Intelligent feature selection using particle swarm optimization algorithm with a decision tree for DDoS attack detection

Aween Abubakr Saeed [a,1,*], Noor Ghazi Mohammed Jameel [b,2]

[a] Department of Information Technology, Technical College of Informatics, Sulaimani Polytechnic University, Sulaimani, Kurdistan Region, Iraq
[b] Department of computer networks, Technical College of Informatics, Sulaimani Polytechnic University, Sulaimani, Kurdistan Region, Iraq
[1] aween.saeed@spu.edu.iq; [2] noor.ghazi@spu.edu.iq
* corresponding author

ARTICLE INFO

Article history
Received September 13, 2020
Revised November 9, 2020
Accepted November 18, 2020
Available online March 31, 2021

Keywords
Distributed denial of service
Binary particle swarm optimization
Decision tree algorithm
Wrapper feature selection
Swarm intelligent

ABSTRACT

The explosive development of information technology is increasingly rising cyber-attacks. Distributed denial of service (DDoS) attack is a malicious threat to the modern cyber-security world, which causes performance disruption to the network servers. It is a pernicious type of attack that can forward a large amount of traffic to damage one or all target's resources simultaneously and prevents authenticated users from accessing network services. The paper aims to select the least number of relevant DDoS attack detection features by designing an intelligent wrapper feature selection model that utilizes a binary-particle swarm optimization algorithm with a decision tree classifier. In this paper, the Binary-particle swarm optimization algorithm is used to resolve discrete optimization problems such as feature selection and decision tree classifier as a performance evaluator to evaluate the wrapper model's accuracy using the selected features from the network traffic flows. The model's intelligence is indicated by selecting 19 convenient features out of 76 features of the dataset. The experiments were accomplished on a large DDoS dataset. The optimal selected features were evaluated with different machine learning algorithms by performance measurement metrics regarding the accuracy, Recall, Precision, and F1-score to detect DDoS attacks. The proposed model showed a high accuracy rate by decision tree classifier 99.52%, random forest 96.94%, and multi-layer perceptron 90.06 %. Also, the paper compares the outcome of the proposed model with previous feature selection models in terms of performance measurement metrics. This outcome will be useful for improving DDoS attack detection systems based on machine learning algorithms. It is also probably applied to other research topics such as DDoS attack detection in the cloud environment and DDoS attack mitigation systems.

This is an open access article under the CC–BY-SA license.

1. Introduction

The rapid development of internet services made network security issues more critical with many network attacks, such as the denial of service (DoS) attack [1]. Denial of DoS attack exhausts the network's resources and avoids authorized users accessing the network services. Recently, another variant of the DoS attack is called Distributed Denial of Service (DDoS), an enormous problem for large corporations. Attackers launch distributed botnets to maximize the attack impact into all resources. The DDoS attack's major aim is to suspend the services to legitimate users, which in turn causes financial losses and reputational damage to victims or target companies. So the early detection of these attacks is

critical, which helps the cyber team mitigate them in time [1][2]. There are many kinds of DDoS attacks, such as ACK-flood, DNS Reflect, UDP-flood, Slowloris, SYN-flood, and ICMP flood. Attackers use several strategies to achieve their goal, one of which is by flooding the network with bogus requests. The DDoS attack is distributed so that the attacker uses multiple computers to launch the denial of service attack [3]. According to the latest records, the most significant DDoS attack in February of 2020 to Amazon web service is gorilla cloud computing. This attack saw incoming traffic at its peak at a rate of 2.3 terabits per second (Tbps) [4]. Furthermore, GitHub assaulted by another significant DDoS attack, which was occurred in February 2018. The attack created over 1000 different independent systems crosswise 10,000 exclusive endpoints. It was an amplification attack using Memcached-based that peaked at 1.35 Tbps, which made significant parts of the internet down [4]. The DDoS attack is implemented by developing a software application and installing it on a machine or computer, called a botnet. Then, the attacker will control the infected machine to launch the attack on all other botnets. The procedure of the DDoS attack involves three steps. At first, the attacker sends the implementation message to the controlled botnet which is also called master, when the master receives the message, it will run the second step and generates a newer execution command and delivers it to the "software application". Finally, when the message is received, the software starts to attack the target network or server which is known as the victims [5][6]. There are many DDoS attack detection methods based on machine learning. Machine learning methods mainly include unsupervised learning and supervised learning [7]. In the research on ML-based DDoS detection techniques, the focus is not only on detection models but also comprises Feature selection (FS) techniques. Feature selection is a vital step that can progress the classification method's performance by removing irrelevant, redundant, and noisy features. When designing a feature selection model, two problems should be taken to consideration. The first problem is the solution representation. Where FS is a binary optimization problem, the solution should be denoted as a binary vector. The 1 value represented that feature is selected, and 0 represented that feature is not selected. The second problem is the design fitness function. Hence, we should consider classification accuracy and the number of selected features [8]. Swarm intelligence is an artificial intelligence (AI) method that depends on group behavior instigated by nature. The most common swarm-intelligence algorithms are Ant Colony Optimization (ACO), Swarm-Optimization (PSO), and Artificial-Bee-Colony (ABC). They can solve complex optimization problems, such as feature selection, objective-function, constraints, and instance problems [9]. Feature selection is considered a crucial key in machine learning problems to select the best features that perform the highest accurate performance with the lowest error rate [8].

Many researchers proposed different DDoS attack detection models based on conventional algorithms. Prasad *et al.* [10] Designed a new DDoS attack detection model depending on different machine learning algorithms. First, the authors created two different datasets, balanced and imbalanced datasets taken from three open datasets (CIC-DoS, CICIDS-2017, and CSE-CIC-IDS-2018), the Canadian Institute for cybersecurity. Then They used the Random Forest feature importance with the entire dataset, including over 12 million samples where a binary classifier detected DDoS and Benign classes. The system's outcome showed that the decision trees classifier accuracy rate was 99.94 %, including 84 features of the dataset. Patil and Kshirsagar [11] proposed an architecture that contains network data and features with machine learning classifiers. The proposed system used the information gain and ranker method as an FS model to improve the features efficiency.

The selected features were evaluated using machine learning techniques such as Logistic Model Tree (LMT), random forest, and J48 classifier for detecting DDoS attacks by taking benefit from the novel CICIDS-2017 dataset. The experimentation results confirmed J48 classifier achieved a detection rate than the random forest and logistic model tree with fewer features. Lima Filho *et al.* [12] proposed a smart online system to detect DoS\DDoS attacks. The Random Forest Tree algorithm was created to detect both types of attacks based on customized data set nominated (CIC-DoS, CICIDS-2017, and CSE-CIC-IDS-2018) datasets. The feature selection model reduced available features from 28 to 20 relevant features, and the outcome of the online system was evaluated by detection rate, Recall, and precision metrics. The performance evaluation of each CIC-DoS and CSE-CICIDS-2018 datasets obtained a detection rate and Precision of more than 93%. It has been found many researchers designed

DDoS detection systems; hence it is seen that distributed denial of service attacks has offensive acts on internet service providers (ISP)s and web-services [13][14].

This study's key objective is to construct a feature selection model that selects the least and effective features for a DDoS attack detection system with fast and high accuracy. The swarm intelligence algorithms can select fewer features and give a higher classification accuracy than the traditional algorithms. The primary aim of this work is to develop a credible wrapper feature selection model using a binary version of the Particle Swarm Optimization Algorithm with a Decision Tree classifier as an evaluator. Feature selection enhances classification algorithm performance by selecting the most relevant features and reducing the required computational time. Furthermore, this study designs a DDoS detection model using machine learning supervised algorithms such as multi-layer perceptron, random forest, and decision tree for accurate and fast DDOS attack detection with optimal selected features. To classify DDoS and Benign classes in a balanced dataset is prepared by Prasad et al. [10]. The paper's organization includes: Section 2 explains the proposed feature selection model using the binary-particle swarm-optimization algorithm with decision tree classifier including proposed model steps. The steps are loading data set, pre-processing, feature selection with (B-PSO), and (DT), and evaluation using different machine learning algorithms. The results of the experiments and their discussion are clarified in section 3. Lastly, section 4 explains the conclusions.

2. Method

This section explains the primary four steps of the proposed wrapper feature selection model by applying Binary Particle Swarm Optimization and Decision Tree classifier. At the first step, the dataset should be loaded into python, then the pre-processing of the dataset executed, which is the second step. The third step will be started by entering the dataset features into the proposed feature selection by Binary Swarm Optimization as an optimization method with Decision Tree classifier as a performance evaluator. The model attempts to select smaller and relevant DDoS attack features with the shortest computational time. At the last and fourth step, the optimal selected features were evaluated by different (ML) techniques such as Multi-Layer Perceptron (MLP), Decision Tree (DT), and Random Forest (RF) within performance evaluation metrics. Also, the results are compared with previous models suggested by researchers by tabular form. The stages of the suggested model, are illustrated in Fig 1.

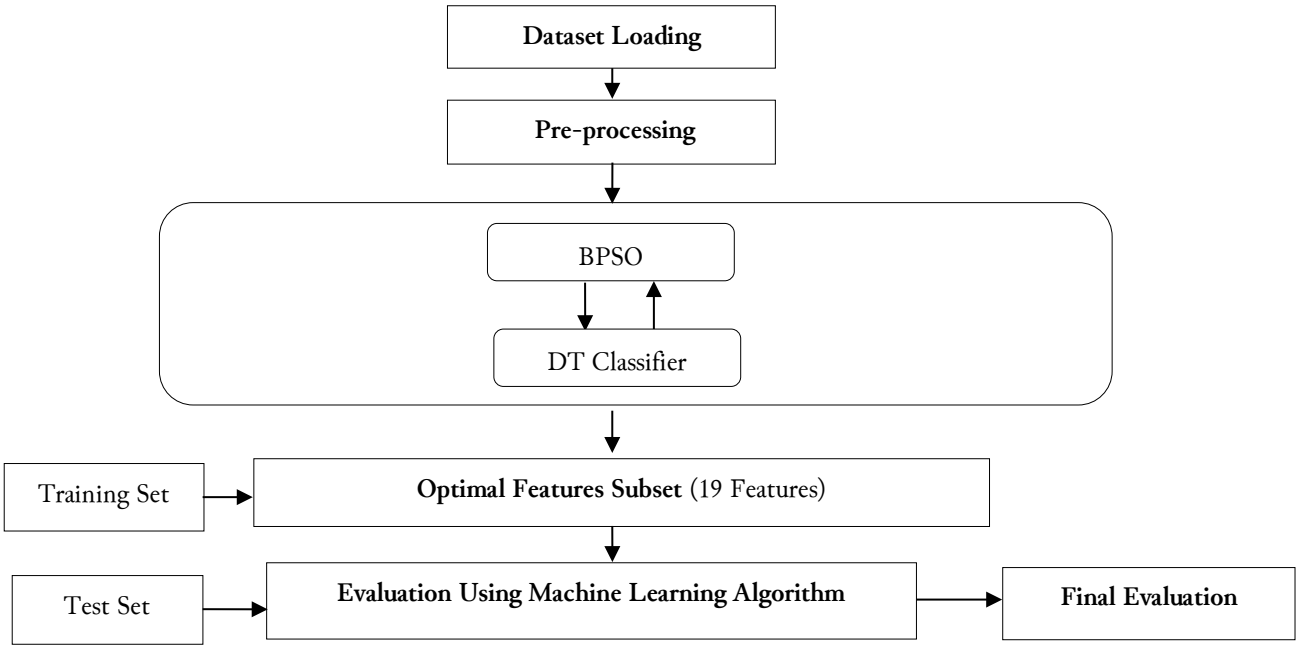

Fig. 1. Proposed model architecture

2.1. Dataset Loading

In this section, the results acquired by the suggested model are discussed and presented. The proposed FS model implemented using Anaconda Python 3 Open Source, and all tests have been run on a Laptop with the description of Core i7 Intel, 7th generation, 2.7 GHz CPU, and RAM 32 GB. The dataset loaded into Jupyter then pre-processing steps executed on it, the results of pre-processing steps were saved in a data frame as shown in (Fig. 6). Then, 1 Million data records have been separated from the dataset. Next, the dataset records were segmented into two sets, the training set 70% of the data records, the test set 30%. The proposed feature selection model is deployed on a balanced dataset downloaded from Kaggle, an open-source [15]. The dataset is a combination of three datasets established by a cyber-security institute in Canada (CCI) [16][17]. The purpose of creating this dataset is to mimic real-time DDoS traffic. The three datasets have been generated using various attack tools in different years (2016, 2017, and 2018). Table 1 explains the number of dataset records in each "DDoS" and "Benign" classes.

Table 1. Number of flows in a balanced dataset

Dataset name	Label: DDoS	Label: Benign	Total Flows	Data Points size
Balanced	6472647	6321980	12794627	12.79 Million

2.2. Pre-processing

In this section, pre-processing stages have been explained. At the first stage, categorical features were dropped from the dataset since some machine learning algorithms often require only numerical data. The dataset now contains 76 features with the class label, which equals 77 columns. At the second stage, a label encoder is used to convert class labels into 0 and 1 [18]. Label encoding has an easy implementation in data science projects, and it is a fast approach to change class labels from categorical values into binary. In the third stage, 500k data records are separated in each class of "DDoS" and "Benign", as a result, 1 million data records are utilized for the feature selection process. The dataset features, are presented in Table 2 [15].

Table 2. Data set features and indexes

Index	Feature	Index	Feature	Index	Feature
1	Source-port	27	Backward-IAT-std	53	Forward-Seg-Size-avg
2	Destination-port	28	Backward-IAT-max	54	Forward-Seg-Size-avg
3	Protocol	29	Backward-IAT-min	55	Forward-Bytes/b-avg
4	Total-Forward-pkts	30	Forward-PSH-flags	56	Forward-Pkts/b-avg
5	Tot-Backward-pkts	31	Backward-PSH-flags	57	Forward-Blk Rate-avg
6	Tot-Len-Forward-pkts	32	Forward-URG-flags	58	Backward-Bytes/b-avg
7	Tot-Len-Backward-pkts	33	Backward-URG-flags	59	Backward-Pkts/b-avg
8	Forward- pkt- Len- max	34	Forward-Header-len	60	Backward-Blk-Rate-avg
9	Forward- pkt- Len- min	35	Backward-Header-len	61	Subflow-Forward-pkts
10	Forward- pkt- Len- mean	36	Forward-pkts/s	62	Subflow-Forward-bytes
11	Forward- pkt- Len- std	37	Backward-pkts/s	63	Subflow-Backward-pkts
12	Backward- pkt- Len- max	38	Pkt-Len-min	64	Subflow-Backward-bytes
13	Backward- pkt- Len- min	39	Pkt-Len-max	65	Init-Forward-Win-bytes
14	Backward- pkt- Len- mean	40	Pkt-Len-mean	66	Init-Backward-Win-bytes
15	Backward- pkt- Len- std	41	Pkt-Len-std	67	Forward-Act-Data-pkts
16	Flow-IAT-mean	42	Pkt-Len-var	68	Forward-Seg-Size-min
17	Flow-IAT-std	43	FIN-Flag-cnt	69	Active- mean
18	Flow-IAT-max	44	SYN-Flag-cnt	70	Active- std
19	Flow-IAT-min	45	RST-Flag-cnt	71	Active- max
20	Forward-IAT-tot	46	PSH-Flag-cnt	72	Active- min
21	Forward-IAT-mean	47	ACK-Flag-cnt	73	Idle- mean
22	Forward-IAT-std	48	URG-Flag-cnt	74	Idle- std
23	Forward-IAT-max	49	CWE-Flag-count	75	Idle- max
24	Forward-IAT-min	50	ECE-Flag-cnt	76	Idle- min
25	Backward-IAT-tot	51	Down/Up-ratio		
26	Backward-IAT-mean	52	Pkt-Size-avg		

2.3. Feature selection

The major step before the classification procedure is feature selection. This step aims to choose an optimal subset of features in existence and drop irrelevant features. Generally, feature selection is categorized into three method types [19][20]. The wrapper method generates a feature subset by finding the search space between the features using searching algorithms and select the best subset by evaluating all the generated subsets. The Filter approach is similar to the wrapper approach, but it uses a simple filter model for evaluation instead of running all the models. The first approach produces accurate results but takes more time for execution, the second approach reduces the execution time, but the accuracy is lesser than the wrapper approach. While the embedded approach takes advantage of both the above approaches, it improves accuracy and reduces execution time. It takes the dataset's intrinsic characteristics and uses predefined mining algorithms for the subset generation and evaluation process [21].

Feature Selection can be attempted with Swarm Intelligence algorithms because it has been verified that Swarm Intelligence algorithms can solve NP-Hard problems. So, selecting an optimal feature subset is a type of that computational problem. Nowadays, Swarm Intelligence algorithms have been prevalent, and there two most used algorithms Ant Colony Optimization and Particle Swarm Optimization [22]. This study focuses on deploying the SI algorithm as the usual option for the wrapper feature selection approach, while wrapper feature selection integrates machine learning classifiers with swarm optimization algorithms to select the most valuable features [23]. Particle-swarm optimization is a renowned swarm-intelligent algorithm. It was introduced by Dr. Kennedy and Eberhart in 1995 [24]. It is an exploratory algorithm that mimics the movement of the fish schooling and bird flocking. PSO is a simple algorithm within few rules that can obtain a new solution from previous solutions [25]; The important idea about PSO is taking information from social communication in the population from personally found solutions [26]. However, it is simple, it has an appropriate computational cost, and it can reach the best solution in high dimensional spaces [27].

In PSO, each particle is considered as a point in a d-dimensional search space. Each particle or candidate solution can memorize its best solution in the search space and its best experiences among the whole swarm [26]. Basically, in the PSO a vector $[x_{i1} = (x_{i1}, x_{i2}, \ldots\ldots, x_{iD})]$ as a position representation of particles i is used, and Di is search space's dimensionality. Additionally, the velocity of particle i is denoted as $[v_i = (v_{i1}, v_{i2}, \ldots\ldots, v_{iD})]$ vector. The best position founded by particles is personal best denotes as *pbest*, the best positions will be founded by swarm is global best denotes as g_{best} [28]. The swarm changes their position based on (1), and velocity based on (4)

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

$$v_{id}^{(t+1)} = w * v_{id}^{(t)} + c_1 r_1 \left(p_{id} - x_{id}^{(t)}\right) + c_2 r_2 (p_{gd} - x_{id}^{(t)}) \tag{2}$$

Where the number of iterations in the search process is denoted by t, d^{th} the dimension of the search space dented by d , acceleration constants are denoted by c_1,c_2c2. Random values of r1 and r2 are distributed uniformly between [0,1]. The best solutions *pbest* and *gbest* are represented as p^{id} and p_{gd}, and the inertia weight is denoted as w [28]. Originally, PSO worked as an optimization algorithm to solve continuous problems, but optimization problems are discrete such as feature selection, which occurs in discrete search space. To expand particle swarm optimization applications Kennedy and Eberhart [29] proposed a different form of particle swarm optimization, called binary-PSO (BPSO), to optimize discrete problems such as traveling salesman, job-scheduling, and other sequence-based problems. In BPSO, velocity is updated similarly as standard PSO with the probability of taking position value [10][18] within binary representation space, (2) is utilized to change the velocity of each particle before transferring its values by sigmoid function into a range between {0,1}. As a result, the value of x_i, p^{id}, and p_{gd} are limited to 0 or 1, each particle could change its position regarding (3) with probability value $T(v_t)$obtained from (4) [30].

$$S(v^{(t)}) = \frac{1}{1+e^{-v(t)}} \quad (3)$$

$$x^{(t+1)} = \begin{cases} 1 \text{ if rand } < S(v^{(t+1)}) \\ 0 \text{ otherwise} \end{cases} \quad (4)$$

Fig. 2 shows the sigmoid function, which transfers velocity values into 0,1 denoted as $S(v^{(t)})$[30].

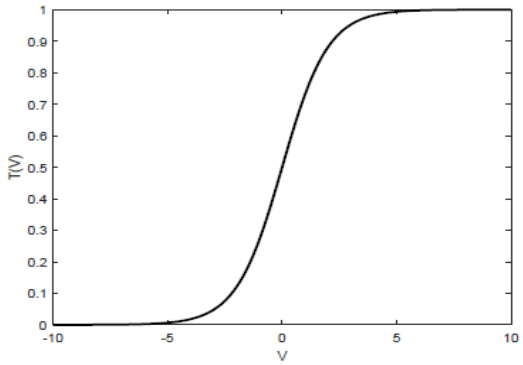

Fig. 2. Sigmoid transfer function

2.4. Feature Selection With BPSO and Decision Tree

This paper aims to develop an intelligent binary-particle swarm optimization algorithm that effectively uses feature selection problems to detect DDoS attacks. It proposed a wrapper feature selection that employs a search strategy by binary particle swarm optimization and a decision tree algorithm as a classifier. In order to optimize the feature selection process, two problems should be solved. The first problem is representing the solutions; The features should be represented within the binary vector because feature selection with binary PSO is a binary optimization problem. Where value is 1 the feature has been selected, otherwise value 0 represents non-selected feature. As a result, the solution's size is equal to the total number of 1 value. The second problem is constructing a fitness function. In this paper wrapper approach is implemented, one of the performance measurement metrics can be used to evaluate features. Here, the accuracy of the decision tree classifier has been used as a performance evaluator within the number of features in fitness function construction. So, the number of subset features and accuracy of the classifier has been considered in fitness function as (5) [28] [31].

$$f(x) = \alpha(1-P) + (1-\alpha)\left(1 - \frac{N_t - N_f}{N_t}\right) \quad (5)$$

Where the total size of features is denoted as (N_t), the size of the feature subset presented as (N_f), performance measurement of classifier denoted by (P), and α is a constant number between {0,1}. More details about B-PSO initialization parameters are explained in Table 3.

Table 3. BPSO parameter setting

Parameter	Value
Population-size	76
Number of iterations	100
Dimension-size	number of features
Fitness function	see (5)
α in the fitness function	0.99
(c1, c2)	c1=2, c2=2
Inertia weight	0.3
Number of neighborhoods (k)	76
Degree of connectivity (p)	2

Algorithm 1, as shown in Fig. 3, presents the pseudocode of the binary-particle swarm optimization (B-PSO) algorithm. The first step started is initializing the population of particles arbitrarily. In the

second step fitness value of every particle will be evaluated, the *pbest* and *gbest* values will set. The position and velocity of particles were changed respected to (2), (3), and (4). Then the fitness of the next particles will be evaluated, also *pbest* and *gbest*. The steps of the algorithm were repeated until reaching the extreme iteration numbers. In the end, the solution of the global best will be the optimal selected subset features [32].

Algorithm 1: Processed Feature Selection By bPSO
Input: numpy.ndarray of shape (n_particles, dimensions)
Output: Best positions array of [0,1]
1. **Begin**
2. split Dataset into Training and Test set;
3. Initialize the velocity and position of each particle randomly;
4. **while** *maxiterations* reached or the stop condition is not met do
fitness function of particles are evaluated depend on (5); accuracy of the test set by DT classifier
5. **for** i=1 **to** *populationsize* **do**
update the pbest of particle *i*;
update the gbest of particle *i*;
6. **for** i=1 **to** *populationsize* **do**
7. **for** d=1 **to** number of inputted features **do**
velocity of particle *i* based on (2) will update;
position of particle *i* based on (3), (4) will update;
compute classification accuracy of test set of the selected feature subset;
return gbest position of the particles (the selected feature subset);
return the best cost;
8. **End**

Fig. 3. Binary particle swarm optimization

2.5. Optimal Feature Subset

This section provides an optimal feature subset selected by wrapper FS that employs the B-PSO algorithm as a search strategy and DT classifier as a performance evaluator. In this work, feature selection is used to improve detection model performance. The problem in this study is a classification problem. The goals are to maximize the DDoS detection model's performance and minimize the number of used features in the dataset. Feature selection is a binary optimization problem. The solution was represented with a binary vector, where the value 1 indicates that the corresponding feature is selected; otherwise, it is not selected. The solution size is the number of features in each dataset. Since we adopted the wrapper approach, the best subset was generated by finding the search space between the features using the searching algorithm by (B-PSO) and selects the best subset by evaluating all the generated subsets by (DT). We have tried various tests with different iterations during the feature selection process. The minimum number of selected features was 19 features out of 76 features with 100 iterations of B-PSO and decision tree classifier with depth= 5. Then the selected features were trained by different machine learning algorithms and got the highest classification accuracy. Table 4 presents indexes of selected features with their description in detail. Fig. 4 demonstrates the cost history, and Fig. 5 shows the Error Rate of the selected features regarding each iteration of B-PSO.

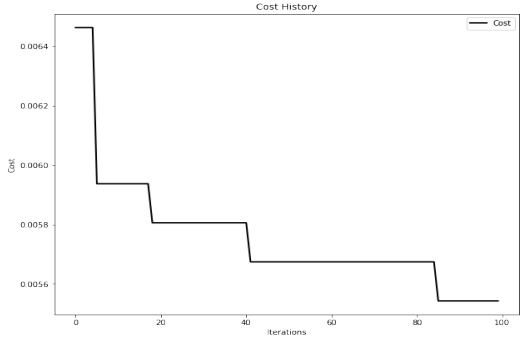

Fig. 4. Cost History in each Iteration

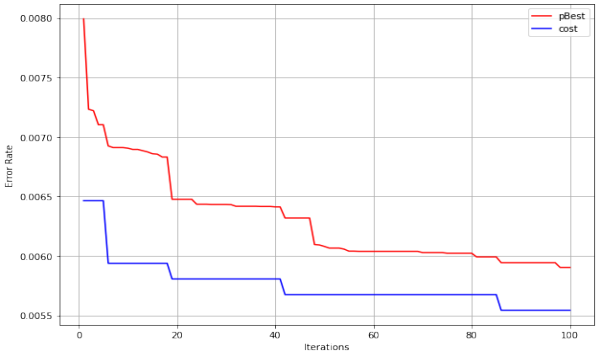

Fig. 5. Error Rate in each Iteration

Table 4. Optimal features selected by the proposed model

Index	Feature	Description
1	Tot-Forward-Pkts	All number of forwarded packets
2	Forward-Pkt-Len-Max	Max size of the forwarded packet
3	Forward-Pkt-Len-Std	Std size of the forwarded packet
4	Backward-Pkt-Len-Std	Std size of the back warded packet
5	Flow-IAT-Std	Std time two flows
6	Backward-IAT-Mean	Two back warded packets mean
7	Backward-IAT-Std	Std time between two backward packets
8	Backward-IAT-Min	least time among two backward packets
9	Forward-Header-Len	Total bytes used for headers forwarded
10	Pkt-Len-Mean	Mean length of a flow
11	SYN-Flag-Cnt	Number of SYN packets
12	PSH-Flag-Cnt	Number of PUSH packets
13	Forward-Seg-Size-Avg	The average size observed seg forwarded
14	Forward-Bytes/b-Avg	The average number of forwarded bytes
15	Forward-Blk-Rate-Avg	The average number of packets bulk rate forwarded
16	Backward-Pkts/b-Avg	The mean size of packets that back warded
17	Init-Forward-Win-Bytes	The number of forwarded bytes sent by the initial window
18	Active-Std	Std time a flow was active- before becoming idle
19	Idle-Std	Std time a flow was idle -before becoming active

2.6 Evaluation using Machine Learning Algorithm

The proposed FS model efficiency is represented by selecting 19 features among 76 dataset features. In this study, the best 19 features were trained and tested using different machine learning algorithms for DDoS attack detection, such as multi-layer perceptron, decision tree, random forest. Traditionally, there are many metrics to evaluate classification algorithms, like accuracy, defined as the sum of all good classified samples divided by all available samples. Generally, classification models have two classes as in this work there is (DDoS) and (Benign). Confusion matrix is another ML evaluation metric that is composed of four components true positive, true negative, false positive, and false negative [33][34]. True Positive (TP) indicated the classifier predicted that data flow as DDoS. True Negative (TN) indicated the classifier predicted that data flow as Benign. False Positive (FP) indicated the classifier predicted that data flow is not DDoS. False Negative (FN) indicated that the classifier predicted that data flow as Benign, but actually is DDoS. The Recall is evaluated by (7), which is the number of correctly predicted data flows overall data flow for a specific class. Precision as in (8) is the percentage of correctly predicted data flows overall predicted data flows for a specific class, and F1-score as in (9) is engaging Recall, Precision, and it is a harmonic average of both [35].

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP}+\text{FN})} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})} \tag{8}$$

$$\text{F1}_{\text{Score}} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{9}$$

3. Results and Discussion

The two sets are fed into the FS model, various tests tried with different iteration numbers of B-PSO, also, the parameters of B-PSO have been set properly since they have chosen either based on some test results or based on previous studies. The FS method's outcome generated different subsets of features, but the minimum subset is taken with 19 relevant features among 76 with 100 iterations of B-PSO and Decision Tree classifier with depth 5. The confusion matrix was used to evaluate the classifiers such as multi-layer perceptron, random forest, and decision tree. First, one million data records within 19 features entered to classification process, and the overall accuracy was 96.83 %, 98.82%, and 99.59%

of multi-layer perceptron, random forest, and decision tree respectively as shown in Table 5. The other evaluation metrics precision, Recall, and specificity are equally important because of the balanced data and should be put into consideration. Secondly, the 19 selected features were extracted from the whole dataset records and entered into the same classification algorithms. The overall accuracy was 90.06%, 96.94 %, 99.52 % of multi-layer perceptron, random forest, and decision tree respectively as shown in Table 6. Out of the three algorithms, DT shows better accuracy in terms of accuracy within the two different data record numbers.

	Src Port	Dst Port	Protocol	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	Fwd Pkt Len Mean	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	4504	80	6	29	44	86.0	59811.0	86.0	0.0	2.965517	...	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
1	4504	80	6	1	1	0.0	0.0	0.0	0.0	0.000000	...	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
2	4505	80	6	2	6	86.0	3037.0	86.0	0.0	43.000000	...	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
3	4505	80	6	2	1	0.0	0.0	0.0	0.0	0.000000	...	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
4	4506	80	6	2	5	73.0	1050.0	73.0	0.0	36.500000	...	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1

Fig. 6. A sample of pre-processed data-frame which contains 76 features + Label

Table 5 illustrates that the proposed FS using B-PSO and Decision Tree classifier was efficient because it could select the fewest features among 76 features of the dataset, which was improved by training the selected features with three machine learning algorithms. The 19 features got the highest classification accuracy rate with the Decision Tree algorithm is equal to 99.59 %, Recall is 0.9993, Precision is 0.9924, and F1-score is 0.9959. As a result, the nominated features can represent "DDoS" or "Benign" data. The proposed approach's selected features were evaluated within 1 Million records of the data set with machine learning algorithms such as multi-layer perceptron, random forest, and decision tree for detecting DDoS attack are presented in Table 5.

Table 5. Evaluation metrics' results of classification algorithms (1 Million Records)

No of Instances	No of Features	Algorithms	Accuracy	Recall	Precision	F1-Score
1000000	19	MLP	96.83	0.9959	0.9438	0.9691
1000000	19	RF	98.82	0.9987	0.9780	0.9883
1000000	19	DT	99.59	0.9993	0.9924	0.9959

Then selected features have been separated from the final balanced dataset within 12794627 data-points, trained, and tested within the same classification algorithms. All experimental results of the detection of DDoS attacks are presented in Table 6. According to the results, the DT algorithm outperformed other algorithms. Table 6 shows that the FS model selected the most relevant features because the same 19 features have been fed to three ML classification algorithms within all dataset instances and resulted in a high accuracy rate with DT algorithm is 99.52 %, Recall 0.9992, Precision 0.9914, and F1-score 0.9953.

Table 6. Evaluation metrics' results of classification algorithms (12794627 Records)

No. of Instances	No. of Features	Detection Algorithm	Accuracy	Recall	Precision	F1-Score
12794627	19	MLP	90.06	0.9538	0.8638	0.9066
12794627	19	RF	96.94	0.9496	0.9896	0.9692
12794627	19	DT	99.52	0.9992	0.9914	0.9953

Table 7 compares the DT classification results and the optimal feature subset chosen by the proposed FS model and the previous studies' results. It has been proven that the proposed FS model is effective and intelligent because it reduced the number of features and achieved a high accuracy rate like

previous proposed works in the same area of study. The detection algorithm outcomes high accuracy rate, as explained in Table 7, the Decision Tree algorithm reaches a 99.52 % accuracy rate with only 19 discriminative features. However, in [12] the accuracy of the decision tree got 99.4182 with 25 features and in [10] the accuracy of the decision tree algorithm reached 99.94 with 84 features on the same applied dataset.

Table 7. Comparison among proposed model and other studies

Ref	FS algorithm	No of Features	Detection Algorithm	Accuracy	Dataset
Patil and Kshirsagar [11]	Information gain and ranker algorithm	75	J48	87.44	CICIDS2017
Lima Filho d. *et al.* [12]	RFECV and Random forest	25	DT	99.42	ISCXIDS2012
Prasad *et al.* [10]	-	84	DT	99.94	Balanced dataset
Proposed Model	B-PSO with DT	19	DT	99.52	Balanced dataset

4. Conclusion

Distributed denial of service (DDoS) attacks became a very popular threat that overwhelms target servers and interrupts network services. This paper targeted to develop an intelligent feature selection model to effectively select the most relevant and significant features in detecting this type of attack within a short execution time and less computational cost. In this study, a wrapper FS model utilizing the binary PSO algorithm with the DT classifier as a performance evaluator in detecting DDoS attacks has been proposed. The proposed model was carried out using a balanced DDoS dataset which contains 12794627 network traffic flows. The experimental results presented high performance and intelligently selected significant 19 features among 76 features of the dataset. The relevant features selected by the proposed model were trained and tested with different classification algorithms and achieved the highest 99.52 % accuracy performance with the decision tree classifier. As future work, we plan to develop a DDoS detection model with deep learning algorithms by utilizing the same massive balanced dataset.

Acknowledgment

The authors would like to thank the Canadian Institute of Cybersecurity (CIC-UBN) as they shared their datasets as a public dataset.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. This proposed work has not achieved any fund
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] N. Bindra and M. Sood, "Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset," *Autom. Control Comput. Sci.*, vol. 53, no. 5, pp. 419-428, 2019, doi: 10.3103/S0146411619050043.

[2] R. Saxena and S. Dey, "DDoS attack prevention using collaborative approach for cloud computing," *Cluster Comput.*, pp. 1-16, 2019, doi: 10.1007/s10586-019-02994-2

[3] *What Is an ACK Flood DDoS Attack? | Types of DDoS Attacks*, 2020, available at: cloudflare.com

[4] *Famous DDoS attacks | The largest DDoS attacks of all time*, 2020, available at: cloudflare.com

[5] J. Weiss, "*DDoS Detection Using Deep Neural Networks on Packet Flows*", 2019, available at: http://www.cs.tufts.edu/comp/116/archive/fall2019/jweiss.pdf.

[6] S. Sarraf, "Analysis and Detection of DDoS Attacks Using Machine Learning Techniques," *Am. Sci. Res. J. Eng. Technol. Sci.,* vol. 66, no. 1, pp. 95-104, 2020, Available at: Google Scholar

[7] Y. Gu, K. Li, Z. Guo, and Y. Wang, "Semi-supervised K-means DDoS detection method using hybrid feature selection algorithm," *IEEE Access,* vol. 7, pp. 64351-64365, 2019, doi: 10.1109/ACCESS.2019.2917532

[8] Y. Li, T. Li, and H. Liu, "Recent advances in feature selection and its applications," *Knowl. Inf. Syst.,* vol. 53, no. 3, pp. 551-577, 2017, doi: 10.1007/s10115-017-1059-8

[9] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.,* vol. 33, pp. 1-17, 2017, doi: 10.1016/j.swevo.2016.12.005.

[10] M. D. Prasad, V Prasanta Babu, and C. Amarnath. , "Machine Learning DDoS Detection Using Stochastic Gradient Boosting," *Int. J. Comput. Sci. Eng.,* vol. 7, no. 4, pp. 157-16, 2019, doi: 10.26438/ijcse/v7i4.157166.

[11] A. Patil and D. Kshirsagar, "Towards feature selection for detection of DDoS attack," in *Comput. Eng. Technol.*, 2020, pp. 215-223, doi: 10.1007/978-981-32-9515-5_21.

[12] F. S. d. Lima Filho, F. A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, and L. F. Silveira, "Smart detection: an online approach for DoS/DDoS attack detection using machine learning," *Secur. Commun. Networks,* vol. 2019, 2019, doi: 10.1155/2019/1574749.

[13] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *Int. J. Distrib. Sens. Networks,* vol. 13, no. 12, p. 1550147717741463, 2017, doi: 10.1177/1550147717741463.

[14] R. Jamar, A. Sogani, S. Mudgal, Y. Bhadra, and P. Churi, "E-shield: Detection and prevention of website attacks," in *2017 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol.*, 2017, pp. 706-710: IEEE, doi: 10.1109/RTEICT.2017.8256689.

[15] Devendra, *DDoS Dataset: DDoS Balanced & Unbalanced Datasets*, 2019, available at: kaggle.com

[16] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *ICISSP*, 2018, pp. 108-116, doi: 10.5220/0006639801080116.

[17] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling," *Comput. Networks,* vol. 121, pp. 25-36, 2017, doi: 10.1016/j.comnet.2017.03.018.

[18] J. J. E. M. Geelan, http://virtualization. sys-con. com/node/612375, "Twenty one experts define cloud computing. Virtualization," 2008, Available at: Google Scholar

[19] F. Koumi, M. Aldasht, and H. Tamimi, "Efficient Feature Selection using Particle Swarm Optimization: A hybrid filters-wrapper Approach," in *2019 10th International Conference on Information and Communication Systems (ICICS)*, 2019, pp. 122-127: IEEE, doi: 10.1109/IACS.2019.8809133.

[20] L. Brezočnik, "Feature selection for classification using particle swarm optimization," in *IEEE EUROCON 2017-17th International Conference on Smart Technologies*, 2017, pp. 966-971: IEEE, doi: 10.1109/EUROCON.2017.8011255.

[21] S.Sandhiya and D. U. Palani, "A Novel Hybrid PSBCO Algorithm for Feature Selection," *Int. J. Comput. Organ. Trends,* vol. 10, no. 3, May-June 2020 2020, doi: 10.14445/22492593/IJCOT-V10I3P305.

[22] A. E. Hassanien and E. Emary, *Swarm intelligence: principles, advances, and applications*. CRC Press, 2018, doi: 10.1201/9781315222455

[23] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Appl. Sci.,* vol. 8, no. 9, p. 1521, 2018, doi: 10.3390/app8091521.

[24] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43: IEEE, doi: 10.1109/MHS.1995.494215.

[25] J. Barrera and C. A. C. Coello, "A review of particle swarm optimization methods used for multimodal optimization," in *Innovations in swarm intelligence*: Springer, 2009, pp. 9-37, doi: 10.1007/978-3-642-04225-6_2.

[26] M. Mafarja, R. Jarrar, S. Ahmad, and A. A. Abusnaina, "Feature selection using binary particle swarm optimization with time varying inertia weight strategies," in *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, 2018, pp. 1-9, doi: 10.1145/3231053.3231071.

[27] Y. Chen, L. Li, J. Xiao, Y. Yang, J. Liang, and T. Li, "Particle swarm optimizer with crossover operation," *Eng. Appl. Artif. Intell.,* vol. 70, pp. 159-169, 2018, doi: 10.1016/j.engappai.2018.01.009.

[28] B. Xue, M. Zhang, and W. N. Browne, "New fitness functions in binary particle swarm optimisation for feature selection," in *2012 IEEE congress on evolutionary computation*, 2012, pp. 1-8: IEEE, doi: 10.1109/CEC.2012.6256617.

[29] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, 1997, vol. 5, pp. 4104-4108: IEEE, doi: 10.1109/ICSMC.1997.637339.

[30] B. Tran, B. Xue, and M. Zhang, "Improved PSO for feature selection on high-dimensional datasets," in *Asia-Pacific Conference on Simulated Evolution and Learning*, 2014, pp. 503-515: Springer, doi: 10.1007/978-3-319-13563-2_43.

[31] S. M. Vieira, L. F. Mendonça, G. J. Farinha, and J. M. Sousa, "Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients," *Appl. Soft Comput.,* vol. 13, no. 8, pp. 3494-3504, 2013, doi: 10.1016/j.asoc.2013.03.021.

[32] J. Too, A. R. Abdullah, N. Mohd Saad, and W. Tee, "EMG feature selection and classification using a Pbest-guide binary particle swarm optimization," *Computation,* vol. 7, no. 1, p. 12, 2019, doi: 10.3390/computation7010012.

[33] K. J. Singh and T. De, "Efficient classification of DDoS attacks using an ensemble feature selection algorithm," *J. Intell. Syst.,* vol. 29, no. 1, pp. 71-83, 2017, doi: 10.1515/jisys-2017-0472.

[34] Y. S. Hussain, "Network Intrusion Detection for Distributed Denial-of-Service (DDoS) Attacks using Machine Learning Classification Techniques," 2020. Available at: https://dspace.library.uvic.ca/bitstream/handle/1828/11679/Hussain_Yasar%20Shahid_MEng_2020.pdf?sequence=3&isAllowed=y.

[35] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based DDoS detection system in software-defined networking (SDN)," *arXiv preprint arXiv:.07400,* 2016, doi: 10.4108/eai.28-12-2017.153515.