Received: February 26, 2023. Revised: May 10, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Novel Approach of DDOS Attack Classification with Optimizing the Ensemble Classifier Using A Hybrid Firefly and Particle Swarm Optimization (HFPSO)

Anuradha Pawar[1]* Nidhi Tiwari[1]

[1]*Department of Electrical and Communication Engineering, Sage University, Indore, Madhya Pradesh, India*
* Corresponding author's Email:er.anuradhapawar@gmail.com

Abstract: Cyberspace is fraught with dangers, one of which is the distributed denial of service attack (DDoS). This type of attack is particularly concerning as it can disrupt vital services, prevent authorized users from accessing them, and result in financial losses. The aim of this research is to present an optimized AdaBoost classifier that has been fine-tuned using the HFPSO algorithm. The data is pre-processed to ensure that it conforms to standard features by normalizing it. Additionally, cross-correlation techniques are used to select features in order to eliminate redundancy. Finally, the constructed signals are used to train and test an HFPSO-optimized AdaBoost classifier. The result indicates the possibility of anticipating the attacks is fairly accurate. The system accuracy is 99.97%.

Keywords: Adaboost, DDoS, HFPSO, Firefly algorithm, PSO.

1. Introduction

The internet and advancements in networking technology have enabled connections between individuals from all corners of the world. With the help of various applications and services, it has provided an impetus for the development of innovative businesses. Consequently, computer network-related services have become increasingly important in both proficient and individual spheres. On the other hand, this rising significance has also piqued the interest of cybercriminals who seek to launch attacks for individual gain. The distributed denial of service (DDoS) attack ranks as one of the most substantial threats in cyberspace [1]. This type of cyber-attack involves flooding a targeted website or server with traffic, rendering it unavailable to authorized users, using a large number of compromised systems. Such attacks can lead to service interruptions and are now commonly sold as a service [2] to individuals who wish to cause harm to their adversaries.

DDoS attacks pose a severe threat to cybersecurity defenses as they launch a high volume of traffic suddenly and with great aggression, giving security measures little time to react. In 2022, an attack of this nature even compromised Internet access for almost all of Andorra's population [3]. The attack often entails leveraging numerous hacked devices to transmit millions of messages, connection requests, or corrupted packets to a target in a consistent way. The goal is to use the victim's computing assets, decrease functionality, and deny service to authentic users [4, 5]. Defence tools must quickly recognize the attack to adopt the necessary measures to contain it and reduce the damage.

Various strategies can be employed to counter DDoS attacks, such as prevention, mitigation, detection, reaction, and prediction [6]. This study focuses on predicting DDoS attacks, as it has been developed to complement other defense mechanisms and provide more reaction time to victims. Deep learning is a subdivision of machine learning that has been used to create cybersecurity solutions against different types of threats, including DDoS attacks [7].

The proposed system analyzes victim's network traffic using supervised machine learning to look for early warning signals (EWS), which are used to identify sudden changes in dynamic systems and anticipate phenomena [8]. The contribution of this study is to propose the optimal AdaBoost classifier tuned by the HFPSO algorithm. Firstly, data is pre-processed by normalizing it to establish a standard feature set. Further features are selected using cross-correlation techniques to avoid redundancy. To construct the signals, the skewness and kurtosis values of each attribute are calculated for one-second traffic windows. Finally, the constructed signals are used to train and test the HFPSO-optimized AdaBoost classifier, evolving the literature and increasing defence time against DDoS attacks.

The combination of firefly and particle swarm optimization algorithms has been used in other research papers. However, the innovation of this paper lies in its application to the specific problem of DDoS attack classification. This paper proposes a novel approach that optimizes the ensemble classifier using the HFPSO algorithm, which has not been previously used for DDoS attack classification. This approach is expected to improve the accuracy of DDoS attack classification and reduce the number of false positives, which are significant challenges in traditional classification methods. The proposed approach also considers the feature selection process to identify the most relevant features for classification, which is crucial in enhancing the classification performance. Furthermore, the study evaluates the proposed approach using a benchmark dataset and compares it with other state-of-the-art methods to demonstrate its effectiveness. The proposed approach's novelty lies in its optimization process and the feature selection method's incorporation to improve the accuracy and efficiency of DDoS attack classification.

This research is structured in the following manner. The second section presents a review of the literature on anticipating DDoS attacks. In section 3, a comprehensive explanation of the proposed attack categorization approach is given. The fourth section contains the system evaluation, followed by the study's conclusion in section 5.

2. Literature review

In [8], a system is presented that analyzes data found on the Internet, particularly social media texts such as tweets, to identify potential cyber threats. The focus is on identifying campaigns to spread malware or targets of possible attacks by filtering tweets related to DDoS attacks. The researchers behind [9] utilize information acquired from the internet, including records of reputation and security occurrences, to instruct the support vector machine (SVM) and foresee security incidents. They anticipate events linked to cybersecurity with a true positive rate averaging at 69%, but do not concentrate on DDoS attacks. In their work [10], the authors endeavor to recognize and model the standard conduct of botnets in a Markov chain to forecast attacks dependent on the probability of change from the existing state to an attack state. Their system predicts C&C communication with 99% precision, which could be used to predict DDoS attacks in the future. The attack forecast can differ from a couple of seconds to 18 hours before the onset of the attack. The authors of [11] employ notifications created by an intrusion detection system (IDS) to forecast attacks using the hidden markov chain (HMM). They carry out offline training, during which the system matches IDS notifications with formerly demarcated historical data. This technique forecasts a DDoS attack about 11 minutes before it occurs.

A real-time detection system that applies machine learning methods like naive Bayes, SVM, and multilayer perceptron was introduced by the authors of [12]. They employed a dataset that included 5 different categories and 27 characteristics and discovered that the J48 algorithm had 98.64% greater accuracy than MLP, Naive Bayes, and Random Forest [20, 22]. They found that DDoS is a prevalent attack that inflicts damage to network resources utilized by legitimate users. A new detection structure for DDoS that includes the usage of Bi-Directional long short-term memory and the Gaussian mixture model is put forth by the authors of [13]. They conclude that this framework is effective in detecting DDoS attacks. They also acknowledge the difficulty of accurately detecting new instances of DDoS attacks that do not fit the distribution of the training data, a problem known as open set recognition. The authors of [14] focused on detecting DDoS attacks in the cloud using various machine learning models. They found that the ensemble model achieved an accuracy of 97.86% using 16 attributes obtained from the regression analysis and feature selection. In [15], the authors proposed a novel model and detection method using RDF-SVM to detect both known and unknown DDoS attacks. They evaluated the algorithm using experimental results and found that the algorithm performed better with optimal features.

In their study, the authors of [16] utilized a range of machine learning algorithms, including artificial neural networks (ANN), K-Means, and logistic regression (LR), to identify DDoS attacks. They obtained a promising accuracy of 94.00% after performing data preprocessing and manipulation processes. The study conducted in [17] presents a proposed system for anomaly detection that utilizes LSTM (long short-term memory) with an attention mechanism (AM) to improve network training performance. The CIC-IDS 2018 data set was utilized

Received: February 26, 2023. Revised: May 10, 2023.

for training the proposed system, and the results analysis reported an accuracy of 96.22%. Additionally, the detection rate was reported as 15%, while the recall rate was 96%. A technique for identifying different types of attacks is suggested by the authors of [18]. The approach involves ranking the detection capacity of classifiers and constructing an ensemble. This approach aims to improve the accuracy of attack detection by combining the strengths of multiple classifiers in an ensemble. In contrast, the authors of [19] recommended utilizing a broad learning system (BLS) to identify DoS attacks in telecommunication networks. The BLS is reported to achieve good performance with less training time, which can be beneficial in real-world scenarios where timely detection of attacks is crucial. The paper [20] proposes a system for detecting network intrusions using deep learning technology, specifically utilizing the LSTM method to build a neural network. The system is trained and tested on the CSE-CIC-IDS2018 real dataset to detect intrusions throughout data flow. Paper [21] introduced a feature reduction method that utilized time comparison on CICIDS 2017 with PART to decrease the number of features in CICIDS 2017 and KDDCup 99 from 77 to 24 and 41 to 12, respectively. This resulted in an enhanced accuracy of 99.95% using the PART classifier. On the other hand, in article [22], the authors proposed a combination of the grasshopper optimization algorithm (GOA) with a machine learning algorithm, GOIDS, to construct an intrusion detection system (IDS) that can effectively distinguish between normal and attack traffic in a monitored environment. The GOA is a nature-inspired optimization algorithm that mimics the search behaviour of grasshoppers in finding optimal solutions. In this article, it is used as a method to enhance the efficiency and accuracy of the IDS. In [23] the results reported in the paper indicate that the proposed IDS achieves a high detection accuracy of 97.59%. This suggests that the DNN-based approach is effective in accurately identifying DDoS attacks in real-time. Additionally, the proposed IDS is designed to use fewer resources and less time, which can be beneficial in terms of computational efficiency and practical implementation in SDN environments. The paper [24] introduces a framework that combines different classifier methods, including K-Nearest neighbour (KNN) classifier, Naïve Bayes classifier, Adaboost with decision tree classifier, support vector machine (SVM) classifier, random forest classifier, and Artificial Intelligence techniques, for detecting botnet attacks on the CSE-CIC-IDS2018 dataset, which is a recent and genuine cyber dataset. The framework utilizes these classifiers to learn and identify patterns and features that are indicative of botnet attacks in the dataset. By combining multiple classifier methods in the framework, the proposed approach aims to enhance the accuracy and effectiveness of botnet attack detection on the CSE-CIC-IDS2018 dataset. The results of the classification achieved is nearly 99 %. Overall, in the paper [25] presents a robust approach for DoS attack detection using RF and MLP models, implemented with the Scikit ML library and the Spark ML library, achieving high accuracy and optimized prediction time. The results reported in the paper indicate that both the RF and MLP models achieved a high mean accuracy of 99.5% for the detection of DoS attacks, both with and without the use of the Spark ML library. The paper [26] proposes a novel classification model that combines Kernel extreme learning machine (KELM) with enhanced grey wolf optimizer (EGWO) for network intrusion detection. The model leverages the dimensionality reduction ability of deep belief network (DBN) to extract features from complex and high-dimensional network intrusion data.

To further evolve the area of DDoS attack prediction, metaheuristic algorithms can be used to present new solutions with good results, overcoming the limitation of using labeled data.

The drawback of each traditional approach employed in the literature review is explained below:

- The drawback of utilizing support vector machine for classifying DDOS attacks is that it may face difficulty in efficiently handling significant amounts of data, resulting in extended processing times and reduced accuracy. Moreover, identifying suitable kernel functions and parameters for SVM-based classification can be a demanding task that can impact its ability to accurately detect DDOS attacks.
- The drawback of employing hidden Markov chain for classifying DDOS attacks is that it may not be capable of capturing the full dependencies between network traffic patterns, resulting in a decrease in accuracy in detecting DDOS attacks. Moreover, the efficacy of hidden Markov chain-based classification can be limited by the appropriate selection of model parameters and the number of states used, which can affect its ability to precisely identify DDOS attacks.
- The drawback of using Naive Bayes for DDOS attack classification is that it entails a sufficient extent of training data to estimate the probabilities of different features accurately. In cases where training data is limited or unrepresentative of the actual network traffic,

Naive Bayes may not be able to effectively model the underlying patterns and accurately identify DDOS attacks. Moreover, Naive Bayes can be sensitive to irrelevant or redundant features, which can negatively impact its performance in classifying DDOS attacks.

- The drawback of using multilayer perceptron (MLP) for DDOS attack classification is that it can be prone to overfitting if the size of the training dataset is small or unrepresentative of the actual network traffic, leading to reduced generalization performance. Furthermore, MLP may require extensive computational resources and longer training times compared to other machine learning models, which can limit its scalability and practicality for real-time DDOS attack classification.
- The interpretability of random forest can be limited, making it challenging to gain insights into the underlying network traffic patterns and the features that are most relevant for detecting DDOS attacks.
- The drawback of using artificial neural networks (ANNs) for DDOS attack classification is that they can be sensitive to the choice of network architecture and hyperparameters, which can affect their performance in detecting DDOS attacks accurately. Moreover, ANNs can be computationally expensive and require significant computational resources to train and optimize, which can limit their scalability and practicality for real-time DDOS attack classification.
- The drawback of using K-Means for DDOS attack classification is that it entails preceding knowledge of the number of clusters, which can be challenging to determine for complex and dynamic network traffic patterns. Furthermore, the performance of K-Means can be affected by the initialization of cluster centroids, and it may result in suboptimal solutions if the initialization is not suitable.
- The drawback of using logistic regression for DDOS attack classification is that it supposes a direct correlation between the features and the target output, which may not hold in complex and non-linear network traffic patterns. This can cause reduced precision in detecting DDOS attacks, especially when the connection between the target variable and the features is nonlinear.

A hybrid firefly and particle swarm optimization (HFPSO) optimized ensemble classifier is one approach to detect DDoS attacks. This approach involves using an HFPSO optimization algorithm to optimize the combination of multiple classifiers, such as bagging and boosting, to improve the accuracy of the detection system. This type of classifier can also be trained using historical data on past DDoS attacks to better recognize patterns and identify new, previously unseen attacks.

The research paper addresses the problem of preventing DDoS attacks in cyberspace. The paper proposes an optimized AdaBoost classifier, fine-tuned using the HFPSO algorithm, for forecasting DDoS attacks accurately. The paper also explores the inadequacies of conventional k-fold cross-validation approaches when evaluating machine learning models for predicting properties of materials that are beyond the scope of the training set. To overcome these drawbacks, the paper introduces a novel series of k-fold validation approaches and a new accuracy metric for exploration to evaluate exploratory power. The problem, therefore, is to improve the accuracy of DDoS attack prediction and to develop effective machine learning models for predicting properties of materials beyond the training set.

3. Proposed methodology

Fig. 1 represents the proposed structure of DDOS attack detection model. Once data is pre-processed then optimal features are selected form feature selection methods. Further data is divided into two parts training and testing data according to their learning rate. It can be seen in Fig. 1 trained model is further evaluated with test data and performance parameter is calculated with performance metrics.

3.1 Pre-processing

Data mining for the prediction of DDoS attacks requires a crucial initial step of pre-processing, which deals with various issues such as irregularities, errors, and missing values in the actual databases. Knowledge discovery in databases (KDD) plays a critical role in making these databases suitable for data mining processes.

3.2 Data normalization

Feature vector attributes in data mining processes are standardized to the same scale through the process of normalization. It is essential to maintain the implicit information of each attribute while ensuring that the distribution of data remains the same. Typically, the normalized data has a range between 0 and 1 or -1 and 1. This process is crucial in making data suitable for neural network training, as it helps prevent neuron saturation and speed up classifier

Received: February 26, 2023. Revised: May 10, 2023.

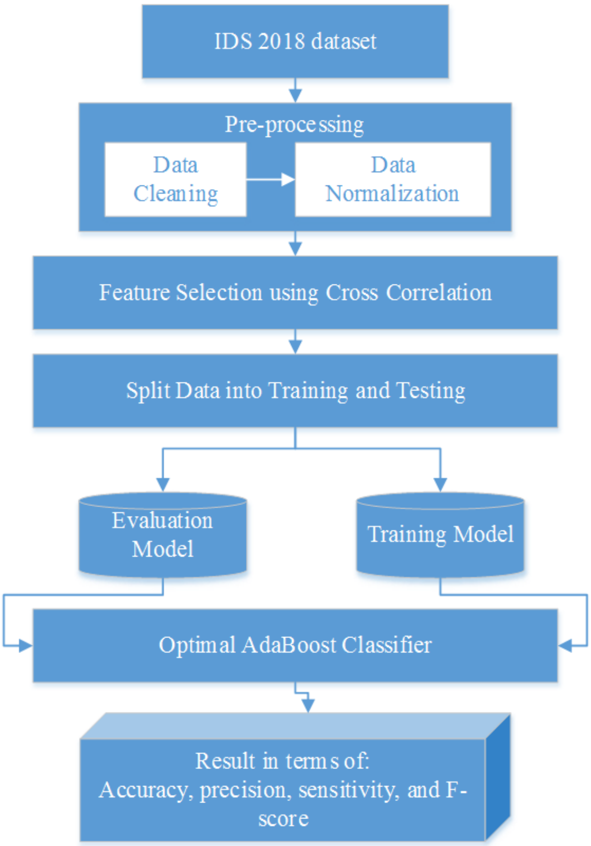
IDS 2018 dataset
Pre-processing
Data Cleaning
Data Normalization
Feature Selection using Cross Correlation
Split Data into Training and Testing
Evaluation Model
Training Model
Optimal AdaBoost Classifier
Result in terms of:
Accuracy, precision, sensitivity, and F-score

Figure. 1 Proposed flow diagram

training time. This process not only increases precision but also improves the efficiency of mining algorithms that employ distance measurements.

3.2.1. Data Normalization with z-score

To normalize the data, z-score normalization can be used. This method involves calculating the standard deviation and mean of each attribute in the feature vector, and then normalizing each attribute based on these values using Eq. (1):

$$x = \frac{x-\mu_x}{\sigma_x} \tag{1}$$

Where σ_x is the standard deviation, and μ_x is the mean of the attribute values.

3.3 Feature selection

In order to build machine learning models, it is important to perform feature selection which involves selecting the most relevant features from a vast pool of potential features to obtain high classification accuracy. This holds true for DDOS attack classification models as well. One effective approach for feature selection is cross-correlation analysis, which is used to identify correlations between features and the target variable. In essence, cross-correlation analysis measures the similarity between two signals, where one signal is the feature and the other is the target variable. This analysis is done by applying a time lag to one of the signals and measuring the correlation between them. Cross-correlation analysis is a standard method utilized in time series analysis, signal processing, and other applications.

To identify the most informative features for classification, cross-correlation analysis is employed in feature selection. It is utilized to compute the correlation between each feature and the target variable. The aim is to identify the features that have high correlation with the target variable since these features are likely to be the most informative for classification.

The cross-correlation between a feature vector x and the target variable y can be computed utilizing the subsequent formula:

$$C_{xy}(l) = \frac{1}{N}\sum_{i=1}^{N-1}(x_i y_{i-l})$$

Where $C_{xy}(l)$ is the cross-correlation at time lag l, N is the length of the vectors x and y. The time lag l represents the delay between the feature vector and the target variable, and can take positive or negative values.

To conduct feature selection using cross-correlation, we begin by calculating the cross-correlation between the target variable and each feature at different time lags. Based on the resulting cross-correlation values, we rank the features according to their correlation with the target variable, selecting those with the highest correlations for use in the classification model.

However, it is crucial to bear in mind that cross-correlation analysis only identifies linear correlations between features and the target variable. Nonlinear relationships may exist, which cross-correlation analysis may not capture. Therefore, to achieve optimal results, it is often necessary to use other feature selection techniques in conjunction with cross-correlation analysis.

3.4 Classification

The hybrid firefly and particle swarm optimization (HFPSO) method is a new approach to optimizing ensemble classifiers that associates the firefly algorithm [27] and PSO [28] to discover the best arrangement of individual classifiers in an ensemble. The firefly algorithm is an optimization technique inspired by the flashing patterns of fireflies, while the PSO is a bio-inspired optimization method that mimics the behaviour of flocks of birds or swarms of insects. The HFPSO method integrates these two algorithms by using firefly movement to update particle positions and velocities in the PSO algorithm. The ensemble is represented as a swarm, where each particle represents an individual classifier, and the fitness of each particle is evaluated based on its accuracy on a validation set.

At the outset, the procedure initiates by creating a population of particles, with each particle representing a potential ensemble classifier. The binary representation of the presence or absence of each individual classifier in the ensemble denotes the position of every particle. Each particle's velocity is updated depending on the best solution it has discovered so far and the best solution found by the swarm. To update the position of each particle, the firefly algorithm comes into play. The attractiveness of the particle's light, which depends on its fitness and the distance to other particles, determines the firefly's movement. AdaBoost ensemble classifier, a machine learning algorithm, builds a strong classifier by combining multiple weak classifiers. The weak classifiers are trained on several subsets of data and are weighted depending on classification accuracy. Finally, the decision for the classification is made by a weighted sum of the weak classifiers.

The mathematical formulation of the AdaBoost ensemble classifier can be represented as follows:

Let X be the input data, Y be the output labels, and M be the number of weak classifiers. The AdaBoost algorithm assigns an importance weight α_m to each weak classifier h_m , and iteratively updates the weights based on the classification accuracy. At each iteration t, the weights for the next weak classifier h_m are calculated as follows:

1. The probability distribution D_t is updated over the training set (X, Y) so that the misclassified samples in the previous iterations receive higher weights.
2. The weak classifier h_m is trained on the weighted training set.
3. The error ε_m is calculated as the weighted sum of the misclassified samples:

$$\varepsilon_m = \sum_{i=1}^{N} D_t(i)\left(1 - Y(i)h_m\big(X(i)\big)\right) \tag{2}$$

4. The weight α_m is calculated as:

$$\alpha_m = \frac{1}{2}\ln\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right) \tag{3}$$

5. The weights in D_t are updated as:

$$D_{t+1}(i) = D_t(i)\mathrm{e}^{-\frac{\alpha_m Y(i) h_m(X(i))}{Z_t}} \tag{4}$$

Where Z_t is a normalization constant that ensures that D_{t+1} is a valid probability distribution.

6. The weighted sum of the weak classifiers is utilized to make the final classification decision:

$$H(X) = sign \sum_{m=1}^{M} \alpha_m h_m(X) \tag{5}$$

The goal of optimizing the AdaBoost ensemble classifier using a hybrid firefly and particle swarm optimization (HFPSO) is to find the optimal set of hyperparameters for the weak classifiers, such as the maximum number of trees in a random forest classifier. Iteratively modifying the velocity and position of each firefly and particle based on their respective fitness values, the HFPSO technique explores the hyperparameter space.

The objective function for optimizing the AdaBoost ensemble classifier using the HFPSO algorithm is the accuracy of the classifier.

Let y be the true class labels of the samples, and $\hat{y}$ be the forecast class labels of the samples using the AdaBoost classifier. Then, the accuracy (ACC) can be defined as:

$$ACC = \frac{(number\ of\ correctly\ classified\ samples)}{(total\ number\ of\ samples)} \tag{6}$$

$$ACC = \frac{\sum_{i=0}^{N}(y_i = \hat{y}_i)}{N} \tag{7}$$

Where N is the total number of samples.

The objective function is a measure used to assess the fitness of the potential solutions (i.e., hyperparameters) generated by the fireflies and particles in the search space, in the context of the HFPSO algorithm. The accuracy of the corresponding AdaBoost classifier, which is trained using the hyperparameters, determines the fitness of a candidate solution. The HFPSO algorithm searches for the optimal hyperparameters that maximize accuracy (i.e., minimize error rate) using the objective function as a guide.

The mathematical formulation of the HFPSO algorithm for optimizing the AdaBoost ensemble classifier is as follows:

1. Initialization
    - Set the maximum number of iterations T and the stopping criterion.
    - Generate an initial population of N solutions.

- Set the initial values for the parameters of the AdaBoost classifier.

2. Repeat for $t = 1$ to T or until the stopping criterion is met:
   a. Particle swarm optimization (PSO) Phase
   - Update each particle's velocity depending on its personal best solution and the swarm's best solution.

$$v_i(t) = wv_i(t-1) + c_1 r_1 \left(pbest_i - x_i(t-1)\right) + c_2 r_2 \left(gbest - x_i(t-1)\right) \quad (8)$$

   - Each particle's location is updated dependent on its velocity:

$$x_i(t) = x_i(t-1) + v_i(t) \quad (9)$$

   b. Firefly algorithm (FA) phase
   - Update the position of each solution based on the attractiveness of its light and the distance to other solutions:

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma d\left(x_i(t), x_j(t)\right)\left(x_{j(t)} - x_{i(t)}\right)} + \alpha r \quad (10)$$

   - Calculate the fitness of each solution:

$$f_i(t) = CV\left(x_i(t)\right) \quad (11)$$

   c. Update each particle's velocity depending on its personal best solution and the swarm's best solution:

$$v_i(t+1) = v_i(t) + wv_i(t-1) + c_1 r_1 \left(pbest_i - x_i(t)\right) + c_2 r_2 \left(gbest - x_i(t)\right) \quad (12)$$

   d. Update each particle's personal best solution and the swarm's global best solution:

$$\begin{aligned} &if\ f_i(t) < f_{pbest_i} \\ &pbest_i = x_{i(t)} \\ &if\ f_{i(t)} < f_{gbest} \\ &gbest = x_i(t) \end{aligned} \quad (13)$$

   e. Adjust the parameters of the AdaBoost classifier according to the best solution found: Update AdaBoost Parameters using $gbest$.
   f. Increase the iteration counter $t = t + 1$

The HFPSO algorithm employs several parameters, including N for population size, t for the current iteration, w for the inertia weight, c_1 and c_2 for acceleration coefficients, r, r_1 and r_2 are random numbers, $pbest_i$ and $gbest$ for personal best and global best solutions, β_0 and γ for the attractiveness and distance parameters of the firefly algorithm, and α for a random perturbation term. The Hamming distance between the solutions $x_i(t)$ and $x_j(t)$ is determined by the function $d\left(x_i(t), x_j(t)\right)$, while $CV\left(x_i(t)\right)$ is the fitness function, calculated by means of cross-validation.

It can be seen in Fig. 2 the flow of HFPSO for optimization of AdaBoost classifier. The HFPSO algorithm combines the PSO and FA algorithms for performance improvement of the AdaBoost classifier. The PSO algorithm updates the velocity of each particle by considering both its own best solution and the best solution discovered by the swarm, while the FA algorithm updates the position of each solution based on the attractiveness of its light and the distance to other solutions. The HFPSO algorithm optimizes the parameters of the AdaBoost classifier by updating the personal best and global best solutions found by the swarm, and using the best solution to update the AdaBoost parameters.

4. Simulation and results

According to the literature, using a value of K equal to 10 provides a reliable estimate of classifier accuracy [24]. However, this approach can be computationally expensive since it requires K trainings to evaluate the model [24].

4.1 Dataset

4.1.1. IDS 2018 Dataset

The IDS 2018 Dataset [29] is a publicly available collection of network traffic data resulting from various types of DDoS attacks . Its purpose is to serve as a benchmark for evaluating the efficacy of intrusion detection systems (IDS) in detecting and mitigating DDoS attacks, and is intended for use by security professionals and researchers. The dataset was produced by capturing traffic from a testbed network that emulates multiple DDoS attack scenarios. The testbed comprises a victim machine and several attacker machines under the control of a master machine. The victim machine runs a range of services such as HTTP, SSH, FTP, and DNS, while the attacker machines produce traffic aimed at the victim machine in order to overload it and cause a denial-of-service. The IDS 2018 Dataset was developed for the advancement and evaluation of IDS

Received: February 26, 2023. Revised: May 10, 2023.

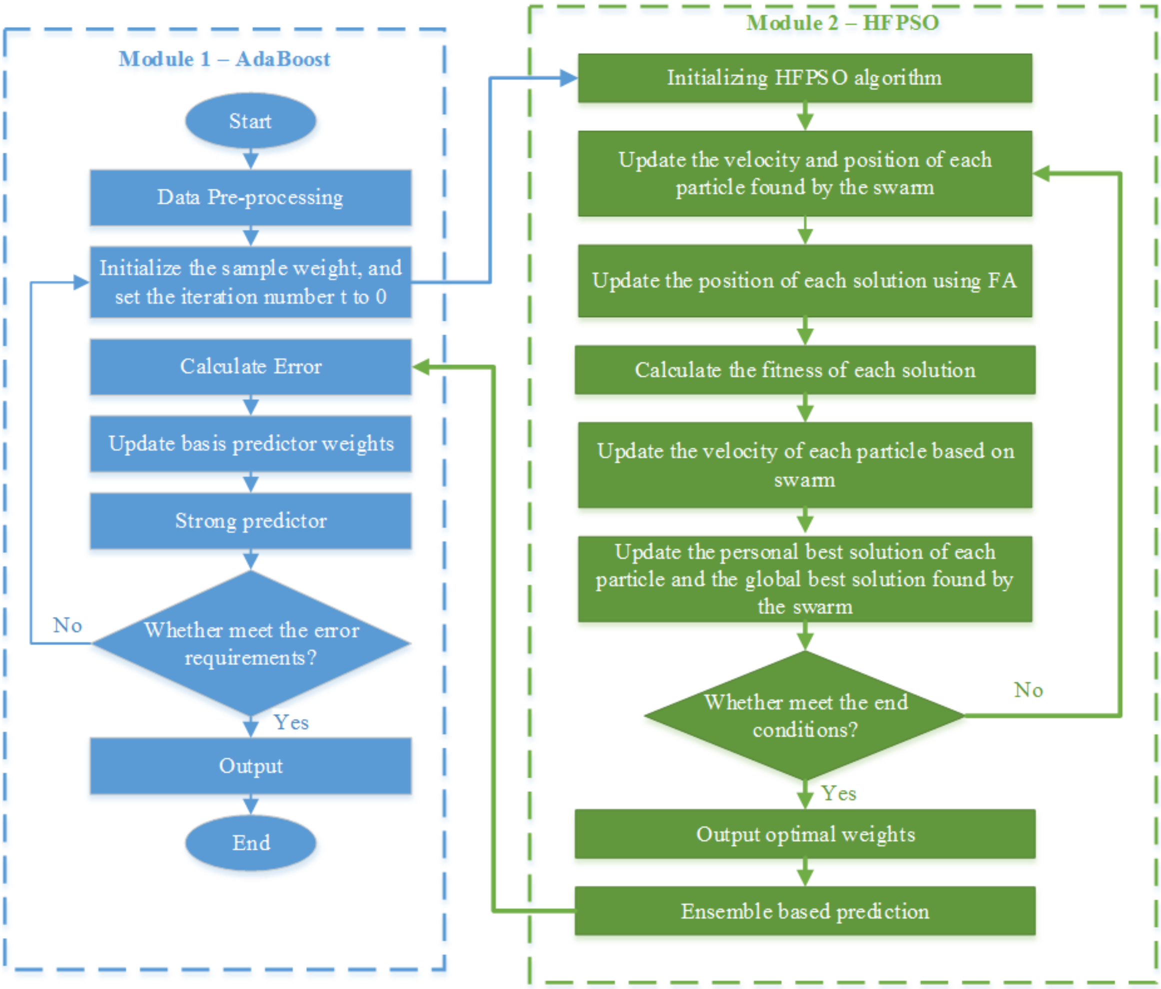
Module 1 – AdaBoost
Start
Data Pre-processing
Initialize the sample weight, and set the iteration number t to 0
Calculate Error
Update basis predictor weights
Strong predictor
No
Whether meet the error requirements?
Yes
Output
End
Module 2 – HFPSO
Initializing HFPSO algorithm
Update the velocity and position of each particle found by the swarm
Update the position of each solution using FA
Calculate the fitness of each solution
Update the velocity of each particle based on swarm
Update the personal best solution of each particle and the global best solution found by the swarm
Whether meet the end conditions?
No
Yes
Output optimal weights
Ensemble based prediction

Figure. 2 HFPSO algorithm for optimizing the AdaBoost ensemble classifier

systems designed for detecting DDoS attacks. It can be utilized to train machine learning algorithms to identify patterns in network traffic indicative of a DDoS attack. Additionally, the dataset can be used to assess the effectiveness of current IDS systems in detecting and mitigating DDoS attacks.

4.1.2. Brute force

The Brute force dataset is a publicly available dataset designed for the classification of DDoS attacks resulting from brute force attacks. In these types of cyber-attacks, an attacker tries to gain access to a system by guessing login credentials or passwords through the use of automated scripts or tools. The Brute force dataset is developed by generating network traffic in a testbed network that simulates a realistic environment of brute force attacks. This testbed network consists of a victim machine running various services and an attacker machine that generates traffic aimed at the victim machine with the aim of overwhelming it and causing a denial-of-service. The dataset is divided into two classes of network traffic: legitimate network traffic generated by regular user activity and DDoS attack traffic generated by brute force attacks.

4.1.3. DoS Slowloris

The DoS slowloris dataset is a publicly accessible dataset used for classifying DDoS attacks that are generated by slowloris attacks. Slowloris is a category of DDoS attack that attempts to overload a web server by sending HTTP requests in small fragments at a slow pace, causing the server to keep the connection open. The DoS slowloris dataset is produced by creating network traffic in a testbed network that simulates a real-world scenario of slowloris attacks. The testbed network comprises a victim machine that operates multiple web services and an attacker machine that generates traffic aimed at the victim machine to inundate it and create a denial-of-service. The dataset includes two types of network traffic: legitimate network traffic and DDoS attack traffic. The legitimate network traffic category contains network traffic generated by typical user behavior, while the DDoS attack traffic category includes network traffic generated by slowloris attacks. The dataset is publicly accessible to facilitate the development and assessment of machine learning algorithms for identifying and mitigating DDoS attacks. By training machine learning models on the dataset, researchers can recognize traffic patterns indicative of a slowloris attack and employ this knowledge to identify and prevent future attacks. Additionally, the dataset can be used to evaluate the effectiveness of existing DDoS detection and mitigation tools and methods.

4.1.4. Http unbearable load king (HULK)

HULK is a type of DoS attack that functions in a similar manner to an HTTP flood. The main goal of this attack is to overwhelm web servers by continuously requesting one or multiple URLs. HULK generates a unique pattern on each request, which helps increase the load on the servers and evade intrusion detection and prevention systems.

4.2 K-fold cross validation

Cross-validation using the K-fold technique is an extensively adopted approach in the area of machine learning for appraising model efficacy. The key principle underlying this methodology is to bifurcate the dataset into k equivalent parts. Subsequently, the model undergoes training on k-1 folds while being evaluated on the remaining fold. The process iterates k times, where each partition serves as the validation set once, and the residual segments are used for training. The model's comprehensive efficiency is evaluated by computing the average of performance indicators obtained in each of the k runs. The adoption of K-fold cross-validation reduces performance estimate variances relative to a single train/test split, thus rendering a more dependable evaluation of the model's efficiency. Fig. 3 provides a better illustration of how this technique works.

4.2.1. Forward holdout validation

Within the domain of machine learning, forward holdout validation represents a mechanism utilized to evaluate a model's performance on a given dataset. This approach entails partitioning the dataset into two distinct subsets, namely the training set and the test set. The training set is utilized to impart knowledge to the model, whereas the test set is employed to evaluate the model's efficacy. Divergent from k-fold cross-validation, wherein the dataset is distributed into k equal segments and each segment is exploited as a test set, forward holdout validation solely employs a solitary test set. The test set is chosen at the beginning of the experiment and is not modified during the training process. The benefit of using a single test set is that it provides a more realistic assessment of the model's performance. This is because the test set is independent of the training set and more closely represents real-world data that the model may encounter.

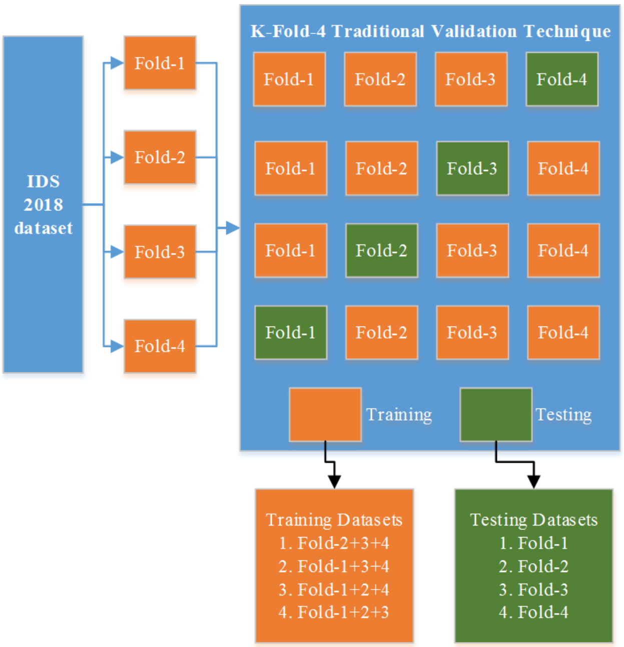

Figure. 3 K-Fold-4 validation technique

Table 1. Simulation parameters for HFPSO

Firefly	PSO
alpha=0.2;	$c1$, $c2$ = 1.49445
beta0=2;	Inertia weight (w), wi = 0.9, wf = 0.5
m=2;	--
gamma=1;	--
alpha=0.2;	--

4.3 Simulation parameters

4.4 Simulation results

The analysis in Fig. 4 compares the roc curve of an ensemble classifier that uses both AdaBoost and bagging. The graph shows that the HFPSO optimized ensemble classifier outperforms the traditional classifier with a higher accuracy. In Fig. 5 and Figure 6, the confusion matrix plots for 3-class and 2-class DDOS attack classifications are presented. Fig. 5 shows that there was one misclassification, while in the 2-class classification, there was no misclassification. As the classification labels increase, there is a greater likelihood of misclassification.

Received: February 26, 2023. Revised: May 10, 2023.

Table 2. Performance evaluation of DDOS attack under different classifiers

Parameters	SVM	KNN	HFPSO Ensemble (Bagging)	Ensemble Classifier	HFPSO Ensemble (AdaBoost)
Accuracy	0.9800	0.9990	0.9996	0.9995	0.9997
Error	0.0200	0.0016	4.5113e-04	7.3801e-04	3.6908e-04
Sensitivity	0.9818	0.9713	0.9993	0.9991	0.9995
Specificity	0.9951	0.9989	0.9998	0.9996	0.9998
Precision	0.9800	0.9713	0.9995	0.9991	0.9995
False Positive Rate	0.0049	0.0011	0.2851e-04	3.8204e-04	1.9104e-04
F1_score	0.9799	0.9713	0.9994	0.9991	0.9995
Matthews Correlation Coefficient	0.9757	0.9703	0.9991	0.9987	0.9994

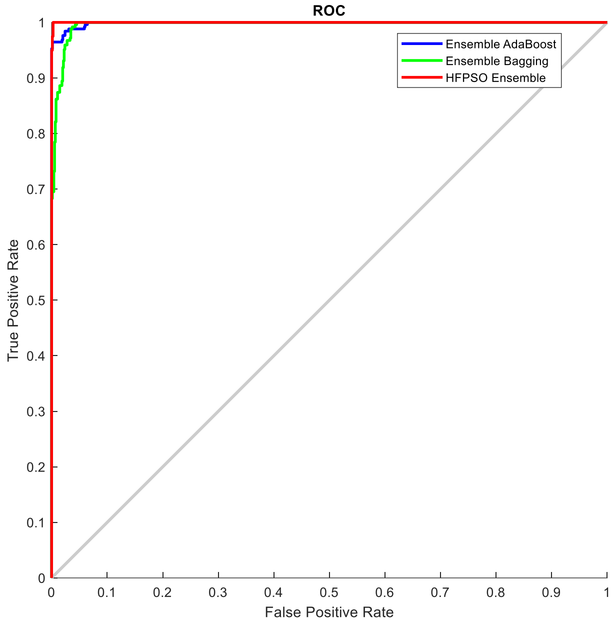

Figure. 4 ROC curve for various classifiers

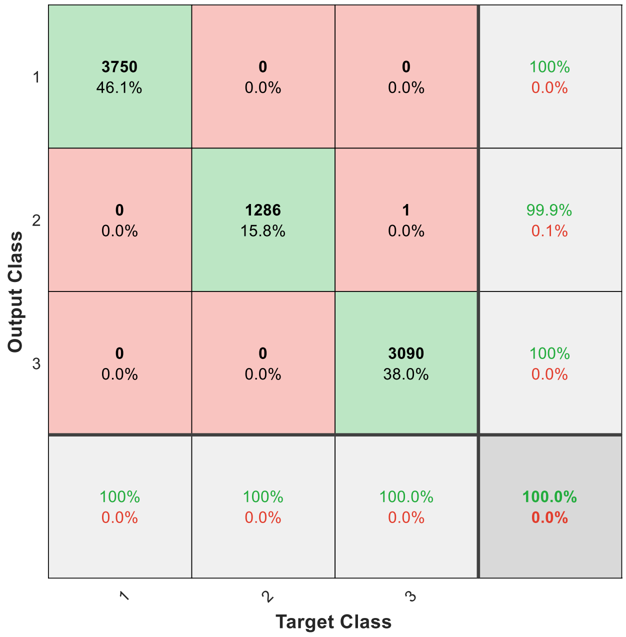

Figure. 5 Confusion matrix plot for 3 class DDOS attack

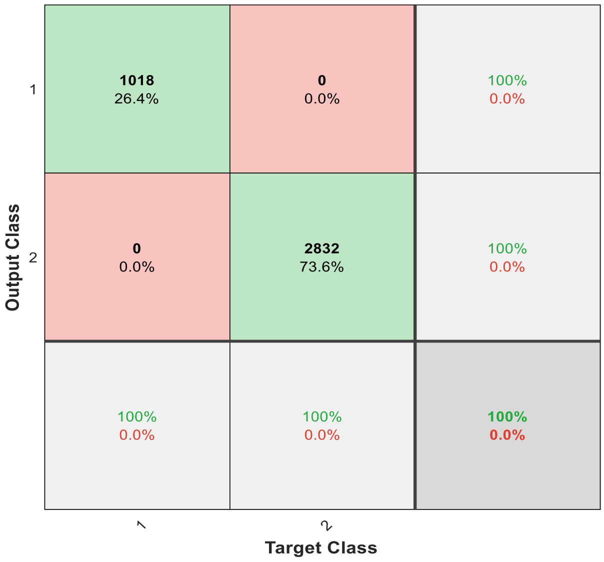

Figure. 6 Confusion matrix plot for 2 class DDOS attack

Table 2 represents the performance comparison of DDOS attacks with different classifiers. It can be seen accuracy achieved by HFPSO is a higher side than another classifier.

HFPSO Bagging yield 99.96% of accuracy whereas HFPSO AdaBoost gives 99.97% of accuracy. Achieved sensitivity of the classifier is 99.95 % in AdaBoost classifier. F-score of HFPSO is 99.95 % is higher side as compared to other classifier in DDOS classifier for 3 class.

Fig. 7 shows the HFPSO ensemble classifier used with three learning classifications: AdaBoost, Bagging, and Subspace. It is apparent that AdaBoost outperforms the other classifiers as an ensemble classifier.

In [20], the reported accuracy of the detection model is 99%, which is a high accuracy rate. However, the paper also acknowledges some challenges and problems associated with the CSE-CIC-IDS2018 dataset. One challenge is the issue of

Table 3. Comparison with previous research works

Studies &Year	Methods	Dataset	No.of Features	Accuracy In %
P. Lin, K. Ye, and C. Z. Xu [17], 2019	LSTM +AM	CSE-CIC-IDS 2018	77	96.2%
S. Seth, K. K. Chahal, and G. Singh [18], 2021	Light GBM + HBGB	CSE-CIC-IDS 2018	72	97.5%
A. L. G. Rios, Z. Li, K. Bekshentayeva, and L. Trajković [19], 2020	CFBLS and BLS	CSE-CIC-IDS 2018	78	97.46%
B. I. Farhan, and A. D. Jasim [20], 2022	LSTM	CSE-CIC-IDS 2018	78	99%
D. Kshirsagar, and S. Kumar [21], 2021	IGR, CR, and ReF +PART classifier	CICIDS 2017 DoS	24	99.9591 %
S. Dwivedi, M. Vardhan, and S. Tripathi [22], 2022	GOIDS	CIC-IDS 2017	71	99.96%
A. Makuvaza, D. S. Jat, and A. M. Gamundani [23], 2021	DNN	CICIDS 2017	86	97.59%.
V. Kanimozhi, and T. P. Jacob [24], 2021	Artificial Neural Network (MLP)	CSE-CIC-IDS2018	77	99.97 %
M. J. Awan, U. Farooq, H. M. A. Babar, A. Yasin, H. Nobanee, M. Hussain, and A. M. Zain [25], 2021	Random Forest (RF) and MultiLayer Perceptron (MLP)	CSE-CIC-IDS2018	77	99.5%
Z. Wang, Y. Zeng, Y. Liu, and D. Li [26], 2021	DBN-EGWO-KELM	CICIDS 2017	78	97.07%
Proposed Method		CSE-CIC-IDS 2018	64	99.97%

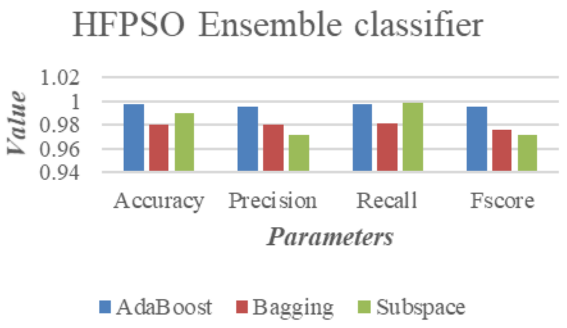

Figure. 7 Comparative analysis

dataset imbalance, which may affect the accuracy computation of the model. Imbalanced datasets, where certain classes have significantly fewer samples than others, can lead to biased results and affect the performance of the model.

Moreover, the article highlights the challenges faced while creating the LSTM model, especially with respect to scaling up the number of nodes and establishing interconnections between multiple layers. These obstacles could have a significant impact on the model's efficacy and need to be addressed with meticulousness during the model's design and training phase.

In [21], the researchers have presented a technique to identify distinct categories of assaults by assessing the classification ability of classifiers and constructing an ensemble. This approach involves using multiple classifiers and combining their outputs to enhance the overall accuracy and effectiveness of the attack detection system.

The idea is to evaluate the performance of different classifiers in detecting various types of attacks and rank them based on their detection ability. Classifiers that exhibit higher detection performance are given more weightage in the ensemble, while those with lower performance are given lower weightage or may be excluded from the ensemble. This way, the strengths of different classifiers can be leveraged to increase the overall precision and robustness of the attack detection system.

The ensemble of classifiers can be built using different techniques, such as averaging the outputs, combining them through weighted voting, or using more advanced methods like stacking or boosting. The goal is to create a diverse set of classifiers that collectively provide better detection performance compared to individual classifiers [26]. The DBN-EGWO-KELM model is designed to optimize the performance of KELM by incorporating the enhanced grey wolf optimizer, which is a meta-heuristic optimization algorithm inspired by the hunting behaviour of grey wolves. The optimizer is used to fine-tune the parameters of the KELM model, resulting in improved classification performance.

The proposed approach is expected to effectively handle the challenges associated with network intrusion data, such as high dimensionality and complex patterns. By utilizing the dimensionality reduction ability of DBN and the optimization competence of EGWO, the model aims to achieve accurate and efficient classification of network intrusion data.

The proposed method HFPSO ensemble classifier where learning rate is 60 % for train data with lower overhead on machine model gives 99.97% accuracy. It can be seen F-Score and recall values are 0.9998 and 0.9998 at the higher side as compare to the other models. Hence the propose model is much optimize and stable for the prediction of DDOS attack.

5. Conclusion

Our research on predicting properties of materials for new materials discovery revealed a distinctive issue, which is to predict property values that are beyond the scope of the training set. We examined the inadequacies of conventional k-fold cross-validation approaches when evaluating machine learning models for such scenarios. To overcome these drawbacks, we introduce a novel series of k-fold validation approaches, along with a new accuracy metric for exploration to evaluate exploratory power. Our comprehensive benchmark results demonstrate that the HFPSO-optimized ensemble classifier has better exploration capability. Our highest precision model was generated using ensemble classifier techniques, with a precision of 99.97%, which is the best outcome compared to other models evaluated.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

This paper conceptualization, software simulation, verification of results and original draft preparation has been done by Anuradha Pawar. The supervision and final approval have been done by Nidhi Tiwari.

References

[1] B. A. Khalaf, S. A. Mostafa, A. Mustapha, M. A. Mohammed, and W. M. Abduallah, “Comprehensive Review of Artificial Intelligence and Statistical Approaches in Distributed Denial of Service Attack And Defense Methods”, *IEEE Access*, Vol. 7, pp. 51691-51713, 2019.

[2] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A Ghorbani, “Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset And Taxonomy”, In: *Proc. of International Conf. On Security Technology (ICCST)*, pp. 1-8, 2019.

[3] A. Singh and B. B. Gupta, “Distributed Denial-of-Service (DDoS) Attacks and Defense Mechanisms in Various Web-Enabled Computing Platforms: Issues, Challenges, and Future Research Directions”, *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol. 18, No. 1, pp. 1-43, 2022.

[4] A. Aljuhani “Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments”, *IEEE Access*, Vol. 9, pp. 42236-42264, 2021.

[5] A. Cheema, M. Tariq, A. Hafiz, M. M. Khan, F. Ahmad, and M. Anwar “Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review”, *Security and Communication Networks (Hindawi)*, Vol. 2022, pp. 1-15, 2022.

[6] G. Baldini and I. Amerini, “Online Distributed Denial of Service (DDoS) Intrusion Detection Based on Adaptive Sliding Window and Morphological Fractal Dimension”, *Computer Networks*, Vol. 210, pp. 1-13, 2022.

[7] S. Velliangiri, P. Karthikeyan, and V. V. Kumar, “Detection of Distributed Denial of Service Attack in Cloud Computing using the Optimization-Based Deep Networks”, *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 33, No. 3, pp. 405-424, 2021.

[8] R. K. Batchu and H. Seetha, “An Integrated Approach Explaining the Detection of Distributed Denial of Service Attacks”, *Computer Networks*, Vol. 216, pp. 1-14, 2022.

[9] S. Sokkalingam, and R. Ramakrishnan, “An Intelligent Intrusion Detection System for Distributed Denial of Service Attacks: A Support Vector Machine With Hybrid Optimization Algorithm Based Approach”,

Concurrency and Computation: Practice and Experience, Vol. 34, No. 27, pp. 1-18, 2022.

[10] M. B. Bharatwaj, M. A. Reddy, T. S. Kumar, and S. Vajipayajula "Detection of DoS and DDoS Attacks Using Hidden Markov Model", In: *Proc. of International Conf. On Inventive Communication and Computational Technologies*, Springer Singapore, pp. 979-992, 2022.

[11] M. Qian, "Evaluation and Prediction Method of System Security Situational Awareness Index Based on HMM Model", *Scientific Programming (Hindawi),* Vol. 2022, pp. 1-11, 2022.

[12] R. R. Nuiaa, S. Manickam, A. H. Alsaeedi, and E. S. Alomari, "A New Proactive Feature Selection Model Based on the Enhanced Optimization Algorithms to Detect DRDoS Attacks", *International Journal of Electrical and Computer Engineering (IJECE),* Vol. 12, No. 2, pp. 1869-1880, 2022.

[13] M. Mittal, K. Kumar, and S. Behal, "Deep Learning Approaches for Detecting DDoS Attacks: A Systematic Review", *Soft Computing (Springer),* pp. 1-37, 2022.

[14] M. Beulah and B. P. Manickam, "Detection of DDoS Attack Using Ensemble Machine Learning Techniques", In: *Proc. of International Conf. On Soft Computing for Security Applications (ICSCS),* Springer Singapore, pp. 889-903, 2022.

[15] G. O. Anyanwu, C. I. Nwakanma, J. M. Lee, and D. S. Kim, "RBF-SVM Kernel-Based Model For Detecting DDoS Attacks In SDN Integrated Vehicular Network", *Ad Hoc Networks (Elsevier),* Vol. 140, No. 103026, pp. 1-12, 2023.

[16] R. Qamar, B. A. Zardari, A. A. Arain, F. H. Khoso, and F. A. Jokhio, "Detecting Distributed Denial of Service Attacks Using Recurrent Neural Network", *University of Sindh Journal of Information and Communication Technology (USJICT),* Vol. 5, No. 1, pp. 86-94, 2022.

[17] P. Lin, K. Ye, and C. Z. Xu, "Dynamic Network Anomaly Detection System by Using Deep Learning Techniques", In: *Proc. of Cloud Computing–CLOUD 2019: 12th International Conference, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, Springer International Publishing*, pp. 161-176, 2019.

[18] S. Seth, K. K. Chahal, and G. Singh, "A Novel Ensemble Framework For An Intelligent Intrusion Detection System", *IEEE Access*, Vol. 9, pp. 138451–138467, 2021.

[19] A. L. G. Rios, Z. Li, K. Bekshentayeva, and L. Trajković, "Detection of Denial of Service Attacks in Communication Networks", In: *Proc. of International Conf. On Circuits And Systems (ISCAS)*, pp. 1-5, 2020.

[20] B. I. Farhan and A. D. Jasim, "Performance Analysis of Intrusion Detection for Deep Learning Model Based On CSE-CIC-IDS2018 Dataset", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 26, No. 2, pp. 1165-1172, 2022.

[21] D. Kshirsagar and S. Kumar, "An Efficient Feature Reduction Method for the Detection of DoS Attack", *ICT Express*, Vol. 7, No. 3, pp. 371-375, 2021.

[22] S. Dwivedi, M. Vardhan, and S. Tripathi, "Defense Against Distributed DOS Attack Detection by Using Intelligent Evolutionary Algorithm", *International Journal of Computers and Applications*, Vol. 44, No. 3, pp. 219-229, 2022.

[23] A. Makuvaza, D. S. Jat, and A. M. Gamundani, "Deep Neural Network (DNN) Solution for Real-Time Detection of Distributed Denial of Service (DDoS) Attacks in Software Defined Networks (SDNs)", *SN Computer Science*, Vol. 2, pp. 1-10, 2021.

[24] V. Kanimozhi, and T. P. Jacob, "Artificial Intelligence Outflanks All Other Machine Learning Classifiers in Network Intrusion Detection System On the Realistic Cyber Dataset CSE-CIC-IDS2018 Using Cloud Computing", *ICT Express*, Vol. 7, No. 3, pp. 366-370, 2021.

[25] M. J. Awan, U. Farooq, H. M. A. Babar, A. Yasin, H. Nobanee, M. Hussain, and A. M. Zain, "Real-Time DDoS Attack Detection System Using Big Data Approach", *Sustainability,* Vol. 13, No. 19, p. 10743, 2021.

[26] Z. Wang, Y. Zeng, Y. Liu, and D. Li, "Deep Belief Network Integrating Improved Kernel-Based Extreme Learning Machine For Network Intrusion Detection", *IEEE Access*, Vol. 9, pp. 16062-16091, 2021.

[27] M. Ghasemi, S. K. Mohammadi, M. Zare, S. Mirjalili, M. Gil, and R. Hemmati, “A New Firefly Algorithm With Improved Global Exploration And Convergence With Application To Engineering Optimization”, *Decision Analytics Journal*, Vol. 5, No. 100125, pp. 1-18, 2022.

[28] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, “An Overview of Variants and Advancements of PSO Algorithm”, *Applied Sciences*, Vol. 12, No. 17, pp. 1-21, 2022.

[29] IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018) dataset. Online Available At: https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv