Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.034321
Article

Tech Science Press

Modified Buffalo Optimization with Big Data Analytics Assisted Intrusion Detection Model

R. Sheeba[1,*], R. Sharmila[2], Ahmed Alkhayyat[3] and Rami Q. Malik[4]

[1]Department of Computer Science and Engineering, K. Ramakrishnan College of Engineering, Tiruchirappalli, 621112, India
[2]Department of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, 621212, India
[3]College of Technical Engineering, The Islamic University, Najaf, Iraq
[4]Medical Instrumentation Techniques Engineering Department, Al-Mustaqbal University College, Babylon, Iraq
*Corresponding Author: R. Sheeba. Email: rsheebaphd@gmail.com
Received: 13 July 2022; Accepted: 13 November 2022

Abstract: Lately, the Internet of Things (IoT) application requires millions of structured and unstructured data since it has numerous problems, such as data organization, production, and capturing. To address these shortcomings, big data analytics is the most superior technology that has to be adapted. Even though big data and IoT could make human life more convenient, those benefits come at the expense of security. To manage these kinds of threats, the intrusion detection system has been extensively applied to identify malicious network traffic, particularly once the preventive technique fails at the level of endpoint IoT devices. As cyber-attacks targeting IoT have gradually become stealthy and more sophisticated, intrusion detection systems (IDS) must continually emerge to manage evolving security threats. This study devises Big Data Analytics with the Internet of Things Assisted Intrusion Detection using Modified Buffalo Optimization Algorithm with Deep Learning (IDMBOA-DL) algorithm. In the presented IDMBOA-DL model, the Hadoop MapReduce tool is exploited for managing big data. The MBOA algorithm is applied to derive an optimal subset of features from picking an optimum set of feature subsets. Finally, the sine cosine algorithm (SCA) with convolutional autoencoder (CAE) mechanism is utilized to recognize and classify the intrusions in the IoT network. A wide range of simulations was conducted to demonstrate the enhanced results of the IDMBOA-DL algorithm. The comparison outcomes emphasized the better performance of the IDMBOA-DL model over other approaches.

Keywords: Big data analytics; internet of things; security; intrusion detection; deep learning

1 Introduction

The growth of the Internet of Things (IoT) systems and technologies were rising at an unprecedented rate. The scale of the latest IoT technology goes far beyond the individual level, with IoT gadgets broadly spread across countries or entire cities [1]. With increasing transmission bandwidth and speed, IoT

gadgets can collect, transmit, and process a huge volume of data. Such IoT mechanisms, linked with the gathered data, provided excessive chances to provide and design intellectual services in special applications, like smart cyber-physical systems (CPS), intelligent transportation, and automated surveillance [2]. But, the gathered IoT data also comprises delicate data and thus needs closer attention on reliable data security issues and privacy protection [3]. For dealing with increased security and privacy concerns, the latest IoT or distributed mechanism prevent and detect network intrusion intelligently. Several efforts were contributed to advance deep learning-based (DL) or machine learning (ML) methods for intrusion detection systems (IDS) to prevent any deviation or misappropriation in IoT and frameworks [4]. Even though IDS was well employed in identifying malicious network acts, one such main vulnerabilities of prevailing IDS were the lack of capability for detecting unknown kinds of network intrusion because of the restricted or imbalanced intrusion data at the time of model training processes [5]. Moreover, prevailing ML techniques are to manage multidomain ID that calls for the additional exploration of hybrid DL structures. Fig. 1 illustrates the overview of Big Data in the IoT Environment.

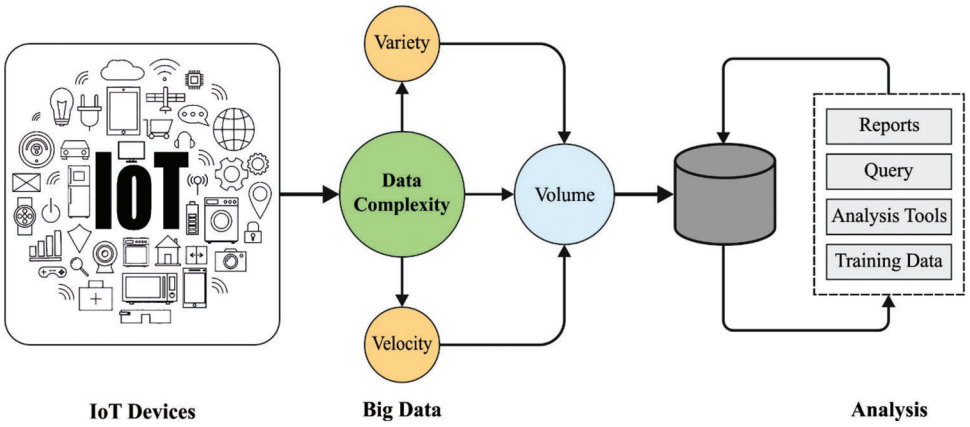

Figure 1: Big data in IoT environment

Because of its heterogeneous nature, the IoT system produces multimodal, temporal, and high-dimensional data [6]. Implementing big data analytics on these data was the potential for discovering hidden paradigms, disclosing hidden relations, and acquiring new insights. Artificial intelligence (AI) is utilized in big data analysis [7]. Specifically, DL methods have proven their success in handling heterogeneous data. It can examine complicated and large-scale datasets to get visions, spot dependencies within datasets, and study previous assault patterns to recognize new and unseen attack patterns [8]. Since IoT gadgets were resource-limited and had inadequate capabilities concerning storage and computation, heavyweight workloads such as big data analysis and constructing learning mechanisms have been offloaded to cloud servers and fog [9]. Therefore, computation offloading could help reduce the performance delay of task and stores energy utilization of battery-powered and mobile IoT gadgets; however, it also imposes certain security concern. Several DL methods were introduced for IDS, and few of them particularly concentrate on IoT [10]. Every method implements its own design choices that may limit its ability to attain better performance of efficiency and effectiveness.

Nie [11] introduce a DL-related intrusion detection (ID) method. Depends on the generative adversarial network (GAN) and formulated a robust ID technique. This ID technique has 3 stages. Firstly, the feature selection approaches were used for processing the collaborative edge network traffic. Secondly, a DL architecture related to GAN was devised for ID, focusing on a single attack. Lastly, a new ID method was proposed by merging numerous ID methods concentrating on a single attack. The presented

GAN-related DL architecture can realize ID targeting for various attacks. In [12], to mitigate the inconsistency among feature retention and dimensionality reduction in imbalanced data, projected a variational long short-term memory (VLSTM) learning method for intellectual anomaly detection (AD) related to reconstructed feature depiction.

Basset et al. [13] modelled a forensics-related DL method for detecting intrusion in industrial IoT (IIoT) traffic. The method studies local representation utilizing a local gated recurrent unit (LocalGRU) and presents a multi-head attention (MHA) layer for capturing and learning global representations. A residual connection among layers can be formulated to prevent information loss. One more difficulty confronting the present IIoT forensics structures was their inadequate scalability, restricting performances in dealing with Big IIoT traffic datasets generated by IIoT gadgets. This difficulty can be sorted by training and deploying the suggested model in a fog computing ecosystem. Idrissi et al. [14] devise a new unsupervised anomaly detection (AD) based Host-IDS for IoT related to adversarial training structure utilizing the GAN. This presented IDS, termed "EdgeIDS", aims most of the IoT gadgets due to limited functionality; IoT gadgets forwards and receive merely detailed data, not like conventional gadgets, like computers or servers, that exchange an extensive range of data.

In [15], a hierarchical intrusion security detection using a stacked Denoised AutoEncoder with Support vector machine (SDAE-SVM) can be built based on the 3-layer neural network (NN) of self-encoder. The sample dataset, after reducing dimensions, was acquired by layer-wise fine-tuning and pretraining. The conventional DL methods deep belief network (DBN) stacked noise autoencoder (SNAE), stacked sparse autoencoder (SSAE), stacked contractive autoencoder (SCAE), stacked autoencoder (SAE)], were presented for executing the comparative simulation with the method in this study. Nie et al. [16] formulated an identifier (ID) method that depends on deep reinforcement learning (DRL) that follows the trend of traffic flow through the extraction of statistical features of previous network traffic for traffic prediction. Afterwards, uses traffic predictors for employing ID.

Though several models are available in the literature, most of the existing works do not focus on feature selection and hyperparameter tuning process concurrently. The hyperparameter values play a vital role in affecting the performance of the DL models. Since trial-and-error hyperparameter tuning is tedious, metaheuristic algorithms can be employed for it. Therefore, this study devises Big Data Analytics with the Internet of Things Assisted Intrusion Detection using Modified Buffalo Optimization Algorithm with Deep Learning (IDMBOA-DL) model. In the presented IDMBOA-DL model, the Hadoop MapReduce tool is exploited for managing big data. The MBOA algorithm is applied to derive an optimal subset of features from picking an optimum set of feature subsets. Finally, the sine cosine algorithm (SCA) with convolutional autoencoder (CAE) model is utilized to recognize and classify the intrusions in the IoT network. A wide range of simulations was conducted to demonstrate the enhanced results of the IDMBOA-DL algorithm.

2 The Proposed Model

This study developed a new IDMBOA-DL approach for intrusion detection in the IoT-enabled big data environment. In the presented IDMBOA-DL model, the Hadoop MapReduce tool is exploited for managing big data. The MBOA algorithm is applied to derive an optimal subset of features from picking an optimum set of feature subsets. Finally, SCA with the CAE model is utilized to recognize and classify the intrusions in the IoT network. Fig. 2 illustrates the IDMBOA-DL approach.

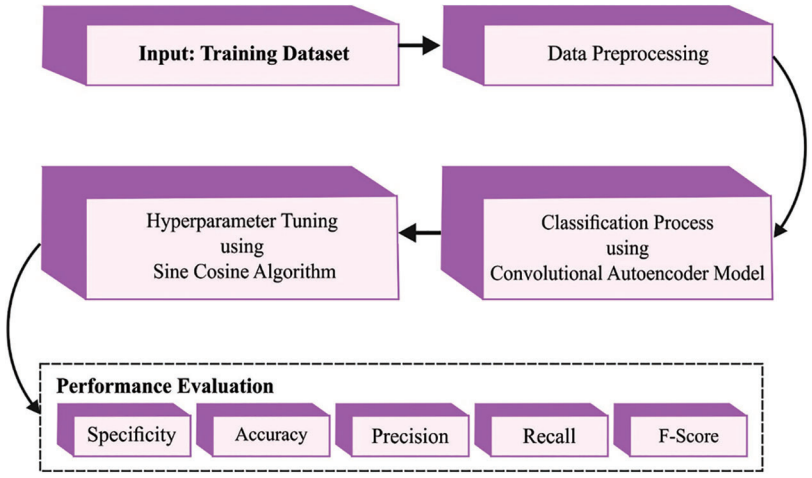

Figure 2: Block diagram of IDMBOA-DL technique

2.1 Hadoop MapReduce

In the presented IDMBOA-DL model, the Hadoop MapReduce tool is exploited to manage big data. MapReduce is a publicly-available software platform for sequential data processing based on the Hadoop Distributed File System (HDFS) [17]. Generally, HDFS comprises a name node and several data nodes. It uses simple data models containing value and key pairs to maximise the parallelism of the data processing and the convenience of horizontal scaling. Also, the simple key-value data model is effective for the parallel data processing on the disk, as HDFS employs disk input-output (I/O)-based batch processing that is better suited for the considerable quantity of data processing than that of memory-based processing.

2.2 Design of MBOA-Based Feature Selection

The MBOA algorithm is applied to derive an optimal subset of features from picking an optimum set of feature subsets. A typical variant of the BOA technique is introduced [18]. The BOA algorithm encompasses the unique abilities of this animal for robust exploitation and exploration in the search space. It tries to resolve the premature convergence problem by ensuring that every individual buffalo is upgrading its position concerning prior experience. Another unique feature of BOA is its sufficient exploitation via reinitializing the whole herd once the leader (the best buffalo) is not improved with iteration.

The fundamental steps of the BOA technique are shown in the following.

1. Initialize the objective function $(x),\ x \in S$, population size n_p, and algorithm parameter, namely l_{p1} & l_{p2}, and so on.
2. Generate the potential population of buffalo randomly and initialize on a random node within the searching space.

$$P_{dis_j}^{Lim}(t) = \begin{cases} 0, & if\ SOC_j(t) \le SOC^{min}\ or\ I_{gs}(t) \le 0 \\ \sqrt{\eta} P_{dis}^{\max}, & if\ SOC_j(t) - \dfrac{P_{dis}^{\max}}{B_j^r} \ge SOC^{\min}\ and\ I_{gs}(t) \forall j \\ \sqrt{\eta} B_j^r \left(SOC_j(t) - SOC^{\min}\right), & if\ SOC_j(t) - \dfrac{P_{dis}^{\max}}{B_j^r} < SOC^{\min}\ and\ I_{gs}(t) > 0 \end{cases} \quad (1)$$

$$P_{charj}^{Lim}(t)=\begin{cases}0, & if\ SOC_j(t)=SOC^{max}\\ P_{char}^{\max}/\sqrt{\eta}, & if\ SOC_j(t)+\dfrac{P_{char}^{\max}}{B_j^r}\le SOC^{\max}\ \ \forall j\\ B_j^r\left(SOC_j(t)-SOC^{\min}\right)/\sqrt{\eta}, & if\ SOC_j(t)+\dfrac{P_{dis}^{\max}}{B_j^r}<SOC^{\max}\end{cases} \tag{2}$$

3. Now upgrade the fitness value of *i-th* buffalo based on the following equation.

$$m_{i+1}=m_i+l_{p1}(h_{best}-w_i)+l_{p2}\left(h_{best,i}-w_i\right) \tag{3}$$

where m_i and w_i represent the exploitation and exploration moves of *i-th* buffaloes $\left(i=1,\ 2,\ 3,\ \ \ldots,\ n_p\right)$, correspondingly. l_{p1} and l_{p2} indicate the learning factor differs from 0.1 to 0.6. Moreover, h_{best} and s_{best} indicate the optimal fitness values of the herd and the best fitness of buffalo i, respectively.

4. Upgrade the location of *i-th* buffalo and $[h_{best},\ sbest,\ i]$,

$$w_{i+1}=\frac{w_i+m_i}{\pm 0.5} \tag{4}$$

5. Is h_{best} improving If yes, go to the next step, or else go to step 2.
6. Repeat steps 3 to 5 until the ending condition is not accomplished, or else go to the next step.
7. Print the optimum solution.

In this study, the MBOA is derived by using the concept of Levy flight. Levy's walk depicts the diffusion pattern of organisms so that searching can be focused on the position of possible solutions. Levy flight foraging hypothesis evaluates the migration from lower-resource to higher-resource environments that consecutively leads to optimum search. Animals with higher memory ability used this algorithm for exploring the search space. The concept of optimum foraging is an extension of Levy's flight foraging hypothesis that organisms give greater consideration to the optimum solution instead of aimless search within the searching space. Levy flight is a random walk that step length can be derived from the Levy distribution, frequently in terms of a simple power law equation as given below.

$L(\zeta)\sim\zeta^{-1-\alpha}$ where $0<\alpha<2$ is an index, and it is arithmetically given in the following

$$L(\zeta,\ \ \omega,\ \ \psi)=\begin{cases}\sqrt{\dfrac{\omega}{2\pi}}\exp\left[-\dfrac{\omega}{2(\zeta-\psi)}\right]\dfrac{1}{(\zeta-\dfrac{\psi)^3}{2}}, & 0<\psi<\zeta<\infty\\ 0 & otherwise\end{cases} \tag{5}$$

$\Psi>0$ is a minimal step, and ω is a scale variable. Ideally, as $\zeta\to\infty$, then.

$$L(\zeta,\ \ \omega,\ \ \psi)\approx\sqrt{\frac{\omega}{2\pi}}\frac{1}{\zeta^3/_2}. \tag{6}$$

The Mantegna algorithm is applied for the implementation of levy flight. Therefore, the step length ζ is evaluated as follows

$$\zeta=\frac{\ell}{|\kappa|^1/\alpha}, \tag{7}$$

where ℓ and κ are derived from the standard distribution,

$$\ell \sim N\left(0,\ \rho_{\ell}^{2}\right),\ \kappa \sim N(0,\ \rho_{\ell}^{2}), \tag{8}$$

The fitness function (FF) employed in the presented technique was planned to contain a balance amongst the amount of chosen features from all the solutions (minimal) and the classifier accuracy (maximal) reached by utilizing these chosen features, Eq. (9) demonstrates the FF for estimating solutions.

$$Fitness = \alpha\gamma_{R}(D) + \beta\frac{|R|}{|C|} \tag{9}$$

Whereas $\gamma_{R}(D)$ indicates the classifier error rate of provided classifier. $|R|$ implies the cardinality of chosen subset, and $|C|$ represents the total number of features from the dataset; α, and β are 2 parameters equivalent to the significance of classifier quality and subset length.

2.3 CAE-Based Classification

To recognize intrusions, the CAE model is exploited in this work. Autoencoder (AE) comprises 2 parts: encoding and decoding [19]. The encoding converts the input x to hidden depiction y (feature code) utilizing a deterministic mapping function. Usually, it can be an affine mapping function after that nonlinearity:

$$y = f(Wx + b) \tag{10}$$

Whereas W refers to the weighted amongst input x and hidden depiction y and b is biased. The decoding executes the procedure of restructuring the outcome z by y, which is formulated as:

$$z = f'(W'y + b') \tag{11}$$

W' signifies the weighted amongst hidden representation y and outcome z, and b' is bias. Compared with the input x, z is assumed as the reconstruction of x.

The principle of training an AE is for minimizing the recreation error that is recognized by minimizing the following cost function J_{AE}:

$$J_{AE} = \frac{1}{p}\sum\nolimits_{i=1}^{p} L\left[x_{i,}z_{i}\right] \tag{12}$$

In which p implies the number of input images, x_i indicates the i^{th} input image, and z_i signifies the reconstructed image equivalent to x_i. $L\left[x_{i,}z_{i}\right]$ stands for the reconstruction error of input images x_i that is evaluated by mean square error (MSE) or cross-entropy (CE). During this case, the MSE among the input image $x_i(\mathrm{i} = 1,\ 2,\ \ldots\ p)$ and the recreated patch of images $z_i(i = 1\ ,\ 2,\ \ldots p)$. Similarly, $L[x_i,\ z_i]$ is formulated as:

$$L_{AE}[x_i,\ z_i] = ||x_i - z_i||^2 \tag{13}$$

Convolutional AE (CAE) integrates the local convolutional linked with the AE, an easy step that adds convolutional function to inputs. Individually, a CAE contains convolutional encoding as well as decoding. The convolutional encoding recognizes the procedure of convolution conversion in the input to the feature map, but convolutional decoding applies the convolution conversion in the feature maps for the outcome. The extracting features and the recreated output in CAE are computed with a convolutional neural network (CNN). Therefore, Eqs. (10) and (11) are rewritten as:

$$y = ReLU(\omega x + b) \tag{14}$$

$$z = ReLU(\omega' y + b') \tag{15}$$

Whereas ω signifies the convolutional kernels among the input and code y, ω' denotes the convolution kernels amongst the code y and the resultant. b and b' are biases. In addition, the encoder and decoder functions are calculated utilizing unsupervised greedy training.

2.4 Parameter Tuning Using SCA

The SCA technique is utilized in the final stage to adjust the hyperparameters. The principle of SCA is easy and simple to implement [20]. It only implements the property of sine and cosine operations for achieving global search and local progress of searching space and continuously optimizing the solution set of main functions with iterative evolutions. Consider that N searches agents from the D dimension searching spaces, whereas the place of *an* i^{th} searching agent is formulated as $x_i = (x_{i1},\ x_{i2},\ \ldots,\ x_{iD}),\ i \in \{1,\ 2,\ \ldots,\ N\}$. The procedure for optimizing the main function with SCA is as follows.

Primarily, the places of N searching agents were arbitrarily initialized from the searching spaces. Secondarily, the individual fitness values were computed dependent upon the objective functions. At last, the present optimum individual places were chosen and stored. During all the iterations of this technique, the individual upgrades the place based on Eq. (16).

$$x_{id}^{t+1} = \begin{cases} x_{id}^{t} + r_1 \times \sin r_2 \times \left| r_3 P_d^t - x_{id}^t \right|, & r_4 < 0.5, \\ x_{id}^{t} + r_1 \times \cos r_2 \times \left| r_3 P_d^t - x_{id}^t \right|, & r_4 \geq 0.5, \end{cases} \tag{16}$$

Whereas t signifies the present iteration, x_{id}^t represents the place of the i^{th} solution from the d^{th} dimensional at $t^{he\ tth}$ iteration, and P_d^t stands for the place of global optimum solutions from the d^{th} dimensional at $t^{he\ tth}$ iteration. There are four important parameters in Eq. (16), whereas $r_1 = 2\left(1 - \frac{t}{T}\right)$ (T signifies the maximal amount of iterations) is the sine-cosine amplitude adjustment feature, and r_1 defines the direction of the next iteration of the i^{th} individual; $r_2 \in (0,\ 2\pi)$, $r_3 \in (0,\ 2)$, and $r_4 \in (0,\ 1)$ are arbitrary numbers, whereas r_2 defines the distance to the next iteration of i^{th} individual r_3 indicates the weighted factor of global optimum individual and r_4 denotes the discriminant co-efficient.

3 Results and Discussion

The proposed model is simulated using Python 3.6.5 tool. The proposed model experiments on PC i5-8600k, GeForce 1050Ti 4 GB, 16 GB RAM, 250 GB SSD, and 1 TB HDD. The experimental validation of the IDMBOA-DL method is tested using a dataset comprising 148517 samples under five classes, as shown in Table 1. Fig. 3 represents the confusion matrices formed by the IDMBOA-DL model on the applied data. The results denoted that the IDMBOA-DL model has effectually recognized all kinds of attacks or intrusions that exist in the IoT data.

Table 1: List of class labels

Class	No. of samples
Normal	77054
DoS	53385
Probe	14410
R2L	3416
U2R	252
Total No. of samples	**148517**

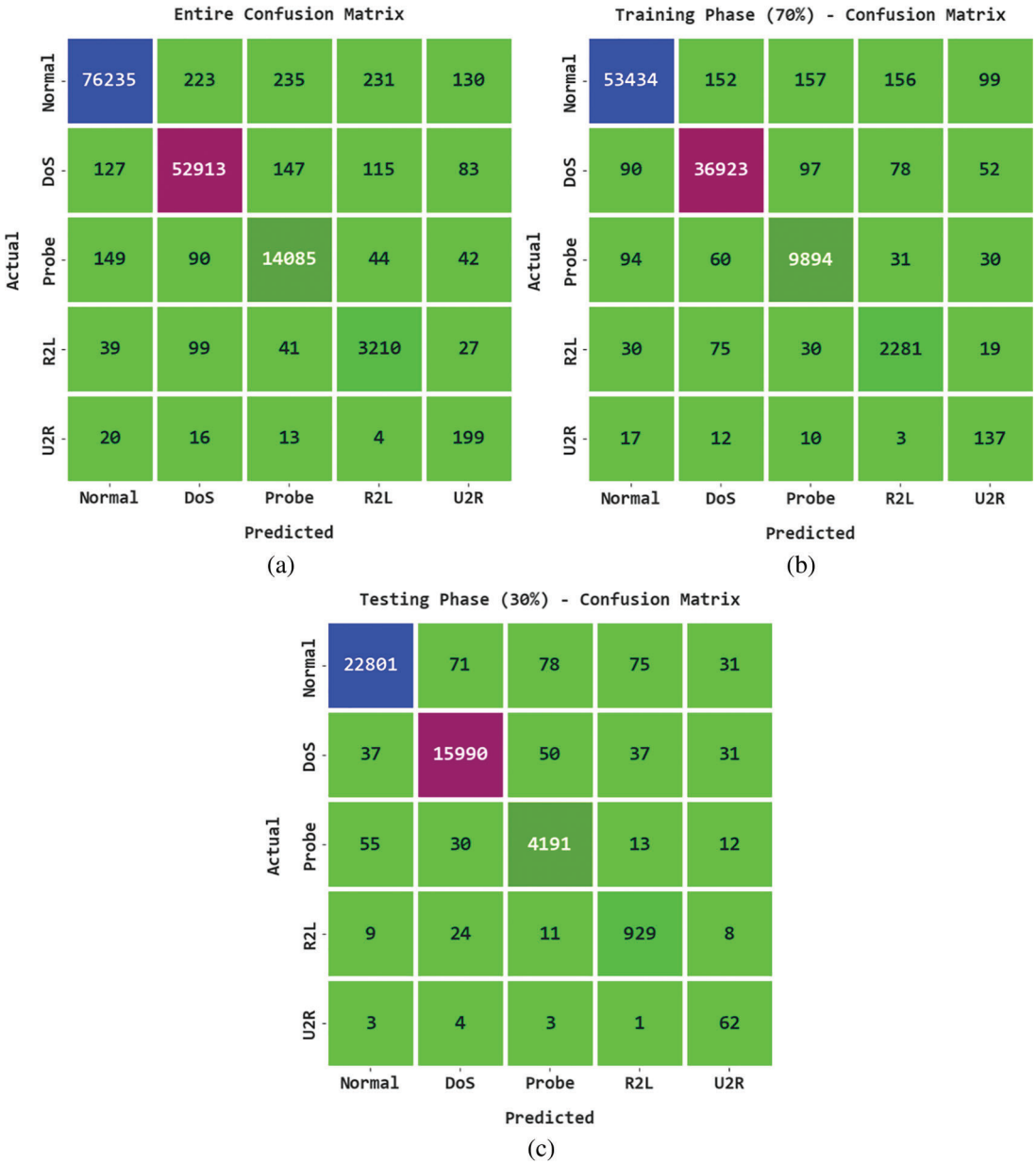

Figure 3: Confusion matrices of IDMBOA-DL approach (a) Entire dataset, (b) 70% of TR data, and (c) 30% of TS data

Table 2 provides an extensive intrusion detection performance of the IDMBOA-DL model on distinct class labels. Fig. 4 reports a brief intrusion classification outcome of the IDMBOA-DL model with several class labels under the entire dataset. The figure shows that the IDMBOA-DL model has improved results under each class. For instance, in a normal class, the IDMBOA-DL model has offered an $accu_y$ of 99.22%, $prec_n$ of 99.56%, $reca_l$ of 98.94%, $spec_y$ of 99.53%, and F_{score} of 99.25%. Furthermore, in the denial of service (DOS) class, the IDMBOA-DL method has provided an $accu_y$ of 99.39%, $prec_n$ of 99.20%, $reca_l$ of 99.12%, $spec_y$ of 99.55%, and F_{score} of 99.16%. Meanwhile, For example, in Probe class, the IDMBOA-DL model has offered $accu_y$ of 99.49%, $prec_n$ of 97.00%, $reca_l$ of 97.74%, $spec_y$ of 99.67%, and F_{score} of 97.37%.

Table 2: Result analysis of IDMBOA-DL methodology with dissimilar class labels and measures

Labels	Accuracy	Precision	Recall	Specificity	F-score
Entire dataset					
Normal	99.22	99.56	98.94	99.53	99.25
DoS	99.39	99.20	99.12	99.55	99.16
Probe	99.49	97.00	97.74	99.67	97.37
R2L	99.60	89.07	93.97	99.73	91.45
U2R	99.77	41.37	78.97	99.81	54.30
Average	**99.50**	**85.24**	**93.75**	**99.66**	**88.31**
Training phase (70%)					
Normal	99.24	99.57	98.96	99.54	99.26
DoS	99.41	99.20	99.15	99.55	99.17
Probe	99.51	97.11	97.87	99.69	97.49
R2L	99.59	89.49	93.68	99.74	91.53
U2R	99.77	40.65	76.54	99.81	53.10
Average	**99.50**	**85.20**	**93.24**	**99.66**	**88.11**
Testing phase (30%)					
Normal	99.19	99.55	98.89	99.52	99.22
DoS	99.36	99.20	99.04	99.55	99.12
Probe	99.43	96.72	97.44	99.65	97.08
R2L	99.60	88.06	94.70	99.71	91.26
U2R	99.79	43.06	84.93	99.82	57.14
Average	**99.48**	**85.32**	**95.00**	**99.65**	**88.76**

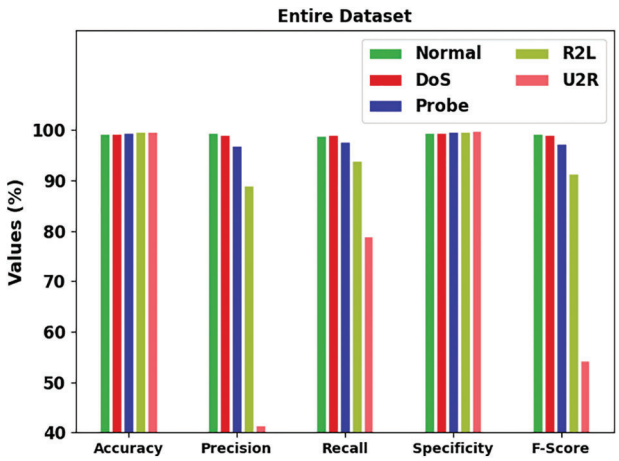

Figure 4: Result analysis of IDMBOA-DL methodology under the whole dataset

Fig. 5 reports a brief intrusion classification outcome of the IDMBOA-DL model with several class labels under 70% of training (TR) data. The figure indicates that the IDMBOA-DL approach has depicted

better outcomes in every class. For example, in a normal class, the IDMBOA-DL technique has given $accu_y$ of 99.24%, $prec_n$ of 99.57%, $reca_l$ of 98.96%, $spec_y$ of 99.54%, and F_{score} of 99.26%. Furthermore, in DOS class, the IDMBOA-DL methodology has offered an $accu_y$ of 99.41%, $prec_n$ of 99.20%, $reca_l$ of 99.15%, $spec_y$ of 99.55%, and F_{score} of 99.17%. Meanwhile, in Probe class, the IDMBOA-DL method has provided $accu_y$ of 99.51%, $prec_n$ of 97.11%, $reca_l$ of 97.87%, $spec_y$ of 99.69%, and F_{score} of 97.49%.

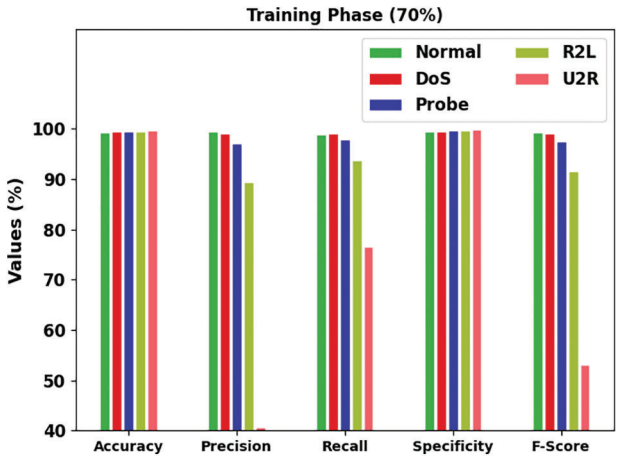

Figure 5: Result analysis of IDMBOA-DL method under 70% of the TR dataset

Fig. 6 reports a brief intrusion classification outcomes of the IDMBOA-DL model with several class labels under 30% of testing (TS) data. The figure shows that the IDMBOA-DL method depicts better outcomes under every class. For example, in a normal class, the IDMBOA-DL technique has provided an $accu_y$ of 99.19%, $prec_n$ of 99.55%, $reca_l$ of 98.89%, $spec_y$ of 99.52%, and F_{score} of 99.22%. Furthermore, in the DOS class, the IDMBOA-DL approach has given $accu_y$ of 99.36%, $prec_n$ of 99.20%, $reca_l$ of 99.04%, $spec_y$ of 99.55%, and F_{score} of 99.12%. Meanwhile, in Probe class, the IDMBOA-DL method has offered an $accu_y$ of 99.43%, $prec_n$ of 96.72%, $reca_l$ of 97.44%, $spec_y$ of 99.65%, and F_{score} of 97.08%.

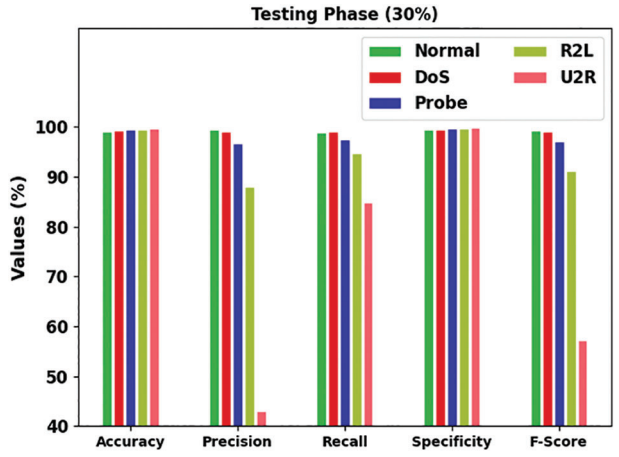

Figure 6: Result analysis of IDMBOA-DL methodology under 30% of the TS dataset

The training accuracy (TA) and validation accuracy (VA) achieved by the IDMBOA-DL approach on the testing dataset is established in Fig. 7. The experimental outcomes implied that the IDMBOA-DL algorithm had accomplished maximal values of TA and VA. In specific, the VA seemed to be greater than TA.

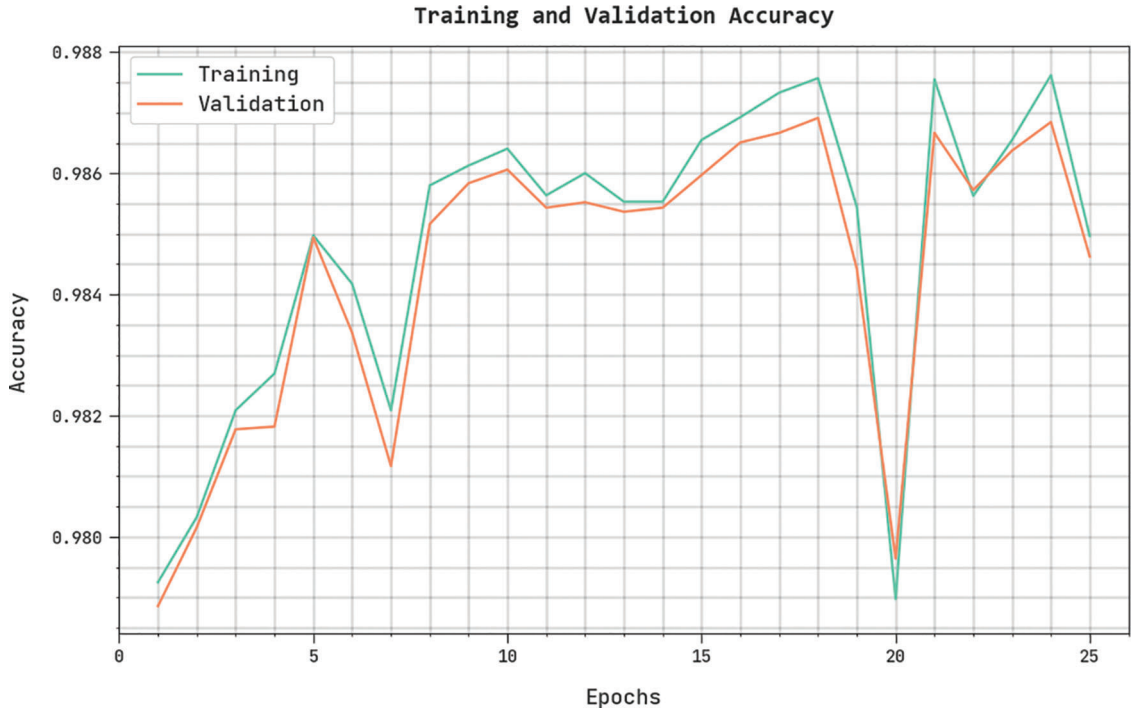

Figure 7: TA and VA analysis of IDMBOA-DL methodology

The training loss (TL) and validation loss (VL) obtained by the IDMBOA-DL method on the testing dataset are illustrated in Fig. 8. The experimental outcomes inferred that the IDMBOA-DL technique had achieved minimum values of TL and VL. Particularly, the VL is lesser than TL.

A clear precision-recall examination of the IDMBOA-DL approach to testing data is depicted in Fig. 9. The figure denoted that the IDMBOA-DL technique has improved precision-recall values under each class.

A brief receiver operating characteristic (ROC) analysis of the IDMBOA-DL approach to testing data is portrayed in Fig. 10. The results indicated the IDMBOA-DL technique had demonstrated its capability in classifying distinct classes on testing data.

A comparative IDS outcome of the IDMBOA-DL model with recent models is made in Table 3. Fig. 11 portrays a detailed $prec_n$ and $reca_l$ validation of the IDMBOA-DL model with existing models. The figure indicated that the IDMBOA-DL model has shown enhanced performance over other models such as DL-based IDS (DL-IDS), D-DL, Deep Feature Embedding Learning with SVM (DFEL SVM), convolutional neural network (CNN), long short-term memory (LSTM), and CNN-LSTM models.

For instance, with respect to $prec_n$, the IDMBOA-DL model has attained a higher $prec_n$ of 85.24%. In contrast, the DL-IDS, D-DL, Deep Feature Embedding Learning with SVM (DFEL SVM), convolutional neural network (CNN), long short-term memory (LSTM), and CNN-LSTM models have obtained reduced $prec_n$ of 84.47%, 80.02%, 79.22%, 78.64%, 78.30%, and 83.50% respectively. Simultaneously, with respect to $reca_l$, the IDMBOA-DL method has accomplished high $reca_l$ of 93.75%, while the DL-IDS, D-DL, DFEL SVM, CNN, LSTM, and CNN-LSTM techniques have achieved minimum $prec_n$ of 88.46%, 86.13%, 89.13%, 89.20%, 90.98%, and 91.55% correspondingly.

Figure 8: TL and VL analysis of IDMBOA-DL methodology

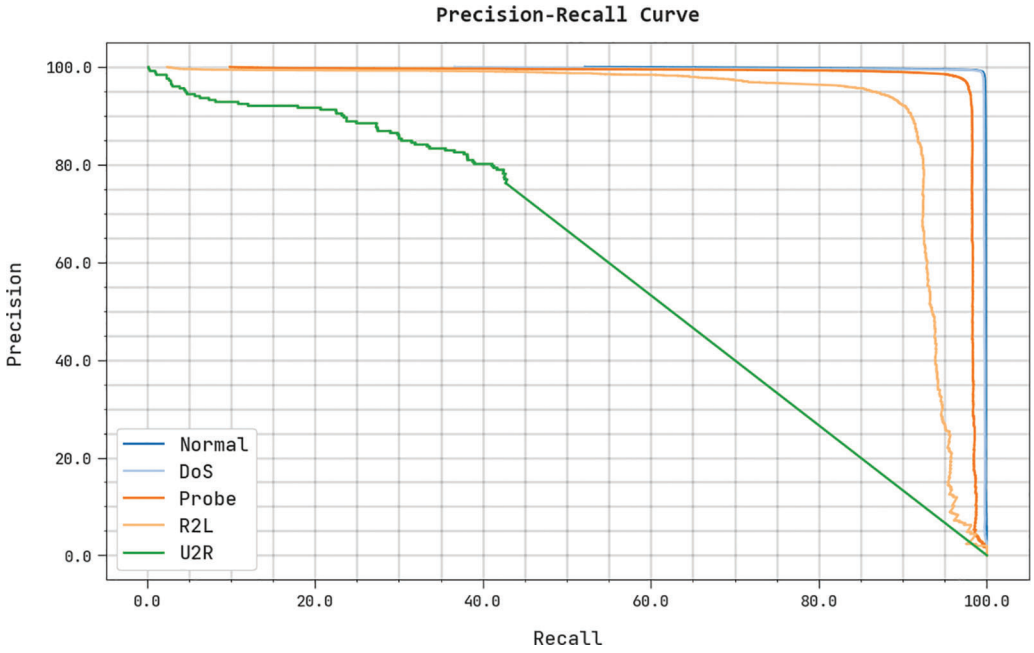

Figure 9: Precision-recall analysis of the IDMBOA-DL method

Fig. 12 demonstrates a detailed $accu_y$ and F_{score} validation of the IDMBOA-DL method with current models. The figure indicates that the IDMBOA-DL approach has improved performance over other techniques. For example, with respect to $accu_y$, the IDMBOA-DL approach has accomplished a maximum $accu_y$ of 99.50% while the DL-IDS, D-DL, DFEL SVM, CNN, LSTM, and CNN-LSTM methodology have attained a minimum $accu_y$ of 98.93%, 98.28%, 98.64%, 98.67%, 97.90%, and 97.68%

correspondingly. Simultaneously, with respect to F_{score}, the IDMBOA-DL method has achieved a maximum F_{score} of 99.66% while the DL-IDS, D-DL, DFEL SVM, CNN, LSTM, and CNN-LSTM techniques have achieved minimum F_{score} of 98.56%, 92.16%, 99.04%, 98.71%, 98.92%, and 99.06% correspondingly.

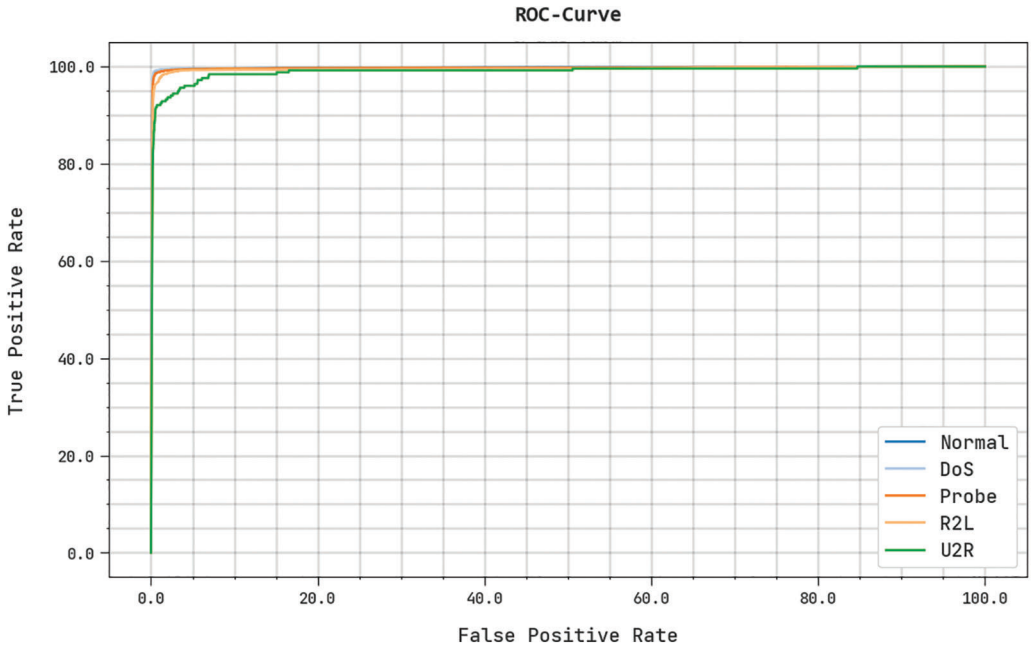

Figure 10: ROC analysis of IDMBOA-DL methodology

Table 3: Comparative analysis of IDMBOA-DL approach with existing methodologies

Methods	Accuracy	Precision	Recall	F1-score
IDMBOA-DL	99.50	85.24	93.75	99.66
DL-IDS	98.93	84.47	88.46	98.56
D-DL	98.28	80.02	86.13	92.16
DFEL SVM	98.64	79.22	89.13	99.04
CNN	96.67	78.64	89.20	98.71
LSTM	97.90	78.30	90.98	98.92
CNN-LSTM	97.68	83.50	91.55	99.06

Therefore, the experimental results reported that the IDMBOA-DL model had accomplished maximum intrusion detection results in an IoT environment.

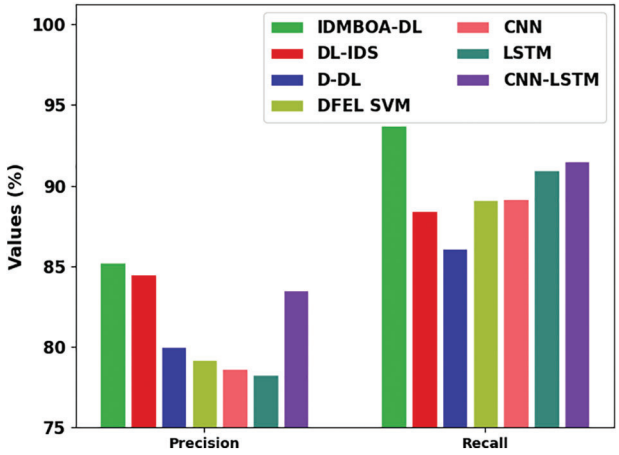

Figure 11: $Prec_n$ and $reca_l$ analysis of IDMBOA-DL approach with existing methodologies

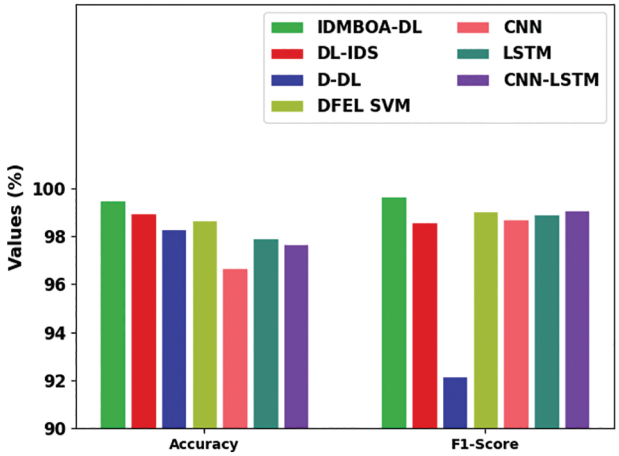

Figure 12: $Accu_y$ and $F1_{score}$ analysis of IDMBOA-DL approach with existing methodologies

4 Conclusion

This study developed a new IDMBOA-DL approach for intrusion detection in the IoT-enabled big data environment. In the presented IDMBOA-DL technique, the Hadoop Mapreduce tool is exploited for managing big data. The MBOA algorithm is applied to derive an optimal subset of features from picking an optimum set of feature subsets. Finally, SCA with the CAE method is utilized to recognize and classify the intrusions in the IoT network. Wide-ranging simulations were conducted to demonstrate the enhanced outcomes of the IDMBOA-DL technique and assessed the outcomes under distinct aspects. The comparison outcomes emphasized the better performance of the IDMBOA-DL algorithm over other approaches. In the future, outlier detection approaches can be derived to enhance detection performance.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] G. Abdelmoumin, D. B. Rawat and A. Rahman, "On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4280–4290, 2022.

[2] S. Zhao, S. Li, L. Qi and L. D. Xu, "Computational intelligence enabled cybersecurity for the internet of things," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 666–674, 2020.

[3] A. K. Bediya and R. Kumar, "A novel intrusion detection system for internet of things network security," *Journal of Information Technology Research*, vol. 14, no. 3, pp. 20–37, 2021.

[4] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan and R. Jain, "Machine learning-based network vulnerability analysis of industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6822–6834, 2019.

[5] A. Khraisat 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. 1, pp. 18, 2021.

[6] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the internet of things," *Sensors*, vol. 19, no. 9, pp. 1977, 2019.

[7] A. Kumari, S. Tanwar, S. Tyagi and N. Kumar, "Verification and validation techniques for streaming big data analytics in internet of things environment," *IET Networks*, vol. 8, no. 3, pp. 155–163, 2019.

[8] S. A. Rahman, H. Tout, C. Talhi and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Network*, vol. 34, no. 6, pp. 310–317, 2020.

[9] J. Asharf, N. Moustafa, H. Khurshid, E. Debie and W. Haider, "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, pp. 1177, 2020.

[10] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider and M. Hamdi, "A survey on security and privacy issues in edge-computing-assisted internet of things," *IEEE Internet of Things Journal*, vol. 8, pp. 4004–4022, 2021.

[11] L. Nie, "Intrusion detection for secure social internet of things based on collaborative edge computing: A generative adversarial network-based approach," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 134–145, 2022.

[12] X. Zhou, Y. Hu, W. Liang, J. Ma and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.

[13] M. A. Basset, V. Chang, H. Hawash, R. K. Chakrabortty and M. Ryan, "Deep-IFS: Intrusion detection approach for industrial internet of things traffic in fog environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 7704–7715, 2020.

[14] I. Idrissi, M. Azizi and O. Moussaoui, "An unsupervised generative adversarial network based-host intrusion detection system for internet of things devices," *Indonesian Journal of Electrical Engineering and Computer Science, Institute of Advanced Engineering and Science*, vol. 25, no. 2, pp. 1140–1150, 2022.

[15] Z. Lv, L. Qiao, J. Li and H. Song, "Deep-learning-enabled security issues in the internet of things," *IEEE Internet Things Journals*, vol. 8, no. 12, pp. 9531–9538, 2021.

[16] L. Nie, W. Sun, S. Wang and Z. Ning, "Intrusion detection in green internet of things: A deep deterministic policy gradient-based algorithm," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 778–788, 2021.

[17] E. A. Mohammed, B. H. Far and C. Naugler, "Applications of the MapReduce programming framework to clinical big data analysis: Current landscape and future trends," *BioData Mining*, vol. 7, no. 1, pp. 22, 2014.

[18] T. Jiang, H. Zhu and G. Deng, "Improved african buffalo optimization algorithm for the green flexible job shop scheduling problem considering energy consumption," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 4, pp. 4573–4589, 2020.

[19] B. Hou and R. Yan, "Convolutional autoencoder model for finger-vein verification," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 2067–2074, 2020.

[20] P. C. Sahu, R. C. Prusty and B. K. Sahoo, "Modified sine cosine algorithm-based fuzzy-aided PID controller for automatic generation control of multiarea power systems," *Soft Computing*, vol. 24, no. 17, pp. 12919–12936, 2020.