Computers, Materials & Continua
DOI: 10.32604/cmc.2023.032972
Article

Tech Science Press

check for updates

IoT-Cloud Assisted Botnet Detection Using Rat Swarm Optimizer with Deep Learning

Saeed Masoud Alshahrani[1], Fatma S. Alrayes[2], Hamed Alqahtani[3], Jaber S. Alzahrani[4], Mohammed Maray[5], Sana Alazwari[6], Mohamed A. Shamseldin[7] and Mesfer Al Duhayyim[8,*]

[1]Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Information Systems, College of Computer Science, Center of Artificial Intelligence, Unit of Cybersecurity, King Khalid University, Abha, Saudi Arabia
[4]Department of Industrial Engineering, College of Engineering at Alqunfudah, Umm Al-Qura University, Saudi Arabia
[5]Department of Information Systems, College of Computer Science, King Khalid University, Abha, Saudi Arabia
[6]Department of Information Technology, College of Computers and Information Technology, Taif University, Taif P.O. Box 11099, Taif, 21944, Saudi Arabia
[7]Department of Mechanical Engineering, Faculty of Engineering and Technology, Future University in Egypt, New Cairo, 11835, Egypt
[8]Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam bin Abdulaziz University, Saudi Arabia
*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa
Received: 03 June 2022; Accepted: 05 July 2022

Abstract: Nowadays, Internet of Things (IoT) has penetrated all facets of human life while on the other hand, IoT devices are heavily prone to cyberattacks. It has become important to develop an accurate system that can detect malicious attacks on IoT environments in order to mitigate security risks. Botnet is one of the dreadful malicious entities that has affected many users for the past few decades. It is challenging to recognize Botnet since it has excellent carrying and hidden capacities. Various approaches have been employed to identify the source of Botnet at earlier stages. Machine Learning (ML) and Deep Learning (DL) techniques are developed based on heavy influence from Botnet detection methodology. In spite of this, it is still a challenging task to detect Botnet at early stages due to low number of features accessible from Botnet dataset. The current study devises IoT with Cloud Assisted Botnet Detection and Classification utilizing Rat Swarm Optimizer with Deep Learning (BDC-RSODL) model. The presented BDC-RSODL model includes a series of processes like pre-processing, feature subset selection, classification, and parameter tuning. Initially, the network data is pre-processed to make it compatible for further processing. Besides, RSO algorithm is exploited for effective selection of subset of features. Additionally, Long Short Term Memory (LSTM) algorithm is utilized for both identification and classification of botnets. Finally, Sine Cosine Algorithm (SCA) is executed for fine-tuning the hyperparameters related to LSTM model. In order to validate the promising

performance of BDC-RSODL system, a comprehensive comparison analysis was conducted. The obtained results confirmed the supremacy of BDC-RSODL model over recent approaches.

Keywords: Internet of things; cloud computing; long short term memory; deep learning; sine cosine algorithm; feature selection

1 Introduction

Internet of Things (IoT) is a paradigm in which billions of intelligent devices are interconnected with each other and are capable of communicating through internet [1]. In recent years, several machines are embedded every day with sensors and there is a tremendous increase can be observed upon communication through internet. As per the paper published in IoT Business News, there is a drastic growth experienced in the number of devices getting interconnected with IoT world. The number is expected to grow up to 24.1 billion by 2030 [2]. ITU Telecommunication Standardization Sector (ITUT) described IoT as a worldwide system that comprises of connected device in accordance to information and communication technology [3]. There is a bigger stream of information exists amidst the connected devices and security is one of the key challenges in IoT [4]. To guarantee the security of IoT network and devices associated with it, appropriate privacy requirements should be met at early stages of design, development and deployment of IoT devices [5,6]. Since IoT model is an emerging phenomenon, it still lacks a strong security mechanism or infrastructure which remains a threat to beneficial data. Current security strategies should be approved for IoT systems to retain the safety of individuals, IoT entities, and organizations [7]. The main security problem in IoT is botnet-based distributed denial of service (DDoS) attacks in which the hacker infects the device with script [8].

At present, Artificial Intelligence (AI) algorithm is utilized for the detection of IoT attacks with high accuracy [9,10]. AI technology has the capability of detecting the variance in methods and channels of attack. This is the major problem confronted by security solutions when it comes to dealing with IoT attacks: attackers introduce slight modifications from the preceding attack which makes the security solution incapable of identifying the threat [11,12]. Researchers and developers utilize AI technology to prevent other risks to the IoT environment by examining system traffic [13,14]. Machine Learning (ML) and Deep Learning (DL) techniques have been converted into security systems to detect the attacks efficiently. DL is an evolution of AI that is contemporary to different real-time applications in order to handle complicated non-linear information.

Waqas et al. [15] inspected cyber security issues from the arrival of distributed denial of service (DDOS), and malware attacks. In this case, various ML techniques such as fuzzy classifier, random forest (RF), support vector machine (SVM), linear regression, Naïve Bayes (NB), decision tree (DT), k-nearest neighbor (KNN), adaptive boosting, tree ensemble, artificial neural network (ANN), and gradient boosting were applied for the recognition of botnet attack. In literature [16], an ensemble learning based approach in IoT (ELBA-IoT) was suggested for botnet attack recognition in IoT networks. This method categorizes the behavioral features of IoT and utilizes ensemble learning for the identification anomalous network traffic in compromised IoT gadgets. Moreover, IoT-related botnet detection technique evaluates three distinct ML methods that belong to DT approaches (RUSBoosted, bagged, and AdaBoosted).

In the study conducted earlier [17], a feature extraction technique was first devised with the help of effective payload from every network packet. Then, a feature selection (FS) technique was projected on the basis of trade-off and by comparing the length of the packets extracted and the

trained performance of models. By selecting a rational sum of packets and a suitable length of bytes as feature vectors, a DL method was projected and assessed for botnet detection. In literature [18], a novel method was introduced to be used in the creation of novel Botnet dataset. This dataset creation is to recognize the anomalous activities in IoT systems. A flow-related Intrusion Detection System (IDS) was tested and examined with the help of flow-related features. Alzahrani et al. [19] suggested a powerful system that is helpful in the detection of botnet attacks on IoT gadgets. This system creatively compiled the convolutional neural network (CNN) with long short term memory (CNN-LSTM) method so as to detect two serious and common IoT assaults (Mirai and BASHLITE) on four different kinds of security cameras.

The current research study devises a model for IoT named Cloud Assisted Botnet Detection and Classification using Rat Swarm Optimizer with Deep Learning (BDC-RSODL) model. The presented BDC-RSODL model involves a series of processes like pre-processing, feature subset selection, classification, and parameter tuning. Primarily, the network data is pre-processed to make it compatible for further processing. Besides, RSO algorithm is exploited to effectively elect a subset of features. Furthermore, LSTM approach is utilized for identification and classification of botnets. Finally, Sine Cosine Algorithm (SCA) is applied to fine tune the hyperparameters related to LSTM model. In order to validate the promising performance of the proposed BDC-RSODL technique, a comprehensive comparison analysis was conducted and the results were discussed under different measures.

2 The Proposed Botnet Detection Model

In this study, an effective BDC-RSODL model has been devised for detection and classification of botnet from IoT cloud environment. The presented BDC-RSODL model includes a series of processes like pre-processing, feature subset selection, classification, and parameter tuning. Initially, the network data is pre-processed to make it compatible for further processing. Besides, RSO algorithm is exploited to effectively elect a subset of features. Moreover, SCA with LSTM model is utilized for identification and classification of botnets. Fig. 1 depicts the overall processes involved in BDC-RSODL algorithm.

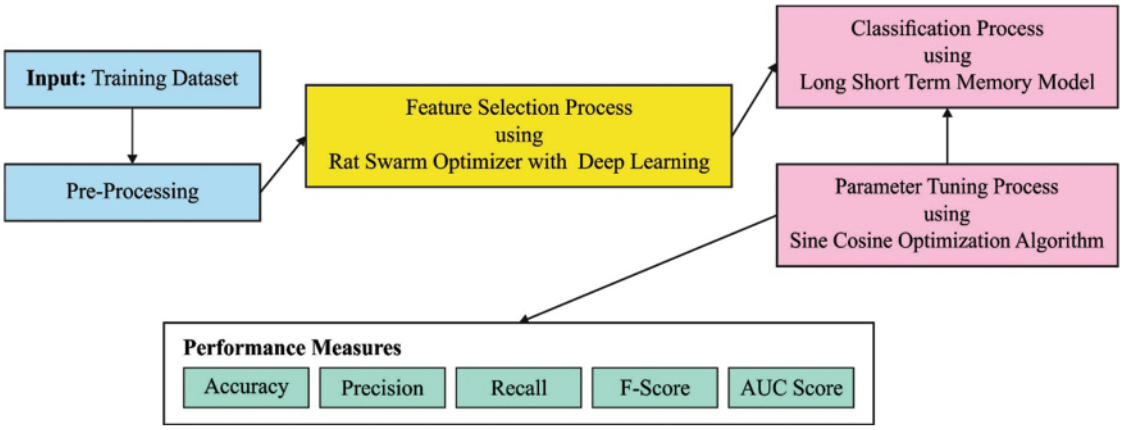

Figure 1: Overall process of BDC-RSODL approach

2.1 Feature Selection Using RSO Algorithm

In this stage, RSO technique is exploited to effectively elect a subset of features. Both chasing and fighting nature of the rats has been arithmetically formulated to develop RSO technique. This technique is used in current study to carry out optimization [20]. Usually, rats are social animals that chase the prey from groups using agonistic nature. In order to describe such behaviours, given that an optimum searching agent is acquaintance of the prey's place.

Furthermore, the searching agents update their position with respect to optimum searching agents obtained until then. The subsequent formulation is used for the above mentioned scenario.

$$\vec{P} = \mathrm{A} \cdot \vec{\mathrm{P}}_i(\mathrm{x}) + \mathrm{C} \cdot \left(\vec{\mathrm{P}}_r(\mathrm{x}) - \vec{\mathrm{P}}_i(\mathrm{x})\right) \tag{1}$$

In Eq. (1), $\vec{\mathrm{P}}_i(\mathrm{x})$ describes the location of the rat and $\vec{\mathrm{P}}_r(\mathrm{x})$ indicates the optimal solution. Hence, A & C parameters are assessed using the formula given below.

$$A = R - x \times \left(\frac{R}{\mathrm{Max}_{Iteration}}\right) \tag{2}$$

If, $x = 0, 1, 2, \cdots, \mathrm{Max}_{Iteration}$

$$C = 2 \cdot rand\,() \tag{3}$$

Thus, R & C denote arbitrary numbers between [1, 5] and [0, 2]. The parameters A & C are accountable for optimum exploitation and exploration at the time of iteration. To arithmetically define the fighting process of rats with prey, Eq. (4) is used.

$$\vec{P}_i(x+1) = \left|\vec{P}_r(x) - \vec{P}\right| \tag{4}$$

Consider $\vec{P}_i(x+1)$ denotes the upgraded position of a rat. It retains the optimum solution and upgrades the location of other searching agents based on the location of the optimum searching agent. The impact of Eqs. (1) and (4) occur in three dimensional environments. Here, the rat's (A, B) position gets upgraded toward the position of prey (A^*, B^*). By adjusting the parameters as presented in Eqs. (2) and (3), numerous quantity of positions are accomplished with respect to existing place. On the other hand, this mechanism is expanded under n-dimensional environments. Then, both exploitation and exploration are ensured by transformed A and C parameter values. The suggested RSO methodology stores the finest solution with the smallest operator. The process followed for existing RSO technique is shown below. The steps and the flowchart of RSO are given herewith.

1. Start the rat population P_i whereas $i = 1, 2, \ldots, n$.
2. Choose the initial variable of RSO, A, C, & R.
3. Now, assess the fitness value of each searching agent.
4. The finest searching agent explores the searching region.
5. Upgrade the location of searching agent using Eq. (4).
6. Check whether the searching agent exceeds the boundary limits of searching region and then regulate it.
7. Also, assess the fitness value of the upgraded searching agent and upgrade the vector P_r later so that there is an optimum solution present compared to early optimal solution.
8. End the procedure when end criteria are satisfied. Otherwise, return to Step 5.
9. Return the obtained optimum solution.

The fitness function of the RSO algorithm assumes classifier accuracy and the amount of chosen features. So, the subsequent FF is utilized in the evaluation of individual solutions as illustrated in Eq. (5).

$$Fitness = \alpha * \ ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \tag{5}$$

Here, ErrorRate represents the classifier error rate that employs the chosen features. ErrorRate is computed as a percentage of incorrect classification to the amount of classifiers made and is formulated as a value between 0 and 1. (ErrorRate refers to the complement of classifier accuracy), $\#SF$ denotes the number of chosen features and $\#All\_F$ signifies the entire amount of attributes from original dataset. α is utilized for controlling the significance of classifier quality and subset length. In this case, α is set to be 0.9.

2.2 Botnet Detection Using LSTM Model

Next, LSTM model is utilized for identification and classification of botnet. Recurrent neural network (RNN) model consists of hidden, output, and input layers. Assume a sequence of length, $x = [x_1, x_2, x_3, \ldots, x_{T-1}, x_T]$, the RNN reads x from x_1 to x_T, and computes the h_t hidden layer and 0_t output by iterating the subsequent equation from $t = 1$ to T

$$h_t = f(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1}), \ t = 1 \ldots T \tag{6}$$

$$0_t = f(W_{hx} \cdot h_t), \ t = 1 \ldots T \tag{7}$$

Let $W_{xh} \in \mathbb{R}^{m \times d}$, $W_{hh} \in \mathbb{R}^{m \times m}$ and $W_{hx} \in \mathbb{R}^{m \times n}$ be the weights of input-hidden, hidden-hidden, and hidden-output whereas the sizes of input, hidden and output are represented by d, m and n respectively. $f(\cdot)$ denotes the activation function that operates on every element. In general, the final output 0_T is applied as a *softmax* classification feature to predict the labels, since it comprises of data for the entire data sequence, x. Owing to the outstanding modelling of data sequence, RNN accomplishes innovative performance on different tasks. LSTM network has three gates such as forget, input, and output gates, besides the cell states [21]. Fig. 2 illustrates the infrastructure of LSTM. An input at present time step is integrated with data on Hidden Layer (HL) from the preceding time step, and is passed onto next step with activation function for input gate as follows.

$$f_t = o(X_t W^f + S_{t-1} U^f + b_f) \tag{8}$$

whereas f_t represents the forget gate. X_t signifies the input at time step t and S_{t-1} implies the HL at preceding time step, $t - 1$. W^f denotes the weight of input layers and U^f represents the recurrent weight of HLs. Here, b_f stands for bias of the input layer. An input gate resolves that data is saved from cell state. Input gate performs two tasks i.e., during the primary task, the input gate layer resolves the upgraded value. But, during the secondary task, *tanh* layer adjusts the network by generating the vector of every novel candidate value. The formulas for the two tasks are as follows.

$$i_t = o(X_t W^i + S_{t-1} U^i + b_j) \tag{9}$$

$$\widetilde{C}_t = \tanh(X_t W^c + S_{t-1} U^c + b_c) \tag{10}$$

$$C_t = C_{t-1} * f_t + i_t * \widetilde{C}_t \tag{11}$$

The resultant gate selects the HLs that are utilized to predict, using a sigmoid activation function. A novel altered cell state is distributed to tanh function and is multiplied to attain the outcome, as follows:

$$O_t = o(X_t W^o + S_{t-1} U^o + b_o) \tag{12}$$

$$S_t = O_t * tanh\,(C_t) \tag{13}$$

The weights are demonstrated by two matrixes, $\{W,\ U\}$. The matrix $W = \{W^i,\ W^c,\ W^o\}$ denotes the input weights connected to input, HL, and output layers correspondingly, but $U = \{U^i,\ U^c,\ U^o\}$ represents the weight of preceding HL i.e., recurrent weight from input layer, HL, and output layer. The bias vectors are gathered and forms as a matrix $B = \{b^i,\ b^c,\ b^o\}$, with the indices equivalent to similar layers.

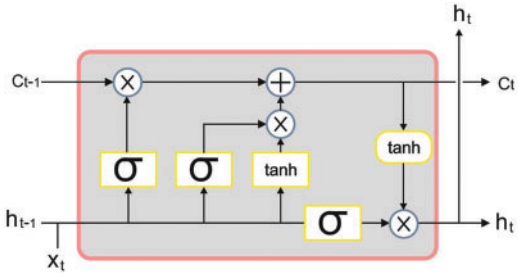

Figure 2: Framework of LSTM

2.3 Hyperparameter Tuning Using SCA

Finally, SCA is applied to fine tune the hyperparameters [22–24] related to LSTM model. The advantage of the proposed SCA algorithm is that one can achieve clear results and use simple parameters [25]. The location update is defined by cosine or sine function as given below.

$$X_{t+1}(i,j) = \begin{cases} X_t(i,j) + r_1(t) sinr_2 \left| r_3 X_t^{best}(i,j) - X_t(i,j) \right|, & r_4 < 0.5 \\ X_t(i,j) + r_1(t) cosr_2 \left| r_3 X_t^{best}(i,j) - X_t(i,j) \right|, & r_4 \geq 0.5 \end{cases} \tag{14}$$

where

$X_{t+1}(i,j)$–refers to i^{th} individual position in $t+1$ round of dimension;

j, $X_t(i,j)$-signifies the i^{th} individual position in t round of j dimension,

$X_t^{besi}(i,j)$—embodies the location of global optimum solution in the preceding t round;

$r_1(t)$-signifies the amplitude factor;

$r_1(t) = \alpha\left(1 - \frac{t}{T}\right)$, $\alpha = 2, r_2 \in [0, 2\pi]$, $r_3 \in [-2, 2]$, $r_4 \in [0, 1]$ and r_2, r_3, r_4 variables are uniform distribution arbitrary integers. The variable r_1 defines the moving direction of $X_{t+1}(i,j)$. This direction is either between the space of $X_t(i,j)$ and $X_t^{besi}(i,j)$ or outside it. Furthermore, r_1 determines the exploration and exploitation of the updating method. Here, r_2 determines how much $X_t(i,j)$ move towards or away from $X_t^{best}(i,j)$. The r_3 variable describes the degree of influence of optimum solution $X_t^{besi}(i,j)$ on the existing solution $X_t(i,j)$. Here, $r3 > 1$ stochastically denotes that the degree influence of X and Y must be weakened, or else it must be strengthened. The r_4 variable controls the switch between sine and cosine transforms.

3 Results and Discussion

The current section assesses the botnet classification outcomes of the proposed BDC-RSODL model using a dataset that is composed of 20,689 samples under two classes as depicted in Tab. 1.

Table 1: Dataset details

Class label	Description	No. of samples
Class 0	Botnet	2554
Class 1	Normal	18135
Total number of samples		20689

Fig. 3 reports the set of confusion matrices generated by BDC-RSODL model on distinct test runs. On run-1, the proposed BDC-RSODL model categorized 2,530 samples under class 0 and 17,959 samples under class 1. Simultaneously, on run-3, BDC-RSODL approach recognized 2,533 samples as class 0 and 17,958 samples as class 1. Concurrently, on run-5, the proposed BDC-RSODL system classified 2,535 samples under class 0 and 17,961 samples under class 1.

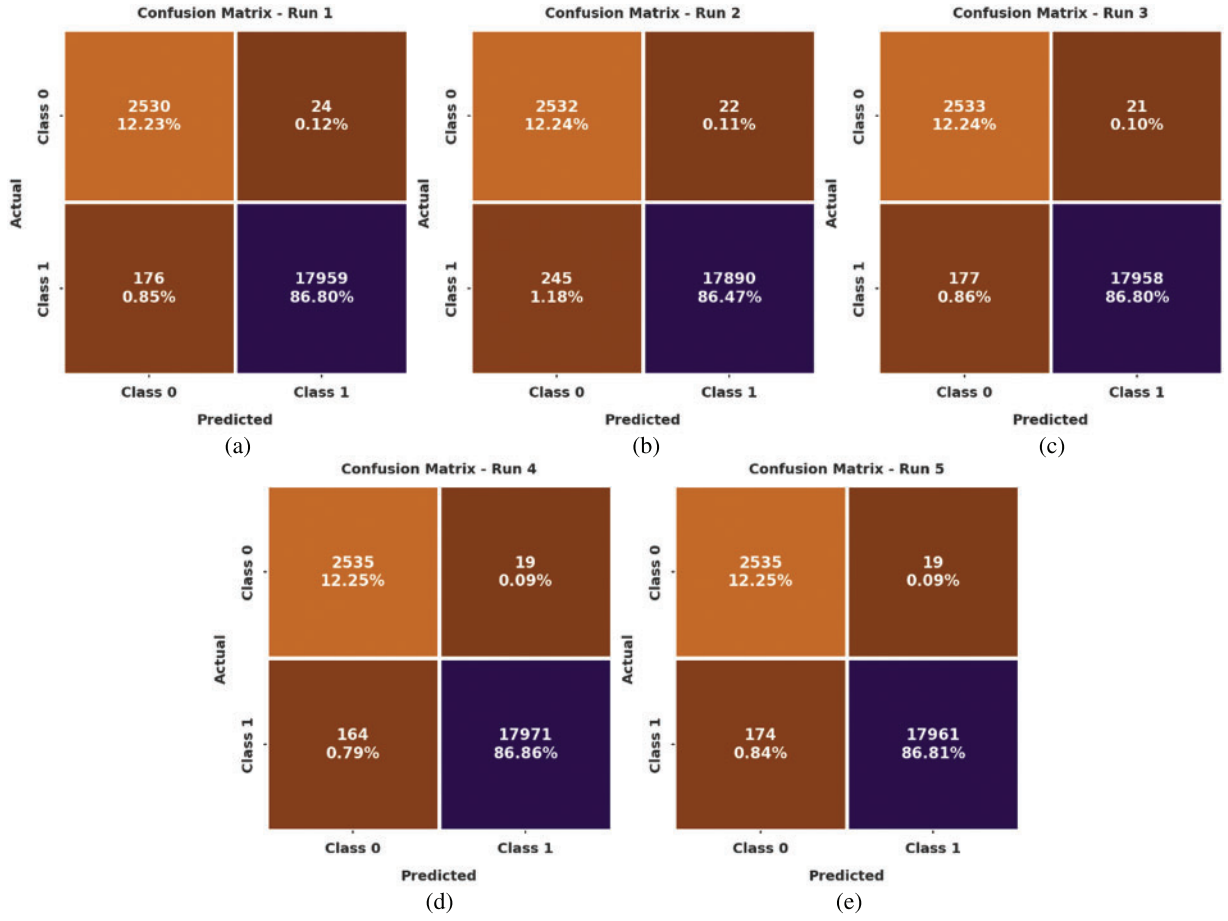

Figure 3: Confusion matrices of BDC-RSODL approach (a) Run-1, (b) Run-2, (c) Run-3, (d) Run-4, and (d) Run-5

Tab. 2 provides an overview of botnet classification performance achieved by BDC-RSODL model under distinct runs of execution.

Table 2: Results of the analysis of BDC-RSODL approach under distinct measures and runs

Labels	Accuracy	Precision	Recall	F-Score	AUC score
			Run-1		
Class 0	99.03	93.50	99.06	96.20	99.04
Class 1	99.03	99.87	99.03	99.45	99.04
Average	99.03	96.68	99.04	97.82	99.04
			Run-2		
Class 0	98.71	91.18	99.14	94.99	98.89
Class 1	98.71	99.88	98.65	99.26	98.89
Average	98.71	95.53	98.89	97.13	98.89
			Run-3		
Class 0	99.04	93.47	99.18	96.24	99.10
Class 1	99.04	99.88	99.02	99.45	99.10
Average	99.04	96.68	99.10	97.85	99.10
			Run-4		
Class 0	99.12	93.92	99.26	96.52	99.18
Class 1	99.12	99.89	99.10	99.49	99.18
Average	99.12	96.91	99.18	98.00	99.18
			Run-5		
Class 0	99.07	93.58	99.26	96.33	99.15
Class 1	99.07	99.89	99.04	99.47	99.15
Average	99.07	96.74	99.15	97.90	99.15

Fig. 4 shows a brief results of the analysis attained by BDC-RSODL model in terms of $prec_n$ and $reca_l$. On run-1, the proposed BDC-RSODL model achieved average $prec_n$ and $reca_l$ values such as 96.68% and 99.04% respectively. In line with this, on run-3, BDC-RSODL system obtained average $prec_n$ and $reca_l$ values such as 96.68% and 99.10% correspondingly. Moreover, on run-5, the proposed BDC-RSODL technique offered average $prec_n$ and $reca_l$ values such as 96.74% and 99.15% correspondingly.

Fig. 5 details about the analysis results accomplished by BDC-RSODL approach with respect to F_{score} and AUC_{score}. On run-1, the proposed BDC-RSODL methodology offered average F_{score} and AUC_{score} values such as 97.82% and 99.04% correspondingly. Likewise, on run-3, the proposed

BDC-RSODL approach achieved average F_{score} and AUC_{score} values such as 97.85% and 99.10% correspondingly. Eventually, on run-5, BDC-RSODL system obtained average F_{score} and AUC_{score} values such as 97.90% and 99.15% correspondingly.

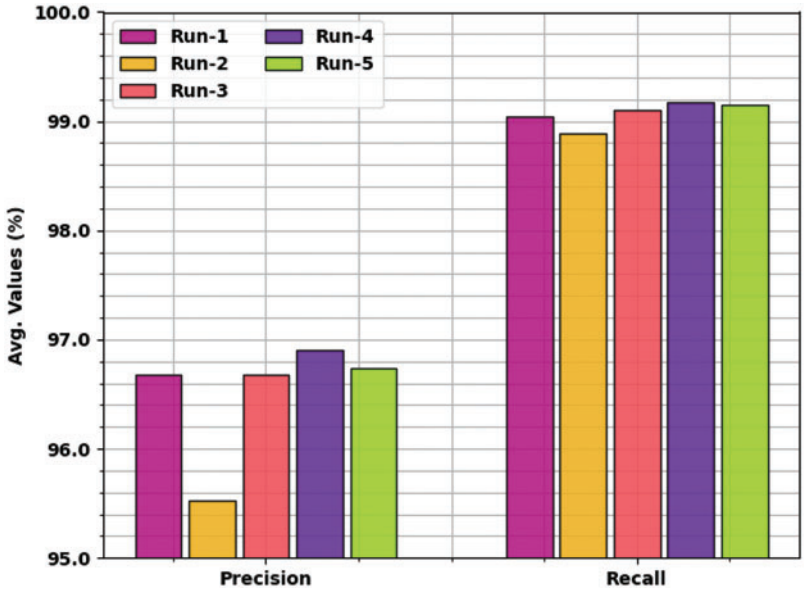

Figure 4: $Prec_n$ and $reca_l$ analysis results of BDC-RSODL approach under distinct runs

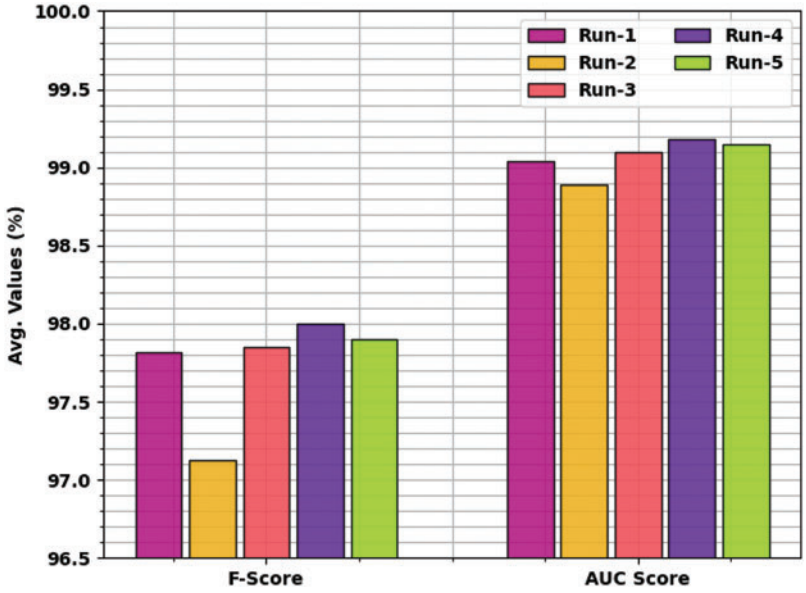

Figure 5: F_{score} and AUC_{score} analysis results of BDC-RSODL approach under distinct number of runs

Training Accuracy (TA) and Validation Accuracy (VA) values, achieved by the proposed BDC-RSODL system on test dataset, are demonstrated in Fig. 6. The experimental outcomes expose that the proposed BDC-RSODL technique gained maximal TA and VA values. To be specific, VA performed higher than TA.

Figure 6: TA and VA analysis results of BDC-RSODL methodology

Training Loss (TL) and Validation Loss (VL) values, attained by BDC-RSODL approach on test dataset, are portrayed in Fig. 7. The experimental outcomes reveal that the proposed BDC-RSODL algorithm achieved minimal TL and VL values. To be specific, VL is lesser than TL.

Figure 7: TL and VL analysis results of BDC-RSODL methodology

A clear precision-recall examination was conducted upon BDC-RSODL approach using test dataset and the results are shown in Fig. 8. The figure reveals that the proposed BDC-RSODL algorithm achieved improved precision-recall values under all classes.

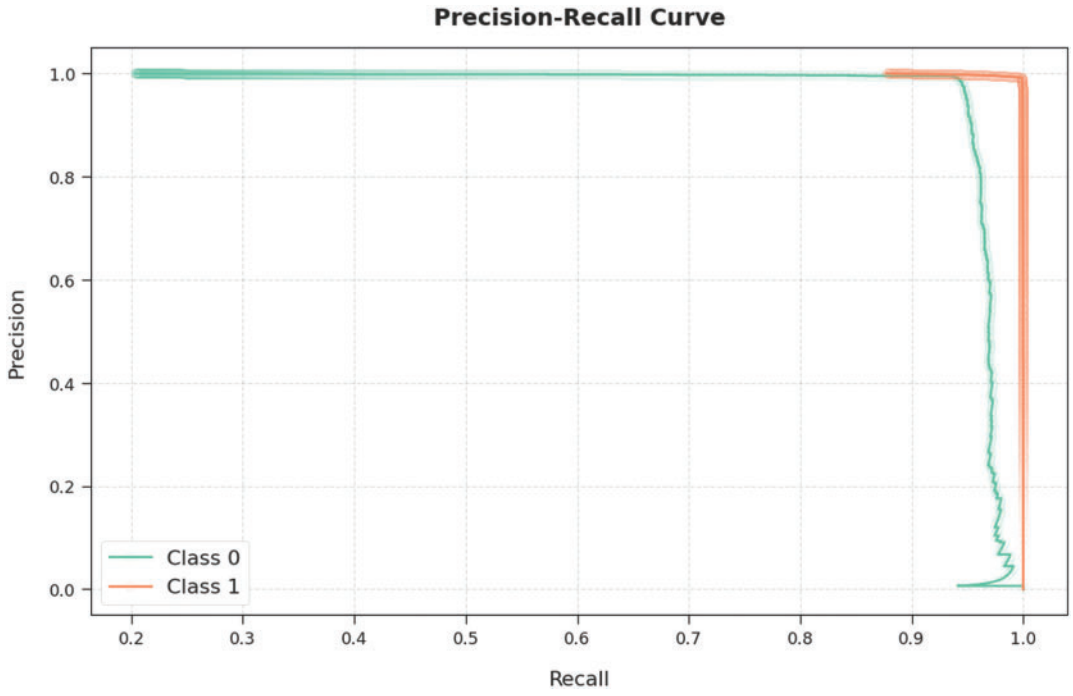

Figure 8: Precision-recall curve analysis results of BDC-RSODL methodology

Tab. 3 shows the comparative analysis results achieved by BDC-RSODL model in terms of botnet classification performance [26]. Fig. 9 shows the detailed $accu_y$ inspection results attained by BDC-RSODL system and other existing models. The figure signifies that the host-based system achieved the least $accu_y$ of 92.90%. Also, peer to peer botnet detection system (P2P-BDS) and MTC-CNN algorithms demonstrated somewhat increased $accu_y$ values such as 94.50% and 95% respectively. Moreover, DT and fuzzy logic based artificial neural network (FL-ANN) models showcased reasonable $accu_y$ values such as 97.90% and 98.94% respectively. But the proposed BDC-RSODL model accomplished the highest $accu_y$ of 99.12%.

Table 3: Comparative analysis results of BDC-RSODL approach and other existing approaches

Methods	Accuracy	Precision	Recall	F-Score
BDC-RSODL model	99.12	96.91	99.18	98.00
P2P-BDS	94.50	95.61	96.67	94.66
MTC-CNN	95.00	95.87	97.77	96.16
Decision tree	97.90	94.94	95.95	95.65
Host-based model	92.90	95.34	96.84	96.59
FL-ANN	98.94	96.29	97.87	97.08

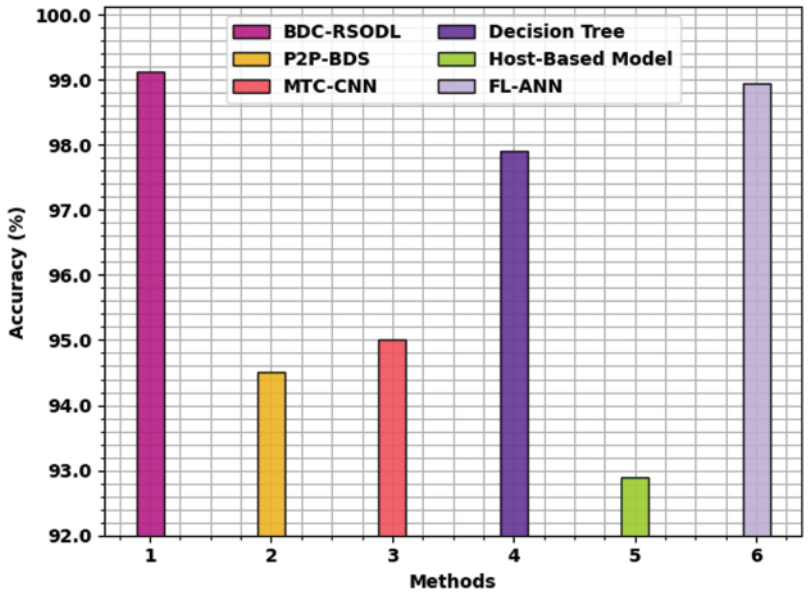

Figure 9: $Accu_y$ analysis results of BDC-RSODL approach and other existing methodologies

Fig. 10 is a brief portrayal of $prec_n$ analysis results achieved by BDC-RSODL approach and other existing models. The figure represents that DT methodology achieved minimal $prec_n$ of 94.94%. Simultaneously, P2P-BDS and host-based systems demonstrated somewhat superior $prec_n$ values such as 95.61% and 95.34% correspondingly. Followed by, MTC-CNN and FL-ANN models showcased reasonable $prec_n$ values such as 95.87% and 96.29% correspondingly. However, the proposed BDC-RSODL model accomplished a reasonable $prec_n$ of 96.91%.

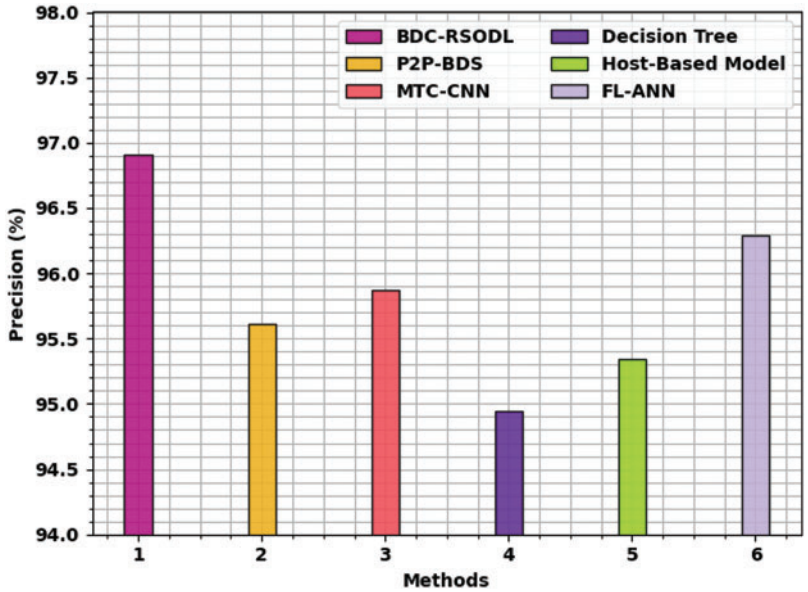

Figure 10: $Prec_n$ analysis results of BDC-RSODL approach and other recent methodologies

Fig. 11 depicts the detailed $reca_l$ investigation outcomes of BDC-RSODL system and other existing approaches. The figure implies that DT system exhibited the least $reca_l$ of 95.95%. Along with that, P2P-BDS and host-based models demonstrated somewhat maximal $reca_l$ values such as 96.67% and 96.84% correspondingly. Furthermore, MTC-CNN and FL-ANN models showcased reasonable $reca_l$ values such as 97.77% and 97.87% correspondingly. Eventually, the proposed BDC-RSODL method accomplished the highest $reca_l$ value of 99.18%.

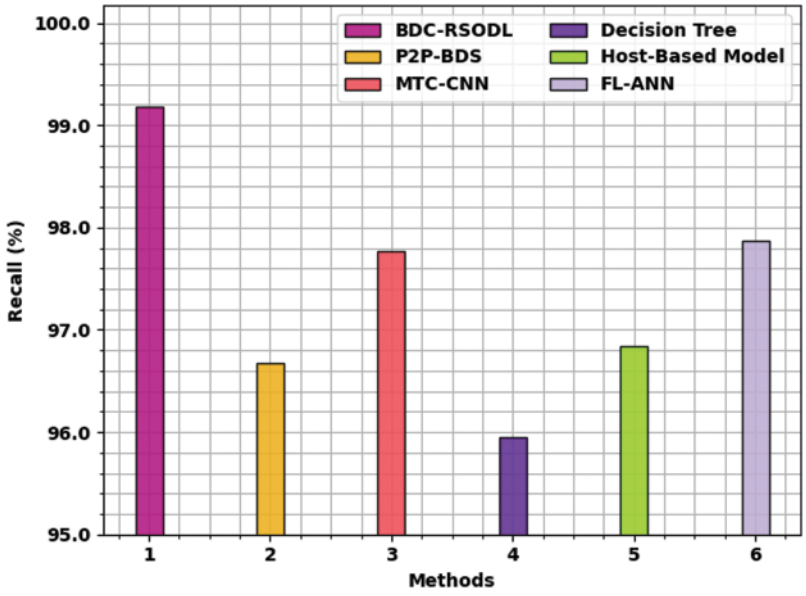

Figure 11: *Reca*$_l$ analysis results of BDC-RSODL approach and other existing methodologies

Fig. 12 is a brief portrays of F_{score} analysis results achieved by the proposed BDC-RSODL system and other existing methodologies. The figure expose that P2P-BDS approach achieved a minimal F_{score} of 94.66%. Similarly, DT and MTC-CNN methods demonstrated somewhat superior F_{score} values such as 95.65% and 96.16% respectively. Besides, the host-based and FL-ANN models depicted reasonable F_{score} values such as 96.59% and 97.08% correspondingly. At last, the proposed BDC-RSODL algorithm accomplished the maximum F_{score} of 98%. Therefore, it is confirmed that the proposed BDC-RSODL approach is effective in botnet classification compared to other existing models.

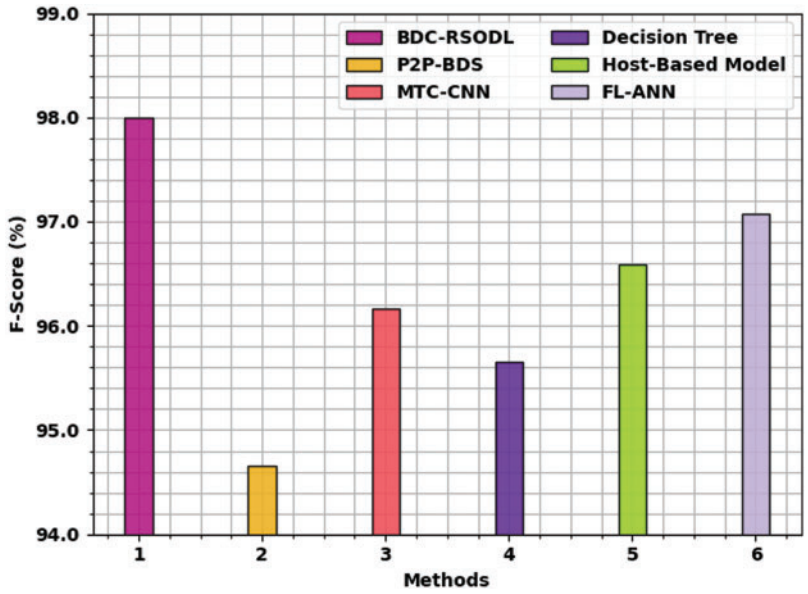

Figure 12: F_{score} analysis results of BDC-RSODL approach and other existing methodologies

4 Conclusion

In this study, an effective BDC-RSODL model has been devised for detection and classification of botnet from IoT cloud environment. The presented BDC-RSODL model includes a series of processes like pre-processing, feature subset selection, classification, and parameter tuning. Initially, the network data is pre-processed to make it compatible for further processing. Besides, RSO algorithm is exploited to effectively elect a subset of features. Moreover, LSTM model is utilized for identification and classification of botnets. Finally, SCA is applied to fine tune the hyperparameters related to LSTM model. In order to validate the promising performance of the proposed BDC-RSODL system, a comprehensive comparative analysis was conducted. The results obtained confirmed the supremacy of the proposed BDC-RSODL model over recent approaches. In future, the performance of the proposed model can be improved by feature reduction models.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (61/43). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R319), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4340237DSR27). The author would like to thank the Deanship of Scientific Research at Shaqra University for supporting this work.

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

References

[1] X. Dong, J. Hu and Y. Cui, "Overview of botnet detection based on machine learning," in *2018 3rd Int. Conf. on Mechanical, Control and Computer Engineering (ICMCCE)*, Huhhot, pp. 476–479, 2018.

[2] W. Ahmad, A. Rasool, A. R. Javed, T. Baker and Z. Jalil, "Cyber security in IoT-based cloud computing: A comprehensive survey," *Electronics*, vol. 11, no. 1, pp. 16, 2021.

[3] V. Kanimozhi and T. Jacob, "Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset cse-cic-ids2018 using cloud computing," *ICT Express*, vol. 5, no. 3, pp. 211–214, 2019.

[4] S. A. Sokolov, T. B. Iliev and I. S. Stoyanov, "Analysis of cybersecurity threats in cloud applications using deep learning techniques," in *2019 42nd Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, pp. 441–446, 2019.

[5] L. Seungjin, A. Abdullah and N. Jhanjhi, "A review on honeypot-based botnet detection models for smart factory," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 418–435, 2020.

[6] A. M. Hilal, H. Alsolai, F. N. Al-Wesabi, M. K. Nour, A. Motwakel *et al.,* "Fuzzy cognitive maps with bird swarm intelligence optimization-based remote sensing image classification," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–12, 2022.

[7] T. Tuan, H. Long, L. Son, R. Kumar, I. Priyadarshini *et al.,* "Performance evaluation of botnet DDoS attack detection using machine learning," *Evolutionary Intelligence*, vol. 13, no. 2, pp. 283–294, 2019.

[8] I. Abunadi, M. M. Althobaiti, F. N. Al-Wesabi, A. M. Hilal, M. Medani *et al.,* "Federated learning with blockchain assisted image classification for clustered UAV networks," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1195–1212, 2022.

[9] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri and L. Cheng, "Internet of things botnet detection approaches: Analysis and recommendations for future research," *Applied Sciences*, vol. 11, no. 12, pp. 5713, 2021.

[10] M. A. Alohali, F. N. Al-Wesabi, A. M. Hilal, S. Goel, D. Gupta *et al.,* "Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment," *Cognitive Neurodynamics*, 2022, https://doi.org/10.1007/s11571-022-09780-8.

[11] N. Koroniotis, N. Moustafa, E. Sitnikova and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[12] A. M. Hilal, M. A. Alohali, F. N. Al-Wesabi, N. Nemri, J. Hasan *et al.,* "Enhancing quality of experience in mobile edge computing using deep learning based data offloading and cyberattack detection technique," *Cluster Computing*, 2021. https://doi.org/10.1007/s10586-021-03401-5.

[13] M. Roopak, G. Y. Tian and J. Chambers, "Deep learning models for cyber security in IoT networks," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conf. (CCWC)*, Las Vegas, NV, USA, pp. 0452–0457, 2019.

[14] B. Padmavathi and B. Muthukumar, "An efficient botnet detection approach based on feature learning and classification," *Journal of Control and Decision*, pp. 1–14, 2022, https://doi.org/10.1080/23307706.2022.2077246.

[15] M. Waqas, K. Kumar, A. A. Laghari, U. Saeed, M. M. Rind *et al.,* "Botnet attack detection in internet of things devices over cloud environment via machine learning," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, pp. e6662, 2022.

[16] Q. A. A. Haija and M. A. Dala'ien, "ELBA-IoT: An ensemble learning model for botnet attack detection in IoT networks," *Journal of Sensor and Actuator Networks*, vol. 11, no. 1, pp. 18, 2022.

[17] C. Li, Y. Zhang, W. Wang, Z. Liao and F. Feng, "Botnet detection with deep neural networks using feature fusion," in *2022 Int. Seminar on Computer Science and Engineering Technology (SCSET)*, Indianapolis, IN, USA, pp. 255–258, 2022.

[18] I. Ullah and Q. H. Mahmoud, "A technique for generating a botnet dataset for anomalous activity detection in IoT networks," in *2020 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, Toronto, ON, Canada, pp. 134–140, 2020.
[19] M. Y. Alzahrani and A. M. Bamhdi, "Hybrid deep-learning model to detect botnet attacks over internet of things environments," *Soft Computing*, 2022, https://doi.org/10.1007/s00500-022-06750-4.
[20] G. Dhiman, M. Garg, A. Nagar, V. Kumar and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 8, pp. 8457–8482, 2021.
[21] R. A. Shabandar, A. Jaddoa, P. Liatsis and A. J. Hussain, "A deep gated recurrent neural network for petroleum production forecasting," *Machine Learning with Applications*, vol. 3, pp. 100013, 2021.
[22] A. Muthumari, J. Banumathi, S. Rajasekaran, P. Vijayakarthik, K. Shankar *et al.,* "High security for de-duplicated big data using optimal simon cipher," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 1863–1879, 2021.
[23] G. N. Nguyen, N. H. L. Viet, M. Elhoseny, K. Shankar, B. B. Gupta *et al.,* "Secure blockchain enabled cyber-physical systems in healthcare using deep belief network with ResNet model," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 150–160, 2021.
[24] M. Elhoseny, M. M. Selim and K. Shankar, "Optimal deep learning based convolution neural network for digital forensics face sketch synthesis in internet of things (IoT)," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3249–3260, 2021.
[25] M. A. Elaziz, D. Oliva and S. Xiong, "An improved opposition-based sine cosine algorithm for global optimization," *Expert Systems with Applications*, vol. 90, pp. 484–500, 2017.
[26] C. Joshi, R. Ranjan and V. Bharti, "A fuzzy logic based feature engineering approach for botnet detection using ANN," *Journal of King Saud University-Computer and Information Sciences*, 2021, https://doi.org/10.1016/j.jksuci.2021.06.018.