed a hybrid intrusion detection system based on machine learning techniques to handle the uncomplete and noisy network data. The DBSCAN was utilized to clean the data from the noise. The K means and K-nearest neighbor algorithms are utilized for dimensional reduction and transform the cleaned data into one-dimensional data. The proposed model utilized the KDD-cup benchmark dataset to evaluate the proposed model.

Krishna & Arunkumar [27] proposed a hybrid intrusion-detection model that combines particle swarms and Gray Wolf optimization algorithms to select highly relevant features. This combination produces a mixed low-level co-evolutionary functionality by merging both variants with low functionalities to improve the computational complexity. The random forest classifier was utilized to predict the intrusion in the dataset.

Table 1. Related work results summary

Proposed solution	Dataset	The results accuracy
TSE-IDS [20]	NSL-KDD UNSW-NB15	72.52 % 88.78%
DBN-IDS [22]	NSL-KDD, KDDTest+ NSL-KDD, KDDTest-21	82.36 % 66.25%
MDPCA-DBN [23]	NSL-KDD, KDDTest+ NSL-KDD, KDDTest-21 UNSW-NB15	82.08% 66.18% 90.21%
GAR-FOREST [24].	NSL-KDD	77.6%
TDTC [25].	NSL-KDD	84.86%
hybrid intrusion-detection model [26]	KDD-cup	98.3%
hybrid intrusion-detection model[27]	NSL-KDD	99.97 %
enhanced firefly optimization [28]	NSL-KDD	96.942%

Alwan et al. [28] proposed an enhanced version of the firefly optimization algorithm by enhancing the original firefly's exploration capabilities. A mutation operation was employed to avoid trapping into local optima. The Naïve Bayes classifier was utilized as a fitness function.

Finally, Table1 presents the summary of the comparison results accuracy of related studies.

3. Proposed method:

The particle swarm algorithm is considered one of the essential evolutionary computational algorithms used for optimization. This algorithm is motivated by the collaborative behavior of individuals living in one environment, such as a flock of birds or fish schooling and a swarm of bees.

The standard PSO algorithm utilizes a swarm of particles to move around the search area to achieve the best solution possible. The PSO creates particles randomly and each particle in the search space updates its movement experience by comparing its status with other particle's movement experiences. The fitness function utilizes to evaluate the particles through two test experiences. The first one compares a particle's experience with itself and the second test compares all particles' experiences in the swarm. These two test experiences call personal best (p-best) and global best (g-best).

The two main parameters of a Particle Swarm are velocity and position. Each particle moves toward the best previous particle's position and global position during each generation. At each iteration, a new velocity value for each particle is calculated based on the current velocity, distance from the previous best position and distance from the global best position. Then, the new velocity value calculates the particle's next position in the search space. This process iterates many times or until a minimum error is achieved.

The implementation of the PSO algorithm depends on the following equations to calculate the velocity and position of each particle. The updated particle i calculates according to Eq. (1) as illustrated below[29]:

$$V_i^d(t+1) = \omega(t)V_i^d(t) + c_1 r_1 \left(pbest_i^d - x_i^d(t)\right) + c_2 r_2 \left(gbest^d - x_i^d(t)\right) \quad (1)$$

Where $r_1 r_2$ are random variables in the range [0, 1], $c_1 c_2$ are positive constants (acceleration coefficients), and ω is the inertia weight. The V_i^t , X_i^t indicate the velocity and particle position at iteration t in dimension, respectively. The PSO uses Eq. (2) to update the value of the particles (candidate solution) [29]:

$$x_i^{t+1} = x_i^t + V_i^{t+1} \quad (2)$$

Where x_i^t is referred to old particle value, and x_i^{t+1} is referred to new particle value.

The use of the particle Swarm algorithm in the feature selection process depends on the principle of the algorithm's ability to select the features highly relevant to the class label of the data. The wrapper is one of the most important models that rely on trial and error in selecting features. Optimization algorithms are used to increase the stability of this model.

As one of the stochastic search algorithms, PSO has a significant limitation. The limitation is the premature convergence of the swarm. Therefore, the PSO finds the solution faster than other algorithms, but the solution's quality cannot be improved because the number of iterations increases.

Moreover, the PSO suffers from stagnation problems. Typically, the stagnation acquires after the midpoint of the search period. For the PSO, to reduce this effect, we need to increase the randomness of PSO during the implementation after the midpoint of the search period. Therefore, adding randomness at this stage is necessary to improve the performance of PSO.

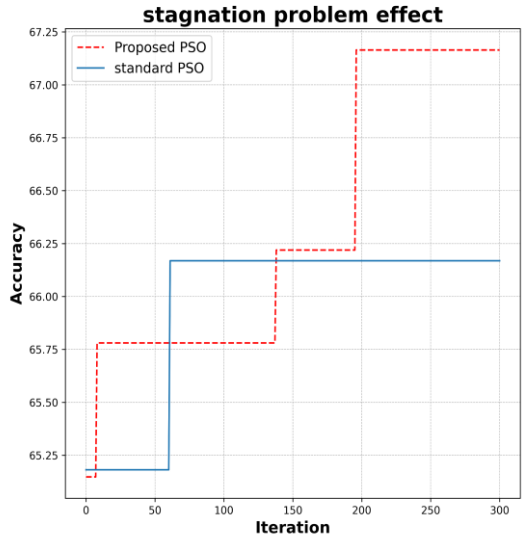

Figure. 3 Stagnation problem effect

To overcome these limitations, the PSO needs to improve two parameters: exploration and exploitation. Exploration depends on increasing the randomness of the search space. At the same time, exploitation depends on directing the solution towards the best solutions to reduce scattering in the algorithm.

The proposed Restoration Particle Swarms Algorithm RPSO improves the exploration and exploitation to generate new populations for feature selection while keeping the best solutions obtained from the previous stages. This improvement leads to increasing exploitation and generating radically new optimal features to increase the randomness in the search space after the algorithm reaches a stagnation stage. The stagnation is when the algorithm cannot change the search flow by changing the threshold, as shown in Fig. 3. This changing of the feature population will increase the possibility of generating new optimal features that have better capabilities of selecting features.

Algorithm (1) illustrates the proposed RPSO algorithm. The RPSO regenerates new populations when the algorithm cannot find particles results better than the global best particle. At the same time, it keeps the value of (Pbest) and (Gbest) that achieves from the last search iteration steps. As result of this step, the exploitation will increase when the algorithm cannot change the search flow by changing the threshold.

We utilize the accuracy metric to evaluate our proposed solution performance based on the confusion matrix, as illustrated in Fig. 4. The confusion matrix provides the best details for

ALGORITHM 1: RPSO ALGORITHM

Input: *Initialization Parameters*
Output: *Optimal Features*

$\Theta \leftarrow 0$
$\sigma \leftarrow 0$
while *iter* $\leq$ *max iteration* **do**:
  Update population
  $local_{best} = f(best)$
  If $Global_{best} < local_{best}$ **then:**
    $Global_{best} \leftarrow local_{best}$
    $\sigma \leftarrow 0$
  Else
    $\sigma \leftarrow \sigma + 1$
    $\sigma_2 \leftarrow \sigma_2 + 1$
  If $\sigma < 2$ *x population size* **then:**
    $\sigma \leftarrow 0$
    If $\sigma_2 <$ *population size /2* **then:**
      $\sigma_2 \leftarrow 0$
      Restart population with keeping $Global_{best}$, $local_{best}$
      $\Theta \leftarrow$ *random(max eqx,min eqx)*
return *Optimal Features*

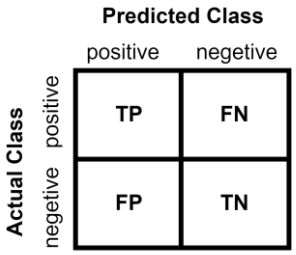

Figure. 4 Confusion matrix

classification results based on Eq. (3). In the Eq. (3), the TP (True Positive) refers to positive instances predicted as positive, FP (False Positive) to negative instances predicted as positive, TN (True Negative) to negative instances predicted as negative, and FN (False Negative) to positive instances predicted as negative.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} \quad (3)$$

4. Experimental results and discussion:

In our experimental results, we utilize Processor Intel ® Core (™)i7-11800H CPU, Ram 64 GB, Storage 1 TB, Freq. 2.3 GHz. As hardware, as *Operating System* Windows 11 64bit, and PyCharm community 2019.3. as *Programming Language*.

Moreover, we consider the NSL-KDD as a benchmark dataset to evaluate our model. The NSL-KDD is the new version of the KDD dataset, where the dataset contains 41 features used to define each

connection in the dataset with a labeling feature. The features explain each connection detail in the dataset and the labeling feature identifies whether the connection is normal or attack with the attack type. In addition, two versions of this data are available for training and testing. The complete version for the training phase is called (KDDTrain+, KDDTest+), while the other version calls (KDDTrain+_20Percent, KDDTest-21), representing 20 % of the dataset. Each one divides into training data and testing data separately[30]. The 20% version is considered more complex than the full version because of the low presence of classes (R2L and U2R). The complexity of the dataset refers to the imbalance when the distribution classes are not uniform. The imbalance dataset causes challenges in guessing the low distribution classes compared to high distribution classes.

Most IDS research has focused on the general performance accuracy without focusing on the detection rate of each class, such as [5, 31-33]. Moreover, other papers combine the original training and testing dataset into one data and then break it into new training and testing data[20, 21]; this action makes overfitting and bias in the model results. Moreover, we focus on the detection rate of each attack class, specifically the low distribution classes, i.e. (U2R and L2R). Moreover, we applied the training and testing phases to the original training and testing with full and reduced dataset versions without combining the original training and testing dataset into one data and then breaking it into new training and testing data; this action makes overfitting and bias in the model results.

The preprocessing data is the first phase to perform the NSL-KDD dataset into the analysis-ready format, as shown in Fig. 5.

To do this, we implement several steps in the preprocessing phase: (i) import training and testing data into python IDE; (ii) Apply encoding (digitization) on categorical data as shown in Fig. 6 ; (iii) and apply normalization on numerical data

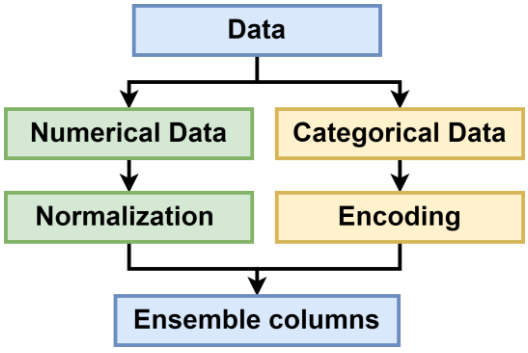

Figure. 5 Preprocessing phase

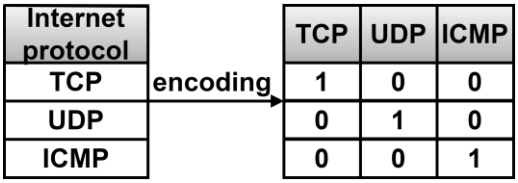

Figure. 6 Encoding categorical data illustration

Table 2. Statistical results of NSL-KDD

	Best	Worst	Mean	STDEV
Standard PSO 20%	66.9	64.7	65.8	0.6
Proposed PSO 20%	68.4	66.05	68	0.48
Standard PSO FULL	83.86	81.7	82.7	82.7
Proposed PSO FULL	85	82.8	84.8	0.47

values in the range [0,1] by utilizing the Z1-Score. The Z1-Score describes in Eq.4, where x, μ, σ reprsent the single value, mean and standard deviation in the specific feature. Moreover, this step is essential to eliminate the biased of the biggest values.

$$Z1_{score} = \frac{x-\mu}{\sigma} \quad (4)$$

Both standard and proposed PSO algorithms are implemented with three classifiers (KNN, SVM, RF) as objective functions were executed individually on the two versions of the dataset 40 times to calculate the results each time.

Comparing the standard PSO with our proposed algorithm, we note that the standard PSO suffers from scattering while our RPSO shows more stability during the running time, as shown in Fig. 7 and 8. Moreover, the standard deviation (STDEV) for the PSO is higher than the RPSO due to the fluctuation of PSO results, as illustrated in Table 2. In the Table 2, the STDEV of the proposed RPSO achieves (0.51) and (0.47) for both versions of the dataset, while the original PSO achieves (0.6) and (0.48).

The standard PSO with RF algorithm detects 41 classes of R2L attack and two classes of U2R attack while applying the proposed RPSO with RF algorithm detects 242 classes of R2L and 21 classes of U2R on the NSL-KDD reduced version. Moreover, applying the PSO and RPSO with RF on the full dataset version, we noticed that the standard PSO algorithm detects 79 classes of R2L attack and seven classes of U2R attack. At the same time, the proposed RPSO detects 521 classes of R2L and 28 classes of U2R.

According to the above mentioned, The RPSO algorithm improves the performance of the RF machine learning algorithm by increasing the general accuracy in predicting attacks.

The proposed and standard PSO algorithms were tested with other machine learning algorithms such as SVM and K-NN. The implementation with SVM achieves the worst results with the proposed and standard PSO because the SVM algorithm is not suitable for high-dimensional data. In contrast, the KNN algorithm achieved better results than SVM, as illustrated in Fig. 9 and 10.

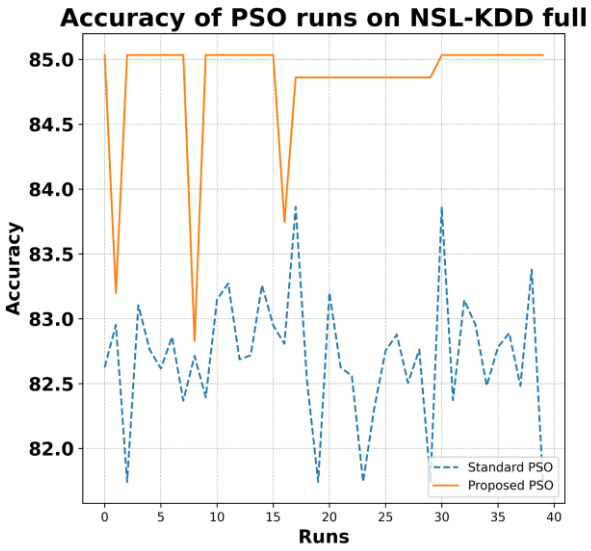

Figure. 7 Result of standard and proposed PSO on NSL-KDD full dataset

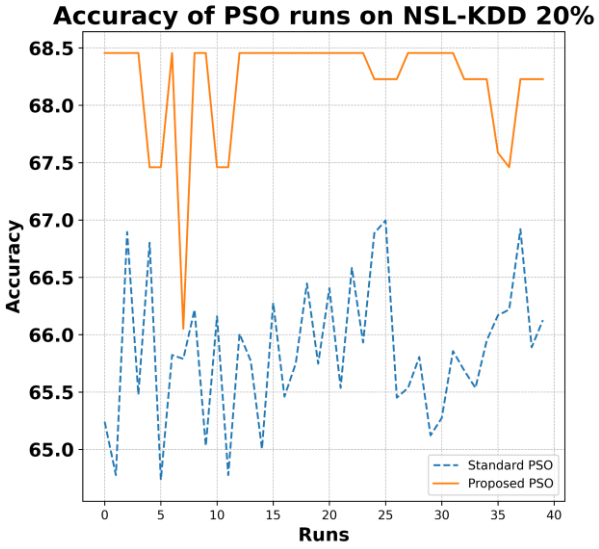

Figure. 8 Result of standard and proposed PSO on NSL-KDD 20%

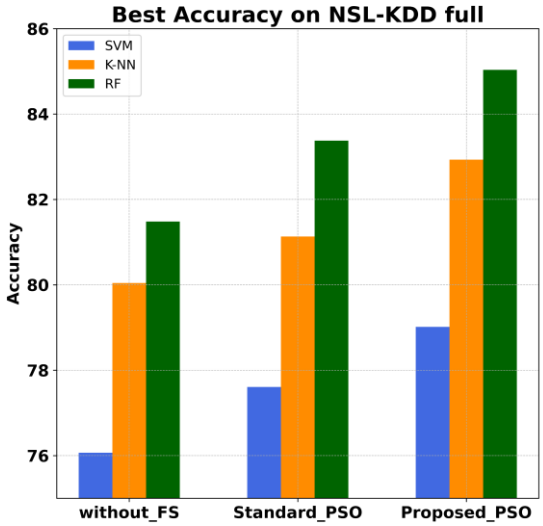

Figure. 9 The best classification accuracy on NSL-KDD full

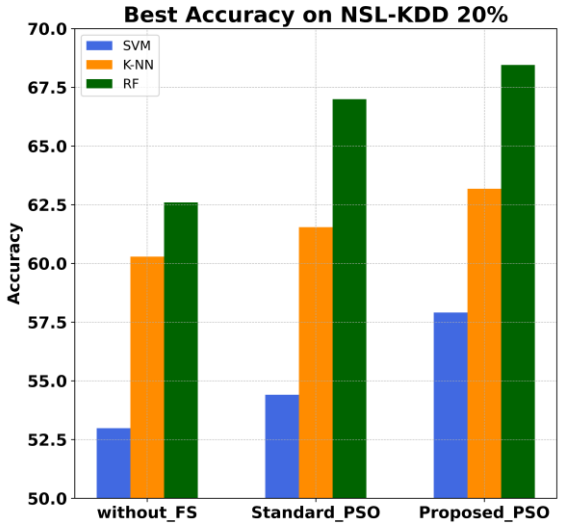

Figure. 10 The best classification accuracy on NSL-KDD full

Table 3. Five-classes classification comparison result with the existing approaches

Model	ACC full dataset	ACC reduced dataset
Hybrid IDS [20]	72.52 %	-
DBN-IDS[22]	82.36 %	66.25%
MDPCA-DBN [23]	82.08%	66.18%
GAR-FOREST [24]	77.6%	-
TDTC [25]	84.86%	-
Proposed RPSO	85%	68.4%

As shown in Table 3, our proposed solution (RPSO) achieves high overall accuracy in both

versions of the NSL-KDD benchmark dataset compared to the related work. Moreover, the proposed solution utilizes the PSO algorithm, which has little processing time complexity resulting in our proposed model having less time complexity.

5. Conclusion:

In fact, the dramatic increase in internet usage has a significant problem in terms of cybersecurity. The intrusion detection systems appear as an optimal solution of cybersecurity threats, but the high dimension of the network payload is a critical issue in terms of intrusion detection. Feature selection is one of the best solutions to overcome this issue. The particle swarm algorithm is considered one of the most popular feature selection techniques in IDS. However, it suffers from instability and the inability to obtain optimal solutions while increasing the number of attritions. In this paper, we proposed Restoration Particle Swarms Algorithm RPSO by utilizing the randomness value to overcome the stagnation problem, reduce data volume and decrease the processing time. The Random Forest algorithm uses to classify the feature selected with our solution. Moreover, the results show high stability, excellent detection rate, and improved general accuracy.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

In this article, each author contributes as follows: conceptualization and methodology, M. Aziz, A. Alfoudi; data validation/curation, M. Aziz, A. Alfoudi; result analysis and conclusion, M. Aziz, A. Alfoudi; writing, original draft preparation, M. Aziz; review and editing, A. Alfoudi.

References

[1] M. Trevisan, D. Giordano, I. Drago, M. M. Munafo, and M. Mellia, "Five Years at the Edge: Watching Internet from the ISP Network", *IEEE/ACM Trans. Netw.*, Vol. 28, No. 2, pp. 561-574, 2020.

[2] S. Hore and K. Raychaudhuri, "Cyber Espionage—An Ethical Analysis", *Adv. Intell. Syst. Comput.*, Vol. 1189, pp. 34-40, 2021.

[3] T. Daniya, K. S. Kumar, B. S. Kumar, and C. S. Kolli, "A survey on anomaly based intrusion detection system", *Mater. Today Proc.*, 2021.

[4] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine Learning and Deep Learning Methods for Cybersecurity", *IEEE Access*, Vol. 6, No. c, pp. 35365-35381, 2018.

[5] S. Sarvari, N. F. M. Sani, Z. M. Hanapi, and M. T. Abdullah, "An Efficient Anomaly Intrusion Detection Method with Feature Selection and Evolutionary Neural Network", *IEEE Access*, Vol. 8, pp. 70651-70663, 2020.

[6] B. Silva, R. Silveira, M. S. Neto, P. Cortez, and D. Gomes, "A comparative analysis of undersampling techniques for network intrusion detection systems design", *J. Commun. Inf. Syst.*, Vol. 36, No. 1, pp. 31-43, 2021.

[7] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu, and J. Li, "Performance comparison and current challenges of using machine learning techniques in cybersecurity", *Energies*, Vol. 13, No. 10, 2020.

[8] S. J. J. Genereux, A. K. H. Lai, C. O. Fowles, V. R. Roberge, G. P. M. Vigeant, and J. R. Paquet, "MAIDENS: MIL-STD-1553 Anomaly-Based Intrusion Detection System Using Time-Based Histogram Comparison", *IEEE Trans. Aerosp. Electron. Syst.*, Vol. 56, No. 1, pp. 276-284, 2020.

[9] R. R. Nuiaa, A. H. Alsaeedi, S. Manickam, and D. E. J. A. Shammary, "Evolving Dynamic Fuzzy Clustering (EDFC) to Enhance DRDoS_DNS Attacks Detection Mechnism", *Int. J. Intell. Eng. Syst.*, Vol. 15, No. 1, pp. 509-519, 2022, doi: 10.22266/ijies2022.0228.46.

[10] L. Sun, Y. Zhou, Y. Wang, C. Zhu, and W. Zhang, "The effective methods for intrusion detection with limited network attack data: Multi-task learning and oversampling", *IEEE Access*, Vol. 8, pp. 185384-185398, 2020.

[11] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective", *J. Big Data*, Vol. 7, No. 1, 2020.

[12] L. S. Reddy and S. Vemuru, "A survey of different machine learning models for static and dynamic malware detection", *Eur. J. Mol. Clin. Med.*, Vol. 7, No. 3, pp. 4299-4308, 2020.

[13] E. A. Gargari, M. S. Reis, U. M. B. Neto, J. Barrera, and E. R. Dougherty, "A fast Branch-and-Bound algorithm for U-curve feature selection", *Pattern Recognit.*, Vol. 73, pp. 172-188, 2018.

[14] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges", *Cybersecurity*, Vol. 2, No. 1, 2019.

[15] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset", *IEEE Access*, Vol. 8, pp. 32150-32162, 2020.

[16] M. A. Mabayoje, A. O. Balogun, A. O. Ameen, and V. E. Adeyemo, "Influence of Feature Selection on Multi - Layer Perceptron Classifier for Intrusion Detection System", *Comput. Inf. Syst. Dev. Informatics Allied Res. J.*, Vol. 7, No. 4, pp. 87-94, 2016.

[17] A. H. Alsaeedi, A. H. Aljanabi, M. E. Manna, and A. L. Albukhnefis, "A proactive metaheuristic model for optimizing weights of artificial neural network", *Indones. J. Electr. Eng. Comput. Sci.*, Vol. 20, No. 2, pp. 976-984, 2020.

[18] G. Kumar, K. Thakur, and M. R. Ayyagari, "MLEsIDSs: machine learning-based ensembles for intrusion detection systems—a review", *Journal of Supercomputing*, Vol. 76, No. 11. pp. 8938-8971, 2020.

[19] M. H. Nasir, S. A. Khan, M. M. Khan, and M. Fatima, "Swarm Intelligence inspired Intrusion Detection Systems — A systematic literature review", *Comput. Networks*, Vol. 205, No. January, p. 108708, 2022.

[20] B. A. Tama, M. Comuzzi, and K. H. Rhee, "TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System", *IEEE Access*, Vol. 7, pp. 94497-94507, 2019.

[21] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches", *Trans. Emerg. Telecommun. Technol.*, Vol. 32, No. 1, pp. 1-29, 2021.

[22] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, "An optimization method for intrusion detection classification model based on deep belief network", *IEEE Access*, Vol. 7, pp. 87593-87605, 2019.

[23] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks", *Appl. Sci.*, Vol. 9, No. 2, 2019.

[24] N. K. Kanakarajan and K. Muniasamy, "Improving the accuracy of intrusion detection using gar-forest with feature selection", *Adv. Intell. Syst. Comput.*, Vol. 404, pp. 539-547, 2016.

[25] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K. K. R. Choo, "A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks", *IEEE Trans. Emerg. Top. Comput.*, Vol. 7, No. 2, pp. 314-323, 2019.

[26] M. Riyadh, B. J. Ali, and D. R. Alshibani, "IDS-MIU: an Intrusion Detection System Based on Machine Learning Techniques for Mixed Type, Incomplete, and Uncertain Data Set", *Int. J. Intell. Eng. Syst.*, Vol. 14, No. 3, pp. 493-502, 2021, doi: 10.22266/ijies2021.0630.41.

[27] E. S. P. Krishna and T. Arunkumar, "Hybrid Particle Swarm and Gray Wolf Optimization Algorithm for IoT Intrusion Detection System", *Int. J. Intell. Eng. Syst.*, Vol. 14, No. 4, pp. 66-76, 2021, doi: 10.22266/ijies2021.0831.07.

[28] K. M. Alwan, A. H. A. Atta, and H. H. Zayed, "Feature Selection Models Based on Hybrid Firefly Algorithm with Mutation Operator for Network Intrusion Detection", *Int. J. Intell. Eng. Syst.*, Vol. 14, No. 1, pp. 192-202, 2021, doi: 10.22266/ijies2021.0228.19.

[29] S. Talukder, "2nd International Conference on Advanced Machine Learning Technologies and Applications, AMLTA 2014", *Commun. Comput. Inf. Sci.*, Vol. 488, No. February, pp. 1-541, 2014.

[30] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", In: *Proc. of 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, 2009.

[31] Y. Shen, K. Zheng, and C. Wu, "A Hybrid PSO-BPSO Based Kernel Extreme Learning Machine Model for Intrusion Detection", *J. Inf. Process. Syst.*, Vol. 18, No. 1, pp. 146-158, 2022.

[32] Kurniabudi, A. Harris, A. E. Mintaria, Darmawijoyo, D. Stiawan, M. Y. B. Idris, and R. Budiarto, "Improving the anomaly detection by combining pso search methods and j48 algorithm", In: *Proc. of Int. Conf. Electr. Eng. Comput. Sci. Informatics*, Vol. 2020-Octob, No. October, pp. 119-126, 2020.

[33] H. Jiang, Z. He, G. Ye, and H. Zhang, "Network Intrusion Detection Based on PSO-Xgboost Model", *IEEE Access*, Vol. 8, pp. 58392-58401, 2020.