Computers, Materials & Continua
DOI:10.32604/cmc.2022.020769
Article

Tech Science Press

Improved Dragonfly Optimizer for Intrusion Detection Using Deep Clustering CNN-PSO Classifier

K. S. Bhuvaneshwari[1], K. Venkatachalam[2], S. Hubálovský[3,*], P. Trojovský[4] and P. Prabu[5]

[1]Department of Computer Sceince and Engineering, Karpagam College of Engineering, Coimbatore, 641032, India
[2]Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore, 560074, India
[3]Department of Applied Cybernetics, Faculty of Science, University of Hradec Králové, Hradec Králové, 50003, Czech Republic
[4]Department of Mathematics, Faculty of Science, University of Hradec Králové, Hradec Králové, 50003, Czech Republic
[5]Department of Computer Sceince, CHRIST(Deemed to be University), Bangalore, 560074, India
[*]Corresponding Author: S. Hubalovsky. Email: stepan.hubalovsky@uhk.cz
Received: 06 June 2021; Accepted: 07 July 2021

Abstract: With the rapid growth of internet based services and the data generated on these services are attracted by the attackers to intrude the networking services and information. Based on the characteristics of these intruders, many researchers attempted to aim to detect the intrusion with the help of automating process. Since, the large volume of data is generated and transferred through network, the security and performance are remained an issue. IDS (Intrusion Detection System) was developed to detect and prevent the intruders and secure the network systems. The performance and loss are still an issue because of the features space grows while detecting the intruders. In this paper, deep clustering based CNN have been used to detect the intruders with the help of Meta heuristic algorithms for feature selection and preprocessing. The proposed system includes three phases such as preprocessing, feature selection and classification. In the first phase, KDD dataset is preprocessed by using Binning normalization and Eigen-PCA based discretization method. In second phase, feature selection is performed by using Information Gain based Dragonfly Optimizer (IGDFO). Finally, Deep clustering based Convolutional Neural Network (CCNN) classifier optimized with Particle Swarm Optimization (PSO) identifies intrusion attacks efficiently. The clustering loss and network loss can be reduced with the optimization algorithm. We evaluate the proposed IDS model with the NSL-KDD dataset in terms of evaluation metrics. The experimental results show that proposed system achieves better performance compared with the existing system in terms of accuracy, precision, recall, f-measure and false detection rate.

Keywords: Intrusion detection system; binning normalization; deep clustering; convolutional neural network; information gain; dragonfly optimizer

1 Introduction

Over the years, computer networks are the area which effectively used in the applications such as business data processing, education and learning, widespread data acquisition and collaboration and entertainment. The connected devices over the internet for this application are increasing day by day and generate large amount of data for transfer and processing. This unauthorized access also aims to access the systems that will break the confidentiality, availability and integrity which is called as attack/intrusion. To monitor the malicious activity of the system and alert if there is any such attack happened is controlled by the Intrusion detection system (IDS). IDS provides the security from the attackers [1]. IDS can be classified either networkbased or hostbased attacks. Network based attacks are the anomaly based attacks that are detected from the computer systems interconnections and the system can communicate to other via routers and switches, attacks also sent through these ones. Host based attacks are found from the single computer system and also it is easy to prevent from attacks. These are occurred from the external devices connected to the systems. Web based attacks are enabled during connecting to the internet and these systems attack to other system over the mail and downloading.

For this IDS development, data mining and Machine learning techniques are widely used. Network feature selection is declared by selecting the most important features for the entire network without the loss of information. In terms of feature selection and classification on IDS, various ML algorithms have been used such as K nearest neighbor (KNN), Support Vector Machine (SVM), Random forest (RF) and Multi-layer perceptron (MLP) [2]. IDS based deep learning algorithm called convolutional neural network are discussed in paper [3] to handle the imbalanced network traffic. The existing deep learning and machine learning algorithms are still lack in improving the IDS performance. The improved deep learning based algorithm has been used to overcome the issues in the existing systems in terms of improving the performance of the IDS and reduce the generalization error in this paper. The benefits of this contribution are as follows:

- The raw input IDS data set is preprocessed with binning normalization process to remove the missing values and handle the data that are out of the range. The high dimension of raw data is then transformed using the discretization method called Eigen-PCA.
- The transmitted data are then used for feature selection process. This proposed work introduced evolutionary algorithm called improved dragonfly optimizer with information gain (IG-DFO) for selecting the relevant features. For feature selection, the weight and the number of iterations will produce the optimal solution. Dragonfly optimizer with information gain is used here to produce the optimal feature selection.
- Classification with deep clustering convolutional neural network has been used. Compare to traditional CNN, deep clusteringcan reduce the clustering loss and improve accuracy of the classification. The network loss can be reduced with the optimization algorithm called PSO. Thus, the PSO optimized by clustering convolutional neural network (PSO-CCNN) can reduce the generalization error, training time and improved the classification accuracy with minimum noise.
- The proposed feature selection based classification on IDS data has been experimented and compared with existing feature selection and classification algorithms in terms of evaluation metrics.

The paper has been organized as follows: Section 2 describes the review of the literature, Section 3 introduces evolutionary based feature selection and deep learning classification approaches,

Section 4 discusses about the experimented results and Section 5 concludes the paper with future directions.

2 Literature Review

This section describes various literature and recent research related to IDS. In paper [4], various classification algorithms are analyzed with NSL-KDD data set. This can analyzethe protocols with the attacks by the intruders using WEKA tool. They used CFS based dimensionality reduction to improve the classification accuracy. Least square support vector machine (LSSSVM) based on IDS was proposed in paper [5]. It used mutual information based on feature selection method to handle linear and nonlinear correlated features. They used KDD cup 99, NSL-KDD and Kyoto 2006 + datasets. The proposed approach obtained better accuracy and computational cost than other existing algorithms.

The Paper [6] proposed an ensemble method to improve the IDS performance. They used two methods such as boosting and bagging with tree based classifiers. It extracted 35 features and used NSL-KDD dataset for evaluation and concluded that bagging with J48 classifier performs better. Recurrent Neural Network (RNN) [7] is used for IDS with supervised learning classifier. The RNN-IDS was compared with the classification models such as J48, Random forest and SVM in terms of accuracy. For binary classification, the RNN consist of 80 hidden nodes with the learning rate of 0.1. RNN-IDS obtains 83.28% accuracy for binary classification and 80 hidden nodes with the learning rate of 0.5 and accuracy of 81.29% for multi class classification. Feature selection based on Ant Colony Optimization (ACO) [8] is used for classification of features with accuracy.

Filter based on feature selection XGBoost [9] is used for detecting network attacks by using machine learning classification techniques with 91% accuracy. Feature selection approach [10] uses correlation techniques as naive bays (NB), Random Forest (RF), J48, and ZeroR. Study of various IDS techniques [11] for identifying attacks using classification algorithms. IDS in the IOT network is very important issue and studiesare conducted [12,13]. This paper gives detail review about the existing IoT based on secure communication on IDS. It also discussed about the IoT based on classifications and discussed about the future research challenges. Best feature selection strategy is implemented [14] using genetic algorithm with logistic regression techniques. Swarm intelligent technique for feature extraction [15] uses Pigeon Inspired optimizer with sigmoid function. Feature extraction using Particle Swarm Optimization (PSO), Firefly Optimization (FO), Grey Wolf Optimization (GO) and Genetic Algorithm (GA) [16] are discussed with accuracy rate. Deep learning clustering techniques for IDS system [17,18] uses contextual deep clustering with Euclidean distance based deep analysis. It can find out clusters accurately for IDS. Convolutional neural network model used for detecting the DoS attacks [19] in big networks are discussed for identifying optimal solution.

3 Proposed Deep Clustering PSO Based IDS Model

The proposed approach includes three phases such as Preprocessing, Feature selection and Classification. The overview of the approach is shown in Fig. 1. Initially the network data set is divided into two datasets-training dataset and testing dataset in the ratio of 6:4. In all classification algorithm approaches, preprocessing plays on vital role on improving the accuracy. From the raw data, the irrelevant and missing data are handled with this preprocessing process.

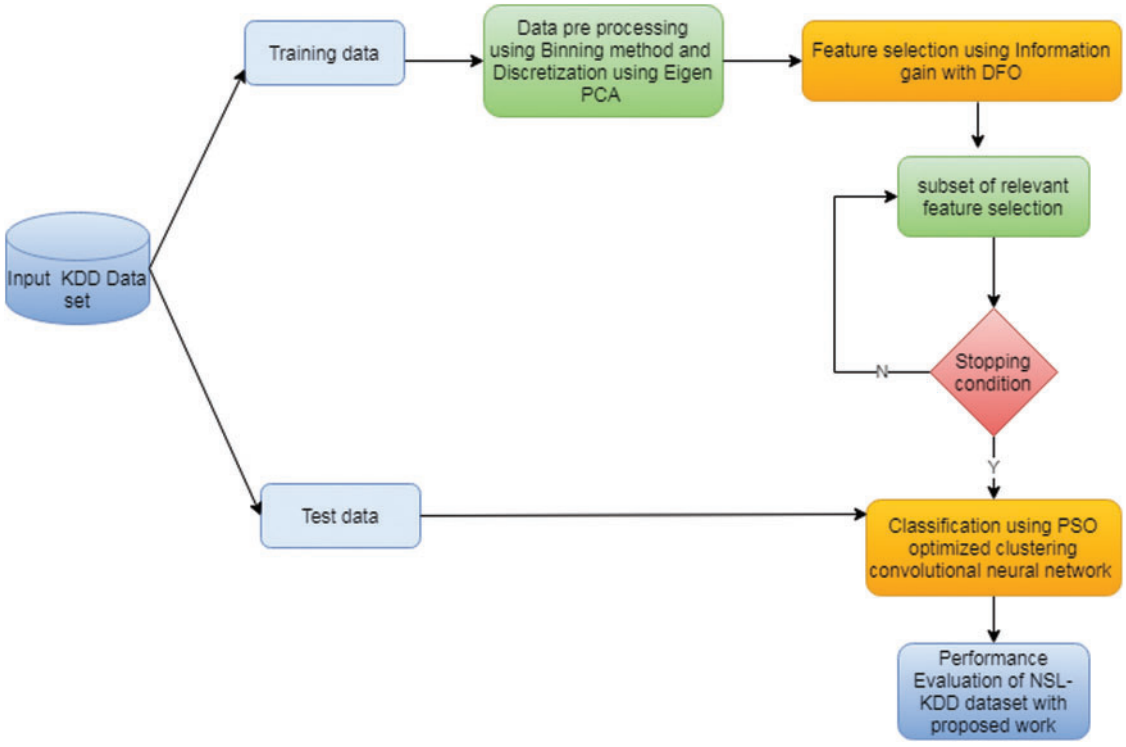

Figure 1: Overview of proposed IDS

In this proposed work, preprocessing stage includes normalization and discretization. The phases are:

(i) Preprocessing using Binning method to handle the missing values. Binning method is used to handle the missing values for both numerical and categorical data. It is used to enhance the model more robust and prevent from overfitting.
(ii) Dimensionality reduction using Eigen PCA. This method is used to reduce the n dimension data into m where $m < n$ based on the non-linear local relationship among the data points.
(iii) Feature selection using proposed information gain with Dragonfly optimizer (IG-DFO). This method is used to select the reduced number of feature set for further processing.
(iv) Classification using proposed PSO optimized-Clustering Convolutional neural network (PSO-CCNN). Among the various machine learning and deep learning techniques, CNN proven to be the generative model that contains multiple layers of latent and stochastic variables. These deep clustering and evolutionary algorithms analyzed with the intrusion detection dataset to prove the efficiency of the proposed work in terms of performance measures.

3.1 Preprocessing Using Binning Normalization

The raw input data set are complex to process due to noise and missing values. It is also difficult to process the dataset with whole features of numeric and non-numeric data. Hence, the raw input data are to be preprocessed before proceeding for further analysis improvement. Initially, the symbols in the data are replaced with the unique numeric value using the mapping function available in Python library which is mathematically represented using the logarithmic equation [20]

of the dataset X which is represented in Eq. (1)

$$X_{normalized} = \log(X_i + 1) \tag{1}$$

The data features having missing values may also have some useful information. Avoiding the missing values data may affect the performance of the classification. Those missing values are handled using the binning method. In this method, the data are smoothened and the missing values are handled. The bin have equal width with the range of each bin are defined as Eq. (2)

$$[\min + w], [\min + 2w] \ldots [\min + n\ w] \tag{2}$$

where

$$w = \frac{(\max - \min)}{\text{no of bins}}. \tag{3}$$

In general, the larger the width, the greater the data get smoothened. In this work, the Binning method combined with discretization to transform the data using the dimensionality reduction technique called Eigen-PCA which is described in the Section 3.2.

3.2 Dimensionality Reduction Using Eigen- PCA

To make the database as an elegant one for processing, dimensionality reduction is important step to reduce the dimension of the input data into low dimensional space. Here, Eigen-PCA has been used to reduce the dimension of feature space. Principal Component Analysis (PCA) has been used to reduce the high dimensional space of the data into low dimension space of features. PCA is used here for calculationof the eigenvector of covariance matrix. The high dimensional data space is transformed into low dimensional space with the eigenvectors with larger eigenvalues. The Nrepresents the number of features in the data set and the set is represented as $[P_1, P_2, \ldots P_N]$. The feature set is represented in Eq. (4)

$$P = \frac{1}{N} \sum_{i=1}^{N} P_i. \tag{4}$$

The covariance matrix C is calculated based on Eq. (5)

$$C = \frac{1}{N} \sum_{i=1}^{N} (P - P_i)(P - P_i)^T. \tag{5}$$

The eigenvalues and eigenvectors are calculated based on Eq. (6)

$$EV = \lambda V, \tag{6}$$

where V are eigen vectors λ are eigen value associated with matrix C.

All the input image setsare projected into the eigen-subspace and they are represented by Eq. (7)

$$y_j^i = w^T(p_i), \tag{7}$$

where $i = 1, 2, 3, \ldots, N$ and y_j^i are projection of p called as the principal components.

The input dataset is the combination of number of principal components. The final reduced dataset dimension is represented as in Eq. (8)

$$F_i^{EPCA} = \{F_1^{EPCA}, F_2^{EPCA}, \ldots, F_n^{EPCA}\}. \tag{8}$$

Algorithm 1: Preprocessing Using Binning- Eigen PCA

Input: Raw dataset X with features (f), n represents number of data
Output: Normalized and reduced dataset.
Step 1: for i = 1 to n // Normalization process
Step 2: if (f_i ==symbol) then
Step 3: map the symbols into numeric using Python library
Step 4: normalization using Eq. (1)
Step 5: Smoothening: Handle missing values using the Binning method as Eq. (2)
Step 6: else
Step 7: Normalization using Eq. (1)
Step 8: Smoothening: Handle missing values using the Binning method as Eq. (2)
Step 7: End if
Step 8: End for
Step 9: Dimensionality reduction using Eigen PCA using Eqs. (4)–(8).

3.3 Optimized Feature Selection Using Enhanced IG-DFO

This section will select the optimal features that are relevant to classify the intruders with improvement. The irrelevant features in the dataset will lead to slow the training and testing process for classification. Reducing the irrelevant features will reduce the complexity of the system, speed up the computation process and improve the overall performance of the Intrusion Detection Systems. The information Gain [21] is formed from the information theory [22] is used to find the features that are relevant to the class. The Dataset X is divided into n number of classes and each feature f_i which have maximum number of non-zero values are selected and the uncertainty of the value X is also called as entropy based on information theory and it is represented as E(X)which is calculated using Eq. (9),

$$E(X) = -\sum_{x \in X} P(x)\log_2(x), \tag{9}$$

where P(x) is probability of x.

The conditional entropy of two random features X and Y are calculated using Eq. (10)

$$CE(X|Y) = -\sum_{x \in X} P(x) \sum_{y \in Y} P(x|y)\log_2(x|y). \tag{10}$$

The enhanced Information Gain is calculated from Eqs. (9) and (10)

$$EIG(X|Y) = E(X) - CE(X|Y). \tag{11}$$

Hence, the feature Y have the stronger correlation to feature X than other features f, if $EIG(X|Y) > EIG(f|Y)$. Those correlated features are then optimized using Dragonfly

Optimizer (DFO) algorithm to select the optimal features set. Among the various meta heuristic algorithms includes Swarm intelligence [23] approaches such as Particle swarm optimization (PSO) [24], Ant colony optimization (ACO) [25], Cat swarm optimization (CSO) [26], Grey wolf optimizer (GWO) [27], Artificial bee colony (ABC) [28], Fitness dependent optimizer (FDO) [29], Donkey and Smuggler Optimization (DSO) [30], Firefly algorithm (FA) [31], Cuckoo search (CS) algorithm [32], Dragonfly optimizer (DFO) [33] is the recent swarm based method that is successfully implemented.

DFO is mimicking the behavior of the dragonfly for the reason of migration or hunting. The swarming behavior may be static or dynamic. In static swarm, small group of dragonflies are moved to hunt other swarms in small area with the local movement of abrupt changes. In dynamic swarm, larger volume of dragonflies is moved towards in one direction for long distance as a group. The dragonflies behavior of static and dynamic swarm is shown in Fig. 2. Artificial dragon flies movement direction is based on five weights such as:

- separation weight (s);
- alignment weight (a);
- cohesion weight (c);
- food factor (f);
- enemy factor (e);
- the inertia weight (w).

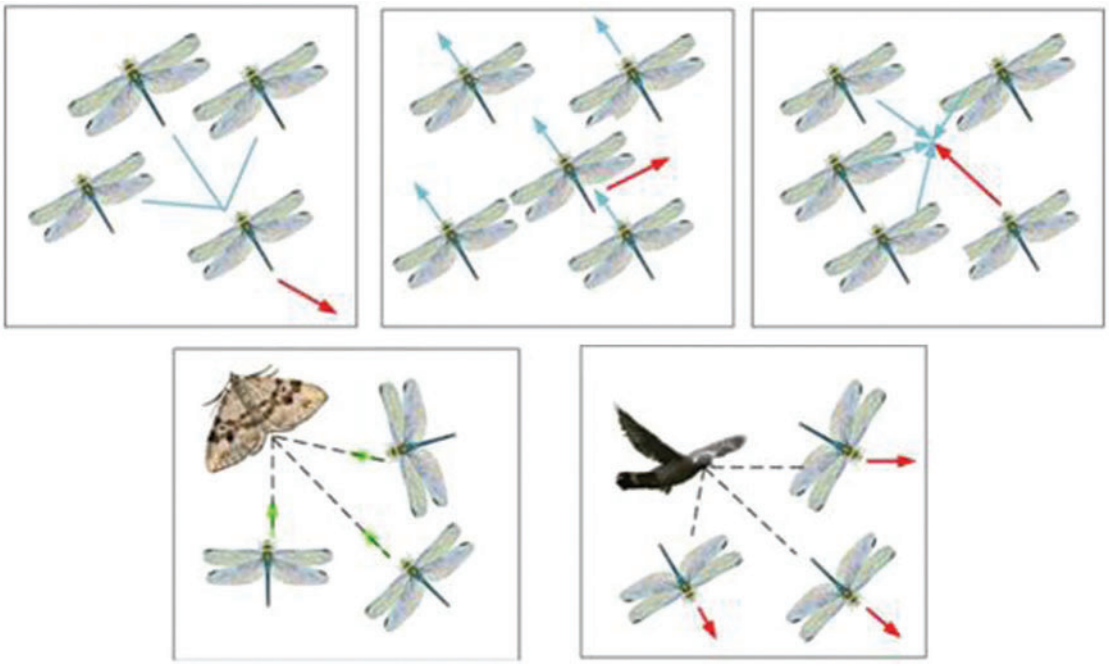

Figure 2: Dragonfly characteristics attraction to food distraction from enemy

To optimize the process of exploration and exploitation, the five weights have to be tuned. The weight factors such as separation, alignment, cohesion, attraction to food and distraction from enemy of DFO has been mathematically explored and computed.

Algorithm 2: IG-DFO

Input: Population of dragonflies and step vector is X_i , where $i = (1, 2, \ldots, n)$
Step 1: while max iteration (tmax)
Step 2: compute enhanced Information Gain using Eq. (5).
Step 3: objective values of dragonflies are calculated
Step 3: update the source of the food and enemy
Step 4: calculate five weight factors such as S, A, C, F, and E using Eqs. (12)–(16)
Step 5: update neighbor radius
Step 6: if dragonfly has neighbor
Step 7: update velocity vector using Eq. (17)
Step 8: update the inertia weight using Eq. (18)
Step 9: update position vector using Eq. (19)
Step 10: else
Step 11: update the position vector using Lévy flight
Step 12: end if
Step 13: new position of the dragonfly are adjusted based on the variable boundaries
Step 14: end while
Output: return optimal features set

The work process of proposed IIW-DFO has been shown in Fig. 3. Until the maximum iterations satisfied the five weight vectors, velocity and position vectors are updated with the improved inertia weight. If the dragonflies have neighbors then the position also are updated. Now the optimized features are given as input to the deep clustering algorithm called PSO optimized clustering convolutional neural network for identification of the intruders.

3.4 Classification Using Proposed PSO Optimized-Clustering CNN

Deep clustering adopts deep neural network to learn the clustering method to reduce the loss. To train the neural network, Clustering deep neural network algorithms are used in the clustering loss. The network can be either fully connected network (FCN) or convolutional neural network (CNN) or deep belief network (DBN). The clustering related loss functions are categorized into principal clustering loss and auxiliary clustering loss. The principal clustering loss contains cluster centroid and sample cluster assignment. This includes the losses such as k means loss, cluster assignment hardening loss, agglomerative clustering loss, and nonparametric maximum margin clustering and so on. Auxiliary clustering loss runs the clustering method after the neural network training with deep clustering loss such as locality-preserving loss, group sparsity loss and subspace clustering loss. The deep clustering loss function is formulated as in Eq. (12)

$$L = \lambda L_n + (1 - \lambda) L_c, \tag{12}$$

where L_n is thenetwork loss, L_c is the clustering loss and $\lambda \in [0, 1]$ is the parameter to balance network and clustering loss. The network loss can be used to learn the relevant features to avoid the confused solutions. The clustering loss groups the feature points to form the clusters. In this proposed work, the clustering loss can be used to train the Convolutional neural network. The loss function of deep clustering with CNN is described by Eq. (13)

$$L = L_c. \tag{13}$$

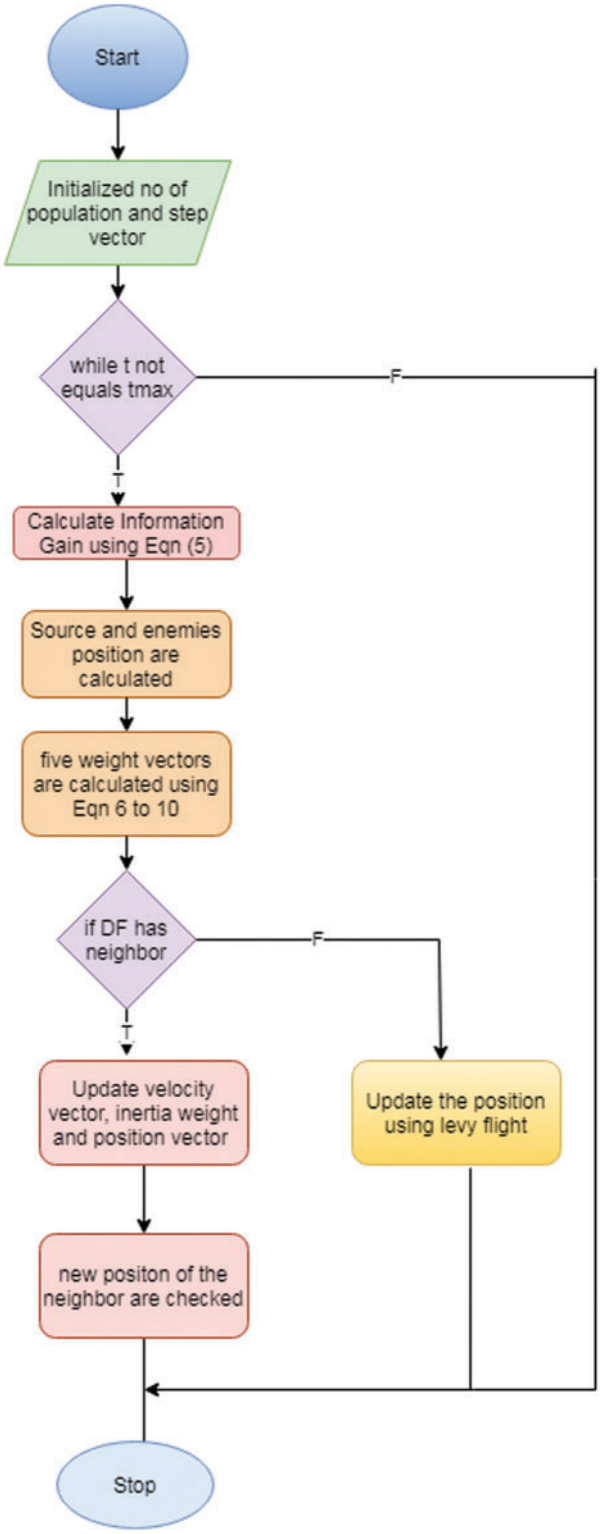

Figure 3: Work flow of proposed IG-DFO

Clustering convolutional neural network of deep clustering is a type of supervised pre-trained network. This uses the CNN framework to learn the clustering and reduce the loss. It picks ' k' samples initially and extract k features as initial cluster Centroid.

Algorithm 3: PSO-CCNN

Input: Transformed dataset X^t, k samples, centroid $c_i \in C$, Swarm Size M, Maximum Iteration tmax, Number of features/Particle dimension D, cognitive learning and social learning factor such as c_l and s_l, random values in the range [0, 1] as r_1 and r_2, internal coefficient w, initialize iteration $t = 1$

Output: detected intruders

Step 1: for particle $i = 1$ to M do
Step 2: random initialization of position X_{id} with values and velocity vector V_{id}
Step 3: end For
Step 4: while $t \neq tmax$
Step 5: for each particle i do
Step 6: if $(EIG(x_i|y_i) > Pbest_i)$ then
Step 8: set $Pbest_i = EIG(x_i|y_i)$
Step 9: end If
Step 10: end For
Step 11: for $t = 1$ to tmax
Step 12: M = feature selection using algorithm 2
Step 13: for $m \in M$ do
Step 14: y,λ(m), d=N(m, c) // label y, distance and learning rate
Step 15: end for
Step 16: M1, y1=k-NN(d,M, c, k)
Step 17: finetune(CCNN, M1, y1, k)
Step 18: set Gbest=Best previous particle fitness value
Step 19: for particle $i = 0$ to M do
Step 20: for dimension D=0 to D do
Step 21: compute Eqs. (14) and (15)

Step 22: activation using $\text{sigmoid}(v_i^{t+1}) = \dfrac{1}{1 + e^{-v_i^{t+1}}}$ (14)

Step 23: $X_i^{t+1} = X_i^t + v_i^{t+1}$ (15)

Step 24: End for
Step 25: End for
Step 26: $t = t + 1$
Step 27: End While

The proposed preprocessing, feature selection and classification algorithms are efficient in terms of accuracy and reducing the error. Accuracy of IDS has been obtained with the pre-processing followed by relevant feature selection using information gain based DFO. This meta heuristic algorithm improves the classification accuracy. The proposed deep clustering method with PSO reduces the network loss error. It improves the QoS of the proposed IDS. Hence, the

proposed preprocessing, feature selection and classification algorithms are efficient and accurate on predicting the intruders around the network communications.

4 Results and Discussions

This section describes about the experimented results and discussions of the proposed feature selection and classification on IDS. This analysis used binary classification on NSL-KDD dataset and it was implemented using Python deep learning library called Keras.

4.1 Dataset Description

To analyze the performance of the proposed IIW-DFO with IDBCNN based IDS system, the benchmark network traffic dataset called NSL-KDD is used. It is proven to be the best dataset for testing the IDS [4]. There are 41 features that are divided as basic, content based and time based attributes. The NSL-KDD dataset consists of three types of features such as Nominal (3 features), Binary (6 features) and Numeric (32 features). Training set consists of 22 attacks and 16 attacks are considered as testing set. The attacks are categorized as:

1) Denial of Service attacks (DoS);
2) Probe Attacks (PA);
3) Remote to Local attacks (R2L);
4) User to Root attacks (U2R).

The IDS attacks with detailed explanation and the training, testing data are shown in Tab. 1. with the binary class.

Table 1: NSL-KDD attacks, training data and testing data

Class	Attack	NSL-KDD dataset	
		Training data	Testing data
Normal	67343	9710	
Abnormal	DoS	45927	7458
	Probe (6 Attacks)	11656	2422
	R2L (16 attacks)	995	2887
	U2R (7 attacks)	52	67

4.2 Optimized Features Selection Using IG-DFO

The NSL-KDD dataset consists of 41 features and 1 class label which is represented with example in Tab. 2. The proposed optimized feature selection approach called IG-DFO select seven relevant optimal features from this feature set which will improve the classification accuracy. This feature selection approach is compared to other existing feature selection approaches which are represented in Tab. 3. And the selected feature names of the proposed approach are represented in Tab. 4.

4.3 Evaluation Using Performance Metrics

The proposed IGDFO-PSOCCNN-IDS system is compared with the existing approaches to analyze the performance using the performance metrics such as Accuracy (ACC), False positive rate (FPR), False negative rate (FNR), Sensitivity/True positive rate (SN), Specificity/True neg-

ative rate (SP) and recall/Attack Detection rate (ADR). The evaluation metrics equations are represented by Eqs. (16)–(18)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{16}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{17}$$

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}, \tag{18}$$

Table 2: NSL-KDD dataset features/attributes

Feature number	Feature name	Feature number	Feature name
1	Duration	22	Is_guest_login
2	Protocol_type	23	Count
3	Service	24	Srv_count
4	Flag	25	Serror_rate
5	Src_bytes	26	Srv_serror_rate
6	Dst_bytes	27	Rerror_rate
7	Land	28	Srv_rerror_rate
8	Wrong_fragment	29	Same_srv_rate
9	Urgent	30	Diff_srv_rate
10	Hot	31	Srv_dif_host_rate

Table 3: Feature selection of different approaches

FS approaches	No of features	Selected features
All features	41	f1, f2, f3, f4, f5, f6.f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17, f18, f19, f20, f21, f22, f23, f24, f25, f26, f27, f28, f29, f30, f31, f32, f33, f34, f35, f36, f37, f38, f39, f40, f41
FMIFS [23]	18	f5, f30, f6, f3, f4, f29, f12, f33, f26, f37, f39, f34, f25, f38, f23, f35, f36, f28
FLCFS [23]	22	f29, f12, f33, f39, f4, f23, f34, f25, f26, f38, f8, f35, f19, f32, f18, f3, f6, f40, f30, f5, f27, f22
SMOTE-ENN [4]	6	f3, f5, f30, f4, f6, f29
Proposed IGDFO-PSOCCNN	7	f3, f5, f30, f4, f6, f29, f35

Table 4: Selected feature names by IIW-DFO

Selected feature	Feature name
f3	Service
f5	Src_bytes
f29	Diff_srv_rate
f4	Flag
f6	Dst_bytes
f29	Same_srv_rate
f35	Dst_host_diff_srv_rate

4.4 Performance Evaluation Based on NSL Feature Selection Approaches

The proposed IDS performance based on the original and selected input features are represented in graph. The original 41 features are normalized using feature selection IGDFO. The selected 7 features are represented. The graph proves the importance of preprocessing method called two step normalization with binning process to avoid the network traffic data. This feature selection process also overcomes the over fitting issue and enhance the overall performance of the IDS to improve the accuracy of classification, decreases the error rate and detection time and also minimize the computational complexity.

The proposed feature selection performance is compared with the existing FS on IDS such as standard DFO, FMIFS, FLCFS, and SMOTE-ENN [3]. The proposed IG-DFO FS obtains higher accuracy with lower complexity analysis than other existing contemporary techniques which indicate that these selected features improves the classification accuracy on protect the computer network from intruders. The graphical illustration from Fig. 4 also clearly shows that the increase percentage of accuracy, sensitivity, specificity and attack detection rateare shown in proposed FS approach more than other techniques.

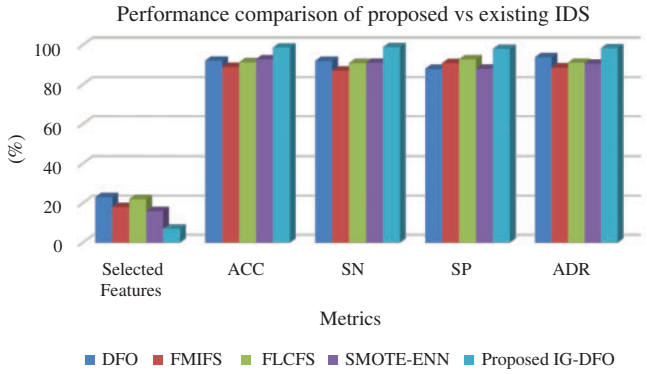

Figure 4: Performance comparison of existing *vs.* proposed approaches

4.5 Performance Evaluation of Proposed IGDFO-PSOCCNN IDS with Existing IDS Systems

In order to prove the deep neural network based IDS systems, our proposed convolutional deep neural network based on IDS is compared with the existing IDS systems such as DMNB, DBN-SVM, PSOM [1], and SMOTE-ENN [3]. The proposed Information Gain Dragonfly opti-

mizer with improved deep clustering convolutional neural network with PSO based on IDS obtains higher accuracy and lowerFPR rate than other existing IDS approaches including our previous work. The proposed system obtains 98.71% of accuracy on detecting the intruders and 0.12 of False Positive Rate. The IGDFO-PSOCCNN IDS obtains this high accuracy due to the optimization process. The graphical representations of these results are shown in Figs. 5 and 6 for better understanding.

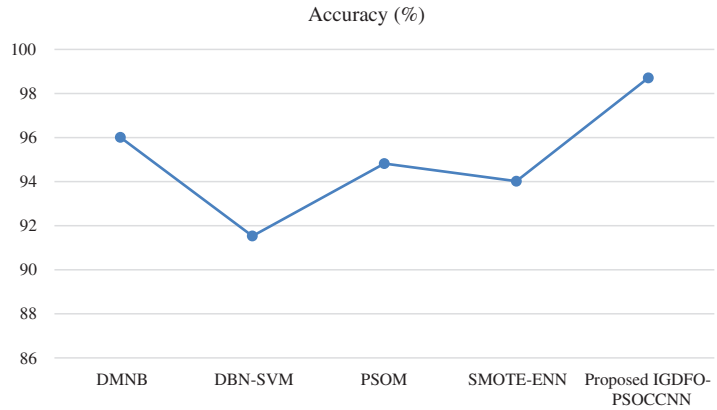

Figure 5: Accuracy comparison of various IDS

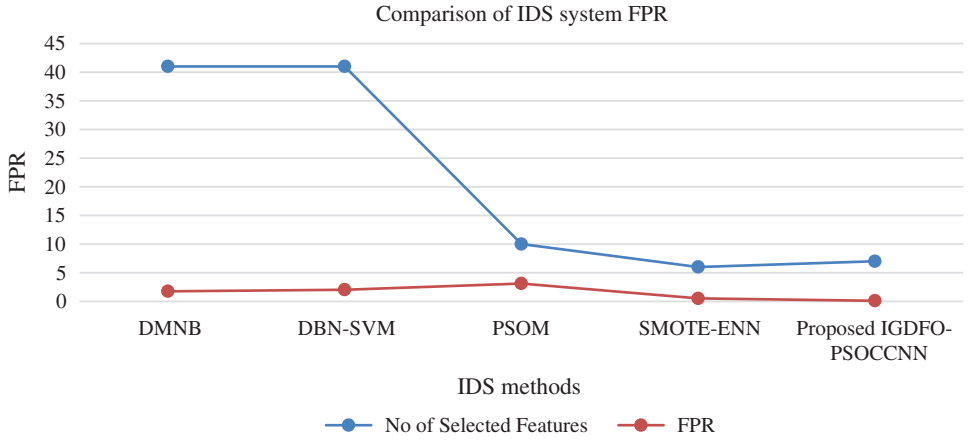

Figure 6: FPR comparison of various IDS

Hence, the various experimented results of IDS process show that our proposed IG-DFO-PSOCCNN have higher classification accuracy, minimum error, and reduce the computational complexity. Even though our previous work performs better on detecting the intruders, this work with optimization process also increases the classification accuracy and reduces the loss than our previous work. The clustering based convolutional neural network withPSO can reduce the loss of the network efficiently. Hence, to detect intruders with high accuracy and low error, the preprocessing based IGDFO feature selection with deep clustering method (PSO-CCNN) is better than other existing algorithms in terms efficiency and accurate classification.

5 Conclusions

An information gain dragonfly optimizer based feature selection and improved Deep bagging convolutional neural network has been proposed in this paper for IDS. Due to the large number of features and large amount of data in the data set, an improved proposed classification technology based on evolutionary and deep learning algorithms is proposed to improve the classification accuracy and prediction of the intruders in the network. Optimization based feature selection algorithm proves that it is the best for finding the relevant features. Deep clustering based CNN with PSO optimization improves the accuracy and stability of the system. Main advantage of using this technique is, swarm intelligence with evolutionary strategy give accurate optimization result. The proposed system is implemented using NSL KDD dataset. The efficiency of the algorithm is proved with the comparative analysis of existing contemporary algorithms in terms of feature selection and classification. The experimental results show that the proposed evolutionary based deep clustering algorithm outperforms in terms of Accuracy (98.71%) and False Positive Rate (0.12). Deep clustering with PSO optimization can reduce the clustering and network loss. Hence, the proposed IDS system will reduce the generalization error, training time, reduced noise and improve the classification accuracy. In the future, the proposed algorithm will be experimented with small number of datasets. The future aim is also to increase the detection rate of the attacks in the NSL KDD dataset.

Funding Statement: The third and fourth authors were supported by the Project of Specific Research PrF UHK No. 2101/2021 and Long-term development plan of UHK, University of Hradec Králové, Czech Republic.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. M. Sakr, M. A. Tawfeeq and A. B. El-Sisi, "Filter versus wrapper feature selection for network intrusion detection system," in *Proc. ICICIS*, Cairo, Egypt, 2019.

[2] S. M. Kasongo and Y. Sun, "A deep learning method with filter based feature engineering for wireless intrusion detection system," *IEEE Access*, vol. 7, pp. 38597–38607, 2019.

[3] X. Zhang, J. Ran and J. Mi, "An intrusion detection system based on convolutional neural network for imbalanced network raffic, in *Proc. ICCSNT*, Dalian, China, pp. 456–460, 2019.

[4] L. Dhanabal and S. P. Shantharadah, "A study on NSLKDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[5] M. A. Ambusaidi, X. He, P. Nanda and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[6] T. Pham, E. Foo, S. Suriadi and H. Jeffrey, "Improving performance of intrusion detection system using ensemble methods and feature selection," *Australasian Computer Science Week Multi-Conference*, vol. 2, pp. 1–6, 2018.

[7] C. Yin, Y. Zhu, J. Fei and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[8] T. Mehmod and H. B. M. Rais, "Ant colony optimization and feature selection for intrusion detection," *Advances in Machine Learning and Signal Processing*, vol. 387, pp. 305–312, 2016.

[9] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW NB15 dataset," *Journal of Big Data*, vol. 7, 2020.

[10] M. Hammad, W. E. Medany and Y. Ismail, "Intrusion detection system using feature selection with clustering and classification machine learning algorithms on the UNSW NB15 dataset," in *Proc. ICIIICT*, Sakheer, Bahrain, pp. 1–6, 2020.

[11] N. Misiko Jacob and M. Yusuf Wanjala, "A review of intrusion detection systems,", *Global Journal of Computer Science and Technology: C Software and Data Engineering*, vol. 17, no. 3, 2017.

[12] A. Khraisa and A. Alazab, "A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy validation strategy attacks public datasets and challenges, *Cybersecurity*, vol. 4, no. 18, 2021.

[13] A. Chauhan, R. Singh and P. Jain, "A literature review: Intrusion detection systems in internet of things," in *Proc. CMVIT*, Sanya, China, 2020.

[14] C. Khammassi and S. A. Krichen, "A GA LR wrapper approach for feature selection in network intrusion detection," *Computer Security*, pp. 255–277, 2017.

[15] H. Alazzam, A. Sharieh and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert System with Application Expert*, vol. 148, 2020.

[16] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 6, no. 12, pp. 1–18, 2020.

[17] B. Sudhakar, V. B. Narsimha and G. Narsimha, "A contextual deep clustering based intrusion detection method for cloud," *International Journal of Advanced Science and Technology*, vol. 28, no. 18, pp. 349–359, 2019.

[18] A. Genevay, G. Dulac Arnold and V. Jean-Philippe, "Differentiable deep clustering with cluster size constraints," *Cornell University, Machine Learning*, vol. 3, pp. 1–14, 2019.

[19] J. Kim, H. Kim, M. Shim and E. Choi, "CNN-Based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, pp. 1–21, 2020.

[20] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. ICNN*, Piscataway, Newyork, vol. 4, pp. 1942–1948, 1995.

[21] Z. Gao, Y. Xu, F. Meng, F. Qi and Z. Lin, "Improved information gain based feature selection for text categorization," in *Proc. ICWC*, Chennai, India, pp. 1–5, 2011.

[22] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computer. Communication Review*, vol. 5, no. 1, pp. 3–55, 2001.

[23] X. S. Yang and X. He, "Swarm intelligence and evolutionary computation: overview and analysis," in *Proc. Studies in Computational Intelligence*, Cham, Switzerland, pp. 1–23, 2014.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN*, Perth, Australia, 1995.

[25] M. Dorigo and D. Carro, "Ant colony optimization: A new meta-heuristic," in *Proc. CEC99*, Wisconsin, United States, 2002.

[26] S. C. Chu, P. W. Tsai and J. S. Pan, "Cat swarm optimization," in *Lecture Notes in Computer Science*, Springer, Berlin, pp. 854–858, 2006.

[27] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[28] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[29] J. M. Abdullah and T. Ahmed, "Fitness dependent optimizer: Inspired by the bee swarming reproductive process," *IEEE Access*, vol. 7, pp. 43473–43486, 2019.

[30] A. S. Shamsaldin, T. A. Rashid, R. A. Al-Rashid Agha, N. K. Al.Salihi and M. Mohammadi, "Donkey and smuggler optimization algorithm: A collaborative working approach to path finding," *Journal of Computational Design and Engineering*, vol. 6, no. 4, pp. 562–583, 2019.

[31] X. Yang, "Firefly algorithms for multimodal optimization," *Stochastic Algorithms: Foundations and Applications*, pp. 169–178, 2009.
[32] X. Yang and S. Deb, "Cuckoo search via levy flights," in *Proc. NaBIC*, Coimbatore, India, 2009.
[33] R. Storn and K. Price, "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.