A Novel Intrusion Detection Framework (IDF) using Machine Learning Methods

Shereen H. Ali[*1]

[1] Department of Communication & Electronics Engineering, Delta Higher Institute for Engineering & Technology, Mansoura, Egypt

Emails: drshereen.2016@gmail.com

Abstract

An intrusion detection system is a critical security feature that analyses network traffic in order to avoid serious unauthorized access to network resources. For securing networks against potential breaches, effective intrusion detection is critical. In this paper, a novel Intrusion Detection Framework (IDF) is proposed. The three modules that comprise the suggested IDF are: (i) Data Pre-processing Module (DPM), (ii) Feature Selection Module (FSM), and Classification Module (CM). DPM collects and processes network traffic in order to prepare data for training and testing. The FSM seeks to identify the key elements for recognizing DPM intrusion attempts. An Improved Particle Swarm Optimization is used (IPSO). IPSO is a hybrid method that uses both filter and wrapper approaches to generate accurate and relevant information for the classification step that follows. Primary Selection Phase (PSP) and Completed Selection Phase (CSP) are the two consecutive feature selection phases in IPSO. PSP employs a filtering approaches to quickly identify the most significant features for detecting intrusion threats while eliminating those that are redundant or ineffective. In CSP, the next level of IPSO, this behavior reduces the computing cost. For accurate feature selection, CSP uses Binary Particle Swarm Optimization (Bi-PSO) as a wrapper approach. Based on the most effective features identified by FSM, The CM aims to identify intrusion attempts with the minimal processing time. Therefore, a K-Nearest Neighbor KNN classifier has been deployed. As a result, based on the significant features identified by the IPSO technique, KNN can accurately detect intrusion attacks with the least amount of processing time. The experimental results have shown that the proposed IDF outperforms other recent techniques using UNSW_NB-15 dataset. The accuracy, precision, recall, F1score, and processing time of the experimental outcomes of our findings were assessed. Our results were competitive with an accuracy of 99.8%, precision of 99.94%, recall of 99.85%, F1-score of 99.89%, and excursion time of 59.15s when compared to the findings of the current works.

Keywords: Intrusion Detection System; Machine Learning; Feature Selection; Particle Swarm Optimization.

1. Introduction

There has been a considerable rise in the number of malicious actions in recent decades as a result of the extensive use of computers and networks as well as the introduction of emerging technologies like internet of things, big data, and cloud computing [1,2]. Network resources need to be protected from cyberattacks, and intrusion detection systems (IDS) are essential for cybersecurity to achieve a strong defence against cyberattacks [3]. IDS helps to discover, describe, and realize anomalous actions prompted by attackers in networks and computer systems [4,5].

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

By memorizing features beforehand, machine learning techniques can be utilized for prediction and categorization [6,7]. IDS can be classified into supervised and unsupervised categories according on the classifier's training strategy; Unsupervised learning is the process of learning training set without being able to uncover the structure information included in the training set. Supervised learning learns annotated training set as much as possible in attempting to predict data from beyond training set [8 ,9]. KNN demonstrates exceptional aptitude and high reliability in the solution of challenging model categorization issues. KNN is a highly efficient and quick approach [10]. Due to this, it is preferable to categorize and evaluate IDS with high classification accuracy and few errors. It might even get trapped. KNN entrapment could occur for a variety of reasons. Several of them are caused by the properties of KNN, whereas others are brought on by the setting and the categorization issue it addresses. The three main causes of KNN entrapment are (i) the presence of outliers in training instances, (ii) the fact that KNN is a distance-based classifier and uses other criteria to make a judgment, and (iii) the fact that its effectiveness is dependent on the value of K.

In this paper a novel Intrusion Detection Framework (IDF) to detect intrusion attacks is introduced. IDF consists of three main modules, which are; (i) Data Pre-processing Module (DPM), (ii) Feature Selection Module (FSM) and Classification Module (CM). During DPM, the network traffic is collected and processed in order to prepare data for training and testing. During FSM, input features are extracted from the input dataset, and then meaningful features are picked from those extracted features using improved Particle Swarm Optimization (IPSO). IPSO is a novel proposed method that combines filter and wrapper methods. It consists of two stages: (a) Primary Selection Phase (PSP), which applies a variety of filter methods, and (b) Completed Selection Phase (CSP), which incorporates Binary Particle Swarm Optimization (Bi-PSO) as a wrapper method. IPSO intends to take the benefits of both filter and wrapper approaches in order to overcome their limitations. Actually, filter methods can provide quick selection, but they can't deliver good performance if they ignore feature requirements. However, because it is dependent on feature dependencies and interaction with the applied classifier, Bi-PSO as a wrapper approach can provide accurate detection, but it cannot provide quick selection. As a result, IPSO can pick the most meaningful subset of features because (i) it uses filter approaches to give quick selection, (ii) it uses wrapper methods to provide accurate selection, and (iii) it incorporates feature dependencies and connections with the classifier. On the other hand, during the CM, fast and accurate detection intrusion attacks based on the selected features is provided by the KNN classifier. It considers the K closest neighbours to the tested item in the feature space. Recent intrusion detection techniques were compared to the suggested IDF. According to the findings of the experiments, IDF outperforms all competitors because it introduced the highest detection accuracy. The paper is structured as follows: Section 2, addresses the relevant research on intrusion detection techniques. The suggested Intrusion Detection Framework is explained in depth in section 3. In Section4, the experiments are discussed, and their findings are investigated. In Sect. 5, the paper is concluded.

2. Related Work

The most recent IDS studies are featured in this section. In [11], an effective decision tree-based network intrusion detection method with improved data efficiency has been presented. To improve the data performance and facilitate appropriate training, network data pre-processing and feature selection relying on entropy are notably done. A decision tree classifier is then constructed for accurate intrusion detection. With the CICIDS2017 dataset, the presented method has a 98.80% overall accuracy.

In [12], a hybrid paradigm with two phases has been suggested. a filter-based feature selection approaches has been utilized in the first stage to minimise the dimensionality of the input data. On the other hand, the deep learning model for classification has been deployed after obtaining the ideal feature subset and were able to boost accuracy while reducing processing time. With the NSL-KDD dataset, the suggested paradigm has 99.73% overall accuracy.

In [13], a robust intrusion detection framework with five stages; pre-processing stage, autoencoder stage, database stage, classification stage, and feedback stage was suggested. The autoencoder stage

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

compresses the data processed by the pre-processing stage to create a reduced-dimensional feature, and the classification result is produced by the classification stage. Each traffic's compressed features are saved in the database stage, where they can be restored to the entire traffic for investigations and post-event assessment as well as used for training and validating the classification stage. Simulated network traffic from the CICIDS2017 dataset was used to assess the performance of the proposed framework. According to the experimental findings, binary classification is more accurate than earlier studies, and high accuracy was attained for the recovered traffic.

In [14], Deep reinforcement learning and recursive feature reduction are the foundations of the network intrusion recognition system reported in this study. The suggested model chooses the best set of features using the recursive feature reduction technique, uses a neural network to extract feature data from it, and deep reinforcement learning to create a classifier to identify network intrusions. The CSE-CIC-IDS2018 dataset, which contains a substantial quantity of actual network traffic, has been used to assess the model's performance. The results of the experiments show that the proposed model is capable of choosing the best set of features, eliminating about 80% of noisy data, and then using deep reinforcement learning to identify the chosen features to improve the performance of the intrusion detection model for identifying intrusion attempts.

3. The Proposed Intrusion Detection Framework (IDF)

Figure. 1 illustrates the proposed Intrusion Detection Framework (IDF) which is composed of three modules namely: (i) Data Pre-processing Module (DPM), (ii) Feature Selection Module (FSM), and (iii) Classification Module (CM). During the next sections, the three modules of the proposed IDF will be explained in more details.

3.1. Data Pre-processing Module (DPM)

DPM gathers and modifies network activity in order to make it suitable for use in training and testing. The required dataset can be gathered using any packet filtering technique. The data obtained is then saved in a log file or a database. The data is then subjected to additional analysis in order to be ready for use in the following steps. Data analysis consists of three steps: (i) data reduction that erases duplicated instances from the dataset; (ii) attack categories transformation, that allocates each attack type to its basic attack category; and (iii) data normalization [15], which converts non-numeric data elements into a standardized numeric representation.

3.2. Feature Selection Module (FSM)

The choice of the best features for intrusion detection via FSM in the proposed IDF is the key challenge. In reality, since doing so can reduce the detection model's accuracy, it is crucial to exclude the minimally impacted features from the dataset. In order to enhance the identification model's performance and make it a quicker and more efficient model, feature selection should be done before learning the identification model. First, intrusion features from the input dataset should be retrieved, and then feature selection can be done on such extracted features to choose the most useful features.

This section introduces the Improved Particle Swarm Optimization (IPSO) process as a novel feature selection method. In order to efficiently and precisely choose the primary subset of features that incorporates utmost effective features for intrusion detection, IPSO is a fusion methodology that combines filter and wrapper techniques. It basically consists of two phases; the Primary Selection Phase (PSP) and the Completed Selection Phase (CSP), which use PSO as a wrapper approach to precisely choose the optimal set of features. PSO can choose useful features, but it has a long calculation time and its completion is highly reliant on the starting population of the particles in the swarm. Because of this, the primary goal of PSP is to establish the starting population of PSO by employing the outcomes of rapid selection techniques in PSP as a starting population in CSP in order to shorten PSO's computational time and enable it to pick an ideal subset of features. The effectiveness of the detection model for intrusion is then enhanced by using the optimum subset of features. Although PSO was mainly developed to address issues in the discrete numbers search area, several

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

optimization issues, such feature selection, also arise in binary search area [16, 17]. To address the finite optimization issues, PSO is altered to become Binary PSO (Bi-PSO). The sigmoid transfer function, which converts the velocity's value from the continuous search area into finite search area, is actually how Bi-PSO expanded the classic PSO.

This transformation allows velocity to represent the likelihood that a particle in the position vector will be equal one. Therefore, particle position, best position, and global best position in the swarm can only have (0 or 1) values while particle velocity in Bi-PSO is still updated in the same way as in the classic PSO. Bi-PSO can effectively choose the most important features for intrusion detection in the binary area, but because it starts the population of the particles in the swarm so slowly and randomly, it is difficult to achieve converge. As a result, IPSO is offered as an innovative selection procedure to quickly and effectively choose the best features for intrusion detection by exploiting the

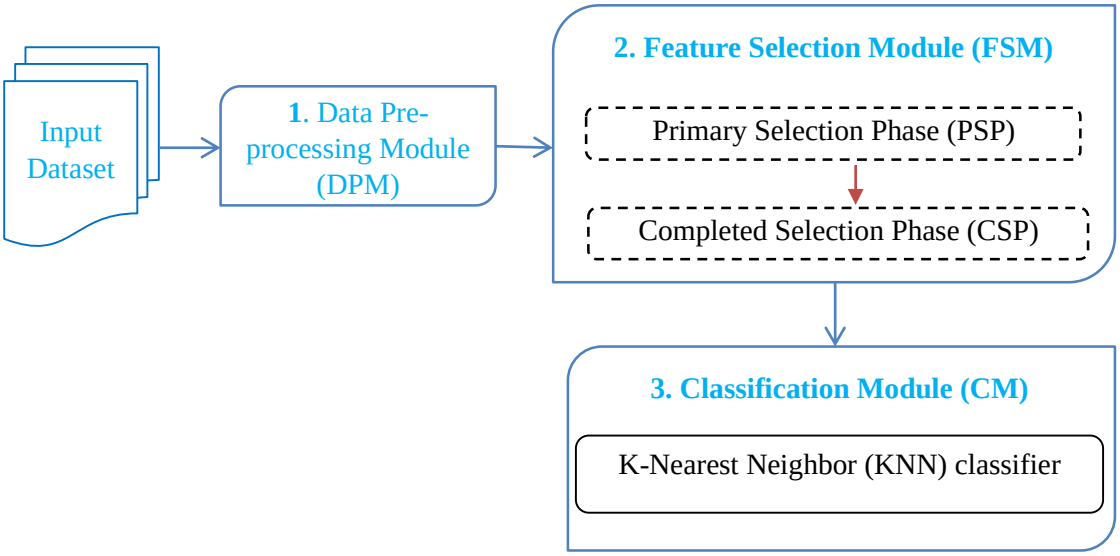

Figure1: The proposed Intrusion Detection Framework

advantages of Bi-PSO method and addressing its drawbacks. Prior to beginning to perform the BiPSO in CSP, the PSP filter techniques are used to create the number of particles and corresponding initial values in the original population of the swarm.

First, the dataset for the intrusion should be supplied to PSP following the extraction of features in DPM so that 'x' filter techniques can be applied to it in concurrently. Following that, CSP will receive the output of these filtering techniques to create the initial population of Bi-PSO. Second, until a termination condition is met, Bi-PSO repetitions will be carried out. In the end, the best subset of features given by the swarm's global best position should be evaluated using a classifier like Naive Bayes [18].

According to its fitness value, Bi-PSO is an optimization method with biological inspiration that was inspired by the social behaviour of fish schooling or flocking birds to provide the optimum solution to the optimization issue [16, 17]. Therefore, Bi-PSO can offer close to ideal solutions for an optimization problem's fitness function. Bi-PSO eventually starts with a swarm (Sm), which is a group of particles.

Each particle in a Bi-PSO search space (i.e., a subset of effective features) indicates a potential solution (i.e., a subset of informative features) in an e-dimensional search area (i.e., e, "the number of extracted features from the input dataset"). As a result, each particle indicates a set of features as a binary string with a length equivalent to the number of features in the input dataset. Particle bits can have a value of 0 or 1. One indicates the pick of the j^{th} feature, while zero indicates the exclusion of the j^{th} feature from the specific subset. In CSP, "x" particles are depicted as a Swarm (Sm), and the

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

fitness function of Bi-PSO is then used to evaluate each particle's fitness level based on a classifier accuracy value. Each particle's fitness value can be determined via (1).

$$\text{Fit}(P_i) = \text{Accuracy}(P_i) \quad (1)$$

Where Accuracy (P_i) is a measure of how accurately a subset of features in an ith particle can be classified. The method seeks out the ideal particle in order to maximize Fit (P_i). Due to fitness values for the particles in Sm, both $P_{Personal}$ and P_{Global} in each particle memory will be updated utilizing (2) and (3) [10].

$$P_{Personal}(P_i) = P_{P_i} = \begin{cases} P_i & if\ (fit(P_i) > fit(P_{P_i}) \\ P_{P_i}\ otherwisw \end{cases} \quad (2)$$

$$P_{Global} = P_G = \begin{cases} P_{P_i} & if\ (fit(P_{P_i}) > fit(P_{P_{i+1}}) \\ P_{P_{i+1}} & otherwise \end{cases} \quad (3)$$

Where $P_{Personal}(P_i)$ denotes the optimal solution for each ith particle and P_i denotes the ith particle's current position. Furthermore, P_{Pi}denotes the ith particle's optimum personal position. Fit (P_i) indicates the ith particle's fitness rating based on its present position. Fit (P_{Pi}) represents the ith particle's fitness rating due to its optimal position.P_{Global} is the optimal particle in swarm (Sm) and Fit (P_{Pi}+ 1) represents the fitness rating of the (i + 1)thparticle due to its optimal position. P_{Pi}+ 1 denotes the personal optimal position of (i + 1)th particle. Moreover,$P_{Personal}$ and $P_{Global}$are utilized to modify every particle's velocity VP_i in the next iteration (t + 1) via (4) [10].

$$VP_i\ (t + 1) = w * VP_i\ (t) + (c_1 r_1(P_{Pi} - P_i(t))) + (c_2 r_2(P_G(t) - P_i(t))) \quad (4)$$

Where t denotes the current iteration and VP_i (t + 1) denotes the velocity of ith particle at the next iteration. VP_i (t) is the velocity of ith particle at the current iteration and P_i (t) denotes the personal best position of ith particle at the current iteration; $P_{personal}(P_i)$. Moreover, P_G(t) denotes the global best position in the swarm (Sm) at the current iteration; P_{Global} . P_i(t) denotes the current position of ith particle at the current iteration. w is the inertia weight; w ∈ [0.9-1.2] [10]. w is employed to influence how the history of prior velocities affects the existing velocities. c1 and c2 are the cognitive and social acceleration constants; c_1, c_2 ∈ [2-4]. Additionally, r_1 and r_2 are uniformly distributed random numbers in the range [0,1]; r_1, r_2 ∈ [0,1]. As a result, three key terms determined the particle's modified velocity VP_i (t + 1). The first term is w ∗VP_i (t) as a current motion term, the second term is c_1 r_1 (P_{Pi} (t) - P_i (t)) as a cognitive term, and the third term is c_2 r_2 (P_G(t) - P_i (t)) as a social term. The particle velocity can be used to show the probability distribution with the primary function of producing the particle position at random after computing the velocity VP_i for each particle in Sm. In order to determine a new particle position based on binary values, the sigmoid function is utilized, and the particle position is altered as a result via (5).

$$P_i^j\ (t + 1) = \begin{cases} 0\ if\ rand(0,1) \geq sig(VP_i^j) \\ 1\ otherwise \end{cases} \quad (5)$$

Where P_i^j(t + 1) denotes the value of ith particle at jth position in the next iteration t + 1. In other words P_i^j(t + 1) denotes to the value of jth feature in ith particle; j = 1,2, 3, ..,e. rand(0,1) a value chosen at random between [0, 1]. As well as, sig (VP_i^j) is the sigmoid transfer function that denotes likelihood that bit j will take a value of 0 or 1. Sig (VP_i^j) is evaluated by via (6).

$$sig(VP_i^j) = \frac{1}{1+e^{-VP_i^j}} \quad (6)$$

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

Every particle is assessed using the fitness function in Sm based on its new position $P_i(t + 1)$ in (1). Afterwards, these computations are carried out until all of the generations have been calculated. The process ends when the best particle from the entire swarm P_{Global} is output. The best features for intrusion detection are all features represented by one in this particle. Six distinct features will be selected as the optimal subset of features after the IPSO method has been applied to the input dataset containing the features. Algorithm 1 demonstrates the systematic process of the IPSO approach utilizing the "x" filter approaches. Suppose that PSP employs three filter approaches: Information gain [19], Chi square [20], and Fisher score [21] to demonstrate the concept. Moreover, take into account that the input dataset has six features (e = 6); FS = {f_1, f_2, f_3, f_4, f_5, f_6}. It is assumed that the subset of features chosen using IG, CHI, and F after implementation on the dataset are; {f_1, f_3, f_5 , f_6 }, {f_3 , f_4 , f_6 }, and {f_1 , f_2 , f_3 , f_4 , f_6 } sequentially. Therefore, in the initial swarm (Sm) of Bi-PSO in CSP, these three subsets of features are used as three particles (P_1, P_2, and P_3). Then, based on many of the criteria in Table 1, BiPSO is performed. These presumptions assume that Bi-PSO is executed over two iterations, resulting in a new swarm with updated values for the three particles; P_1 = {0,1,1,1,1,0}, P_2 = {1,1,0,1,1,1}, and P_3 = {1,1,1,0,0,1}. P_3 is the particle in the swarm P_{Global}that offers the best subset of features after P_1 and P_2 have been evaluated, and P_3 has been determined to obtain the maximum fitness value. Last but not least, the most impacted features in the input dataset are {f_1, f_2, f_3, f_6}.

Table 1: The presumptions for using Bi-PSO in FSM

No.	Assumption	Value
1	No. of generations to process	2
2	Swarm size (no. of particles)	3 (i.e., no. of filter approaches in PSP)
3	Initial Ppersonal (*Pi*)	P_i
4	Initial P_{Global}	0
5	Initial VP_i	0
6	Particle size “P”	6 (i.e., no. of features)
7	Fitness function	Accuracy of NB classifier
8	Initial swarm	P_1 = {1, 0, 1, 0, 1, 1} P_2 = {0, 0, 1, 1, 0, 1} P_3 = {1, 1, 0, 0, 1, 1}
9	w	1.1
10	c_1=c_2	2
11	r_1=r_2	0.6

3.3. Classification Module (CM)

One of the most well-known machine learning and pattern categorization approaches is K Nearest Neighbor (KNN). It is employed frequently due to its advantages over more complex supervised machine learning techniques in terms of simplicity and ease of implementation [18]. The KNN method is a wrapper technique that develops classification rules from training examples. It adjusts new

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

examples in the test set that are the closest neighbors to the examples used in the training set based on the training process, and it categorizes a given new instance per the highest category likelihood. The distance metric known as Euclidean distance is used to gauge how similar two points are [18]. However, while training KNN, the selection of k must be made carefully and is therefore chosen after numerous trials and errors iterations. Algorithm 2 illustrates the KNN algorithm's classification procedure.

Inputs: Y: training data, Z: class labels of X, K: number of nearest neighbors.
Output: class of test sample y
Start
Classify (Y,Z,y)
1. for each sample x do
Calculate the distance: $d(y,Y) = \sqrt{\sum_{i=1}^{n}(y_i - Y_i)^2}$
End for

2. classify y in the majority class: $S(y_i) = argmax_k \sum y_{j \in KNN} S(y_j, Y_K)$

End

Algorithm 2: pseudo code of KNN method

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

Feature Selection using IPSO algorithm

- ***Inputs:***
  - *x=No. of particles in swarm "swarm size" equals no. of filter approaches.*
  - *P=P_1….P_x; group of particles in swarm.*
  - *TR_A= (A, FS); Training dataset.*
  - *TE_A= (Q, FS); Testing dataset.*
  - *e=|F|; No. of features in training and testing dataset.*
  - *Initiate c_1,c_2∈[2-4]; the cognitive and social acceleration constants.*
  - *Initiate r_1,r_2∈[0-1]; uniformly distributed random numbers.*
  - *Initiate w∈[0.9-1.2]; inertia weight.*
  - *Initiate rand∈[0-1]; uniformly distributed random number.*
- ***Output:***
  - *Os*= the ideal swarm particle overall (P_{Global}).
- ***Steps:***

//implementing 'g' filter approaches on training and testing dataset

1: For every filter approach fil ∈ x

2: identify the subset of chosen feature for each approach as subset(fil)

3: End for

// build the swarm's initial population.

4: In an initial population of Swarm(Sm)n with particles denoted by, set 'x' subsets as the value of 'x' particles (P).

// estimate each particle's fitness value.

5: **For every P_i∈P**

6: $Fit(P_i) = Accuracy\ (P_i)$

7: **End for**

// Modify each particle's ideal solution. ($P_{Personal}$).

8: **For every P_i∈P**

9: $$P_{Personal}(P_i) = P_{pi} = \begin{cases} P_i & if(\ fit(P_i) > fit(P_{pi})) \\ P_{pi} & Else \end{cases}$$

10: **End for**

// Modifying the swarm's ideal particle (P_{Global}).

11: **For every P_i∈P**

12: $$P_{Global} = P_G = \begin{cases} P_{pi} & if(\ fit(P_{pi}) > fit(P_{pi+1})) \\ P_{pi+1} & Else \end{cases}$$

13: **End for**

// estimate each particle's updated velocity.

14: **For every P_i∈P**

15: $$VP_i(t+1) = w * VP_i(t) + \left(c_1 r_1 \left(P_{pi}(t) - P_i(t)\right)\right) + \left(c_2 r_2 \left(P_G(t) - P_i(t)\right)\right)$$

16: **End for**

// determine each particle solution's sigmoid function.

17: **For every Pi∈P**

18: $sig\ (VP_i^j) = \frac{1}{1+e^{-VP_i^j}}$

19: **End for**

Algorithm Parameters	
x	swarm size = no. of filter approaches
P	set of particles in swarm; P=P_1….P_x.
TR_A	Training data set contents and its features FS, TR_A= (A, FS).
TE_Q	Testing data set contents and its features FS, TE_A= (Q, FS).
M	No. of classes in the training dataset, M=|TC|.
e	No. of features in training and testing data set, e=|F|.
Subset (fil)	The subset of chosen feature at filter approach fil
c_1,c_2	The cognitive and social acceleration constants; c_1,c_2∈[2-4]; c_1+c_2=4.
r_1,r_2	uniformly distributed random numbers r_1,r_2∈[0-1].
w	inertia weight; w∈[0.9 - 1.2].
Os	Ideal swarm particle overall (P_{Global}).
P	Group of particle in swarm; P=P_1,P_2,P_3,…..P_x
P_i	i^{th} particle in the swarm.
rand	randam distributed numbers;r_1,r_2 ∈ [0,1]
Fit(P_i)	the fitness value of P_i particle.
$P_{Personal}$(P_i)	the best solution of P_i particle; P_{Pi}
P_{Global}	the best particle of the whole swarm; P_G.
VP_i(t+1)	the new velocity of particle P_i for iteration t+1.
P_i(t+1)	the new position of particle P_i for iteration t+1.
sig(VP_i^j)	The sigmoid function of i^{th} particle velocity at j^{th} position

// determine each particle's new position based on binary values.

20: **For every Pi ∈P**

21: $$P_i^j = \begin{cases} 0 & if\ (rand\ (0,1) > sig(\ VP_i^j) \\ 1 & Else \end{cases}$$

22: **End for**

23: modifying the values of w, vc_1,vc_2,vr_1, and r_2 parameters accordance with their appropriate ranges

24: **if (there are more generations to process) then**

25: *Go to step5.*

26: **Else**

27: *Return P_{Global} in Os, where the best features have been chosen to depict all of the ones in this particle*

28: **End if**

29: verified the chosen features utilizing NB classifier.

Algorithm 1: Feature selection using IPSO

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

4. Experimental Results

The suggested Intrusion Detection Framework (IDF) will be assessed in this section. The Data Preprocessing Module (DPM), Feature Selection Module (FSM), and Classification Module (CM) are the three successive modules used to perform IDF. Network traffic is gathered and analyzed in DPM in order to generate the data for training and testing. To choose the most important features from the input dataset in DPM, IPSO was presented as a novel feature selection approach in FSM. Then, the elected features are feed the KNN to make accurate and correct decision in CM. The accuracy, precision, recall, and f1-score performance measures will be employed in the subsequent trials to assess each component of the suggested framework [10]. Table 2 shows the relevant parameters together with the matching implementation values.

4.1 Dataset Description

For implementing the proposed IDF as well as the considered competitors, UNSW_NB-15 Dataset [22] has been deployed. This intrusion detection dataset was created using the IXIA Perfect Storm tool which records the network packets moving across the defined network dataset [22] Along with regular network traffic, the collection includes network traffic that identifies security flaws and attacks. Additionally, the UNSW NB-15 dataset includes separate training and test datasets with 175,341 and 82,332 data samples, correspondingly [22].

4.2 Evaluating the proposed improved particle swarm optimization (IPSO)

The proposed Improved Particle Swarm Optimization (IPSO) will be assessed in this part using the NB classifier as a standard classifier. To prove the effectiveness of the proposed feature selection method, many features selection techniques are compared to the proposed features selection technique IPSO. The most recent feature selection techniques used for evaluation are: [23], [24], and [25]. The accuracy, precision, recall, and f-measure for the employed classifier using the different feature selection techniques are depicted in Table 3. The accuracy, precision, recall, F1-score and execution time for IPSO is 98.88%, 99.89%, 98.71% 99.30% and 120.36s respectively. The findings suggest that the proposed IPSO outperformed the existing feature selection techniques.

Table 2: The applied parameters with the implemented values

Parameter	Description	Applied value
e	No. of obtained features	12
w	Inertia weight	1.1
c_1	The cognitive acceleration	2
c_2	The social acceleration	2
r_1	Uniformly distributed	0.6
r_2	random number	0.6
k	No. of neighbors	5

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

Table 3: Comparison between IPSO and recent feature selection techniques

Technique	No. of selected features	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Execution time (s)
[23]	20	81.94	81.91	98.46	89.42	360.11
[24]	30	92.65	99.59	91.24	95.23	279.63
[25]	23	94.90	95.10	94.30	94.70	355.01
Proposed IPSO	16	98.88	99.89	98.71	99.30	120.36

4.3 Evaluating the proposed Intrusion Detection Framework (IDF)

Through this subsection, it is the time to test the proposed IDF. All capabilities proposed are used in our IDF, hence, IPSO is employed for feature selection, and the KNN classifier is used for classification. Also, to argue the effectiveness of our proposed strategy for intrusion detection, it is compared against some of the recently used intrusion detection techniques which are; [11], [12], [13], and [14]. Results are shown in Table 4. The accuracy, precision, recall, F1-score and execution time for IDF is 99.8%, 99.94%, 99.85%, 99.89%, and 59.15s. According to Table 4, it is concluded that the performance of IDF is much better than [11], [12], [13], and [14]. The reason is that IDF gives fast and accurate detection for the intrusion attack based on using the KNN in CM depending on the most effective and significant features for intrusion detection which are selected through FSM.

Table 4: Comparison between IDF and other recent intrusion detection techniques

Techniques	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)	Execution time (s)
[11]	97.78	97.77	97.00	97.32	322.51
[12]	98.00	98.89	97.77	98.32	205.21
[13]	98.00	98.17	99.82	98.98	320.52
[14]	98.9	99.9	98.7	99.30	220.25
Proposed IDF	99.8	99.85	99.94	99.89	59.15

6. Conclusion

In this work, we have presented precise and intelligent intrusion detection strategy. The Intrusion Detection Framework (IDF) is comprised of three main modules: (i) Data Pre-processing Module (DPM), (ii) Feature Selection Module (FSM), and (iii) Classification Module (CM). Network traffic is collected and processed in order to prepare the data for training and testing in DPM. IPSO is a suggested feature selection methodology that involves the integration of both filter and wrapper selection methods. From the collected features from the input dataset, IPSO selects the most informative and effective features in FSM. Finally, the elected features are feed the KNN classifier which has to make accurate and correct decision in CM. Using the UNSW NB-15 dataset, the experimental results showed that the proposed IDF exceeded the state-of-the-art in terms of accuracy, recall, precision, f1-score, and inference time, obtaining values of 99.8%, 99.85%, 99.94%, 99.89%,

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

and 59.15s, respectively. Adopting an ensemble model with a novel dataset appropriate for the IoT environment will be the focus of future work.

Funding: "This research received no external funding"

Conflicts of Interest: "The author declare no conflict of interest."

References

[1 Deshmukh, M.S., Alvi, A.S. (2022). Detection and Prevention of Malicious Activities in Vulnerable Network Security Using Deep Learning. In: Gunjan, V.K., Zurada, J.M. (eds) Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications. Lecture Notes in Networks and Systems, vol 237. Springer, Singapore. https://doi.org/10.1007/978-981-16-6407-6_29. https://doi.org/10.1007/978-981-166407-6_29.

[2] S. Sadhasivam, P. Valarmathie and K. Dinakaran, "Malicious activities prediction over online social networking using ensemble model," Intelligent Automation & Soft Computing, vol. 36, no.1, pp. 461–479, 2023. https://doi.org/10.32604/iasc.2023.028650

[3] Mahadik, S., Pawar, P.M. & Muthalagu, R. Efficient Intelligent Intrusion Detection System for Heterogeneous Internet of Things (HetIoT). J Netw Syst Manage 31, 2 (2023). https://doi.org/10.1007/s10922-022-09697-x.

[4] Ashiku, L., Dagli, C. Network Intrusion Detection System using Deep Learning. Procedia Computer Science, 2021, 185, 239-247.

[5] Jadhav, A. D., Pellakuri, V. Highly Accurate and Efficient Two Phase-Intrusion Detection System (TP-IDS) using Distributed Processing of HADOOP & Machine Learning Techniques, 2021.

[6] Ali, S.H., El-Atier, R.A., Abo-Al-Ez, K.M. et al. A Gen-Fuzzy Based Strategy (GFBS) for Web Service Classification. Wireless Pers Commun 113, 1917–1953 (2020). https://doi.org/10.1007/s11277-020-07300-7

[7] A. Thakkar, R. Lohiya, Attack classification using feature selection techniques: a comparative study, J. Ambient Intell. Humaniz. Comput. 12 (1) (2021)1249–1266.

[8] Rabbani, M. et al. A review on machine learning approaches for network malicious behavior detection in emerging technologies. Entropy 23(5), 529 (2021).

[9] Ali, S. H., A New Intrusion Detection Strategy Based on Combined Feature Selection Methodology and Machine Learning Technique, MEJ. Mansoura Engineering Journal, Vol. 46(4),27-35(2021).

[10] Rabie, A.H., Ali, S.H., Saleh, A.I. et al. A fog based load forecasting strategy based on multiensemble classification for smart grids. J Ambient Intell Human Comput 11, 209–236 (2020). https://doi.org/10.1007/s12652-019-01299-x.

[11] Azidine Guezzaz, Said Benkirane, Mourade Azrour, and Shahzada Khurram, "A Reliable Network Intrusion Detection Approach Using Decision Tree with Enhanced Data Quality", Security and Communication Networks,2021. https://doi.org/10.1155/2021/1230593.

[12] Muhammad Naveed, Fahim Arif, Syed Muhammad Usman, Aamir Anwar, Myriam Hadjouni, Hela Elmannai, Saddam Hussain, Syed Sajid Ullah, and Fazlullah Umar, A Deep Learning-Based Framework for Feature Extraction and Classification of Intrusion Detection in Networks, Wireless Communications and Mobile Computing, Vol. 2022. https://doi.org/10.1155/2022/2215852.

[13] Chongzhen Zhang, Yanli Chen,1 Yang Meng, Fangming Ruan, Runze Chen, Yidan Li, and Yaru Yang, "A Novel Framework Design of Network Intrusion Detection Based on Machine Learning Techniques", Security and Communication Networks Volume 2021. https://doi.org/10.1155/2021/6610675.

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022

[14] Kezhou Ren, Yifan Zeng, Zhiqin Cao & Yingchao Zhang, "ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model" Scientific Reports (2022) 12:15370. https://doi.org/10.1038/s41598-022-19366-3.

[15] I.S. Thaseen and C.A. Kumar, Intrusion detection model using a fusion of chi-square feature selection and multiclass SVM. Journal of King Saud University - Computer and Information Sciences, 2017. vol. 29, pp. 462-472. [16] Brezočnik, L.; Fister, I.; Podgorelec, V. Swarm Intelligence Algorithms for Feature Selection: A Review. Appl. Sci. 2018, 8, 1521. https://doi.org/10.3390/app8091521.

[17] Binsaedan, W., Alramlawi, S., CS-BPSO: Hybrid feature selection based on chi-square and binary PSO algorithm for Arabic email authorship analysis, kowlegde based systems, Vol.27(5), 2021. https://doi.org/10.1016/j.knosys.2021.107224.

[18] Saleh, A. I., El Desouky, A. I., Ali, S. H., Promoting the performance of vertical recommendation systems by applying new classification techniques, kowlegde based systems, Vol.75, 192-223, 2015.

[19] M. I. Prasetiyowati, N. U. Maulidevi, K. Surendro. (2021, June). Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest. Prasetiyowati et al. J Big Data. 8(84). Available: https://doi.org/10.1186/s40537-021-00472-4

[20] S. Bahassine, A. Madani, M. Al-Sarem, M. Kissi. (2020). Feature selection using an improved Chi-square for Arabic text classification. Journal of King Saud University – Computer and Information Sciences.32, pp. 225-231. Available:
https://doi.org/10.1016/j.jksuci.2018.05.010

[21] H. Djellali, N. Zine, N. Azizi. (2016). Two stages feature selection based on filter ranking methods and SVMRFE on medical applications. Modelling and Implementation of Complex Systems Lecture Notes in Networks and Systems. 1, pp. 281–293.

[22] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference (MilCIS), IEEE, 2015, pp.1–6.

[23] Albulayhi, K.; Abu Al-Haija, Q.; Alsuhibany, S.A.; Jillepalli, A.A.; Ashrafuzzaman, M.; Sheldon, F.T. IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method. Appl. Sci. 2022, 12, 5015. https://doi.org/10.3390/app12105015.

[24] Saif S. Kareem, Reham R. Mostafa, Fatma A. Hashim and Hazem M. El-Bakry, "An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection" Sensors 2022, 22, 1396. https://doi.org/10.3390/s22041396.

[25] Faezah Hamad Almasoudy, Wathiq Laftah Al-Yaseen, Ali Kadhum Idrees, "Differential Evolution Wrapper Feature Selection for Intrusion Detection System", Procedia Computer Science, Volume 167, 2020, Pages 1230-1239. https://doi.org/10.1016/j.procs.2020.03.438.

Doi : https://doi.org/10.54216/JCIM.100103
Received: April 09, 2022 Accepted: July 20, 2022