WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM

e-ISSN: 2788-5879 p-ISSN: 2788-5879

Hybrid Feature Selection Approach to Improve the Deep Neural Network on New Flow-Based Dataset for NIDS

Rawaa Ismael Farhan[1,*], Dr. Abeer Tariq Maolood[2] and Dr.NidaaFlaih Hassan[2]

[1]Department of Computer Science, University of Technology, Wasit University, Iraq

[2]Department of Computer Science, University of Technology, Iraq

*Corresponding Author: Rawaa Ismael Farhan

DOI: https://doi.org/10.31185/wjcm.Vol1.Iss1.10

Received: December 2020; Accepted: February 2021; Available online: March 2021

ABSTRACT: Network Intrusion Detection System (NIDS) detects normal and malicious behavior by analyzing network traffic, this analysis has the potential to detect novel attacks especially in IoT environments. Deep Learning (DL)has proven its outperformance compared to machine learning algorithms in solving the complex problems of the real-world like NIDS. Although, this approach needs more computational resources and consumes a long time. Feature selection plays a significant role in choosing the best features only that describe the target concept optimally during a classification process. However, when handling a large number of features the selecting such relevant features becomes a difficult task. Therefore, this paper proposes Enhanced BPSO using Binary Particle Swarm Optimization (BPSO) and correlation–based (CFS) classical statistical feature selection approach to solve the problem on BPSO feature selection. The selected feature subset has evaluated on Deep Neural Networks (DNN) classifiers and the new flow-based CSE-CIC-IDS2018 dataset. Experimental results have shown a high accuracy of 95% based on processing time, detection rate, and false alarm rate compared with other benchmark classifiers.

Keywords: Network Intrusion Detection System, Feature Selection, Deep Learning, CFS, Binary PSO, CSECIC-IDS2018

1. INTRODUCTION

The exponentially growing number of security breaches, cyberattacks on Internet of things IOT highly required reliable security solutions. Network Intrusion Detection System (NIDS) used as defense of network infrastructure by detecting malicious activities and preventing attacks [1]. NIDS can be divided into misuse detection is also called signature-based detection and anomaly detection NIDS that are monitoring the network pattern and learning the normal behavior of a system and distinguish each network activity detect it as an intrusion when deviate from the normal pattern [2]. We focus on anomaly detection NIDS because its ability to detect unknown attacks despite it have high false alarm rate because inability to determine reasons of an abnormality.

Traditional machine learning (ML) approaches have been supplied for cyber security such as Bayesian Belief Networks (BBN), Random forest, Support Vector Machines (SVM) and others, but the generation of large scale data in IoT required a deep learning based approach which performs better with large data sizes and can learn representation of feature from raw data so it is adaptable to different attack scenarios [3]. They proposed a new malware prediction model that could detect the coming future malware by the implementing a deep learning method of Mal Generative Adversarial Network (Mal-GAN) [4]. showed that the LSTM classifier outperform over previously published results of other static classifiers on KDD Cup '99 dataset challenge for long time which prove the benefit of LSTM networks to intrusion detection, because the ability of LSTM to learn from look back in time and link connection records consecutively [5].The RNN, Stacked

RNN, and CNN are supervised deep learning techniques applied to classify common five attack types using Keras .This technique used packet header information without need any user payload then compared its results with Snort IDS .The results showed that this technique gave superior results compared Snort [6]. Variant-Gated Recurrent Units (GRU) with encoders performed on ISCX2012 dataset to make preprocessing on packets of payload-aware intrusion detection. It could learn features of network packet header and payload automatically and improved the detection rate of the IDS [7]. proposed RNN-RBM model which take input data as byte-level without feature engineering. At first, RBM model used network packets to extract the feature vectors. Then RNN model extracted the flow feature vector which sent to the Softmax layer to detect result [8].

Recently, the large growth of data makes big challenge to the task of data classification. The feature selection is an option to solve this challenge by reducing the dimensionality of the data and achieve higher accuracy in data classification.

In terms of feature selection, it plays an important role in improving NIDS performance. This is because anomaly detection uses a large number of time-consuming features. Therefore, choosing the method for selecting the feature affects the improvement of the level of accuracy and the time required to check traffic behavior. There are three types of features selection: filter, wrapper and embedded techniques. The filter technique tries to classify a subset from the original set containing of several selected features based on the evaluation criteria. While the wrapper technique, chose the features that have high predictive accuracy from different learning algorithms. The embedded technique where the feature selection embeds into the training step [9]. Paper performed Experiments on NSL-KDD datasets using log2 and PCA on deep learning algorithms. Results proved the effect of dimensionality reduction on the accuracy ratio about 97.9%. Thus, minimizing features in dataset and select optimal subset of most relevant features for each class to reduce processing time, improve detection accuracy rate, reduce false alarm rate. As result, the efficiency for intrusion detection in IOT environment improved because the irrelevant and redundant features cause overfitting and poor generalization during the classification [10].

Optimization means finding the optimal solution from a set of choices with regards to an objective function and some conditions. Intelligent applications that using Swarm Intelligence algorithms are becoming famous because of their ability to handle any real time complex and uncertain situation. Swarm intelligence is kind of algorithms which simulate the behavior of living organisms such as birds, insects, and fish. These individuals able to complete complex tasks in real world when working in unity that would be very difficult to achieve it [11].

In today's world, application of Swarm intelligence and Deep learning have been provided in many fields successfully such as image classification, pattern recognition and intrusion detection system. this paper has designed seven layer CNN which commonest deep learning approach, called ConvNet performed to classification of handwriting digit. The Particle Swarm Optimization algorithm (PSO) is used to improve the input parameters of processing layers [12]. This work proposed approach of swarm intelligence for parameter setting in deep neural network. through providing this approach to the phishing websites classification. As a result, the proposed algorithm improves their detection compared to other algorithms [13]. The contributions of this paper as following:

Using a swarm intelligence for features selection by implementing a Binary PSO algorithm.

Improving the NIDS by optimized deep learning models with pre-processing phase employing a Binary PSO algorithm. This approach optimized detection rate (DR) of deep learning models while reducing false alarm rate(FAR)compared with corresponding values of deep learning models without preprocessing phase.

Evaluating this approach by using new CSE-CIC-IDS2018 real datasets for classification tasks.

Presenting four comparative analyses between our results and the literature best results. Also, employing several evaluation metrics to depict analysis performance of deep learning models on our approach.

However, Swarm intelligence are often limited by weak points of computation time and local solution for large and complex problems. While, Deep learning algorithms are often limited by weak points of data and parameters

2. RELATED WORK

The Paper proposed a new algorithm to optimize the structure of DBN network. At first designed a PSO next used the fish behavior to optimize the PSO and find the initial solution of optimization. Then, used the genetic operators (crossover probability and mutation)on the PSO to search the global solution for optimization which used to construct the network structure for intrusion detection on NSL-KDD [14].The researcher aimed to improving the performance of NIDSs on UNSW- N15 dataset by proposed four feature selection models based on the particle swarm optimization (PSO), firefly optimization (FFA),genetic algorithm (GA)and grey wolf optimizer (GWO).The derived features from this model are evaluated on the J48 ML and support vector machine(SVM) classifiers [15].A double PSO-based algorithm proposed to select subset of features and hyper parameters both in the same work . Three deep learning models (Deep Neural Networks (DNN), Deep Belief Networks (DBN) and Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN) utilized to show the differences in performance on CICIDS2017dataset [16]. PSO-RF is an intrusion detection mechanism based

on binary particle swarm optimization (BPSO) and random forests (RF) algorithms to find best features set by BPSO and RF as a classifier for classifying intrusions of networkonKDD99Cup dataset [17].

3. METHODOLOGY

This section proposed Swarm-based intrusion detection method which is a hybrid feature selection approach used correlation–based (CFS) classical statistical feature selection approach to enhance the performance of BPSO feature selection. This method improves the previous work by applying Enhanced BPSO-based algorithm for feature selection stage. The proposed algorithm will optimize the detection performance of deep learning.

3.1 CSE-CIC-IDS2018 DATA SET SPECIFICATION

The traditional NSL-KDD dataset and others dataset not reflect situations of real world. According to Gharib et al. [18], determined 11 essential criteria for each dataset to be reliable dataset, but none of previous NIDS dataset covered all criteria. While, our CSE-CIC-IDS2018 dataset covered all 11 criteria. CSE-CIC-IDS2018 dataset represents a shift from static data to dynamically generated data available on AWS cloud [19, 20]. This paper evaluated the NIDS on a real traffic captured from AWS network and machines log files with 80 extracted features from 50 terminals represent Attacking infrastructure and 30 servers and 420 computers represent the infected organizations comprised. Seven types of attacks occurred: DOS, DDoS, Botnet, Web attacks, Brute-force, infiltration and Heartbleed [20].

CSE-CIC-IDS2018 dataset contains details of intrusions with details of protocols. The applications and lowest level entities of network are representing best approach of testing and evaluation, also it refers to shifting from Static data to Dynamic data which is real-time traffic on the Amazon platform (AWS). To download this dataset, the following description is applied:

"Resource type
S3 Bucket
Amazon Resource Name (ARN)
arn:: aws:: s3: cse-cic-ids2018
AWS Region
ca-central-1" [87]

The dataset has 80 features divided into 8 classes contained 1 normal class and 7 classes of attacks as following:

1. DOS: this type of network attack is a famous in which attackers deny the legitimate users by sending or overwhelming number of bogus requests target a service. Attackers used to generate DoS attack traffic such as, Goldeneye, Slowloris, Hulk and Showhttptest commonly available.

2. DDoS: represent more sophisticated Distributed Denial of Service where attackers flood the target systems or services using multiple botnets from around the world composed of thousands of compromised systems possibly to the overwhelming amount of network traffic.

3. Brute-force: Attackers in this type of network attack using each key combination to estimate online passwords or examine the existence of hidden web pages such as admin login pages. It consists of two common network services FTP and SSH occurs as FTP-Brute force and SSH-Brute force.

4. Heartbleed: This attack-type also categorized into Brute-force where Heartbleed attacks generate Traffic against the notorious Heartbleed Bug.

5. Botnet: Attackers use group of botnet as a single virtual network contained network systems and devices working implement several Internet attacks Attackers use a group of a botnet as a single virtual network contained network systems and devices working implement several Internet attacks such as (phishing attacks, send spam, sniffing and stealing data, provide backdoor access to compromised systems, etc.) by sniffers and key loggers.

6. Infiltration: This attack-type represents internal network Attacks.

7. Web attacks: BruteForce-XSS, SQL-Injection, cross-site scripting (XSS) are categories of web attacks in modern web applications.

Attacking infrastructure comprised 50 terminals and the infected organizations comprised 30 servers and 420 computers. This dataset represented the captured traffic of AWS network and machines log files with 80 extracted features represented bidirectional information of flow forward (source to destination) and backward (destination to source) directions by using

CICFlowMeter-V3. It is a flow – based dataset where flow refer to a set of IP packets passing the network through certain point during a specific interval of time. Each flow containing packets with common properties. Table 3.1 shows A sub set of extracted traffic features.

3.2 PROPOSED WORK

In this paper, the detailed aspects of the proposed system for Optimized Deep Learning with Enhanced Binary PSO for Feature Selection and Classification shown in Figure.1 work is given as framework consist from five phases, they are:

1. In the first phase of the research, which including real-world AWS-flow (CSE-CIC-IDS2018) dataset capturing. The cybersecurity dataset developed in collaboration between the Canadian Cyber Security Institute (CIC) and the Communications Security Corporation (CSE). Thus created a systematic method that uses profiles to develop a detailed description of intrusions associated with protocols and lower levels of network entities.

2. In the second phase, preprocessing steps comprised of feature encoding and feature normalization. False Alarms (FAR) arise during the classification because of rough features. Therefore, the preprocessing phase on the dataset is an essential part to reduce FAR. While, in data preparation, raw data transformed into a more suitable form for modeling. Therefore, floating-point numbers must be entered in a range from 0 to 1 to the input layer of the Deep Neural Network (DNN). There are three steps for data preprocessing:

    - Feature Encoding: convert categorical features into numerical values.
    - Feature Normalization: used for changing scale, type, and probability distribution of variables in the dataset.
    - Feature Selection: A crucial step that focuses on feature selection with swarm intelligence using Enhanced Binary particle swarm optimization algorithm (BPSO). Due to Big Data challenges, the feature selection in intrusion detection increase efficiency of the classification by reducing computational processes. Feature selection play significant role with high quality real data sets compared with traditional KDD data sets for Intrusion Detection, due to select only most correlated features with certain class.

3. In the third phase, to evaluate these built models available data into training and testing data needed to be divided. Cross-validation technique has been used for evaluating how to generalize the statistical analysis results on an independent dataset. It is the famous approach to computing the learning model accuracy which improve the reliability of classifier. It estimates how the predictive model will perform in practice accurately. A 5-fold approach is used in this work. Thus the data is divided through 5 parts randomly and each run used one partition of these for testing and the remaining partitions used for training iteratively. The training data is used to fitting the model, and the test data to evaluate it, while the labels of the test data are known.

4. In the fourth phase, directing experiments towards Deep Neural Network (DNN) modeling. then perform classification on the training set, which is explained in this chapter in detail.

5. Finally, Attack detection and performance evaluation on the testing set with different metrics to highlight whole strong perspectives of our proposed system and other works comparison.

3.3 FEATURE SELECTION

Set of techniques used to select optimal subset of input features which are most relevant to the target variable need to predict. Dimensionality of the data mean the number of input features for a dataset, but the problem is more dimensions in space make dataset representation a very sparse and unrepresentative for samples in that space. So, this motivates feature selection that remove irrelevant and redundant input features which leading to lower predictive performance. Thus, models could develop by using the data that is required to make a prediction only.

this paper employed feature selection for reducing irrelevant attributes using random forest algorithm. Therefore, the task of NIDS became efficient. While, PSO algorithm performed on the selected features of the NSL-KDD dataset. This minimize the false alarm rate and increase the detection rate and the accuracy of the NIDS compared with machine learning classifiers such as SVM, KNN, DT and LR algorithms [21]. A new method has been presented based on particle swarm optimization with multiple criteria linear programming that improve attacks detection accuracy. During training phase, PSO used for tuning parameters to optimize the MCLP classifier performance [22]. Due to PSO has advantages that are simple implementation, fewer number of parameter, and no calculation of mutation. It is considered as best search algorithm for optimization. PSO classified into Standard PSO and binary PSO. Standard PSO assigns real numbers to particles, but binary PSO assigns binary numbers to particles. But, there is a possibility that binary PSO (BPSO) will

quickly fall into local minima so it has been suggested to enhanced it by correlation–based (CFS) classical statistical feature selection approach.

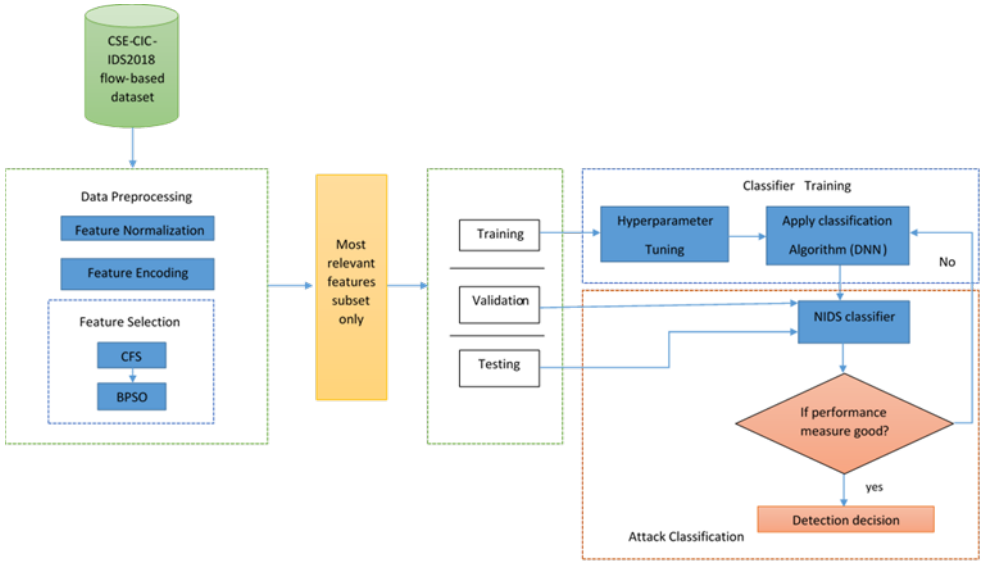

FIGURE 1. General Flowchart of proposed Model

3.4 CORRELATION–BASED FEATURE SELECTION (CFS) APPROACH

CFS is a filter- based technique where the major idea behind this approach is to evaluate the relevance of selected feature to class and the redundancy between the selected feature subset to obtain the optimal solution in search space. Features are choosing according to the result of the feature subset assessment using the function of correlation. This means the chosen features are maximally related to the class but not related to each other. Each feature with a high score predicts classes in the instance space more than different features as following Eq. 1:

$$Cs = \frac{f\ dcf}{\sqrt{f + f(f-1) + dff}} \tag{1}$$

Where CS is the evaluation for s feature subset comprised f features, dcf is the degree of correlation mean between features and the class label, and dff is the degree of inter-correlation mean between features. The evaluation of CFS is a method of correlation based on feature subsets. Thus, higher evaluation value come from bigger dcf or smaller dff in selected subsets. Finally, as shown in Figure. 2 the selected subsets of features which have highest value used to reduce both the training and testing set.

3.5 BINARY PARTICLE SWARM OPTIMIZATION (BPSO)

In [23] PSO after the population initialization each particle update its velocity and its position in each iteration based on their own experience (pbest) and the best experience of swarm (gbest) as in Equations (2) & (3). Then the performance of all particles evaluated by predefined cost functions at end of each iteration.

$$Vj(st+1) = W * Vj\ (st) + F\ 1\ d\ 1\,(Pj\ best\ (st) - Pj(st)) + F\ 2\ d2\,(G\ best\,(st) - Pj(st))$$

$$Pj\,[st+1]\ =\ Pj\,[st] +\ Vj\,[st+1] \tag{3}$$

Where:

At each iteration **st** each particle **j** Acquire three vectors velocity, position and personal best all in length **N** which refer to the problem dimension. When either the improved value of the global best is smaller than stopping value (ε) or reached the maximum iteration number the stop condition is met and PSO terminates.

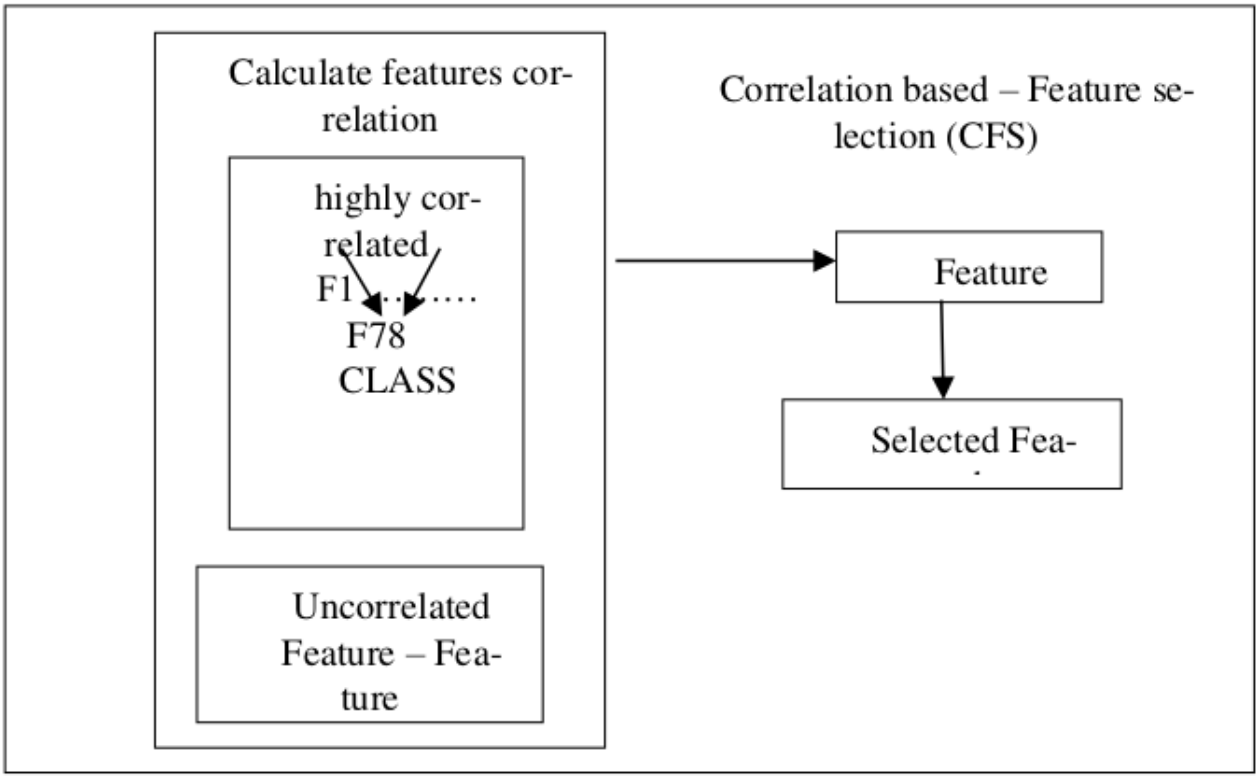

FIGURE 2. Correlation based – Feature selection (CFS)

The standard PSO used in continuous domains well, while in discrete space it gives poor effects on the results. Usually, the binary PSO in the feature selection problem outperforms the standard PSO because that the problem of feature selection occurring with a discrete search space. BPSO search space is seemed as a hypercube where a particle moves to nearer and farther corners of the hypercube through flipping bits into various numbers. The moving velocity represent changes of probabilities for the bit which may be in one state or the other. So, a particle in each dimension moves in a state space limited to 0 and 1 [24]. Therefore, we will exploit the binary PSO in our design for the feature selection method.

To implement the BPSO, the selected number of population is 100 and the number of iteration is considered to be 10, Initialize swarm randomly where X = (x1, x2, …., x n) is a particle as feature vector and y ε [0,1] represent class label which 0,1 respectively refer to normal and abnormal. Then, setting parameter as following:

W is the constant refer to Inertia weight that controls the velocity impact of particle during the current iteration it is usually ranged in [0.4,0.9]. **F 1** and **F 2** are acceleration coefficients constants ranged in [0.5]. while, **d 1** and **d 2** which are values ranged randomly in [0,1]. These parameter scale both of personal knowledge and swarm knowledge on the velocity changes. Consequently, calculate Activation Function to measure fitness value of each particle as in equation (4) to select particle with best value and called gbest.

$$F(X) = (1 - Pr) + (1 - \alpha)(1 - \frac{Ns}{Nv}) \tag{4}$$

X is the input variable where $\mathbf{P_r}$ is the measure of classifier performance and **Ns** is the feature subset size have been tested and $\mathbf{N_v}$ is the total number of available input variables. The term on the left side of the equation refer to the total accuracy and the term on the right for the used features percentage.

BPSO is resulted by adapting equations in standard PSO to be suitable to binary space. The velocity vector in BPSO shows the probability of taking value 1 for element in the position vector. Moreover, the sigmoid function in Eq. (5) used to convert $V_j^{(st+1)}$ to the range of [0,1].

$$S(v\ jst\ +1\ ) = \frac{1}{1\ +\ e\ -(v\ jst\ +1\ )} \tag{5}$$

where rand ( ) is a random selected value from range [0,1].

$$P_j^{st+1} = \begin{cases} 1 \text{ if rand } () < S\left(V_j^{st+1}\right) \\ 0 \text{otherwise} \end{cases} \quad (6)$$

Update position and velocity of each particle as equation (5) and (6). Finally, PSO output optimal solution that is global best vector next checks the stop condition when one met it the PSO will terminates.

3.6 ENHANCED BPSO FEATURES BASED ON CFS SELECTION

To enhancing the feature selection of standard BPSO Algorithm has been proposed CFS classical statistical method, where Algorithm 1 proposed Enhanced BPSO Algorithm as the following procedure:

1. Determine score for each feature using CFS –based correlation equation.
2. Put certain threshold then select all features that larger than threshold.
3. Perform BPSO on selected features subset.
4. Further selection to reduce the redundant features and select optimal features subset.

Algorithm 1Enhanced BPSO Algorithm
Input: Training set, Testing set.
Output: selected feature subset Xbest
Begin
1.Initialize population of N particles Xj best =(X1, …,DT) , j=1,2,3,….,N
2. Initialize fit(Xj) and Xbest // eq (3.2) correlation –based equation//
3. Initialize Fit temp (j) and X temp (j) for solution storage //select solution from set of best solutions and generate local solution around it //
4. while j<=max no. of iteration do
5. For j=0 to n do
6. Update Xj and Vj// check population for optimal order by iterative update for location and velocity //
7. IF rand() < Vj then
8. Select Xj from X best
9. Generate a new X new
10. end if
11. Calculate Fit (X new)// eq (3.2) correlation –based equation//
12. if Fit (X j) <= Fit (X new) then
13. Fit temp (j) = Fit (X new)
14. X temp (j) =X new
15. end if
16. if Fit (X new)>=max of Fit tempthen
17. X best= X new
19. end if
20. end for
21. t = t+1
22. end while
23. End

3.7 DEEP LEARNING CLASSIFIER

Our DNN model implemented on Windows10 using Visual Studio 2019 contain python 3.7 and installed Keras on top of Tensorflow using (Numpy, Scikit-learn, Panda) libraries ,8GB Memory, CPU core i7,512GB Hard disk, seaborn library for visualization results.

Deep learning models classified into two types supervised and unsupervised learning models. comprise, deep neural networks (DNNs), deep brief networks (DBNs), recurrent neural networks (RNNs)and convolutional neural networks (CNNs) as supervised learning models. In other hand, restricted Boltzmann machines (RBMs), auto encoders and generative adversarial networks (GANs) as unsupervised learning models [25].

Deep learning methods plays a significant role for flow-based datasets compared with machine learning models because do not required manual feature engineering. Thus, it can learn feature representations automatically from raw data. The deep structure of deep learning represents comparable characteristic where used multiple hidden layers compared with shallow models, which contain one hidden layer or none [26].

The 55 optimal features selected by the Enhanced BPSO algorithms from preprocessing phase will be provide to our DNN classifier to improve the performance of DNN contain three fully connected layers are used, they described in Algorithm 2 as following:

- dense1 layer with 55 neurons use ReLu Activation function.
- dens2 layer with 64 neurons use ReLu Activation function.
- dense3 layer with 10 neurons use Softmax Activation function.
- Regularization method with two dropout ratio (0.2) are used to avoid overfitting.

Table 1 shows that good tuning of hyper parameter values is important to avoid overfitting.

Table 1. Experimental hyper parameter of proposed DNN model

Parameters	Value
Epoch	100
Batch size	500
Activation function	ReLu , Soft max
Loss function	categorical_crossentropy
Optimizer	Adam

Algorithm 2 NIDS Framework

Input: CSE-CIC-IDS2018 & Preprocessing 78 features
Output: Display results of evaluation metrics (Accuracy, Precision, Recall, Detection Rate, False Alarm Rate, F1-score)
Begin
1. Function process –model()
2. Perform Feature selection on CSE-CIC-IDS2018 using Enhanced BPSO //it select 55feature vector//
3.Calculate the classification accuracy through DNN classifier on selected feature subset only
4.Compare the results with others from Literature
End

In this paper used two types of non-linear Activation function are ReLu and softmax. ReLu is faster than other non-linear Activation function which maximize the deep learning efficient while Softmax used for multi classification as output layer in DNN model because it outputs the probability of each class then choose biggest value for accurate result.

Loss function represent the difference between the predicted and actual output.

The Optimizer Adam used to minimize Loss function by calculate gradients of a loss after that apply gradients to update values and therefore enhance the DNN results.

4. EXPERIMENTAL RESULTS AND DISCUSSION

We directed comparative analyses by comparing our results to the previous results in the literature. In addition to that, we proposed various evaluation Measurement in order to investigate the differences in performance in different approaches and focus on performance of deep learning models through using our approach.

4.1 EVALUATION MEASUREMENTS

We used for model evaluation various performance metrics to give powerful view on our model which based on BPSO with DNN as following:

1. Confusion Matrix

In the intrusion detection, Confusion Matrix is a good tool to predict the network attack type where TP normal data and TN refer to the abnormal data correctly classified, while FP the normal data and FN refer to abnormal data of the misclassification. Figure .2 show Confusion Matrix as resulted from the seaborn library in Python.

2. Accuracy: A percentage of positive detection of all data cases.

3. Precision: How many attacks are properly returned.

4. Recall: How many attacks the system returns.

5. F1-score: Rate of Precision and Recall in our model.

6.Detection Rate (DR) and False Alarm Rate(FAR): DR and FAR depicts how the classifier distinguishes well the positive and the negative classes, respectively.

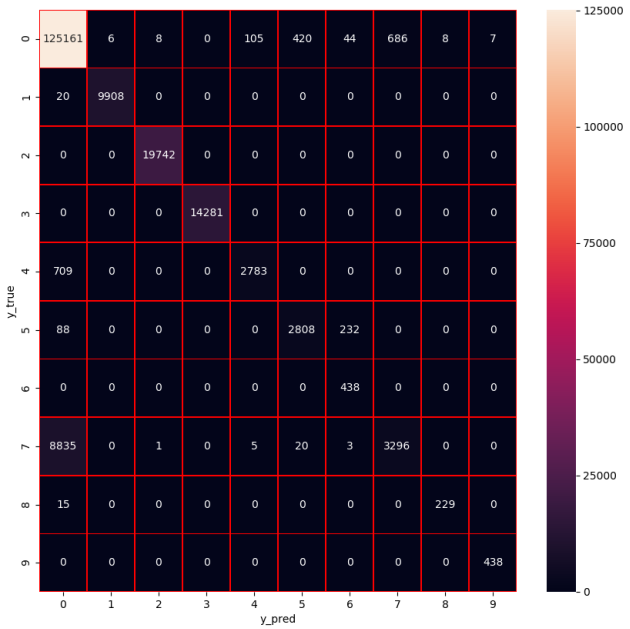

FIGURE 3. Confusion Matrix of DNN based on BPSO

These metrics described in Table 2 from which We determine the detection rate (DR) and false alarm report (FAR).

Accuracy, Precision, recall these criteria are limited, especially if one class among 10 classes is much larger than the other. With an imbalanced classification problem, the classification error in the minority class will not have much effect on the accuracy value. If the dataset is unbalanced, then in such cases, you only obtain very high accuracy by predicting the majority class, but you fail to capture the minority class, which is often the goal of creating the model in the first place .as shown in Table 3

The relation between training accuracy and testing accuracy showing in Figure.3, where the model accuracy reached to 94% only with 10 Epoch, while when increased Epoch to 100 We noticed that accuracy has settled on 95%.

Table 2. **Performance analysis of our Model**

Attack	PRECISION	RECALL	F1-SCORE
infiltrations	0.93	0.99	0.96
Benign	1.00	1.00	1.00
DDOS attack_HOLC	1.00	1.00	1.00
DDOS attack_LOTC_UDP	1.00	1.00	1.00
BOT	0.88	0.93	0.90
SQL Injection	0.74	0.89	0.84
FTP_Brutforce	0.00	0.00	0.00
SH_Brutforce	0.87	.27	0.41
DOS atack_slow HTTP Test	0.89	0.94	0.91
DOS attack_HULK	0.98	1.00	0.99

Table 3. **Performance quality assessment**

Attack type	0	1	2	3	4	5	6	7	8	9
DR %	0.144	2.217	7.035	0.0	0.002	0.005	0.0	0.003	0.000	5.267
FAR%	0.942	0.999	0.999	1.o	0.996	0.993	0.997	0.950	5.999	0.999

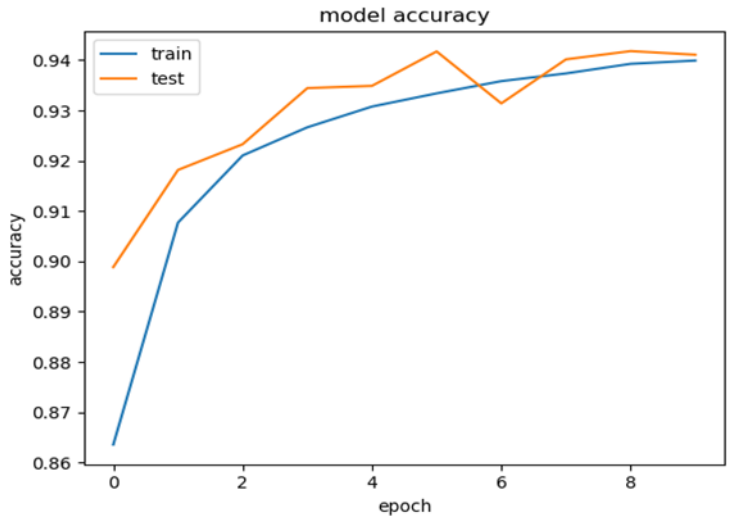

FIGURE 4. **Model Accuracy**

The Loss of Model showing in Figure.4, where in training phase beginning from o.6 and decreased through time, while The Loss of Model in testing phase beginning from o.4 and decreased through time. The elapsed Detection time about 580.27 sec.

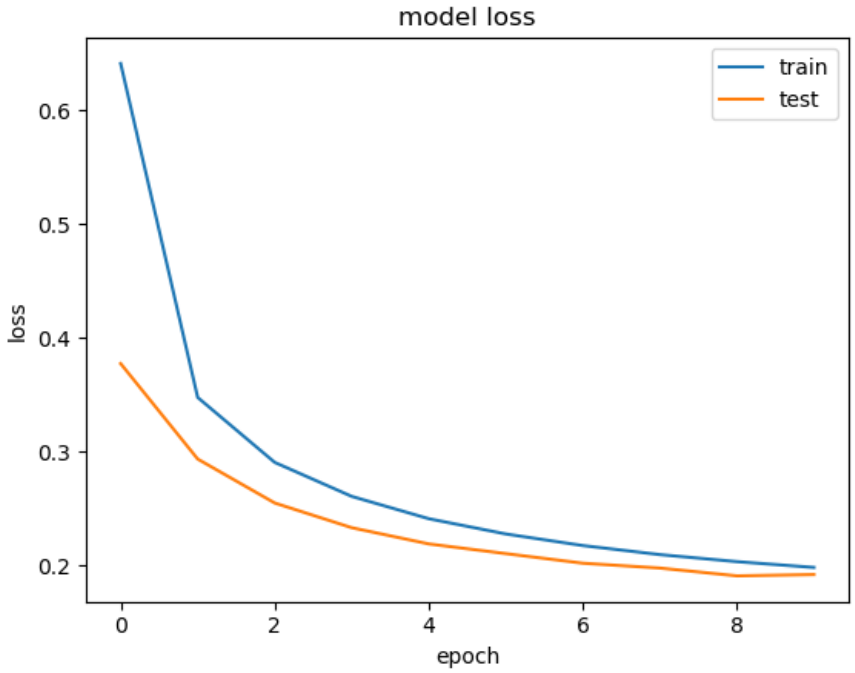

FIGURE 5. Loss of Model

4.2 COMPARATIVE ANALYSIS

A comparative study in Table 4 is directed for showing the differences between our implemented Deep Neural Network (DNN) with Binary PSO and other previous methods. Our approach used Binary PSO as preprocessing phase that select only 55 best features from 80 features contained in real CSE-CIC-IDS2018 dataset. The selected features feed into DNN classifier. The results showed superiority over the previous classifier without feature selection. Moreover, no research used our dataset.

Table 4. Comparative Analysis

Classification algorithm	Feature selection method	Dataset	Accuracy	DR	FAR
Our DNN	Binary PSO	CSE-CIC-DS2018	95%	1.464	0.982
[14] DBN	At first PSO, Then Fish	NSL-KDD	83.86%	20.94	2.4
[15] SVM,J48	PSO,GWO,FFA,GA	UNSW-N15	89.01 85.67 86.03 86.87	80.84 93.79 96.58 96.70	2.817 20.95 22.59 21.16
[16] three deep classifier DNN,LSTM,DBN	Double PSO	CI-CIDS2017	88.04 92.41 95.81	88.04 92.41 95.81	98.62 99.31 99.79
[27]Our Previous DNN	-	CSE-CIC-DS2018	90.25%	0.95	-

The accuracy of proposed model is 95% after 100 Epoch which consider good, because we used Binary PSO which is one of the available feature selection methods that reduce the dataset dimensionality and select the most relevant features

only. The proposed approach increase system accuracy, decrease false alarm report(FAR) and reduce the computation time where the elapsed time for Detection time about 580.27 sec

5. CONCLUSION

From the final results of proposed NIDS implementation reached during this thesis, concluding from proposed NIDS scenario the following:

1. The first proposal is Enhanced BPSO (Binary Particle Swarm Optimization as feature selection) is an efficient approach for feature selection because the fact that problem of the features selection occurred in discrete space, and this leads to select the best features which enhance the classification performance.

2. The proposed algorithm for features selection reduces the elapsed time consumed in the training and testing processes during classification process. Therefore, accuracy, detection rate increased and False Alarm Report (FAR)decreased.

3. Fully connected dense Deep Neural Network(DNN)is proposed, its used for flow-based intrusion detection on real-world dataset CSE-CIC-IDS2018 available at AWS platform, few papers addressed this dataset yet. The performance of proposed approach is examined on off line (NSL-KDD) and on-line (CSE-CIC-IDS2018) datasets, and the results for performance evaluation showed that the both types were good indicating that our model generalized well and did not deviate to a specific type of data.

4. Hyperparameters tuning for DNN model is recommended to be used for further efficiency on our proposed DNN. So, we used two dropout layers

FUNDING

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] S. Mishra, R. Sagban, A. Y. & Niketa, and Gandhi, “Swarm intelligence in anomaly detection system: an overview,” *International Journal of Computers and Application*, 2018.
[2] B. B. Zarpelao, “A Survey of Intrusion Detection in Internet of Things,” *Journal of Network and Computer Applications*.
[3] H. Liu and B. Lang *Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey*, vol. 9, pp. 439–439, 2019.
[4] S. Lu, “New Era of Deep learning -Based Malware Intrusion Detection: The Malware Detection and Prediction Based On Deep Learning,” *ArXiv*, 2019.
[5] R. C. Staudemeyer, “Applying long short-term memory recurrent neural networks to Intrusion detection,” *SACJ*, no. 56, 2015.
[6] N. Chockwanich and Vasakavisoottiviseth, “Intrusion Detection by Deep Learning with TensorFlow,” *International Conference on Advanced Communications (ICACT)*, 2019.
[7] Y. Hao, “Variant-Gated Recurrent Units with Encoders to Preprocess Packets for Payload-Aware Intrusion Detection,” *IEEE*, vol. 7, 2019.
[8] C. Li, “Using a Recurrent Neural Network and Restricted Boltzmann Machines for Malicious Traffic Detection,” *Neuro Quantology*, vol. 16, 2018.
[9] A. Nisioti, A. Mylonas, P. D. Yoo, and V. Katos *From Intrusion Detection to Attacker Attribution: A Comprehensive Survey of Unsupervised Methods*.
[10] M. K. Ibraheem *Network Intrusion Detection Using Deep Learning Based On Dimensionality Reduction*.
[11] C. Kolias, V. Kolias, and G. Kambourakis, “TermID : a distributed swarm intelligence-based approach for wireless intrusion detection,” *Int. J. Inf. Secur*, 2016.
[12] H. Mujahid and Khalifa, “Particle Swarm Optimization for Deep learning of Convolution Neural Network,” *Sudan Conference on Computer Science and Information Technology (SCCSIT)*, 2017.
[13] G. Vrbančič, I. Fister, and V. Podgorelec, “Swarm Intelligence Approaches for Parameter Setting of Deep Learning Neural Network: Case Study on Phishing Websites Classification,” in *International Conference on Web Intelligence, Mining and Semantics*, ACM, 2018.
[14] P. Wei, “An Optimization Method for Intrusion Detection Classification Model based on Deep Belief Network,” *IEEE*.
[15] O. Almomani, . Pso, Gwo, and Algorithms, “A Feature Selection Model for Network Intrusion Detection System Based on,” *Symmetry 2020*, vol. 12, 1046.
[16] W. Elmasry, A. Akbulut, and A. H. Zaim, “Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic,” *Computer Networks*, 2019.

[17] . J. Arif, W. Malik, F. A. Shahzad, and Khan, "Network intrusion detection using hybrid binary PSOand random forests algorithm," *Networks*, vol. 8, pp. 2646–2660, 2015.
[18] I. Sharafaldin, A. Lashkari, and A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pp. 108–116.
[19] I. Sharafaldin, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Journal of Software Networking*, pp. 177–200.
[20]
[21] N. Kunhare, R. Tiwari, and J. Dhar, "Particle swarm optimization and feature selection for intrusion detection system," *Sadhana (2020) 45:109*.
[22] S. M. H. Bamakan, "A New Intrusion Detection Approach using PSO based Multiple Criteria Linear Programming," *Procedia Computer Science*, vol. 55, pp. 231–237, 2015.
[23] D. A. A. G. Singh, "Enhancing the Performance of Classifier Using Particle SwarmOptimization (PSO) - based Dimensionality Reduction," *International Journal of Energy, Information and Communications*, vol. 6, pp. 19–26, 2015.
[24] H. Nezamabadi-Pour, M. Rostami-Shahrbabaki, and M. M. Farsangi, "Binary Particle Swarm Optimization: challenges and New Solutions"," *The Journal of Computer Society of Iran (CSI) On Computer Science and Engineering (JCSE)*, vol. 6, pp. 21–32, 2008.
[25] S. Mahdavifar and A. A. Ghorbani *Application of deep learning to cybersecurity: A survey*, vol. 347, pp. 149–176, 2019.
[26] M. Mohammadi, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," *IEEE Communications Surveys & Tutorials*, 2018.
[27] R. Ismael, F. Abeer, T. M. Nidaa, and F. H, "Performance Analysis of Flow-Based Attacks Detection on CSE-CIC-IDS2018 Dataset Using Deep Learning," *IJEECS*, vol. 20, no. 3, 2020.