INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

Received: 10-07-2023

Accepted: 04-10-2023

Published: 14-11-2023

Citation: Sumathi S, Rajesh R (2023) A Dynamic BPN-MLP Neural Network DDoS Detection Model Using Hybrid Swarm Intelligent Framework. Indian Journal of Science and Technology 16(43): 3890-3904. https://doi.org/10.17485/IJST/v16i43.1718

[*] **Corresponding author**.

sumock123@yahoo.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

A Dynamic BPN-MLP Neural Network DDoS Detection Model Using Hybrid Swarm Intelligent Framework

S Sumathi[1]*, R Rajesh[2]

1 Assistant Professor, Computer Science and Engineering, University V.O.C College of Engineering, Tamil Nadu, India
2 Research Scholar, Department of Engineering Design, Indian Institute of Technology, Madras, Tamil Nadu, India

Abstract

Background/Objectives: The most untreated and severe cyber security issue in cloud computing is DDoS attack, this is being under research to find novel findings with less complexity and better efficiency to detect and mitigate this issue. In this research article, Artificial Neural Network (ANN) algorithms like Backpropogation neural network (BPN) and Multilayer perceptron (MLP) are implemented and their performance on intrusion detection by utilizing NSL-KDD dataset is demonstrated. **Methods:** Initially, NSL-KDD benchmark dataset construction is carried out in the range of (0-1) using min-max normalization technique. Following this, hybrid Harris Hawks optimization particle swarm optimization (HHO-PSO) is employed to reduce the dataset size by selecting significant features that represents anomaly in network traffic. This hybrid algorithm is also employed to tune the features selected which is assigned as initial weight vectors for both BPN and MLP intrusion detection system (IDS) models. These selected optimally tuned features are trained using 10-fold cross validation technique and the number of hidden neurons is fixed using thumb rule. After training, the hybrid BPN-MLP neural network IDS model is validated on test dataset and its performance is validated using performance metrics such as accuracy, precision, sensitivity, specificity and F1 score. **Findings:** The proposed hybrid HHO-PSO BPN and HHOPSO MLP IDS model has achieved detection accuracy of 97.08% and 97.74% with F1 score of 0.9743 and 0.9800 respectively. **Novelty:** In ANN based intrusion detection schemes, the stochastic nature of model parameters is an important problem of concern. To handle this issue, a hybrid swarm intelligent algorithm called Harris hawks optimization particle swarm optimization (HHOPSO) is proposed to tune the model parameters, so that the network performance is enhanced.

Keywords: Backpropogation Neural Network; Multilayer perceptron; Harris Hawks Optimization; Particle Swarm Optimization; Intrusion Detection System

1 Introduction

The tremendous growth in Information Technology (IT) sector has satisfied the online demand of users and organizations over internet. Cloud computing offers huge data storage facility which can be accessed both by public and private sectors using a common service provider. This leads to a large number of cyber-attacks and among these Distributed Denial of Service (DDoS) is a severe intrusion attack as it completely paralyzes the victim. Distributed Denial of Service (DDoS) is a significant security concern in the cloud computing environment, which consumes the entire network resources disrupting its regular traffic[1]. This attack causes traffic congestion using zombies and it is a highly difficult task to distinguish this attack traffic from a legitimate one. Nagarajan et.al (2023)[2] proposed an Intrusion Detection System (IDS) system using Back Propagation Network (BPN) optimized by Particle Swarm Optimization (PSO) which detects intrusions based on system calls collected from KDD cup 99 dataset. Narengbam et.al (2023)[3] proposed an anomaly IDS by using hybrid Harris Hawk Optimization (HHO)-Artificial Neural Network (ANN) algorithm on AWID, CIDDS001 and NSL-KDD datasets. This hybrid IDS achieved better convergence with high reliability. Siva Shankar et.al[4] proposed an optimized Artificial Intelligence (AI) approach IDS which uses backpropagation technique to tune the precondition parameters. This methodology effectively identifies intrusions on UNSW-NB 15 and NSL-KDD datasets. So, this research article aims to develop an intelligent neural network based IDS model that detects DDoS in cloud computing domain. DDoS can easily be carried out by hackers using tools which are freely available in internet.

1.1 Research Motivation

DDoS can easily be carried out by hackers using tools which are freely available in internet. DDoS attack packets behaves very much similar to normal packets and hence its detection is a highly challenging task. Machine Learning (ML) algorithms pattern learning technique helps in DDoS attack detection. Various research works has been carried out in the past to mitigate this attack. These research works failed to achieve DDoS detection with higher performance measures. This research article proposes a neural network based IDS model which performs DDoS detection with ideal performance measures. Backpropagation Neural Networks (BPN) and Multilayer Perceptron (MLP) are chosen in this study as they possess the following properties:

1. Prior Knowledge is not required in BPN and it is highly efficient as well as adaptable.
2. MLP has the capacity to make quick predictions irrespective of the dataset size.

1.2 Contribution of this research

The key contributions of this research article are as follows:

1. Experimental modeling of neural network based IDS model using NSL-KDD dataset
2. Feature selection is done using 10 -fold cross validation technique
3. Hybrid swarm intelligent algorithm is used in fine tuning of selected parameters
4. The proposed neural network IDS model is trained and tested with NSL-KDD benchmark dataset
5. The proposed neural network IDS model performance is compared with other IDS models in literature.

2 Review of state of the art algorithms

2.1 Backpropagation Neural Networks (BPN)

The Backpropagation Neural Networks (BPN) are feed forward neural networks with a learning mechanism of error backpropagation. The weight values are updated by employing a gradient descent learning rule with the objective of reducing the mean square error between the actual and estimated output.

The training process of BPN network has four entities such as,

1. Weight Initialization
2. Feed forward phase
3. Error backpropagation phase
4. Weight and bias vector updating phase

The initial process of the algorithm is random initialization of network parameters such as weight and bias vectors, then the input vector. $X_i = (x_1, x_2 \ldots x_n\}$ is transmitted from the input layer to hidden layer during the feed-forward phase, where the net input for each neuron is estimated and activation function in hidden neurons $(z_1, z_2 \ldots z_n\}$ is employed over the given net input the obtained output becomes input pattern for the output layer of neurons $(y_1, y_2 \ldots y_n\}$ in the network. The final output attained in the output layer is compared with the actual output, the estimated error factor is backpropagated and is utilized in the weight updating phase of the learning algorithm. The training algorithm is presented as follows:

Step 1: Initialize the BPN network parameters such as weight and bias vector randomly in the range of $0 - 1$, the stopping criteria is specified.

Step 2: For every input vector $X_i = (x_1, x_2 \ldots x_n\}$, do steps 3 to 8 until stopping criteria are attained.

Step 3: The input is transmitted from the input layer to the hidden units in the hidden layer.

Step 4: For all hidden neuron z_j find the net input and output of hidden layer by the following equations,

$$Z_{-inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij} \tag{1}$$

$$Z_j = f\left(Z_{-inj}\right) \tag{2}$$

Step 5: The network output is estimated at the output units as,

$$y_{-ink} = w_{oj} + \sum_{j=1}^{p} z_j w_{jk} \tag{3}$$

$$Y_k = f\left(y_{-ink}\right) \tag{4}$$

Step 6: At output layer the error between the actual and estimated output is calculated,

$$\delta_k = \left(t_k - y_k\right) f^{'}\left(y_{-ink}\right) \tag{5}$$

Step 7: The error term in hidden units are computed as,

$$\delta_{-inj} = \delta_j w_{jk} \tag{6}$$

$$\delta_j = \delta_{-inj} f^{'}\left(z_{-inj}\right) \tag{7}$$

Step 8: The weight and bias vectors at the output layer are updated by,

The weight correction term,

$$\Delta W_{jk} = \alpha \delta_k z_j \tag{8}$$

The bias correction term,

$$\Delta W_{ok} = \alpha \delta_k \tag{9}$$

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{jk} \tag{10}$$

The weight and bias vectors in hidden layer are updated by,

The weight correction term,

$$\Delta V_{ij} = \alpha \delta_j x_i \tag{11}$$

The bias correction term,

$$\Delta V_{oj} = \alpha \delta_j \tag{12}$$

$$V_{ij}(new) \ = \ V_{ij}\ (old) \ + \ \triangle V_{ij} \tag{13}$$

Step 9: The attainment of stopping criteria is checked, in the proposed study the stopping criteria is the error convergence of 10^{-5}.

2.2 Multilayer perceptron (MLP)

The one of the important class of neural network is multilayer perceptron (MLP) neural network, it is multi-layered feed forward neural network. The highly non-linear complex tasks are handled effectively by the hidden layers and the hidden neurons in the network and the output is computed by employing non-linear activation functions.

The MLP is trained by employing static backpropagation learning rule, the main advantage of MLP is it has many layers which improve the learning ability of the network. Further, the nonlinear activations are employed to handle the highly non-linear tasks. The input signal is processed in each layer of the network, and the output is computed at the output layer. The error gradient is computed and it is backpropagated into the network layer by layer. The training algorithm of the MLP network is presented as follows:

Step 1: Initialize the MLP network parameters such as weight and bias vector randomly in the range of $0-1$, the stopping criteria is specified.

Step 2: For every input, vector $X_i = (x_1, x_2 \ldots x_n\}$ do steps 3 to 5 until stopping criteria is attained.

Step 3: The activation function is employed to estimate the output of the hidden neuron,

The net input of the hidden layer is determined by

$$Z_{-inj} = \sum_{j=1}^{n} X_i V_{ij} \tag{14}$$

The hidden layer output is estimated by

$$Z_j = f\left(\sum_{j=1}^{n} X_i V_{ij}\right) \tag{15}$$

The net input of output layer is estimated by

$$Y_{-ink} = \sum_{k=1}^{n} Z_j W_{jk} \tag{16}$$

Output is estimated by,

$$Y_k = f\left(\sum_{j=1}^{n} \left(Z_j W_{jk}\right)\right) \tag{17}$$

Where V represents the weight vector between the input and hidden layer, W represents the weight vector between the hidden and output layer.

Step 4: The error between the actual and target value is determined and employed in weight updating equations.

Step 5: Error back-propagation learning algorithm is implemented to update the weight vectors.

Step 6: Check for stopping criteria, in the proposed study the error convergence to the rate of 10^{-5} is the stopping criteria, once stopping criteria is attained stop the algorithm and return the solution.

2.3 Harris Hawk Optimization (HHO)

The Harris Hawk Optimization algorithm is developed by Heidari et al. (2019) based on inspiring the hunting behavior of the Harris Hawks. Generally, the Hawks hunts in groups and all the individuals in the population collaboratively involve themselves to take decisions on hunting the prey based on its escape energy. The escape energy of the prey decides the hunting period, based on the energy level of prey, that may get caught immediately or the hawks exhaust the energy level of prey to make a sudden attack of the hunt. The hawks are highly intelligent to make rapid and confusing movements in order to exhaust and attack the rabbit that is considered as prey in this algorithm. The hunting mechanism of HHO is shown in Figure 1.

The pseudo-code of the HHO algorithm is explained as follows:

- **The shift of Exploration to Exploitation:** The exploration phase of the algorithm is the observation of prey by the sharp eyes of the Hawks. This is the phase of high time consumption as the hawks are needed to wait to track prey. The position of Hawks in presented by the following expression.

$$X(t+1) = \left\{ \begin{array}{cc} X_{rand}(t) - r_1\left(X_{rand}(t) - 2rX(t)\right| & q \geq 0.5 \\ \left(X_{rabbit}(t) - X_n(t) - r_3\left(LB + r_4(UB - LB)\right)\right) & q < 0.5 \end{array} \right. \tag{18}$$

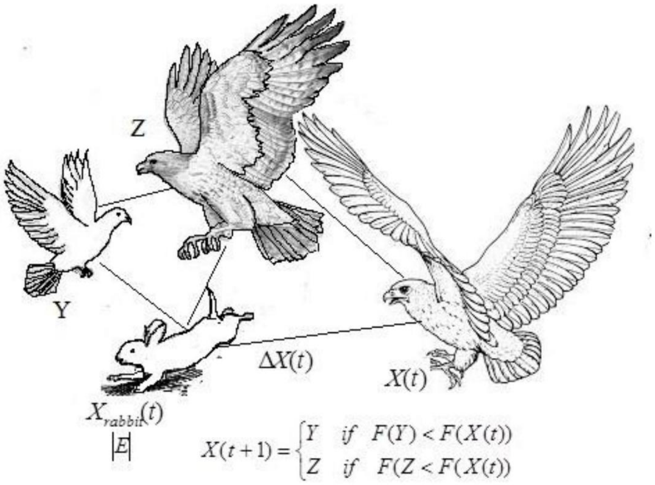
Z
Y
$\Delta X(t)$
$X(t)$
$X_{rabbit}(t)$
$|E|$
$X(t+1) = \begin{cases} Y & if \;\; F(Y) < F(X(t)) \\ Z & if \;\; F(Z < F(X(t)) \end{cases}$

Fig 1. Hunting Mechanism of Harris Hawks

Where $X(t)$ symbolizes the current position of prey, r_1, r_2, r_3, r_4 denotes a factor of random values, $X_{random}(t)$ signifies the random position of Hawks, $X_n(t)$ represents the mean position of Hawks in the current population. In equation (18) two strategies are assumed, if $q < 0.5$ then the position of the Hawks is updated based on the position of the prey, in the other cases of $q \geq 0.5$ then the position of Hawks is considered to the hunting position in the search space.

The position of each Hawk in the population is given by

$$X_n(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \tag{19}$$

- **Exploitation Phase:** The energy level of the prey is adopted for shifting of exploration to exploitation based on the energy level of prey. The energy level of pray is presented as,

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \tag{20}$$

Where E denotes the escape energy level of the prey, initially, the energy is assumed to be E_0 that is varied from - 1 to 1 for every iteration. The rabbit is assumed to be exhaust when the energy level is between 0 to -1 and is assumed to be highly energetic when it is 0 to 1. When the energy level of prey is high then the hawks will keep observing the prey behavior until $|E| \geq 1$ represented as an exploration phase, when energy level decreases at $|E| < 1$ then exploitation phase of attacking starts. During the exploitation phase, the energy level of prey is decreased so the Hawks start to make a surprise attack over the prey. The escape chance of the rabbit is denoted as r when $r \geq 0.5$ then it cannot escape easily and in another case of the prey gets a high chance of escape. Based on this the exploitation phase of the algorithm has two hunting mechanisms such as soft besiege and hard besiege.

- **Soft Besiege:** When the rabbit possesses the energy which is sufficient to escape from the Hawks, the Hawks try to exhaust the prey's energy by encircling the prey. At this scenario, the $r \geq 0.5$ and $|E| \geq 0.5$ the position of Hawks is presented as,

$$X(t+1) = \Delta X(t) - E\,|JX_{rabbit}(t) - X(t)| \tag{21}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{22}$$

Where $J = 2(1 - r_5)$ denotes the power of jumping, the r_5 varies from 0 to 1.$\Delta X(t)$ express the change of position vector of prey from its current position.

- **Hard Besiege:** Now, the prey would have exhausted its energy completely and become easy to trap, such a case $r \geq 0.5$ and $|E| < 0.5$, the Hawks make an intelligent tactic of a sudden and close dive towards the prey to make a surprise attack. The position of the rabbit is presented as,

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \tag{23}$$

- **Soft besiege with progressive diving:** The new position of Hawks is mathematically expressed based on the previous history of attacks made in soft besiege. The next position of the rabbit is computed in advance by the hawks to make accurate dive. The mathematical expressions that represent the process is presented as follows:

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X(t)| \tag{24}$$

$$Z = Y + SxLF(D) \tag{25}$$

$$X(t+1) = \begin{cases} Y\ if\ F(Y)\ < F(X(t)) \\ Z\ if\ F(Z)\ < F(X(t)) \end{cases} \tag{26}$$

- **Hard besiege with progressive quick dives:** During hard besiege the Hawks start a fast and hard attack with the intention of killing the prey. The mechanism is similar to that of soft besiege but the distance between the Hawks and the prey is very small as compared to that of soft besiege. The Equation (24) is reframed as,

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X_n(t)| \tag{27}$$

2.4 Particle Swarm Optimization (PSO)

Kennedy and Eberhart introduced the PSO optimization algorithm in the middle of 1995 based on inspiring the behavior of flocks of birds, schools of fish, and herds of animals to search for a solution in space. The algorithm is framed such that the randomly generated population is trained to adapt themselves to attain an optimal solution in the space. The PSO is a swarm intelligence based strategy that is aimed to find the global optimal value in the given space. The working principle of the PSO has three steps, the particle generation, the position, and the velocity equations update. The particle in space represents the point that changes its position based on the velocity changes that occur in the space. The population is initially generated with random value of position and velocity, and the design variables are constrained to the lower and upper bounds. The better solution is attained by the influence of entire particles in the population, so the fitness value of all the particles in the neighborhood is utilized to identify the best position that has the optimal solution; let the best particle in a neighborhood be Pl_{bt} the best particle that is identified from the entire population be Gl_{bt}

The position of the particle

$$p_i^{t+1} = p_i^t + v_i^{t+1} \tag{28}$$

Where v_i^{t+1} is the velocity component and is computed as follows:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 \{Pl_{bti} - p_i^t\} + c_2 r_2 \{Gl_{bt} - p_i^t\} \tag{29}$$

Where r_1, r_2 is the random values of (0-1), v_i^t - The velocity of the particle at iteration ' t ' , v_i^{t+1}

• The velocity of the particle at iteration' $t+1$ ', p_i^t - Position of the particle at iteration ' t ', p_i^{t+1} Position of the particle at iteration ' $t+1$ ', Pl_{bti} - Local best among the current individuals, Gl_{bt} - Global best among the swarm. C_1, C_2 - The stochastic acceleration coefficients that pull every particle towards the local and global best values. ω - Inertia factor that adjusts the velocity such that controls the exploitation and exploration ability of the algorithm.

$$\omega = \omega^{max} - \left(\frac{\omega^{max} - \omega^{min}}{iter^{max}}\right) * iter \tag{30}$$

Where, ω^{max}– Maximum inertia weight, ω^{min} - Minimum inertia weight, iter ^{max}– Maximum number of iterations, iter- Current iteration number

The pseudo-code of the PSO algorithm is explained as follows:

Step 1: Initialize the necessary parameters of the algorithm such as population size, stopping criteria, inertia factor, and the acceleration coefficient of the algorithm.

Step 2: The search agents are generated randomly in the search space within the bounded limits.

Step 3: The fitness value of all the particles in the population is computed, the local best and the global best population is identified among the swarm.

Step 4: The position and velocity of all the particles in the population is adjusted by the Equations (28) and (29).

Step 5: The steps 3 to 5 are repeated until the stopping criteria are attained.

Step 6: Return the solution obtained at the end of the algorithm.

3 Proposed Hybrid HHO-PSO Algorithm

In the proposed research contribution, a hybrid swarm intelligent optimization algorithm is developed that serves a two-fold purpose, initially the algorithm is employed to select the significant features that play major contribution for the attack identification, and the algorithm is also implemented to tune the proposed neural network-based IDS models with optimal parameter settings. To accomplish these objectives, the Harris Hawks Optimization algorithm is considered in this study, based on the shortcomings encountered by the algorithm during the training process, the Particle Swarm Optimization algorithm is combined with HHO optimizer to attain better trade off between exploration and exploitation ability of the algorithm. The conventional HHO algorithm suffers from poor exploration ability as the Hawks are needed to wait for prey from several minutes to several hours. To handle this issue, the PSO is incorporated into HHO to improve the convergence speed of the algorithm. The PSO algorithm has been chosen in this study among all other swarm intelligence algorithms because of its simplicity and excellent exploration ability. The best qualities of HHO and PSO have been combined to present a hybrid HHO algorithm so as to attain a better trade off between exploration and exploitation mechanism than the conventional HHO algorithm and other conventional algorithms. In the exploration phase of the HHO optimization algorithm the equation (18) is modified by incorporating the PSO algorithm as follows:

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \left(X_{rand}(t) - 2r_2 X(t)\right| + v(t+1) & q \geq 0.5 \\ \left(X_{rabbit}(t) - X_n(t) - r_3 \left(LB + r_4(UB - LB)\right) + v(t+1)\right) & q < 0.5 \end{cases} \tag{31}$$

$$v(t+1) = \omega v(t) + c_1 r_1 \{Pl_{bt} - X(t)\} + c_2 r_2 \{Gl_{bt} - X(t)\} \tag{32}$$

Pseudo-code for Proposed Hybrid HHO-PSO

```
Input: Population Size, Convergence criteria, random factors, acceleration coefficient, inertia factor, upper and lower bounds.
Output: The Fitness value and the corresponding position of the prey
Initialize the population
While (stopping criteria)
Do
Fitness(all Hawks in Population)
if current_pBest>pBest
then pBset=current_pBest
else
pBest = pBest
end
gBest=particle with best pBest among the population
```

```
    Define the position of the rabbit
    for (all Hawks)
    Update the initial energy level of prey and its jumping power.
    Update the current energy level of prey
     # Exploration Phase
    if ($|E| \geq 1$) then
    The position of each hawk in the population is adjusted by the Equation (31)
    # Exploitation Phase
    if ($|E| \leq 1$) then
            $if(r \geq 0.5$ and $|E| \geq 0.5)$ then # Soft besiege
    Adjust the Hawks position by the Equation (21)
    else if ($r \geq 0.5$ and $|E| < 0.5$) then # Hard besiege
    Adjust the position by the Equation (23)
    else if ($r < 0.5$ and $|E| \geq 0.5$) then # Soft besiege with dives
                    $Y = X_{rabbit}(t) - E\,|JX_{rabbit} - X_m(t)|$
                    $Z = Y + SxLF(D)$
                    $X(t+1) = \begin{cases} Y & if(F(Y) < F(X(t))) \\ Z & if(F(Z) < F(X(t))) \end{cases}$
else if ($r < 0.5$ and $|E| < 0.5$)# Hard besiege with dives
                    $Y = X_{rabbit}(t) - E\,|JX_{rabbit} - X_m(t)|$
                    $Z = Y + SxLF(D)$
                    $X(t+1) = \begin{cases} Y & if(F(Y) < F(X(t))) \\ Z & if(F(Z) < F(X(t))) \end{cases}$
return the solution
```

4 Proposed Methodology

The experimental modeling of the proposed IDS model is depicted in Figure 2. The steps included in the experimental modeling of the proposed study is presented as follows:

Step 1: The first step in the development of the intrusion detection model is the dataset construction. The NSL-KDD benchmark dataset is employed in the study to perform intrusion detection. The dataset has 41 features, the features are not scaled, and also some of the features are in text format which will again impose a burden to the algorithm while training. Initially, the data is encoded to numeric format then min-max normalization is employed to scale the dataset to the range of [0-1], during this process the size of the dataset can be reduced, the model performance is enhanced.

Step 2: The proposed hybrid HHO-PSO optimization algorithm is employed to select the trend features that are significant to present the anomaly in the network traffic. So, the size of the data that is feed into the model is reduced, the performance of the model can be enhanced. The algorithm converges towards the objective of increasing classification accuracy while reducing the number of features, the cost function framed is expressed as,

$$f(x) = \xi A + \xi' \frac{N-L}{N} \tag{33}$$

where $A = \frac{Number\ of\ correctly\ predicted\ samples}{Total\ number\ of\ samples}$, $N-$ number of selected features, L length of selected features, $\xi \in (0,1)$ - weight vector of classification accuracy, $\xi' = 1 - \xi$.

Step 3: The data subset of selected features is trained into the model by employing 10 -fold cross validation, in the proposed study two neural network models are developed to perform effective intrusion detection. The neural network models are

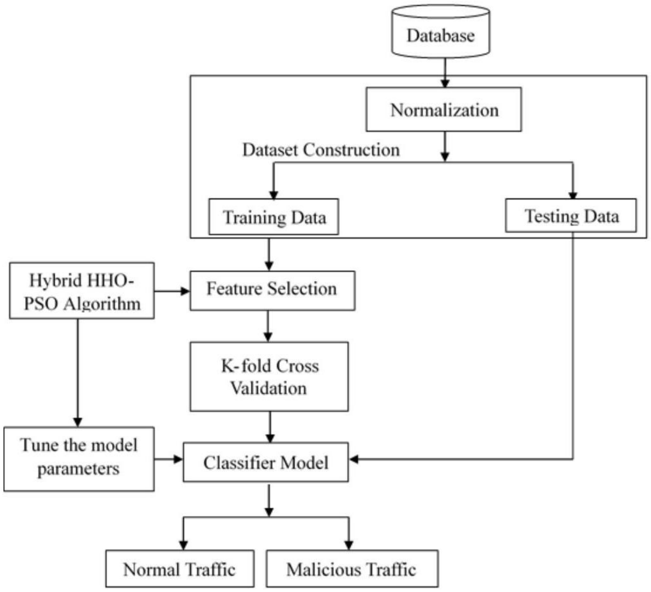

Fig 2. Experimental Modeling of the proposed model

greatly influenced by the weight vectors that are initialized randomly and are updated during the training process. The random initialization of the network greatly affects the performance of the model so the proposed hybrid HHO-PSO optimization algorithm is employed to tune the model parameters and optimally generated values are employed as initial weight vectors for the BPNN and MLP models. In addition to weight vectors the learning rate and momentum factor also play a vital role in attaining better convergence of the neural network models, these values returned at the end of the training process. Numerous trails are made to avoid the biased output, based on better convergence attained the learning rate is fixed to the model, the parameters of the proposed model are presented in Table 1. The number of hidden neuron fixations in the neural network is another problem of concern, if the number is too high it will increase the complexity of the network if it inadequate then the network learning ability is affected. On accounting, this contains, the number of hidden neurons that are initially fixed based on thumb rule, and based on trial and error method the number of neurons is varied and correspondingly the network architecture is framed.

Step 4: Once the training is completed the model is ready for its validation with an unknown dataset. The proposed models are feed with the testing dataset and the corresponding performance is evaluated by the performance metrics such as Accuracy, Precision, Sensitivity, Specificity, and F1 Score.

Step 5: A comparative analysis is made to demonstrate the effectiveness of the proposed models with the existing models in the literature and the other proposed IDS models.

5 Results and Discussion

The proposed intrusion detection models are experimentally validated by the experimental analysis made on benchmark datasets in MATLAB R2014a environment and executed in Intel Duo Core2 Processor with 2GB Ram of speed 2.27 GHz. It is a big challenge to distinguish illegitimate traffic from legitimate traffic. The proposed IDS model of this research article has claimed two objectives, the development of a hybrid optimization strategy for feature selection and the optimal design of the neural network-based IDS model. The proposed hybrid algorithm is employed to select the necessary features and the obtained features for every fold 10 fold cross-validation are depicted in Table 2 and the selected feature subset is employed to train the model. At the end of 10-fold cross validation, the number of times of occurrence of the each selected is presented in Table 3. On rejecting the unselected features, the models are trained with the selected features for 10 trails runs and the average of the obtained outputs are reported to avoid the biased results. The frequency of selected features above 8 is considered as selection criteria and this is made after several trials and error methods. The selected dataset is fed into the proposed models and analyzed their performances as shown in Figure 3. The results obtained confirmed that the proposed hybrid HHOPSO based optimally designed MLP neural network outperformed the conventional models with better performance metrics. The convergence graph

Table 1. Parameters of the proposed model

Neural Network Models		
Parameters	**BPN Network**	**MLP Network**
Weights and Bias	Optimally fed by HHO-PSO	Optimally fed by HHO-PSO
Number of input Neurons	Number of selected Features	Number of selected Features
Number of hidden Layers	2	2
Number of hidden Neurons	07-Oct	07-Oct
Number of output neurons	1	1
Activation Function	Sigmoidal Activation Function	Sigmoidal Activation Function
Learning rate	0.23(Fixed at end trial)	0.3 (Fixed at end trial)
Momentum Factor	0.4	Not applicable
Learning Rule	Gradient descent rule	Perceptron rule
Hybrid HHO-PSO		
Population Size		100
Maximum Number of Iterations		Until convergence attained
(u, v)		(0,1)
β		1.5
Initial Energy State E_0		(0,1)

is plotted in Figure 4, it is observed from the graph that the convergence of the conventional BPN and MLP has been improved by adopting optimization algorithms. The proposed hybrid HHO-PSO based MLP attained convergence at 450^{th} iteration whereas the conventional MLP model has attained convergence at 552^{nd} iteration. The convergence of the proposed hybrid HHO-PSO based BPN attained at 500^{th} iteration whereas the classic BPN based IDS attains the convergence at 600^{th} iteration. The proposed hybrid HHOPSO optimization algorithm improved the convergence speed of the neural network models than the individual algorithms.

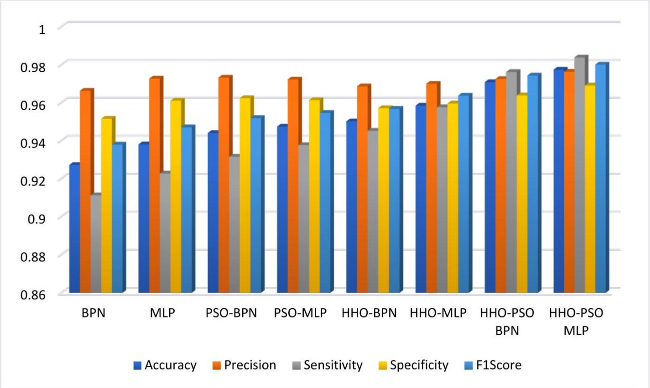

Fig 3. Performance comparisons of the proposed models

The optimization algorithms such as PSO, HHO, and the proposed HHO-PSO are employed to select the trend feature subset. It is observed that the proposed hybrid HHO-PSO optimizer returned minimum number of features as compared to other models presented in this study with better classification accuracy. The C4.5 selected 12 features with better classification accuracy, the selected features include F4 (flag), F5 (src_bytes), F8 (wrong_fragment), F10 (hot), F12 (logged_in), F23 (Count), F25 (serror_rate), F29 (same_srv_rate), F30 (diff_srv_rate), F35 (dst_host_diff_srv_rate), F36 (dst_host_same_src_port_rate), F37 (dst_host_srv_diff_host_rate). The optimal feature subset selected by the proposed hybrid HHO-PSO algorithm has 8 features such as F3 (service), F4 (flag), F5 (src_bytes), F6 (dst_bytes), F12 (logged_in), F25 (serror_rate), F30 (diff_srv_rate), F39 (dst_host_srv_serror_rate). The selected data set is feed into the proposed models and their performances are investigated and reported in Table 4. From the obtained result, it is clear that the proposed hybrid HHO-PSO based optimally designed MLP neural network outperformed the conventional models with better performance metrics which illustrated the significance of

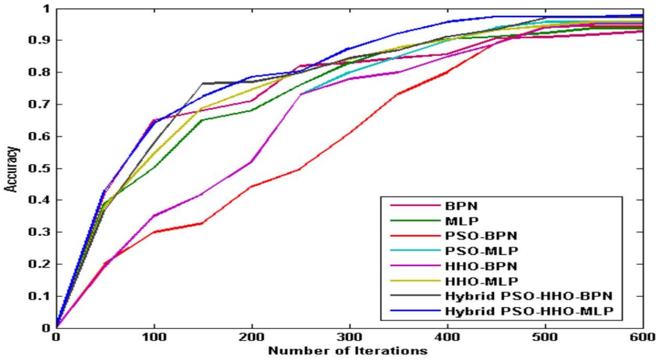

Fig 4. Convergence graph of the proposed models

the proposed hybrid optimization algorithm for optimal network design.

Table 2. Selected features of the proposed algorithms during 10-fold cross-validation

Fold	Selected Features	Fold	Selected Features
	PSO		
#1	F3,F5,F8,F10,F12,F23,F29,F30,F35,F36,F39	#6	F5,F6,F7,F8,F12,F23,F25,F29,F30,F36,F37
#2	F3,F4,F5,F10,F12,F23,F25,F29,30,F35,F36,F37	#7	F3,F4,F5,F10,F23,F25,F29,F30,F35,F36,F37
#3	F3,F4,F5,F6,F8,F12,F23,F29,F30,F35,F36,F39	#8	F4,F5,F6,F8,F10,F12,F25,F29,F30,F35,F36,F37,F38
#4	F3,F4,F5,F6,F8,F12,F23,F29,F30,F36,F37	#9	F4,F8,F10,F12,F23,F25,F35,F36,F37,F38
#5	F4,F5,F8,F10,F12,F23,F25,F30,F35,F36,F37	#10	F4,F5,F8,F10,F12,F23,F25,F29,F35,F36,F37,F39
	HHO		
#1	F2,F3,F4,F5,F8,F10,F11,F12,F23,F25,F29, F30, F35,F37,F38,F39	#6	F3,F4,F5,F8,F10,F11,F12,F23,F25,F29,F30,F36,F37
#2	F3,F2,F4,F5,F8,F10,F12,F23,F25,F29,F30,F35,F36,F39	#7	F3,F4,F5,F8,F10,F12,F23,F25,F29,F30,F35,F36,F37
#3	F3,F4,F5,F8,F10,F12,F23,F25,F30,F35,F37,F39	#8	F3,F4,F5,F6,F8,F10,F11,F12,F23,F25,F30,F35,F36,F37,F38
#4	F3,F4,F5,F8,F12,F23,F30,F36,F37,F39	#9	F5,F6,F8,F10,F12,F23,F25,F30,F35,F36,F38
#5	F4,F5,F6,F8,F10,F12,F23,F25,F30,F35,F36,F37	#10	F4,F5,F6,F8,F12,F23,F25,F30,F35,F36,F38,F39
	Proposed hybrid HHO-PSO		
#1	F3,F4,F5,F6,F8,F10,F12,F25,F26,F29,F30, F35,F36,F37,F39	#6	F3,F4,F5,F6,F8,F9,F12,F23,F25,F30
#2	F4,F5,F6,F8,F12,F25,F26,F30,F35,F36,F39	#7	F3,F4,F5,F6,F8,F12,F23,F25,F29,F30,F38
#3	F3,F4,F5,F6,F8,F10,F12,F23,F25,F26,F30,F36,F39	#8	F3,F4,F5,F6,F8,F12,F23,F25,F30,F35,F39
#4	F4,F5,F6,F8,F10,F12,F23,F25,F26,F36,F39	#9	F3,F4,F5,F6,F12,F23,F25,F30,F35,F36,F39
#5	F3,F4,F5,F12,F23,F25,F26,F30,F35,F39	#10	F3,F4,F5,F10,F12,F25,F30,F35,F39

Chiba et al. (2019)[5] presented the BPN algorithm based NIDS. The proposed network model is optimally framed by an improved GA optimization algorithm. The experimental simulations are carried out with DARPA's KDD cup datasets which achieved lower detection accuracy rate when compared to the proposed hybrid BPN-MLP model. Liu et al. (2019)[6] presented a generalized entropy scheme to pre-determine the traffic based on the SDN controller architecture. In this model, BPN is optimally framed by the PSO algorithm which is employed to classify the traffic. This model performs DDoS attack detection with reduced CPU load with F1 score of 0.9237 but the proposed hybrid BPN-MLP IDS model has F1 score of 0.97 and 0.98 respectively. A deep learning auto encoder strategy is developed to identify the DDoS attack in the smart grid environment by Ali & Li (2019)[7]. In the proposed approach the multilevel stacked encoders are developed and feature selection is made by multiple kernel learning algorithms. The experimental simulation is carried out with two benchmark functions to illustrate the classification ability of this model but it has insufficient accuracy of 89% when compared to the proposed hybrid BPN-MLP model. A convolutional Neural network (CNN) model for DDoS detection in Smart grid application is developed by Ghanbari and Kinsner (2021)[8]. In this proposed architecture, deep learning algorithm has been employed and feature extraction is made by variance fractal dimension trajectory (VFDTv2) as a pre-processing step. The proposed model demonstrated 87.35%

Table 3. Frequency of selected features between the proposed models

Feature	C4.5	KNN	SVM	PSO	HHO	HHO-PSO
F2	3	0	0	0	1	0
F3	3	0	3	5	7	**8**
F4	**9**	**10**	**9**	**8**	**9**	**10**
F5	**10**	**10**	**10**	**9**	**10**	**10**
F6	1	4	1	4	4	**8**
F7	1	0	1	0	0	0
F8	**10**	**10**	**10**	**8**	**10**	7
F10	**10**	**10**	**10**	7	**8**	4
F11	1	0	1	0	3	0
F12	**10**	**10**	**10**	**9**	**10**	**10**
F23	**10**	**9**	**10**	**9**	**10**	7
F25	**10**	**9**	**10**	7	**9**	**10**
F26	0	**8**	0	0	0	0
F29	**10**	**10**	**10**	**8**	4	2
F30	**10**	**10**	**10**	**8**	**10**	**9**
F33	0	**7**	0	0	0	0
F34	0	0	0	0	0	0
F35	**9**	**10**	**9**	**8**	**8**	6
F36	**10**	**10**	**10**	**10**	**8**	5
F37	**8**	**9**	**8**	**8**	7	1
F38	1	3	1	2	4	1
F39	0	**10**	0	3	5	**8**
F40	0	0	0	0	0	0
F41	1	0	1	0	0	0
Total Number of Selected Features	12	14	12	10	10	8

Table 4. The Performance Metric Values of the proposed IDS models

Model Under Study	Accuracy	Precision	Sensitivity	Specificity	F1 Score
BPN	0.9272	0.9663	0.9111	0.9515	0.9379
MLP	0.9380	0.9726	0.9227	0.9611	0.9470
PSO-BPN	0.9440	0.9732	0.9315	0.9623	0.9519
PSO-MLP	0.9473	0.9722	0.9376	0.9613	0.9546
HHO-BPN	0.9501	0.9686	0.9451	0.9571	0.9567
HHO-MLP	0.9584	0.9699	0.9576	0.9596	0.9637
GA-BPN Chiba et al.[2]	0.9212	0.9618	0.9056	0.9450	0.9328
PSO-BPN Liu et al.[3]	0.9096	0.9618	0.8886	0.9434	0.9237
BR-BPN Ali et al.[4]	0.8989	0.9556	0.8777	0.9335	0.9150
Proposed Hybrid HHO-PSO BPN	**0.9708**	**0.9725**	**0.9761**	**0.9638**	**0.9743**
Proposed Hybrid HHO-PSO MLP	**0.9774**	**0.9763**	**0.9838**	**0.9690**	**0.9800**

classification accuracy which is less when compared to the proposed hybrid IDS model in this study. Goparaju & Bandla (2020)[9] presented an ANN-based DDoS intrusion detection model based on open CICIDS2017 dataset that requires higher computational power which is greatly reduced in the hybrid BPN-MLP IDS model. Doriguzzi-Corin et al. (2020)[10] developed a CNN model for DDoS attack detection. This study has addressed four main strategies namely CNN based traffic classification, the dataset pre-processing tool, an activation analysis and experiential validation of the proposed classifier algorithm with other conventional models understudy. This model is significantly slower when compared to the proposed hybrid BPN-MLP IDS model. To mitigate the anomaly in traffic protocols, Saharkhizan et al. (2020)[11] introduced an Adaptive Resource Management Enabling Deception (ARMED) technique which enables intrusion identification before the endpoint gets affected. Detection time is more in this model when compared to the proposed hybrid BPN-MLP IDS model Fisher et al. (2020)[12] proposed an automated self-modelled neural network model for unauthorized traffic classification, the proposed model outperformed the other conventional state of art tools by effective high load optimization policy. The vehicle-to-cloud (V2C) technology is the emerging smart technology in cloud computing and IoT services. To provide better cloud vehicle security, an intelligent artificial intelligence (ISRM-AI) has been presented and it was identified that the proposed CNN mechanism guarantees the reliability of the provided service in the V2C environment at a slower rate but the proposed hybrid BPN-MLP model performs all these functions at a faster rate. Tang et al. (2020)[13] designed a CNN model for Low-rate denial-of-service (LDoS) attack, the LDoS attack is difficult to identify because of its low average rate and multiple feature variation during the course of the attack. The proposed model performs excellent classification on NS2 simulation and test-bed tool kits but it requires larger dataset which is not required in the proposed BPN-MLP IDS model. Kona et al. (2020)[14] proposed a hybrid RNN architecture for DDoS intrusion detection, in the proposed approach a time series algorithm is utilized to pre-process the input data. Then the data is sent to the novel ensemble model which achieves classification accuracy of 92.2 % but the proposed hybrid BPN-MLP model achieved classification accuracy of 97.74% and 97.08% respectively.

Jia et al. (2020)[15] proposed a CNN equipped backup server for detecting suspicious traffic according to the characteristics of attack in the DoS/DDoS environment. This model detects suspicious traffic very slowly when compared to BPN-MLP model. Kupershteine et al. (2019)[16] studied various feed-forward neural network architecture models for DDoS attack detection in the IoT environment but it has no parameters to optimize whereas parameters are effectively optimized in the proposed BPN-MLP model. Lu et al. (2020)[17] combined the improved PSO (IPSO) optimization algorithm with BPN for intrusion classification in WSNs. The network parameters of the proposed approach are optimally tuned by the IPSO algorithm and experimentally validated by employing NSL-KDD and UNSW-NB15 datasets. This model has low convergence rate whereas the proposed BPN-MLP IDS model has higher convergence rate. To detect the malicious traffic in cloud computing infrastructure, Tang et al. (2019)[18] developed a neural network modelby combining CNN, BP, and LSTM to train the security features of the network. In this IDS model, learning is done within a short time interval with better performance measures. But it is computationally very expensive, which is greatly reduced in the proposed hybrid BPN-MLP model. A CNN based DDoS classification strategy was developed by Shaaban et al. (2019)[19] to detect and mitigate the DDoS attack at an early stage. Various benchmark datasets were employed to classify the malicious traffic and the obtained results are compared with the performance of other ML classifier algorithms such as D-Tree, SVM, K-NN, and NN models. This model requires lots of training data whereas less training data is required in the proposed hybrid BPN-MLP model.

Hannache & Batouche (2020)[20] proposed a Traffic Flow Classifier model based on a neural network strategy for DDoS attack detection in the SDN environment with high complexity which is greatly reduced in the proposed hybrid BPN-MLP model. The impact of optimization algorithms in selecting appropriate features for classification in a big data environment is discussed by Maslan et al. (2019)[21]. The abnormal traffic classification is carried out by a neural network model and the obtained results are compared with other classic models. This model shows that selecting essential features has improved the system performance considerably but it suffers from over fitting issue which is rectified in the proposed hybrid BPN-MLP model. To detect and defence the DoS attack in cloud computing, Gao et al. (2019)[22] developed a backpropagation neural network model utilizing the KDDCUP99 dataset and the proposed model performed multiple feature selection with improved classification accuracy. Training time is more in this model when compared to the proposed hybrid BPN-MLP model. Evmorfos et al. (2018)[23] discussed on Random Neural network model for SYN attacks in the IoT environment, the proposed model is trained with deep learning algorithms and investigated for its performance with the performance of classic LSTM model. Time taken to develop this IDS model is longer when compared to the proposed hybrid BPN-MLP model. Lai et al. (2019)[24] developed a flow-based intrusion detection model based on MLP and the performance is compared with the performance of the packet-based detection strategy, the flow-based outperformed the packet-based model. This model lacks hyperparameter tuning which is effectively done in the proposed hybrid BPN-MLP model.

6 Conclusion

This part of the research has solved intrusion classification problem by using Multilayer Perceptron (MLP) and Back Propagation Network (BPN) network models. The hybrid optimization algorithm based on combining the best features of Particle Swarm Optimization (PSO) and Harris Hawks Optimization (HHO) has been employed to optimize the proposed neural network models. The proposed algorithm is validated with benchmark functions. This algorithm is also employed to select the necessary features that represent cyber security activity in the network. The performances of the models are analyzed based on the metric values. The proposed hybrid HHOPSO optimization algorithm has improved the performance of the neural network models by providing the optimal feature subset. Finally, the performance of the proposed models is compared with the existing models and confirmed that the proposed hybrid HHO-PSO-MLP neural network model outperformed with better metric values but with the limitation of delayed convergence. So, in future an attempt can be made to address this issue by utilizing Recurrent Neural Network model and previous hidden states during the learning process. Various setbacks in literature in this field of research such as unstable output and poor false prediction problem can be rectified using auto encoder and decoder based deep learning IDS model.

References

1) Sokkalingam S, Ramakrishnan R. 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*. 2022;34(27):e7334. Available from: https://doi.org/10.1002/cpe.7334.
2) Nagarajan G, Sajith PJ. Optimization of BPN parameters using PSO for intrusion detection in cloud environment. *Soft Computing*. 2023;p. 1–2. Available from: https://doi.org/10.1007/s00500-023-08737-1.
3) Narengbam L, Dey S. Harris hawk optimization trained artificial neural network for anomaly based intrusion detection system. *Concurrency and Computation: Practice and Experience*. 2023;35(23):e7771. Available from: https://doi.org/10.1002/cpe.7771.
4) Shankar SS, Hung BT, Chakrabarti P, Chakrabarti T, Parasa G. A novel optimization based deep learning with artificial intelligence approach to detect intrusion attack in network system. *Education and Information Technologies* . 2023;p. 1–25. Available from: https://doi.org/10.1007/s10639-023-11885-4.
5) Chiba Z. New Anomaly Network Intrusion Detection System in Cloud Environment Based on Optimized Back Propagation Neural Network Using Improved Genetic Algorithm. *International Journal of Communication Networks and Information Security (IJCNIS)*. 2019;11(1):61–84. Available from: https://doi.org/10.17762/ijcnis.v11i1.3764.
6) Liu Z, He Y, Wang W, Zhang B. DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN. *China Communications*. 2019;16(7):144–155. Available from: https://doi.org/10.23919/JCC.2019.07.012.
7) Ali S, Li Y. Learning Multilevel Auto-Encoders for DDoS Attack Detection in Smart Grid Network. *IEEE Access*. 2019;7:108647–108659. Available from: https://ieeexplore.ieee.org/document/8788512.
8) Ghanbari M, Kinsner W. Detecting DDoS Attacks Using Polyscale Analysis and Deep Learning. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*. 2020;14(1):17–34. Available from: https://www.igi-global.com/pdf.aspx?tid=240242&ptid=229511&ctid=4&oa=true&isxn=9781799805311.
9) Goparaju B, Bandla SR. Distributed Denial of Service Attack Classification Using Artificial Neural Networks. *EasyChair Preprint*. 2020;p. 2–10. Available from: https://easychair.org/publications/preprint/D1fr.
10) Doriguzzi-Corin R, Millar S, Scott-Hayward S, Martinez-Del-Rincon J, Siracusa D. Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection. *IEEE Transactions on Network and Service Management*. 2020;17(2):876–889. Available from: https://ieeexplore.ieee.org/document/8984222.
11) Saharkhizan M, Azmoodeh A, Dehghantanha A, Choo KKRK, Parizi RM. An Ensemble of Deep Recurrent Neural Networks for Detecting IoT Cyber Attacks Using Network Traffic. *IEEE Internet of Things Journal*. 2020;7(9):8852–8859. Available from: https://ieeexplore.ieee.org/document/9097894.
12) Fisher D, Chandler A, Greton J, Delport C. Implementing embedded uniqueness for naturally one-to-one monoids in a high-speed learning neural network for cyber defense. *Software Engineering Review*. 2020;1(1). Available from: https://doi.org/10.1177/15501329221084882.
13) Tang D, Tang L, Shi W, Zhan S, Yang Q. MF-CNN: a New Approach for LDoS Attack Detection Based on Multi-feature Fusion and CNN. *Mobile Networks and Applications*. 2021;26(4):1705–1722. Available from: https://doi.org/10.1007/s11036-019-01506-1.
14) Kona SS. Detection of DDoS attacks using RNN-LSTM and Hybrid model ensemble. Dublin, Ireland. 2020. Available from: https://norma.ncirl.ie/id/eprint/4180.
15) Jia W, Liu Y, Liu Y, Wang J. Detection Mechanism Against DDoS Attacks based on Convolutional Neural Network in SINET. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 12-14 June 2020, Chongqing, China. IEEE. 2020;p. 1144–1148. Available from: https://doi.org/10.1109/ITNEC48623.2020.9084918.
16) Kupershtein LM, Martyniuk TB, Voitovych OP, Kulchytskyi BV, Kozhemiako AV, Sawicki D, et al. DDoS-attack detection using artificial neural networks in Matlab. In: Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019, Wilga, Poland;vol. 11176. SPIE. 2019;p. 521–530. Available from: https://doi.org/10.1117/12.2536478.
17) Lu X, Han D, Duan L, Tian Q. Intrusion detection of wireless sensor networks based on IPSO algorithm and BP neural network. *International Journal of Computational Science and Engineering*. 2020;22(2-3):221–232. Available from: https://doi.org/10.1504/IJCSE.2020.107344.
18) Tang X, Chen M, Cheng J, Xu J, Li H. A security situation assessment method based on neural network. In: International Symposium on Cyberspace Safety and Security, CSS 2019: Cyberspace Safety and Security ;vol. 11983 of Lecture Notes in Computer Science. Springer, Cham. 2020;p. 579–587. Available from: https://doi.org/10.1007/978-3-030-37352-8_52.
19) Shaaban AR, Abd-Elwanis E, Hussein M. DDoS attack detection and classification via Convolutional Neural Network (CNN). In: 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), 08-10 December 2019, Cairo, Egypt. IEEE. 2020;p. 233–238. Available from: https://ieeexplore.ieee.org/document/9014826.

20) Hannache O, Batouche MC. Neural Network-Based Approach for Detection and Mitigation of DDoS Attacks in SDN Environments. *International Journal of Information Security and Privacy*. 2020;14(3):50–71. Available from: https://www.igi-global.com/article/neural-network-based-approach-for-detection-and-mitigation-of-ddos-attacks-in-sdn-environments/256568.

21) Maslan A, Mohammad KM, Foozy FBM, Rizki SN. DDoS Detection on Network Protocol Using Neural Network with Feature Extract Optimization. In: 2019 2nd International Conference on Applied Information Technology and Innovation (ICAITI), 21-22 September 2019, Denpasar, Indonesia. IEEE. 2020;p. 60–65. Available from: https://ieeexplore.ieee.org/document/8982136.

22) Gao L, Li Y, Zhang L, Lin F, Ma M. Research on Detection and Defense Mechanisms of DoS Attacks Based on BP Neural Network and Game Theory. *IEEE Access*. 2019;7:43018–43030. Available from: https://ieeexplore.ieee.org/abstract/document/8674748.

23) Evmorfos S, Vlachodimitropoulos G, Bakalos N, Gelenbe E. Neural network architectures for the detection of SYN flood attacks in IoT systems. In: PETRA ’20: Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments. ACM. 2020;p. 1–4. Available from: https://doi.org/10.1145/3389189.3398000.

24) Lai YC, Zhou KZ, Lin SR, Lo NW. F1ow-based Anomaly Detection Using Multilayer Perceptron in Software Defined Networks. In: 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 20-24 May 2019, Opatija, Croatia. IEEE. 2019;p. 1154–1158. Available from: https://doi.org/10.23919/MIPRO.2019.8757199.