Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Evaluated bird swarm optimization based on deep belief network (EBSO-DBN) classification technique for IOT network intrusion detection

A. Biju & S. Wilfred Franklin

To cite this article: A. Biju & S. Wilfred Franklin (2024) Evaluated bird swarm optimization based on deep belief network (EBSO-DBN) classification technique for IOT network intrusion detection, Automatika, 65:1, 108-116, DOI: 10.1080/00051144.2023.2269646

To link to this article: https://doi.org/10.1080/00051144.2023.2269646

Published online: 29 Nov 2023.

AUTOMATIKA
2024, VOL. 65, NO. 1, 108–116
https://doi.org/10.1080/00051144.2023.2269646

Evaluated bird swarm optimization based on deep belief network (EBSO-DBN) classification technique for IOT network intrusion detection

A. Biju[a] and S. Wilfred Franklin[b]

[a]Department of Computer Science and Engineering, Maria College of Engineering and Technology, Attoor, India; [b]Department of Electronics and Communication Engineering, C.S.I Institute of Technology, Thovalai, India

ABSTRACT
Because of the recent development of various intrusion detection systems (IDS), which defend computer networks from security as well as privacy threats. The confidentiality, integrity and also availability of data may be compromised in the case that IDS prevention efforts fail. The amount of private, delicate and crucial data travelling over the worldwide network has expanded tremendously as a result of the recent development of Internet of Things (IoT) devices. Developing a better edge-based feature selection strategy, a deep learning technique for identifying and blocking malicious traffic, is the goal of intrusion detection. The classification method Evaluated Bird Swarm Optimization based Deep Belief Network (EBSO-DBN) has shown to be the most successful in this study. A variation of performance criteria have been used to critically assess deep learning techniques for IDS (accuracy, precision, recall, f-1 score, false alarm rate and detection rate). To ascertain the optimal performance of IDS models, this study focuses on building an ensemble classifier utilizing the suggested EBSO-DBN classification algorithm with 98.7% of accuracy, 99.4% of precision and 98.8% of recall.

ARTICLE HISTORY
Received 22 August 2023
Accepted 16 September 2023

KEYWORDS
Intrusion detection systems (IDS); deep learning; internet of things (IoT); malicious traffic; evaluated bird swarm optimization based deep belief network (EBSO-DBN)

1. Introduction

Among the most common types of network security technologies utilized to protect the network is indeed an IDS. The Iot has clearly progressed over the past few years and will soon play a crucial role in our daily lives. The risks to this sensitive data increase along with the volume of transactions in a network; as a result, an IoT network must have a smart mechanism to detect any illegal upgrades and avert such hazards. This system detects but instead presents intrusion possibilities based on a few attributes obtained through classification techniques. It is put to the test if an intrusion detection system can spot malicious activities in IoT networks.

Real guard [1–3], a DNN-based IDS, had been implemented at an IoT network gateway to detect different intrusions, such as LR, NB and also DT with hard voting. The robustness of another P-ResNet model is assured through its capacity and classifies attack events inside multiple heterogeneous IoT networks [4]. To reliably identify various information security, researcher developed the IMIDS attack data generator, which is powered by either a CNN-based IDS or a generative neural network [5]. This article suggested a particular IDS, termed "Edge IDS", for IoT devices by utilizing [6] the generative adversarial network (Skip-GAN anomaly). In an effort to identify as finest effective model on an ensemble classifier to use to identify rather meticulous attacks utilizing deep learning but also machine learning technologies, it is suggested [7] that the proposed IEM be used in conjunction towards the effective Ranking Best Selection Method (RBSM). Several security [8,9] and integrity aspects, including denial of service (DoS), data type probing, scanning, spying, malicious operation, intrusion detection, brute force, web attacks, and incorrect configuration towards thoroughly analysed but instead found by a comprehensive prediction model using sparse evolutionary training (SET). Based on a newly established MH methodology named Reptile Search Algorithm (RSA) [10], which is modelled after crocodile hunting techniques, a novel feature selection procedure has been provided.

When a type of attack is under represented in the dataset, typical in IDS datasets, the resulting model performs poorly on the detection of attack variants that belong to the infrequent attack type. Several attempts have been proposed to mitigate the issues caused by imbalanced IDS datasets, focusing mainly on the data sampling and class balancing techniques.

However, the evaluation metrics were only limited to accuracy, with no discussion around recall and precision. The proven ability of along with the lack of an in-depth analysis of DBNs and the limited work on tackling imbalanced cyber-security datasets.

CONTACT A. Biju bijua.cse@yahoo.com Department of Computer Science and Engineering, Maria College of Engineering and Technology, Attoor, Kanyakumari, India

The following is a list of this paper's main contributions:

(1) Given an NSL-KDD input dataset for pre-process the date, which can reduce or eliminate the noise from the input data
(2) Classifying the intrusion in a specific dataset using the Enhanced Bird Swarm Optimization based Deep Belief Network (EBSO-DBN) classifier algorithm for malicious attack, i.e. ID in IoT Environment
(3) Performance assessment of the new model in terms of execution time, precision, accuracy, detection rate and false alarm rate
(4) To combine deep learning as well as the EBSO-DBN classification technique to develop a high-performance IDS.

Following is the arrangement of a remaining portions of the paper. In Section 3, the suggested method for data collection, pre-processing, classification in a deep learning system is presented. In Section 4, the experimental findings and analyses are given. A conclusion and research proposals for the future are provided in Section 5.

2. Literature review

To effectively detect vulnerabilities in IoT contexts, Muthanna et al. [11] highly suggested a robust, SDN-enabled hybrid architecture using the cuLSTMGRU (cuda Long Short Term Memory Gated Recurrent Unit). A novel red deer-bird swarm approach (RD-BSA) was created in this study by Balashunmugaraja et al. [12] to improve convergence while reducing the use of control components in solution development. To reduce temporal complexity, Onah et al. [13] introduced a Genetic Algorithm Wrapper-Based feature selection and Nave Bayes for Anomaly Detection Model (GANBADM) in a Fog Environment (NSL-KDD). Task scheduling using the improved bird swarm algorithm (IBSA) approach has suggested by Fan et al. [14] as a solution to the problems with improved work through the cloud computing environment, scheduling with high components energy usage. Mokbal et al. [15] claim that an accurate strategy towards detecting malicious is created utilized an embedded feature selection method as well as Extreme Gradient Boosting (XGBoost). Additionally, the most recent Canadian Institute for Cybersecurity's real-world intrusion dataset is used to derive the most efficient uniform feature subset for all attacks.

Because reinforcement learning may enhance the capacity of the learning process to make decisions, Tharewal et al. [16] have employed it in place of supervised and unsupervised learning. By reducing the dimension of data characteristics and enhancing the efficiency of anomaly identification, the method put forward by Bacha et al. [17] is utilized. The Otoum et al. [18] module's suggested combination towards spider monkey optimization (SMO) approach with the stacked-deep polynomial network (SDPN) results on best possible detection and identification. The best features in the data sets are chosen by SMO, and the data are classified as normal or anomalous by SDPN. Three attack detection modules with three different classifiers make up the proposed system. The Hybrid Detection Module (HDM) employs the Meta-AdaboostM1 method, the Anomaly Detection Module (ADM) uses the Naive-based classifier, and the Signature Detection Module (SDM) uses the C4.5 classifier, according to Singh et al. [19] proposed approach (HDM). By identifying novel, unidentified assaults with a low FAR, the created EHIDF can solve the current detection issues. Feature Extraction (FE) technique that uses a Sea Turtle Foraging Algorithm with Explorated Particle Swarm Optimization (PSO) as its core and offered by Jeyaselvi et al. [20] efficient computing speed and accuracy (EXPSO-STFA).

Ogwara et al. [21] use a novel hybrid ensemble feature selection (FS) technique that has been proposed. Three different types of FS algorithms are included in the ensemble (filter, wrapper and embedded algorithms). A probable hybrid feature selection (HFS) method involving an ensemble approach recently proposed by Jaw et al. [22]. To choose correlated subsets of attributes effectively, combine the advantages of genetic searching, CfsSubsetEval, and even a rule-based engine. The innovative neighbourhood search-based particle swarm optimization (NSBPSO) methodology became initiated by Baniasadi et al. [23], who often improved the utilization but also investigation of such PSO technique. By relocating the FFA processes into the binary space, the V-shaped function, which would be a component of the described technique by Naseri et al. [24], transforms the prolonged position of another FFA algorithm's solutions towards binary mode. Harris Hawks optimization metaheuristics, which were developed by Zivkovic et al. [25] and abd et al. [26] lately but are already well-known, and a deep neural network machine learning model are combined in this research.

3. Proposed methodology

Intrusion detection aims to identify any anomalous behaviour that system intruders might have generated. An effective detection algorithm is utilized to monitor as well as analyse the nodes to identify the intrusions. The proposed EBSO-DBN approach besides detection of attacks in the IoT is illustrated in Figure 1. In this study, a novel EBSO-DBN approach for recognizing and categorizing intrusions in the IoT environment has indeed been developed. First, a relevant format is generated by pre-processing the networking data. Then, implementing EBSO-DBN, the deep learning

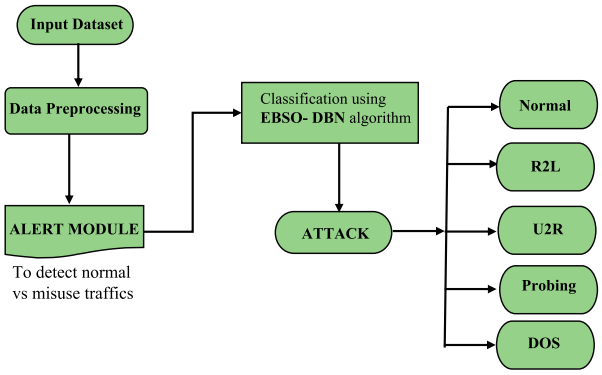

Figure 1. Schematic view of proposed EBSO-DBN model for intrusion detection in IoT.

(DL) technique is used during alert generation to identify as well as classify intrusions in the IoT environment. The stages for establishing an EBSO-DBN model for intrusion detection in which it seems to be more accurate as well as efficient are as follows:

1. Selecting a suitable dataset with high-quality data NSL KDD
2. Towards the research, the dataset was divided into 10% test as well as 90% train
3. The pre-processing stage: This stage essentially allows any noise imposed upon that data to be reduced or eliminated. This would be done with the intention to always collect important information. On the other hand, several of the most popular methods of normalization function is utilized in an effort to simplify the subsequent approaches.
4. To identify as well as categorize intrusions in the IoT environment, provide an alert module for the detection of intrusion.
5. In the end, due to alert generation classify the pre-processed data by using EBSO-DBN algorithm for characterization whether data is normal or attacks like R2L Attack, U2R Attack, Probing Attack, and also DOS Attack.

3.1. Data collection

The NSL-KDD dataset is based on an actual data extraction from a database that covers a wide range of simulated intrusions in a defence network environment. For data from Internet traffic was analysed, and four types of simulated attacks were identified: R2L Attack, U2R Attack, Probing Attack and also DOS Attack.

It represents a reasonable ratio of 100,917 training samples to 47,600 testing samples. Although containing considerable intrinsic problems such a lack of malicious attack scenarios, the NSL-KDD dataset is still the most often used IDSs evaluation dataset since it has the special ability to maximize predictions for classifiers. There are four attack categories with a total of 41 qualities, as well as a single labelled class that separates harmful from legitimate network data. The NSL-KDD dataset's comprehensive theoretical and technical documentation is available in reference for those who are still interested.

3.2. Date pre-processing

The training as well as testing sets were created using dataset. Following its training with the training set, the model uses the information to learn the mapping function. To assess the model's effectiveness, let use the testing set. Data preparation is the most time-consuming but essential phase in data extraction since it can make the process simpler as well as more efficient. Additionally, data might be noisy, excessive, incomplete, as well as conflicted and typically originates from different platform. As a result, it is crucial to transform raw data into knowledge that may be used for research and disclosure. The NSL KDD dataset's network traffic data provide values for each feature corresponding to the network packet properties. The 148,517 network traffic records (packets) together compose the entire dataset are classified as either normal packets or attack packets. The collection includes four categories of well-known attack packets.

i. Denial of Service Attack (DoS): This form of attack prevents users of a system from accessing resources or services they have requested.
ii. User to Root Attack (U2R): Due to a compromised user account, this sort of attack results in the hijacking of the host system.
iii. A remote to local attack (R2L): delivers a network packet to a computer in order to attack a user account and gain unauthorized access to the system.
iv. In a probe attack, the host ports are inspected to see if there are any open ports that could be used to exploit security flaws in the system.

3.2.1. Data normalization function

Both discrete as well as continuous characteristics are present in the NSL-KDD dataset, similar to those in KDD99. Features become more diverse and incongruous when their values differ. It is therefore necessary to normalize the data and scale all feature values into the same range during the pre-processing stage. The mean approach utilized for feature scaling is described by Equation (1).

$$\text{Mean} = \frac{1}{t \times \sum_{k=1}^{n}(x_k)} \tag{1}$$

The min–max algorithm, which can even convert the current range of data normally in the intervals [−1, 1] and [0, 1], is indeed the fundamental basis of the main normalizing function. Data normalization is the

premise for all of this algorithm. Equation gives the normalizing solution (1).

The mean in this circumstance is indeed an arithmetic mean. The total number of rows in a single column that are averaged is denoted by the symbol t. To handle the data dispersion using standard deviation; x_k is the average of each unique result.

The min–max algorithm, which can even convert the current range of data normally in the intervals [−1, 1] and [0, 1], is indeed the fundamental basis of the main normalizing function. Data normalization is the premise for all of this algorithm. Equation gives the normalizing solution (2).

$$P = \frac{((x - x_{min})(max - min))}{(x_{max} - x_{min}) + min} \tag{2}$$

where (max, min) is indeed that input variable's specified value, p relates the converted input value and (x_{min}, x_{max}) specifies the initial range values from input variables.

The following formula can be used to rescale a range between any two numbers [−1,1]:

$$p' = 1 + \left(\frac{(p - min(p))(-1, 1)}{\max(p) - \min(p)}\right) \tag{3}$$

$$\text{Mean normalization:} p' = \frac{p - \bar{p}}{\max p - \min(p)} \tag{4}$$

While p symbolizes the initial value, p' the normalized value, or even $\bar{p} = avg(p)$ the mean of the feature vector. The term "standardization" often refers to a unique way of normalizing means, which divides results by the standard deviation.

3.3. Evaluated Bird Swarm optimization -deep belief network (EBSO-DBN) classification

This portion provides the proposed EBSO-DBN classifier for categorizing IoT intrusions. To select the most appropriate weights for the DBN, the EBSO-DBN is proposed here by including BS into the DBN model. By choosing the best weights, the suggested EBSO aids in altering the performance with DBN. An EBSO-DBN, technique that is primarily utilized to improve the DBN's fundamental network structure. The DBN network structure's input layer count and output layer count are correlated with the number of data characteristics and categories, respectively. The proposed multiple sets of initiating network topologies are based around the number of hidden layer nodes in each layer but instead throughout each dimension of the particle, which corresponds to the integer bird swarm produced by the random technique.

This allows the algorithms to properly predict the class labels of previously unknown test records. Based on their high accuracy rates in the field of intrusion detection, good generalizability, as well as the variety of approaches they take to problem-solving, many categorization techniques are chosen. The performance of EBSO-DBN classification algorithm is analysed in context of the processed dataset by comparing the outputs of these learning algorithms.

Restricted Boltzmann machines (RBMs) as well as MLPs are used at various layers to create the DBN, which is a subset of the DNN. RBMs have hidden and visible units that are connected based on the connections' weights. The MLPs are regarded as input, hidden and output layers in feed-forward networks. A network with many layers can handle any challenging tasks and improve classification efficiency to find intrusions.

The following equation is used to perform gradient descent weight updates when training a single RBM:

$$W_{i,j}(t + 1) = W_{i,j}(t) + \eta \frac{\partial \log(p(V))}{\partial W_{i,j}} \tag{5}$$

where $p(V)$ seems to be the possibility that perhaps a vector can be observed.

The visible layer's input originates from the features in the NSL-KDD dataset, and the first RBM's hidden layer is described as

$$S^1 = \{S_1^1, S_2^1, \ldots., S_T^1, \ldots., S_{12}^1,\}; 12 \geq T \geq 1 \tag{6}$$

$$R^1 = \{R_1^1, R_2^1, \ldots., R_U^1, \ldots., S_V^1,\}; V \geq U \geq 1 \tag{7}$$

where S_T^1 denotes the T^{th} visible neuron in the first RBM, R_U^1 represents the U^{th} hidden neuron and V indicates total hidden neurons (Figure 2).

Training a property layer that can directly gain input data via pixels is the beginning stage. Obtain the characteristics of the preliminary acquired features in a different retired sub caste by using its values as pixels. Each time additional packages or features are added to the network, the lower bound on the log-liability of the training data set gets better.

The training process is mainly divided into two parts for both the EBSO and DBN modules:

For each RBM, a customized training code is created. During transmitting feature vectors across different feature spaces, this then ensures that feature data is retained as much as is practical as well as utilizes unsupervised fully independent features although during training process. The following equation is used to perform gradient descent weight updates when training a single RBM: The EBSA uses simulations of such foraging, flight, as well as vigilance subsystems to solve optimization problems. It was inspired by the social interactions but also behaviour of swarms of birds. Five straightforward guidelines, which are detailed below, can be used to summarize how birds interact with one another.

$$W_{i,j}(t + 1) = W_{i,j}(t) + \eta \frac{\partial \log(p(v))}{\partial W_{i,j}} \tag{8}$$

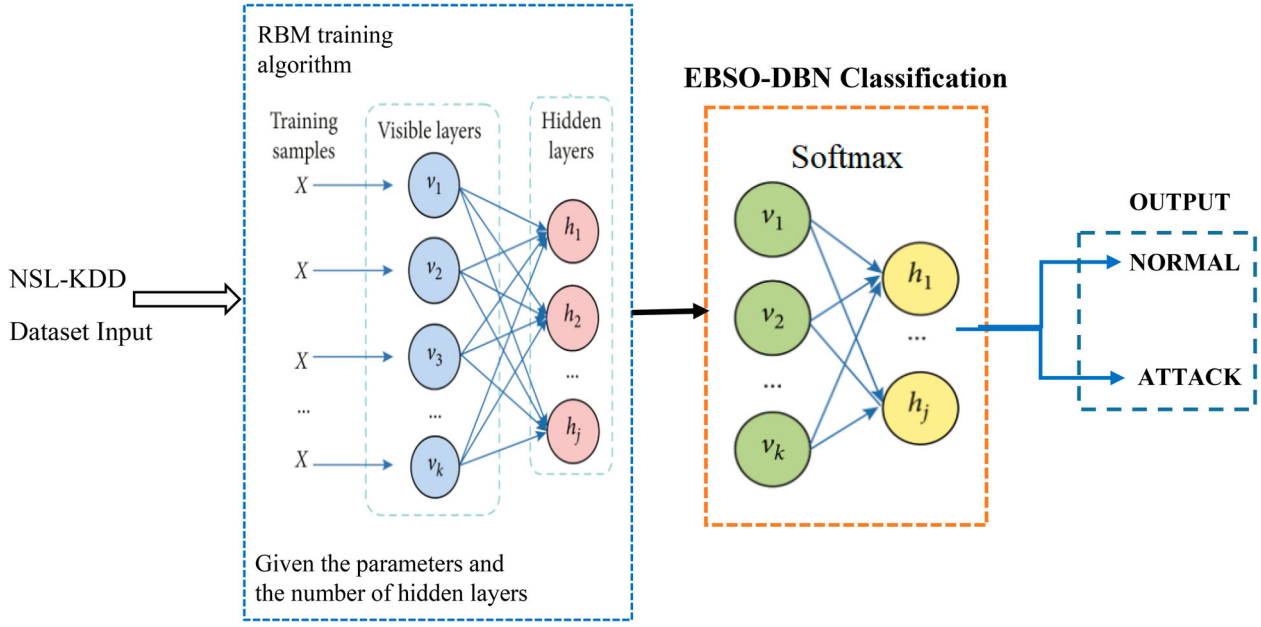

Figure 2. Architecture of EBSO-DBNs.

where $p(v)$ seems to be the probability that a visible vector will occur and therefore is determined by

$$p(v) = \frac{1}{Z}\sum_{H} e^{-E(v,h)} \tag{9}$$

where $E(v,h)$ is the energy function given to its traffic pattern and Z is perhaps partition function.

$$Z = \sum_{v,H} e^{-E(v,h)} \tag{10}$$

The following model represents the observed joint distribution of input value x and hidden layer H_k:

$$P(x, H_1, \ldots, H_N) = \left(\prod_{k=0}^{N-2} p(H_k|H_{k+1}) * (P(H_{k+1}, H_N))\right) \tag{11}$$

where $x = H_0$, $\mathrm{P}(H_{k+1}|H_k)$ is an RBM conditional distribution of hidden units with visible units in the k layer. At the top level of the RBM, the visible-hidden joint distribution is given by $\mathrm{P}(H_{N+1}, H_N)$.

Rule 1: Each bird seems to be in both the vigilant or foraging stage.

This can be signifies as a stochastic determination. The m-dimensional vector can be utilized to express each bird's position throughout the swarm.

$$X_i = \{X_i^1, X_i^2, \ldots., X_i^m\} \tag{12}$$

Rule 2: When foraging, each bird records as well as retains both its own best foraging experiences as well as the swarm in its entirety in terms of food placements. This information will affect the animal's movement and food-finding strategy.

Each bird makes a distinct alert. The procedure for each bird's position transformation during foraging is as follows:

$$X_{j-i}^{t+1} = X_{j-i}^t + (p_{(j-1)} - X_{i-j}^t) * C * rand(0,1) + (g_{(j-1)} - X_{i-j}^t) * S * rand(0,1) \tag{13}$$

X_{j-i}^{t+1} is the next position of the individual i, $g_{(i-j)}$ is the best position of the individual i, $p_{(i-j)}$ is the best position of the group S, C is positive, and mean j is the j^{th} component of own average position of the total bird.

Rule 3: During the alertness stage, each bird competes to move closer towards the flock's centre, presuming that birds with large reserves are closest to the middle. Predators are less likely to target birds in the middle.

The vigilance behaviour is described as

$$X_{j-i}^{t+1} = X_{j-i}^t + B_1(mean_j - X_{j-i}^t) \times rand(0,1) + B_2(p_{(i,j)} - X_{j-i}^t) \times rand(-1,1) \tag{14}$$

where B_1 and B_2 can be described mathematically as

$$B_1 = b_1 \times \exp\left[-\frac{P_{Fit_1}}{sum\ fit + \varepsilon} \times n\right] \tag{15}$$

$$B_2 = b_2 \times \exp\left[\left[\frac{Fit_i - Fit_K}{|Fit_k - Fit_i| + \varepsilon}\right] \times \frac{n \times P_{Fit_K}}{sum\ fit + \varepsilon}\right] \tag{16}$$

where b_1, b_2 and ε are constants.

Rule 4: Birds rotate between producing as well as foraging because they move from one location to another. The algorithm guarantees that producers have the largest reserves, whilst foragers have the smallest reserves. On the other side, producers or foragers are randomly assigned to other birds.

Rule 5: Food producers are constantly searching for novel sources of food. In search of nourishment, the scroungers randomly chase a producer.

The producers as well as scroungers can indeed be discriminated from the swarm. Mathematical descriptions of both the behaviours of a producers as well as scroungers are as follows, combined:

$$X_{j-i}^{t=1} = X_{j-i}^{t} + randn(0,1) * X_{j-i}^{t} \quad (17)$$

$$X_{j-i}^{t=1} = X_{j-i}^{t} + (X_{k-j}^{t} - X_{i-j}^{t}) \times FL \times rand(0,1) \quad (18)$$

where $randn(0,1)$ represents the Gaussian distributed random number with mean 0 and standard deviation1, $K\epsilon[1,2,3,..,n], K \neq i$. $FL(FL\epsilon[0,2])$ denotes that the scrounger would follow the producer to search for food.

ALGORITHM: EVALUATED BIRD SWARM OPTIMIZATION

Input: The population's overall number of packets, given in n.
m-the maximum no. of iteration
f-frequency of the bird's flight behaviour
C,S,FL,b_1, b_2,ε-constant parameters
T = 0; Set the population's initial parameters and also the resultant parameters.
Improve each group's fitness value as well as identify the optimal way to proceed
While($m > t$)
If(t%f $\neq$ 0)
For i = 1:n
If rand(0,1) < p
Birds go on product searches. (Equation 13)
Else
Creatures maintain alertness (Equation 14)
End if
End for
Else
Organize the swarm across two groups: producers as well as scroungers.
For i = 1:n
If I is a producer
Producing (Equation 17)
Else
Scrounging (Equation 18)
End if End for
End if Explore innovative solutions
Update them if the new solutions are superior to the past iterations.
Choose the optimal solution.
$T = t + 1$; End while
Output: the individual in the population with the highest objective function value

4. Performance evaluation

4.1. Performance measures

The EBSO-DBN algorithm-enhanced IDS methodology is trained in a variety of attack scenarios and assessed for its performance. The most frequently utilized parameters to assess a given behaviour DL-based IDS function are classification accuracy, true positive rate (recall or detection rate), then false alarm rate. Either the false positive rate or the sum of the false positive rate and false negative rate is used to calculate false alarm rates. Precision and the harmonic mean of recall and precision are additional measures (F-score).

Table 1. Analysis of the NSL-KDD dataset using the EBSO-DBN technique in compared to other advanced techniques.

Methods	Accuracy (%)	Recall (%)	Precision (%)
CuLSTMGRU [11]	95.40	93.28	94.25
RDBSA [12]	95.70	95.47	94.72
IBSA [14]	96.55	95.55	95.58
SMO-SDPN [18]	97.70	96.70	96.25
NSBPSO [23]	98.25	97.2	96.62
EBSO-DBN [PROPOSED]	98.96	98.87	99.4

Table 2. Results of EBSO-DBN model for IDS.

Class	TP rate	FP rate	Correctly classified	Incorrectly classified
Normal	0.971	0.035	98.765	3.6826
Attack	0.041	0.036		
Weight Avg.	0.978	0.035		
Time taken to build the model			145 s	

The following performance criteria were applied in this study:

(1) Classification Accuracy (CA)

$$= \frac{TP + TN}{TP + TN + FN} \times 100 \quad (19)$$

$$(2)\ \text{Error Rate } (ER) = \frac{FP + FN}{TP + TN + FN + FP} \times 100 \quad (20)$$

$$(3)\ \text{Precision Rate } (PR) = \frac{TP}{TP + Fp} \times 100 \quad (21)$$

(4) Recall (RC)/ Detection Rate (DR)

$$= \frac{TP}{TP + FN} \times 100 \quad (22)$$

$$(5)\ \text{False Alarm Rate } (FAR) = \frac{FPR + FNR}{2} \quad (23)$$

$$(6)\ F_{score} = \frac{2 * \text{precision} * \text{Detection Rate}}{\text{Precision} + \text{detection rate}} \quad (24)$$

4.2. Experimental results

The research findings that were gathered after employing the distributed approach were provided throughout this portion.

Table 1 compares the performance of the EBSO-DBNS model to that of other methods using the test NSL-KDD dataset.

This is a result of such higher classification accuracy rate (98.73), which was greater only at time of implementation in comparison to other classified throughout the field. One can view the data value of normal, attack and weighted average results in Table 2 of the findings.

According to Figure 3, the suggested model's output is (98.75%) for classification accuracy, (98.9%) for detection rate, as well as (93.21%) for false alarm rate.

According to Figure 4, a given degree of network structure produced by a particular form of attack is

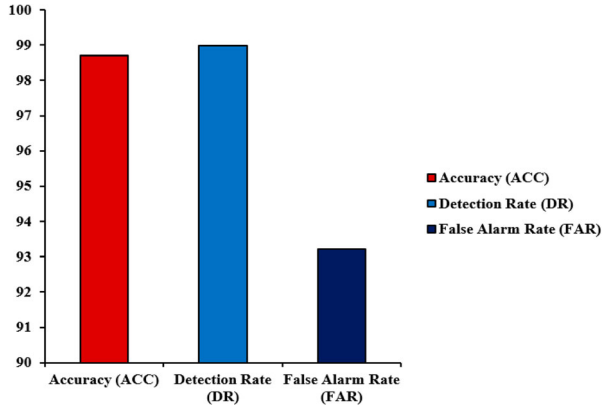

Figure 3. Result of EBSO-DNN classification accuracy, detection and false alarm rate graph.

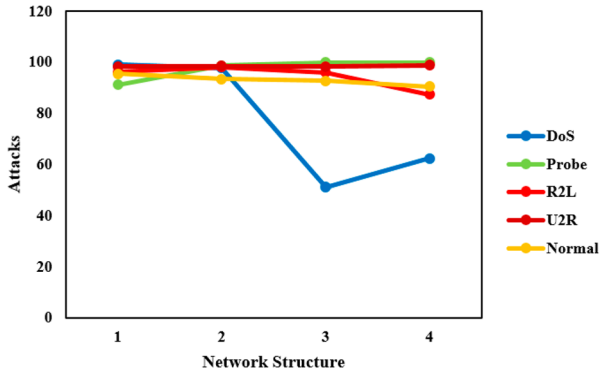

Figure 4. Detection Rate for different class of attacks.

more likely to be discovered than other network structures. As seen, the network structure that the EBSO-DBN algorithm generates adaptively has a higher detection rate than alternative network structures.

The testing time indicates the measurement time required to examine each packet delivered across the network, while the training time is the measurement time required to train the DNN structure. Since a training has a time complexity from 4 to 11 seconds, it should be conducted offline. However, the time complexity in a testing period during packet inspection is only 2–5 ms for categorizing the packets and 8–9 s for processing features per packet, which may be applied to a real-time application depicted in Figure 5.

The original dataset was divided into three separate datasets for the study, as follows: The DL models were trained on 70% of the data (105,132 records), tested on 20% of the data (30,740 records), subsequently validated on 10% of the data (14,644 records). Table 3 displays the distribution of attack incidents across the training, testing and validation datasets.

EBSO-DBN intrusion detection was assessed using the flow-based dataset as seen in Figure 6. There are classifications for both regular and attack in it. Every incident of a traffic record is categorized as normal, suspicious, unknown, aggressor or victim.

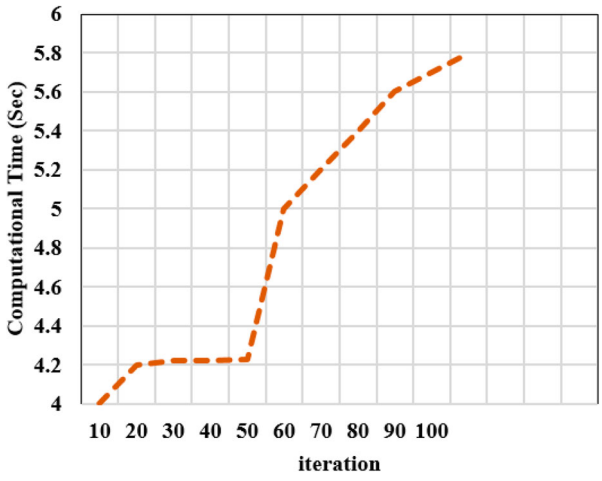

Figure 5. Computation time.

Table 3. The quantity of packets used for training, testing, and validation sets (per packet type).

Packet type	Training set	Testing set	Validation set
Normal Traffic	55,149	16,465	7695
DoS	37,121	10,601	5501
U2R	8312	2396	1230
R2L	2835	762	392
Probe	1715	516	258

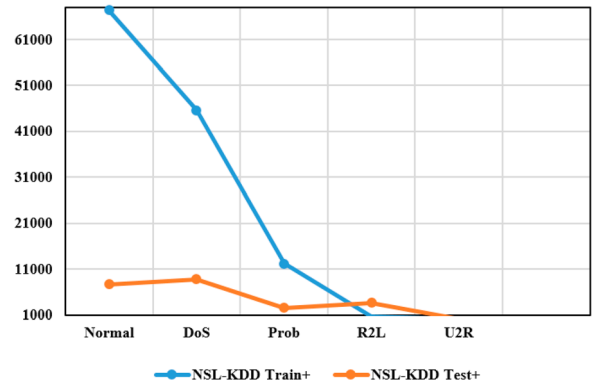

Figure 6. Normal and attack class of training and testing set.

Figure 7 represents the graphical representation that compares the proposed approach with other algorithms. The IDS prevents malicious traffic from making any kind of changes in the network that could be harmful. It protects the system from DDOS (distributed denial of attack), data breach, server shutdown and similar kinds of problems that could lead to hinder production.

The first step would be to identify the false positive and then to determine the root cause of the false positive. Once you have determined the root cause, you can then take steps to mitigate the false positive and to prevent it from happening again in the future.

5. Conclusion and future work

Among the complex aspects for investigators to protect network infrastructure from adversary activities

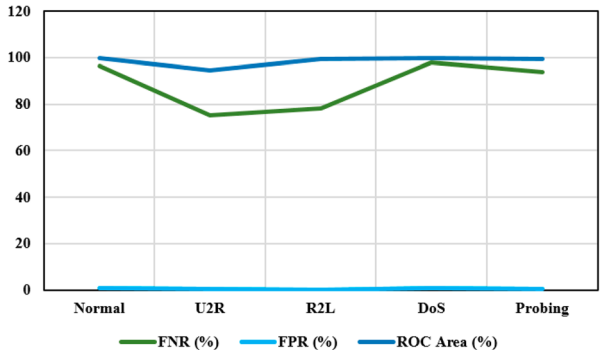

Figure 7. overall efficiency of the proposed approach.

involves detecting abnormal traffic in the Internet of Things (IoT). There are numerous automatic methods that can find unusual traffic. Furthermore, overall efficiency, adaptability, as well as scalability of current Intrusion Detection Systems (IDS) require to be improved in order to identify attack traffic from diverse IoT networks in addition to accuracy. The EBSO-DBN algorithm and the edge based feature selection algorithm will be used with other algorithms to improve the exploration and also exploitation capabilities, further reducing the training time for feature subset and classification detection. To ascertain the optimal performance of IDS models, this study focuses on building an ensemble classifier utilizing the suggested EBSO-DBN classification algorithm with 98.7% of accuracy, 99.4% of precision and 98.8% of recall. The work currently working towards this direction exploring the various capabilities of DBNs when deployed in a distributed manner.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Nguyen X-H, Nguyen X-D, Huynh H-H, et al. Realguard: a lightweight network intrusion detection system for IoT gateways. Sensors. 2022;22(2):432. doi:10.3390/s22020432

[2] Abbas A, Khan MA, Latif S, et al. A new ensemble-based intrusion detection system for internet of things. Arab J Sci Eng. 2022;47(2):1805–1819. doi:10.1007/s13369-021-06086-5

[3] Saba T, Rehman A, Sadad T, et al. Anomaly-based intrusion detection system for IoT networks through deep learning model. Comput Electr Eng. 2022;99:107810. doi:10.1016/j.compeleceng.2022.107810

[4] Mehedi ST, Anwar A, Rahman Z, et al. Dependable intrusion detection system for IoT: a deep transfer learning-based approach. IEEE Trans Ind Inf. 2022;19(1):1006–1017.

[5] Le K-H, Nguyen M-H, Tran T-D, et al. IMIDS: an intelligent intrusion detection system against cyber threats in IoT. Electronics. 2022;11(4):524. doi:10.3390/electronics11040524

[6] Idrissi I, Azizi M, Moussaoui O. An unsupervised generative adversarial network based-host intrusion detection system for internet of things devices. Indones J Electric Eng Comput Sci (IJEECS). 2022;25(2):1140–1150. doi:10.11591/ijeecs.v25.i2.pp1140-1150

[7] Alghamdi R, Bellaiche M. Evaluation and selection models for ensemble intrusion detection systems in IoT. IoT. 2022;3(2):285–314. doi:10.3390/iot3020017

[8] Singh KP, Kesswani N. An anomaly-based intrusion detection system for IoT networks using trust factor. SN Comput Sci. 2022;3(2):1–9. doi:10.1007/s42979-022-01053-9

[9] Mendonça RV, Silva JC, Rosa RL, et al. A lightweight intelligent intrusion detection system for industrial internet of things using deep learning algorithms. Expert Syst. 2022;39(5):e12917. doi:10.1111/exsy.12917

[10] Dahou A, Elaziz MA, Chelloug SA, et al. Intrusion detection system for IoT based on deep learning and modified reptile search algorithm. Comput Intell Neurosci. 2022;2022. doi:10.1155/2022/6473507

[11] Muthanna MSA, Alkanhel R, Muthanna A, et al. Towards SDN-enabled, intelligent intrusion detection system for internet of things (IoT). IEEE Access. 2022;10:22756–22768. doi:10.1109/ACCESS.2022.3153716

[12] Balashunmugaraja B, Ganeshbabu TR. Privacy preservation of cloud data in business application enabled by multi-objective red deer-bird swarm algorithm. Knowl Based Syst. 2022;236:107748. doi:10.1016/j.knosys.2021.107748

[13] Onah JO, Abdullahi M, Hassan IH, et al. Genetic algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment. Mach Learn Appl. 2021;6:100156. doi:10.1016/j.mlwa.2021.100156

[14] Fan X. Cloud computing task scheduling based on improved bird swarm algorithm. Int J Performabil Eng. 2021;17(1):85–94.

[15] Mokbal FMM, Wang D, Osman M, et al. An efficient intrusion detection framework based on embedding feature selection and ensemble learning technique. Int Arab J Inf Technol. 2022;19(2):237–248.

[16] Tharewal S, Ashfaque MW, Banu SS, et al. Intrusion detection system for industrial internet of things based on deep reinforcement learning. Wirel Commun Mob Comput. 2022;2022. doi:10.1155/2022/9023719

[17] Bacha S, Aljuhani A, Abdellafou KB, et al. Anomaly-based intrusion detection system in IoT using kernel extreme learning machine. J Ambient Intell Humaniz Comput. 2022: 1–12. doi:10.1007/s12652-022-03887-w

[18] Otoum Y, Liu D, Nayak A. DL-IDS: a deep learning-based intrusion detection framework for securing IoT. Trans Emerg Telecommun Technol. 2022;33(3):e3803. doi:10.1002/ett.3803

[19] Singh A, Chatterjee K, Satapathy SC. An edge based hybrid intrusion detection framework for mobile edge computing. Complex Intell Syst. 2021;8(5):3719–3746.

[20] Jeyaselvi M, Dhanaraj RK, Sathya M, et al. A highly secured intrusion detection system for IoT using EXPSO-STFA feature selection for LAANN to detect attacks. Cluster Comput. 2022;26(1):559–574.

[21] Ogwara NO, Petrova K, Yang ML. Towards the development of a cloud computing intrusion detection framework using an ensemble hybrid feature selection approach. J Comput Netw Commun. 2022;2022. doi:10.1155/2022/5988567

[22] Jaw E, Wang X. Feature selection and ensemble-based intrusion detection system: an efficient and comprehensive approach. Symmetry. 2021;13(10):1764. doi:10.3390/sym13101764

[23] Baniasadi S, Rostami O, Martín D, et al. A novel deep supervised learning-based approach for intrusion detection in IoT systems. Sensors. 2022;22(12):4459. doi:10.3390/s22124459

[24] Naseri TS, Gharehchopogh FS. A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. J Netw Syst Manage. 2022;30(3):1–27. doi:10.1007/s10922-022-09653-9

[25] Zivkovic M, Bacanin N, Arandjelovic J, et al. Novel Harris Hawks optimization and deep neural network approach for intrusion detection. In: Proceedings of International Joint Conference on Advances in Computational Intelligence : IJCACI 2021. Singapore: Springer; 2022, May. p. 239–250.

[26] Abd Elaziz M, Al-qaness MA, Dahou A, et al. Intrusion detection approach for cloud and IoT environments using deep learning and capuchin search algorithm. Adv Eng Softw. 2023;176:103402. doi:10.1016/j.advengsoft.2022.103402