IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 September 2023, accepted 10 November 2023, date of publication 14 November 2023,
date of current version 27 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332690

RESEARCH ARTICLE

Modeling of Botnet Detection Using Chaotic Binary Pelican Optimization Algorithm With Deep Learning on Internet of Things Environment

FADWA ALROWAIS[1], **MAJDY M. ELTAHIR**[2], **SUMAYH S. ALJAMEEL**[3], **RADWA MARZOUK**[4], **GOUSE PASHA MOHAMMED**[5], **AND AHMED S. SALAMA**[6]

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[2]Department of Information Systems, College of Science & Art at Mahayil, King Khalid University, Abha 62529, Saudi Arabia
[3]SAUDI ARAMCO Cybersecurity Chair, Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman bin Faisal University, Dammam 31441, Saudi Arabia
[4]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[5]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
[6]Department of Electrical Engineering, Faculty of Engineering & Technology, Future University in Egypt, New Cairo 11845, Egypt

Corresponding author: Gouse Pasha Mohammed (g.mohammed@psau.edu.sa)

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number (RGP2/ 29/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R77), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. We Would like to thank SAUDI ARAMCO Cybersecurity Chair for funding this project. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444). This study is partially funded by the Future University in Egypt (FUE).

ABSTRACT Nowadays, there are ample amounts of Internet of Things (IoT) devices interconnected to the networks, and with technological improvement, cyberattacks and security threads, for example, botnets, are rapidly evolving and emerging with high-risk attacks. A botnet is a network of compromised devices that are controlled by cyber attackers, frequently employed to perform different cyberattacks. Such attack disrupts IoT evolution by disrupting services and networks for IoT devices. Detecting botnets in an IoT environment includes finding abnormal patterns or behaviors that might indicate the existence of these malicious networks. Several researchers have proposed deep learning (DL) and machine learning (ML) approaches for identifying and categorizing botnet attacks in the IoT platform. Therefore, this study introduces a Botnet Detection using the Chaotic Binary Pelican Optimization Algorithm with Deep Learning (BNT-CBPOADL) technique in the IoT environment. The main aim of the BNT-CBPOADL method lies in the correct detection and categorization of botnet attacks in the IoT environment. In the BNT-CBPOADL method, Z-score normalization is applied for pre-processing. Besides, the CBPOA technique is derived for feature selection. The convolutional variational autoencoder (CVAE) method is applied for botnet detection. At last, the arithmetical optimization algorithm (AOA) is employed for the optimal hyperparameter tuning of the CVAE algorithm. The experimental valuation of the BNT-CBPOADL technique is tested on a Bot-IoT database. The experimentation outcomes inferred the supremacy of the BNT-CBPOADL method over other existing techniques with maximum accuracy of 99.50%.

INDEX TERMS Internet of Things, privacy, security, botnet detection, metaheuristics, deep learning.

I. INTRODUCTION

The proliferation of Internet of Things (IoT) devices has led to continuous growth in the volume of IoT-based attacks [1]. The IoT botnet attack is one of the most serious IoT threats that attempts to commit profitable, actual, and effective cybercrimes. IoT botnet is a group of Internet-connected IoT devices that are remotely controlled by an attacker and have been infected with malware [2]. IoT systems have major problems in providing approaches to identify security attacks

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

and vulnerabilities due to the fast development of attacks and a variety of attack strategies. There are numerous security attacks targeting the IoT that have several vulnerabilities [3]. Since the IoT is exposed to various attacks, it is significant for classifying the appropriate attacks and vulnerabilities to analyse the IoT [4]. Any studies classify these attacks depending on the IoT layers, attacks themselves, and vulnerabilities that can result in the attack [5]. The study found that routing, man in the middle, virus, worm, DoS, jamming, flooding, sinkhole, and wormhole attacks are the most probably to take place in an IoT platform. Especially, DoS, flooding and attacks arise in the creation of IoT platforms by botnets [6].

Botnet attacks are particularly rising in popularity. A botnet runs a bot on various devices inter-connected with the internet to create a botnet measured by the command and control (C&C) [7]. The botnet causes different kinds of damage namely, resource depletion and service interruption. Now, AI is extensively employed for identifying these IoT attacks. Presently, the IoT is attacked by different channels and techniques [8]. However, it is hard to introduce security solutions and resolve occurrences of hacking on the IoT through an analysis of security attacks utilizing network data. There are several current security problems relevant to the IoT that can raise knowledge of these issues [4]. Nowadays, research on IoT security attacks has focussed on responding and analysing networks however, it has been constraints that prevent identifying direct modifications in hardware. As malware is implemented, it has an improving quantity of developments in deep learning (DL) or machine learning (ML) based identification methods, which utilize time series data [9]. However, the requirement to use time series data seriously parameters present the effort's effectiveness. Alternatively, early detection can allow improved IoT Botnet response suggestions. Therefore, it decreases the damage because of possible attacks. The dynamic analysis technique inspects how malware communicates with its environments while it is being implemented.

This study introduces a Botnet Detection using a Chaotic Binary Pelican Optimization Algorithm with Deep Learning (BNT-CBPOADL) algorithm in the IoT platform. In the BNT-CBPOADL method, Z-score normalization is applied for pre-processing. Besides, the CBPOA technique is derived for feature selection. The convolutional variational autoencoder (CVAE) model is applied for botnet detection. At last, the arithmetical optimization algorithm (AOA) is employed for the optimal hyperparameter tuning of the CVAE algorithm. The experimental valuation of the BNT-CBPOADL method is tested on a benchmark database. The key contributions of the paper is summarized as follows.

- The BNT-CBPOADL algorithm represents a novel and effective method for detecting botnets in IoT environments. It leverages a combination of preprocessing, CBPOA based feature selection, CVAE classification, and AOA based hyperparameter tuning techniques to improve the accuracy and efficiency of botnet detection.
- The design of the CBPOA for feature selection ensuring that relevant features are selected for analysis and enhancing the detection results
- The employment of the AOA for hyperparameter tuning of the CVAE model is a unique and valuable contribution, fine-tuning the model's parameters to optimize botnet detection performance.

The rest of the paper is organized as follows. Section II provides the related works and section III offers the proposed model. Then, section IV gives the result analysis and section V concludes the paper.

II. RELATED WORKS

Wei et al. [10] suggested a lightweight and generic NIDS with the 2-phase technique for detecting botnet activity on the IoT networks, only employing available packet-length features. According to these features, the authors develop a method that depends on an AE to filter out a massive count of traffic flow in the primary phase. Later, the authors introduced a new method to convert the packet length series into 3-channel RGB images for the classification of malicious traffic depending on lightweight CNNs. Alshahrani et al. [11] developed an IoT with Cloud Aided Botnet Detection and Classification using Rat Swarm Optimization with DL (BDC-RSODL) approach. Primarily, the network information was preprocessed to make it compatible for more processing. Also, the RSO method can be used for efficiently selecting the feature subsets. Moreover, the LSTM method has been used for both detecting and classifying botnets. Lastly, SCA is implemented for adjusting the hyper-parameters.

Alrayes et al. [12] introduced a botnet identification method employing the barnacles mating optimization with ML (BND-BMOML) for the IoT platform. The study considers the recognition and detection of botnets in IoT platforms. For botnet identification, this BND-BMOML method in such analysis utilizes an ENN algorithm. Eventually, this introduced BND-BMOML technique employs a chicken swarm optimization (CSO) approach for the parameters tuning method, establishing the innovation of the effort. Dong et al. [13] suggested a botnet identification technique that depends on an extreme learning machine (ELM) called as BotDetector that could directly acquire network stream records and rapidly learn with no data processing for extracting botnet traffic features.

Al-Sarem et al. [14] introduced a combined Mutual Information (MI) based FS technique with ML algorithms for improving the identification of IoT botnet attacks. The FS approach integrates the MI method, PCA and ANOVA f-test at a fine-grained recognition level for selecting the feature to optimize the effectiveness of the IoT Botnet classifier. Popoola et al. [15] suggested an effectual DL-based botnet attack identification technique that should control greatly unbalanced network traffic data. Especially, SMOTE produces more small instances for achieving class balance, whereas Deep-RNN (DRNN) is used to learn hierarchical representation from the balanced traffic data for performing distinctive classification.

Sriram et al. [16] recommended a DL-based botnet identification method, which acts on network traffic flow. The botnet identification technique gathers the network traffic flow, converts it into connectivity records and utilizes a DL algorithm for detecting attacks originating from the co-operated IoT device. In [4], the authors concentrate on the reduction of feature subsets for ML tasks, which can be formulated as 6 various binary and multiclass classification issues depending on the phases of the botnet lifecycle. Particularly, the authors employed wrapper and filter approaches with chosen ML techniques and obtained optimum feature sets for every classification problem.

Malik et al. [17] developed a one-class classifier-based ML technique for identifying IoT botnets in a heterogeneous condition. This introduced a one-class classifier that relies on one-class KNN, to identify the IoT botnets at the earlier phase with higher accuracy. This presented ML-based architecture was a lightweight solution, which acts by selecting the better features leveraging notable filter and wrapper techniques for FS. Alharbi et al. [18] suggested a Local-Global best Bat Algorithm for NNs (LGBA-NN) for choosing both hyperparameters and feature subsets for effectively identifying botnet attacks, acquired from 9 viable IoT devices attacked by 2 botnets: Gafgyt and Mirai. Lekssays et al. [19] developed PAutoBotCatcher, a dynamic botnet identification model dependent upon community behavior analysis between peers controlled by various parameters. PAutoBotCatcher leverages BC to ensure transparency and immutability among all parameters. For enhancing uninterrupted identification but maintaining improved accuracy, the authors developed a type of optimization approach namely preprocessing the communicated network traffic and collecting detection's outputs.

In spite of the ML and DL models existed in the earlier studies, it is still needed to enhance the botnet classification performance. IoT environments are dynamic, and the relevance of features may change over time. Research should explore adaptive and dynamic feature selection approaches that can automatically adjust feature sets based on evolving IoT network characteristics and botnet behaviors. While parameter tuning can significantly affect the performance of deep learning models, there is a need for research on efficient and automated hyperparameter tuning strategies specifically tailored to IoT botnet detection. This includes exploring techniques that balance model complexity and performance. Therefore, in this work, AOA can be used for parameter tuning process.

III. THE PROPOSED MODEL

In this study, automatic botnet detection using the BNT-CBPOADL technique was established for botnet recognition in the IoT platform. The main objective of the BNT-CBPOADL method lies in the correct detection and categorization of botnet attacks in the IoT platform. In the BNT-CBPOADL method, several subprocesses are involved namely Z-score normalization, CVAE-based classification, CBPOA-based feature selection, and AOA-based hyperparameter tuning. Fig. 1 represents the overall flow of the BNT-CBPOADL method.

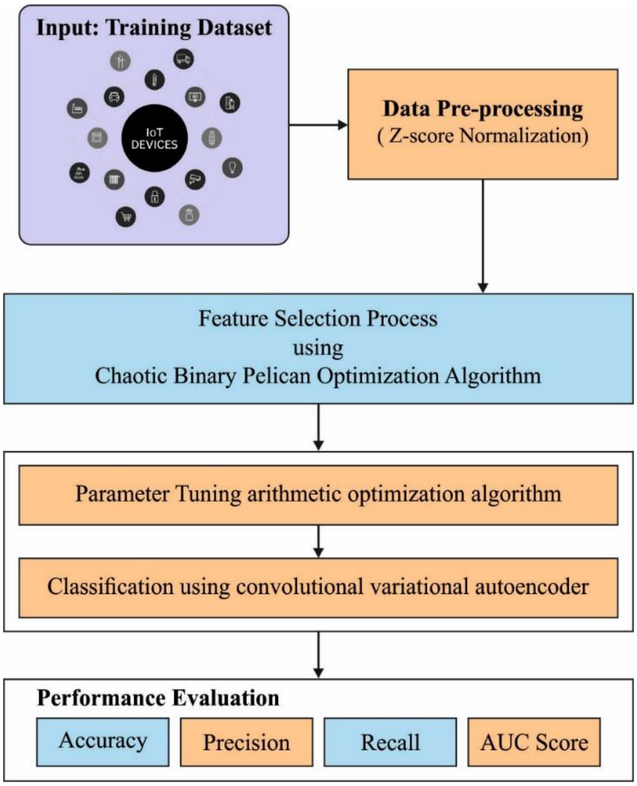
Input: Training Dataset
IoT DEVICES
Data Pre-processing
(Z-score Normalization)
Feature Selection Process
using
Chaotic Binary Pelican Optimization Algorithm
Parameter Tuning arithmetic optimization algorithm
Classification using convolutional variational autoencoder
Performance Evaluation
Accuracy
Precision
Recall
AUC Score

FIGURE 1. The overall flow of the BNT-CBPOADL approach.

A. Z-SCORE NORMALIZATION

To normalize the input data, Z-score normalization is applied. Z-score normalization, otherwise called standardization, is a data preprocessing method exploited in statistics and ML to scale and centre mathematical features. The objective is to convert the information with a mean of 0 and a standard deviation of 1. This helps algorithms that are sensitive to the scale of features work better and make the features comparable.

B. FEATURE SELECTION USING CBPOA

The CBPOA is used in this study for feature selection purposes. POA is the population-based algorithm where the pelican is regarded as the population member [20]. As the POA is continuous, a binary version is introduced to apply feature selection (FS). Further, the CBPOA algorithm is used to improve the optimization efficiency. The mathematical expression of CBPOA for FS is shown below.

Initialization: Similar to the original POA, the CBPOA algorithm begins with setting the parameter. In POA, the population member was randomly initialized to the upper and lower boundaries of the problems by using the subsequent formula.

$$y_{j,k} = m_k + Rand\left(v_k - m_k\right), j= 1, 2, 3, \ldots, O, k = 1, 2, 3, \ldots, n \tag{1}$$

Here, $y_{j,k}$ refers to the value of *the* k^{th} parameter quantified by the j^{th} solution candidate, n denotes the problem variable, O shows the number of members of the population, v_k represents the k^{th} upper bounds, m_k describes the k^{th} lower bounds and *Rand* shows the randomly generated value within [0, 1].

After introducing chaotic mapping, dissimilar chaotic maps are used for interchanging the randomized development in the POA to initialize the features. Eq. (1) can be modified as follows.

$$y_{j,k} = m_k + D_l\left(v_k - m_k\right), j = 1, 2, 3, \ldots, O, k = 1, 2, 3, \ldots, n \tag{2}$$

In Eq. (2), D_l shows the outcomes of chaotic sequence at l^{th} iteration. In this stage, the location of the feature was initialized by using chaotic maps that helped to improve the performance of the global search.

Fitness function: It determines the goodness of the candidate solution. The CBPOA method has applied the fitness function (two cost functions). The initial fitness function is to minimalize G_l for the FS issue. The next one is used for reducing G_l corresponding to the designation for the benchmark function intended for global optimization issues. CBPOA is engaged to define the set of optimum features in a search region with a reduced classifier error rate and small size of feature subset. The formula of cost functions for the FS problem is given as follows,

$$G_l = (1 - \beta) \times Q + \beta \times \left(\frac{M}{Dim}\right) \tag{3}$$

where Dim is the dimension, Q specifies the classifier accuracy, M determines the size of the selected feature subset and β shows the weight parameter set to 0.01.

POA is a continuous version using POA for the FS method, a binary version of POA was defined. In the study, the solution representation for all the features is varied, where the value of all the features is limited [0, 1]. The value 1 indicates the resulting features selected, and the value 0 denotes the neglected or non-selected feature. In all the features, the number of bits represents the size of the features selected.

To transfer the location of all the features from continuous to discrete, different $\overrightarrow{TF}$ are used, and their mathematical formula is given below. All the features are updated corresponding to the following expression:

$$y_{j,k} = \begin{cases} 1, & if\ \left(T_{func}\left(y_{j,k}\right)\right) \geq S \\ 0, & Otherwise. \end{cases} \tag{4}$$

In Eq. (4), S shows the randomly generated value within [0, 1].

Movement towards the features selected (Exploration stage): Modelling the strategy of feature tends to scan the searching region and the exploration effect of CBPOA to determine different areas of the searching region. The location of features selected is produced at random towards search space that is the significant degree in CBPOA. This increases the exploration ability of CBPOA in a specific search for solution space. In this phase, a dynamic weight factor ϖ is adopted to help the feature in continuously updating the location:

$$\varpi = \frac{e^{2\left(1-\frac{l}{L}\right)} - e^{-2\left(1-\frac{l}{L}\right)}}{e^{2\left(1-\frac{l}{L}\right)} + e^{-2\left(1-\frac{l}{L}\right)}} \tag{5}$$

Here, the value of ϖ is larger if the CBPOA can able to achieve a better global search, and the value adaptively diminishes at the iteration end. The CBPOA could implement the best local search while increasing the convergence rate.

$$y_{j,k}^{l+1} = \begin{cases} y_{j,k}^{l} + Rand \cdot \varpi\left(q_k^l - y_{j,k}^l\right), & G_q < G_l; \\ y_{j,k}^{l} + Rand \cdot \varpi\left(y_{j,k}^l - q_q^l\right), & else, \end{cases} \tag{6}$$

Winging on the water surface (Exploitation phase): The CBPOA is used to carry out the best global search. Now, modeling of feature behaviors tends to converge towards the best point in the search range. These behaviors of features increase the local search ability and the exploitation ability. The CBPOA examines the neighborhood point of feature location from the mathematical perspective to converge on the best solution.

$$y_{j,k}^{l+1} = y_{j,k}^{l} + S\left(1 - \frac{u}{U}\right)(2 \cdot Rand - 1) \cdot y_{j,k}^{l} \tag{7}$$

In Eq. (7), U denotes the maximal iteration u shows the iteration count and $\left(1 - \frac{u}{U}\right)$ refers to the neighborhood radius of $y_{j,k}$.

C. BOTNET DETECTION USING CVAE

In this work, the CVAE approach is used for the recognition and classification of botnet attacks. In recent times, there has been increased attention to adjusting the loss function of VAE to increase the disentanglement of a dimension of the hidden space with the objective that the hidden space dimension corresponds to the continuum of significant domain-specific attributes [21]. Higgins et al. formulated this as a constraint optimization problem to increase the marginal log-probability of the data observed as follows:

$$\begin{aligned} &\max_{\theta,\phi} \mathbb{E}_{x\sim D}\left[\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right]\right. \\ &\text{s.t. } KL\left(q_\phi(z \mid x)||p(z)\right) < e \end{aligned} \tag{8}$$

Furthermore, Lagrangian KKT condition can be defined as follows:

$$\begin{aligned} \mathcal{F}(\theta, \phi, \beta, x, z) = &\ \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] \\ &- \beta\left(KL\left(q_\phi(z \mid x)\|p(z)\right) - e\right) \end{aligned} \tag{9}$$

The study found that increasing β enhances the disentanglement of the hidden space dimension, but it reduces the reconstructed quality. Most recent surveys have introduced additional terms to factorize the hidden space and enhance the relationship between hidden space dimensions that enhance the disentanglement of the hidden space. Even though this disentanglement can be easier to validate and quantify while handling imagery information, it became apparent that this disentanglement is not quite as clear for time series data:

$$\mathcal{L}(\theta, \phi, \beta, x, z) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] - \beta KL\left(q_\phi(z \mid x) \| p_\theta(z)\right) \qquad (10)$$

Since β and e in Eq.(11) are both positive constants, $\mathcal{L}$ denotes the lower boundary for $\mathcal{F}$: $\mathcal{F}(\theta, \phi, \beta, x, z) \geq \mathcal{L}(\theta, \phi, \beta, x, z)$. Note that β used as a regularization hyperparameter and to enhance disentanglement in the hidden space. Recall that the KL divergence term in the loss function penalizes the hidden variable posterior. Consequently, hyperparameter β acts as a metric that defines what amount of CVAE is needed to fit on the training dataset. Assuming that there is a degree of freedom in how CVAE fits the mapping to the hidden space, we are applying an unsupervised method and the training dataset comprises anomalous and nominal time series. Therefore, β is treated as a regularization hyperparameter that requires tuning. CVAE can converge towards the regular CAE as β tends towards zero, and if $\beta = 1$, the CVAE model is equal. Fig. 2 depicts the infrastructure of CVAE.

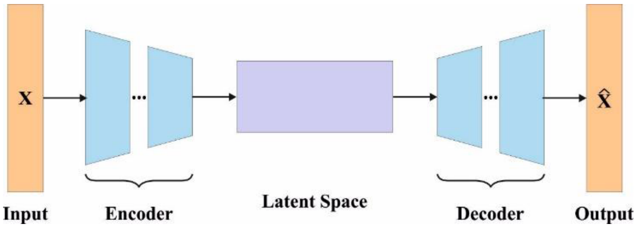

FIGURE 2. Structure of CVAE.

D. HYPERPARAMETER TUNING USING AOA

The hyperparameter tuning of the CVAE model can be chosen by the AOA model. AOA is a metaheuristic optimization approach derived from the distribution behaviors of arithmetical operations (multiplication, addition, division, and subtraction) [22]. It is described by the high accuracy and faster convergence rate of the model. Based on the relationships between r_1 and the values of the MOA function, the AOA approach defines the searching stage. $r_1 \in [0, 1]$. If $r_1 > A$ for global exploration; If $r_1 < A$, AOA is designed locally A is given below:

$$A\left(C_{iter}\right) = \mathrm{Min} + C_{iter} \times \left(\frac{\mathrm{Max} - \mathrm{Min}}{M_{iter}}\right) \qquad (11)$$

In Eq. (11), $A\left(C_{iter}\right)$ shows the value of the existing iteration. Min and Max symbolize the maximal and minimal values of the problem, correspondingly. C_{iter} refers to the present iteration. M_{iter} implies the overall iteration counter. The AOA model exploits division and multiplication operations for the global search process. r_2 is a random integer between [0, 1]. If $r_2 > 0.5$, then enter the multiplication search stage. If $r_2 < 0.5$, then enter the division search phase:

$$X_{i,j}(C_{iter}+1) = \begin{cases} best\left(X_j\right) / (MOP + \varepsilon) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_2 < 0.5 \\ best\left(X_j\right) \times MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_2 > 0.5 \end{cases} \qquad (12)$$

In Eq. (12), $X_{i,j}\left(C_{iter}+1\right)$ denotes the j^{th} location of i^{th} solution at the following iteration, and the best $\left(X_j\right)$ signifies the better solution j^{th} location. ε indicates the smaller integer, UB_j and LB_j shows the upper and lower boundaries at the j^{th} location. μ refers to a control variable that modifies the search stage:

$$P\left(C_{iter}\right) = 1 - \frac{C_{iter}^{\alpha}}{M_{iter}^{\alpha}} \qquad (13)$$

The AOA method exploits subtraction and addition operations for local development. In Eq. (13), α shows the mining accuracy during the iteration process. r_3 refers to a randomly generated value within [0, 1]. If $r_3 > 0.5$, then enter the addition search phase. If $r_3 < 0.5$, then enter the subtraction search phrase:

$$X_{i,j}\left(C_{iter}+1\right) = \begin{cases} best\left(X_j\right) - MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), \ r_3 < 0.5 \\ best\left(X_j\right) + MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), \ r_3 > 0.5 \end{cases} \qquad (14)$$

Fitness selection is an important element in the AOA technique. An encoder solution is employed for estimating the goodness of the solution candidate. Now, the accuracy value is the primary condition used to develop a FF.

$$Fitness = \max(P) \qquad (15)$$

$$P = \frac{TP}{TP + FP} \qquad (16)$$

where TP and FP represent the true and the false positive values.

IV. RESULTS AND DISCUSSION

The botnet detection results of the BNT-CBPOADL method are tested on the Bot-IoT Dataset [23]. The dataset comprises 900 data instances with 2 classes as defined in Table 1. The BNT-CBPOADL technique has chosen 27 features from the available 43 features.

TABLE 1. Description of database.

Class	No. of samples
Botnet	450
Normal	450
Total Samples	900

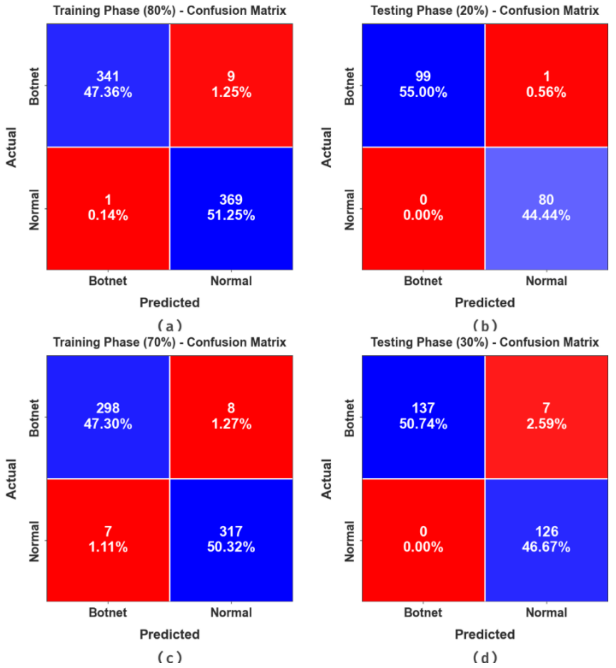

FIGURE 3. Confusion matrices of (a-b) 80:20 of TR set/TS set and (c-d) 70:30 of TR set/TS set.

The botnet detection results can be inspected in the form of a confusion matrix, as shown in Fig. 3. The confusion matrices imply that the BNT-CBPOADL method accurately identifies the botnet and normal data instances.

The botnet recognition outcomes of the BNT-CBPOADL technique can be assessed on 80:20 of the TR set/TS set in Table 2 and Fig. 4. The outcomes ensure the ability of the BNT-CBPOADL technique in the botnet recognition process.

TABLE 2. Botnet recognition outcome of BNT-CBPOADL technique on 80:20 of TR set/TS.

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	AUC_{score}
Training Phase (80%)					
Botnet	97.43	99.71	97.43	98.55	98.58
Normal	99.73	97.62	99.73	98.66	98.58
Average	98.58	98.66	98.58	98.61	98.58
Testing Phase (20%)					
Botnet	99.00	100.00	99.00	99.50	99.50
Normal	100.00	98.77	100.00	99.38	99.50
Average	99.50	99.38	99.50	99.44	99.50

With 80% of TR set, the BNT-CBPOADL technique offers an average $accu_y$, $prec_n$, $recal_l$, F_{score}, and AUC_{score} of 98.58%, 98.66%, 98.58%, 98.61%, and 98.58% correspondingly. As well, with 20% of TS set, the BNT-CBPOADL technique offers an average $accu_y$, $prec_n$, $recal_l$, F_{score}, and AUC_{score} of 99.50%, 99.38%, 99.50%, 99.44%, and 99.50% correspondingly.

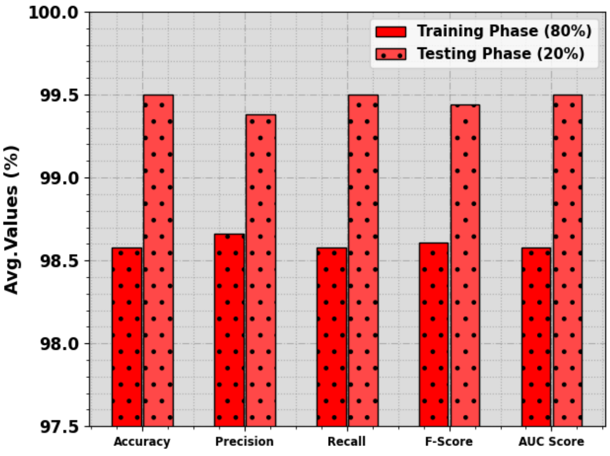

FIGURE 4. Average of BNT-CBPOADL technique on 80:20 of TR set/TS set.

The botnet recognition outcomes of the BNT-CBPOADL method can be measured on 70:30 of the TR set/TS set in Table 3 and Fig. 5. The outcomes ensure the ability of the BNT-CBPOADL technique in the botnet recognition method. With 70% of TR set, the BNT-CBPOADL method offers an average $accu_y$, $prec_n$, $recal_l$, F_{score}, and AUC_{score} of 97.61%, 97.62%, 97.61%, 97.62%, and 97.61% correspondingly. Also, with 30% of TS set, the BNT-CBPOADL technique offers an average $accu_y$, $prec_n$, $recal_l$, F_{score}, and AUC_{score} of 97.57%, 97.37%, 97.57%, 97.40%, and 97.57% correspondingly.

TABLE 3. Botnet recognition outcome of BNT-CBPOADL technique on 70:30 of TR set/TS set.

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	AUC_{score}
Training Phase (70%)					
Botnet	97.39	97.70	97.39	97.55	97.61
Normal	97.84	97.54	97.84	97.69	97.61
Average	97.61	97.62	97.61	97.62	97.61
Testing Phase (30%)					
Botnet	95.14	100.00	95.14	97.51	97.57
Normal	100.00	94.74	100.00	97.30	97.57
Average	97.57	97.37	97.57	97.40	97.57

Fig. 6 shows the training accuracy TR_accu_y and VL_accu_y of the BNT-CBPOADL technique on 80:20 of the TR set/TS set. The TL_accu_y is determined by the evaluation of the BNT-CBPOADL technique on the TR dataset whereas the VL_accu_y is computed by evaluating the performance on a separate testing dataset. The results exhibit that TR_accu_y and VL_accu_y increase with an upsurge in epochs. As a result, the performance of the BNT-CBPOADL technique gets improved on the TR and TS dataset with a rise in the number of epochs.

In Fig. 7, the TR_loss and VR_loss outcomes of the BNT-CBPOADL method on 80:20 of the TR set/TS set are shown. The TR_loss defines the error among the predictive performance and original values on the TR data. The

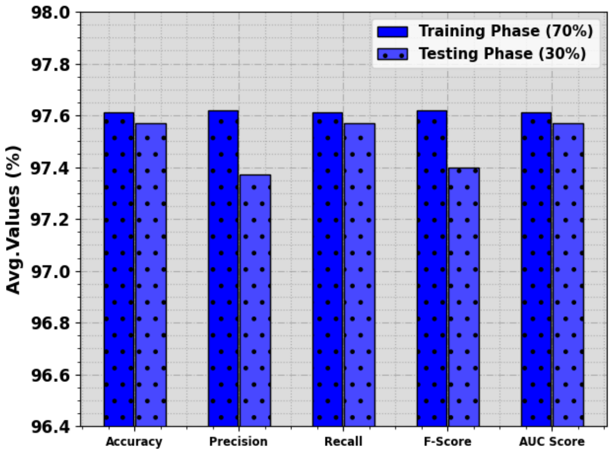

FIGURE 5. Average of BNT-CBPOADL technique on 70:30 of TR set/TS set.

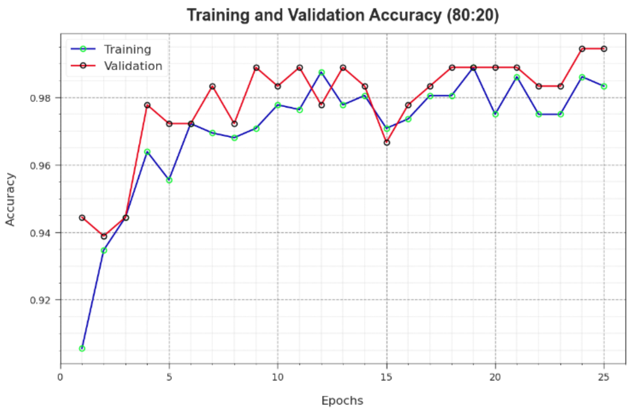

FIGURE 6. ***Accu*$_y$ curve of BNT-CBPOADL method on 80:20 of TR set/TS set.**

VR_loss represent the measure of the performance of the BNT-CBPOADL technique on individual validation data. The results indicate that the *TR_loss* and *VR_loss* tend to decrease with rising epochs. It portrayed the enhanced performance of the BNT-CBPOADL technique and its capability to generate accurate classification. The reduced value of *TR_loss* and *VR_loss* demonstrates the enhanced performance of the BNT-CBPOADL technique in capturing patterns and relationships.

A detailed PR analysis of the BNT-CBPOADL method is illustrated on 80:20 of the TR set/TS set in Fig. 8. The outcomes indicated that the BNT-CBPOADL system results in maximum values of PR. Moreover, the BNT-CBPOADL method reaches high PR values in all classes.

In Fig. 9, a ROC curve of the BNT-CBPOADL method is shown at 80:20 of the TR set/TS set. The outcome showed that the BNT-CBPOADL system resulted in enhanced ROC values. In addition, the BNT-CBPOADL method shows maximum ROC values in all classes.

The enhanced capability of the BNT-CBPOADL technique can be confirmed using a comparison study, as given in Table 4 [11]. Fig. 10 illustrates the comparative results of the BNT-CBPOADL technique in terms of $accu_y$ and F_{score}. The results show that the BNT-CBPOADL technique outperforms the other models. Based on $accu_y$, the BNT-CBPOADL technique indicates an increasing value of 99.50% while the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based model, and FL-ANN models offer to decrease

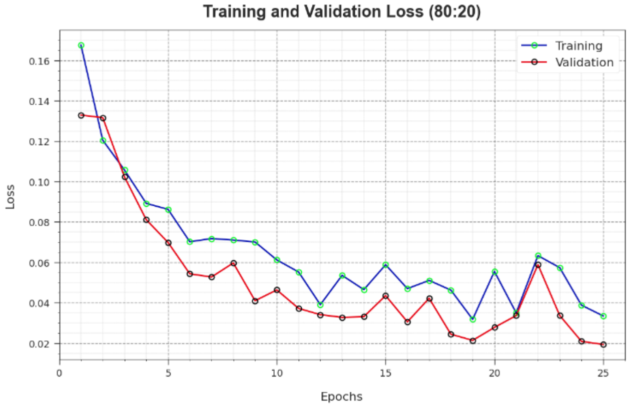

FIGURE 7. Loss curve of BNT-CBPOADL technique on 80:20 of TR set/TS set.

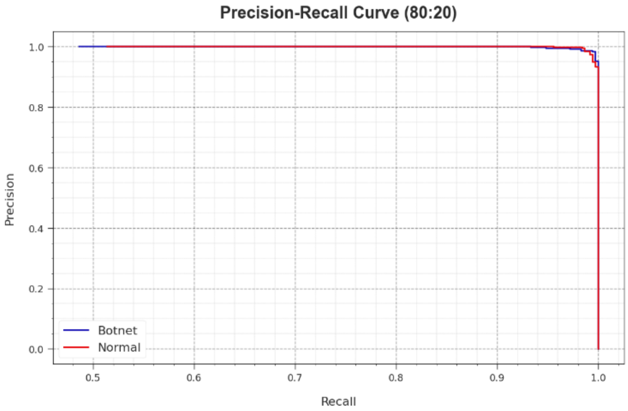

FIGURE 8. PR curve of BNT-CBPOADL technique on 80:20 of TR set/TS set.

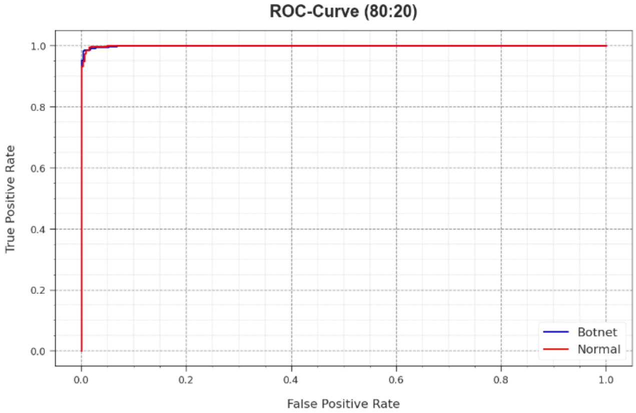

FIGURE 9. ROC of BNT-CBPOADL method on 80:20 of TR set/TS set.

$accu_y$ values of 99.12%, 94.50%, 95%, 97.90%, 92.90%, and 98.94% respectively. In addition, based on F_{score}, the BNT-CBPOADL technique indicates increasing value of 99.44% while the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based model, and FL-ANN models offer to decrease F_{score} values of 98%, 94.66%, 96.16%, 95.65%, 96.59%, and 97.08% correspondingly.

TABLE 4. Comparative outcome of BNT-CBPOADL system with other techniques [11].

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
BNT-CBPOADL	99.50	99.38	99.50	99.44
BDC-RSODL	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
DT	97.90	94.94	95.95	95.65
Host-based	92.90	95.34	96.84	96.59
FL-ANN	98.94	96.29	97.87	97.08

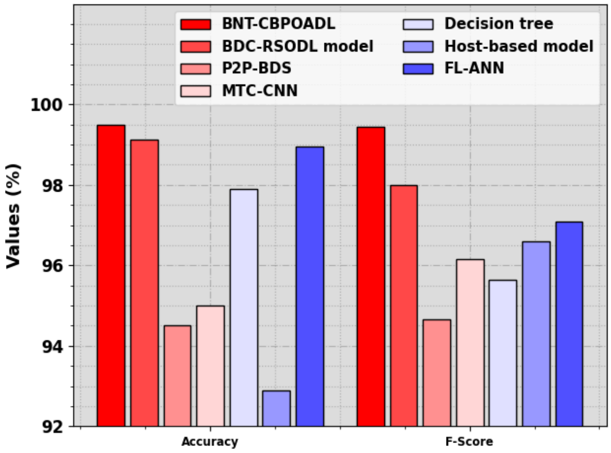

FIGURE 10. $Accu_y$ **and** F_{score} **outcome of BNT-CBPOADL algorithm with other systems.**

Fig. 11 illustrates the comparative results of the BNT-CBPOADL technique in terms of $prec_n$ and $reca_l$. The results show that the BNT-CBPOADL technique outperforms the other models. Based on $prec_n$, the BNT-CBPOADL technique indicates an increasing value of 99.38% while the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based model, and FL-ANN models offer to decrease $prec_n$ values of 96.91%, 95.61%, 95.87%, 94.94%, 95.34%, and 96.29% respectively. In addition, based on $reca_l$, the BNT-CBPOADL technique indicates an increasing value of 99.50% while the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based model, and FL-ANN models offer to decrease $reca_l$ values of 99.18%, 96.67%, 97.77%, 95.95%, 96.84%, and 97.87% correspondingly. These results show that the BNT-CBPOADL technique offers superior performance in the botnet detection technique.

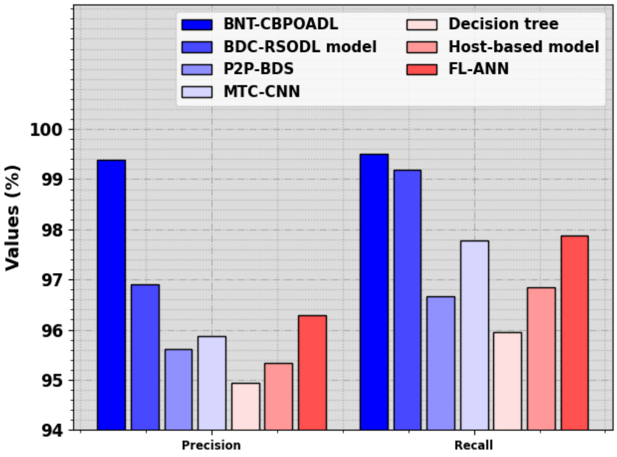

FIGURE 11. $Prec_n$ **and** $reca_l$ **outcome of BNT-CBPOADL algorithm with other systems.**

V. CONCLUSION

In this study, an automatic botnet detection using the BNT-CBPOADL technique was established for botnet detection in the IoT platform. The main objective of the BNT-CBPOADL method lies in the correct detection and categorization of botnet attacks in the IoT platform. In the BNT-CBPOADL method, several subprocesses are involved namely Z-score normalization, CBPOA-based feature selection, CVAE-based classification, and AOA-based hyperparameter tuning. Meanwhile, the CBPOA technique is derived for feature selection which aids in accomplishing improved performance. Finally, the AOA is used for the optimal hyperparameter tuning of the CVAE method. The simulation analysis of the BNT-CBPOADL technique is tested on Bot-IoT Dataset. The experimentation outcomes inferred the supremacy of the BNT-CBPOADL method over other existing techniques with maximum accuracy of 99.50%. Future work can focus on the exploration of multi-modal data fusion approaches to incorporate different data sources from IoT devices, such as network traffic, device behavior, and sensor data, to enhance botnet detection accuracy. In addition, the proposed model can be integrated to the edge computing to allow localized botnet detection, minimizing latency, and resource usage in the IoT environment. Future work can develop mechanisms for real-time threat intelligence sharing among IoT devices and networks to bolster collective defense mechanisms and enable rapid response to evolving botnet threats.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number (RGP2/ 29/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R77), Princess Nourah bint Abdulrahman

University, Riyadh, Saudi Arabia. We Would like to thank SAUDI ARAMCO Cybersecurity Chair for funding this project. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444). This study is partially funded by the Future University in Egypt (FUE).

REFERENCES

[1] Sudhakar and S. Kumar, "ABBDIoT: Anomaly-based botnet detection using machine learning model in the Internet of Things network," in *Proc. Int. Conf. IoT, Intell. Comput. Secur., Select (IICS)*. Singapore: Springer Nature, Apr. 2023, pp. 235–245.

[2] M. M. Alani, "BotStop: Packet-based efficient and explainable IoT botnet detection using machine learning," *Comput. Commun.*, vol. 193, pp. 53–62, Sep. 2022.

[3] G. Kirubavathi and U. K. Sridevi, "Detection of IoT Botnet using machine learning and deep learning techniques," Tech. Rep., 2023.

[4] R. Kalakoti, S. Nõmm, and H. Bahsi, "In-depth feature selection for the statistical machine learning-based botnet detection in IoT networks," *IEEE Access*, vol. 10, pp. 94518–94535, 2022.

[5] F. Hussain, S. G. Abbas, I. M. Pires, S. Tanveer, U. U. Fayyaz, N. M. Garcia, G. A. Shah, and F. Shahzad, "A two-fold machine learning approach to prevent and detect IoT botnet attacks," *IEEE Access*, vol. 9, pp. 163412–163430, 2021.

[6] S. Verma, N. Sharma, A. Singh, A. Alharbi, W. Alosaimi, H. Alyami, D. Gupta, and N. Goyal, "DNNBoT: Deep neural network-based botnet detection and classification," *Comput., Mater. Continua*, vol. 71, no. 1, pp. 1729–1750, 2022.

[7] K. Shinan, K. Alsubhi, A. Alzahrani, and M. U. Ashraf, "Machine learning-based botnet detection in software-defined network: A systematic review," *Symmetry*, vol. 13, no. 5, p. 866, May 2021.

[8] Y. Masoudi-Sobhanzadeh and S. Emami-Moghaddam, "A real-time IoT-based botnet detection method using a novel two-step feature selection technique and the support vector machine classifier," *Comput. Netw.*, vol. 217, Nov. 2022, Art. no. 109365.

[9] I. Apostol, M. Preda, C. Nila, and I. Bica, "IoT botnet anomaly detection using unsupervised deep learning," *Electronics*, vol. 10, no. 16, p. 1876, Aug. 2021.

[10] C. Wei, G. Xie, and Z. Diao, "A lightweight deep learning framework for botnet detecting at the IoT edge," *Comput. Secur.*, vol. 129, Jun. 2023, Art. no. 103195.

[11] S. M. Alshahrani, F. S. Alrayes, H. Alqahtani, J. S. Alzahrani, M. Maray, S. Alazwari, M. A. Shamseldin, and M. Al Duhayyim, "IoT-cloud-assisted botnet detection using rat swarm optimizer with deep learning," *Comput., Mater. Continua*, vol. 74, no. 2, pp. 3085–3100, 2023.

[12] F. S. Alrayes, M. Maray, A. Gaddah, A. Yafoz, R. Alsini, O. Alghushairy, H. Mohsen, and A. Motwakel, "Modeling of botnet detection using barnacles mating optimizer with machine learning model for Internet of Things environment," *Electronics*, vol. 11, no. 20, p. 3411, Oct. 2022.

[13] X. Dong, C. Dong, Z. Chen, Y. Cheng, and B. Chen, "BotDetector: An extreme learning machine-based Internet of Things botnet detection model," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 5, May 2021, Art. no. e3999.

[14] M. Al-Sarem, F. Saeed, E. H. Alkhammash, and N. S. Alghamdi, "An aggregated mutual information based feature selection with machine learning methods for enhancing IoT botnet attack detection," *Sensors*, vol. 22, no. 1, p. 185, Dec. 2021.

[15] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, and A. A. Atayero, "SMOTE-DRNN: A deep learning algorithm for botnet detection in the Internet-of-Things networks," *Sensors*, vol. 21, no. 9, p. 2985, Apr. 2021.

[16] S. Sriram, R. Vinayakumar, M. Alazab, and K. P. Soman, "Network flow based IoT botnet attack detection using deep learning," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 189–194.

[17] K. Malik, F. Rehman, T. Maqsood, S. Mustafa, O. Khalid, and A. Akhunzada, "Lightweight Internet of Things botnet detection using one-class classification," *Sensors*, vol. 22, no. 10, p. 3646, May 2022.

[18] A. Alharbi, W. Alosaimi, H. Alyami, H. T. Rauf, and R. Damaševičius, "Botnet attack detection using local global best bat algorithm for industrial Internet of Things," *Electronics*, vol. 10, no. 11, p. 1341, Jun. 2021.

[19] A. Lekssays, L. Landa, B. Carminati, and E. Ferrari, "PAutoBotCatcher: A blockchain-based privacy-preserving botnet detector for Internet of Things," *Comput. Netw.*, vol. 200, Dec. 2021, Art. no. 108512.

[20] R. K. Eluri and N. Devarakonda, "Chaotic binary pelican optimization algorithm for feature selection," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 31, no. 3, pp. 497–530, Jun. 2023.

[21] M. Memarzadeh, B. Matthews, and I. Avrekh, "Unsupervised anomaly detection in flight data using convolutional variational auto-encoder," *Aerospace*, vol. 7, no. 8, p. 115, Aug. 2020.

[22] J. Yang, Q. Yu, S. Chen, and D. Yang, "Underwater image color constancy calculation with optimized deep extreme learning machine based on improved arithmetic optimization algorithm," *Electronics*, vol. 12, no. 14, p. 3174, Jul. 2023.

[23] (Jun. 2, 2021). *The Bot-IoT Dataset*. [Online]. Available: https://research.unsw.edu.au/projects/bot-iot-dataset