Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052804

ARTICLE

# An Optimized Approach to Deep Learning for Botnet Detection and Classification for Cybersecurity in Internet of Things Environment

**Abdulrahman Alzahrani***

Department of Computer Science and Engineering, College of Computer Science and Engineering, University of Hafr Al Batin, P.O. Box 1803, Hafr Al Batin, 39524, Saudi Arabia

*Corresponding Author: Abdulrahman Alzahrani. Email: aalzahrani@uhb.edu.sa

Received: 16 April 2024 Accepted: 28 June 2024 Published: 15 August 2024

**ABSTRACT**

The recent development of the Internet of Things (IoTs) resulted in the growth of IoT-based DDoS attacks. The detection of Botnet in IoT systems implements advanced cybersecurity measures to detect and reduce malevolent botnets in interconnected devices. Anomaly detection models evaluate transmission patterns, network traffic, and device behaviour to detect deviations from usual activities. Machine learning (ML) techniques detect patterns signalling botnet activity, namely sudden traffic increase, unusual command and control patterns, or irregular device behaviour. In addition, intrusion detection systems (IDSs) and signature-based techniques are applied to recognize known malware signatures related to botnets. Various ML and deep learning (DL) techniques have been developed to detect botnet attacks in IoT systems. To overcome security issues in an IoT environment, this article designs a gorilla troops optimizer with DL-enabled botnet attack detection and classification (GTODL-BADC) technique. The GTODL-BADC technique follows feature selection (FS) with optimal DL-based classification for accomplishing security in an IoT environment. For data preprocessing, the min-max data normalization approach is primarily used. The GTODL-BADC technique uses the GTO algorithm to select features and elect optimal feature subsets. Moreover, the multi-head attention-based long short-term memory (MHA-LSTM) technique was applied for botnet detection. Finally, the tree seed algorithm (TSA) was used to select the optimum hyperparameter for the MHA-LSTM method. The experimental validation of the GTODL-BADC technique can be tested on a benchmark dataset. The simulation results highlighted that the GTODL-BADC technique demonstrates promising performance in the botnet detection process.

**KEYWORDS**

Botnet detection; internet of things; gorilla troops optimizer; hyperparameter tuning; intrusion detection system

## 1 Introduction

The production of IoT devices caused the stable growth of IoT-based attacks. Currently, dangerous IoT risks are nothing but IoT Botnet attacks that attempt to pledge effective, profitable and actual cybercrimes [1]. IoT botnets are groups of Internet-connected IoT devices infected by malware and accomplished slightly by attackers. IoT networks have essential tasks in providing models to identify safety vulnerabilities and attacks owing to the fast development of threats and the assortment

of attack strategies [2]. If malware is implemented, there will be a growing number of advances in DL/ML based recognition models that use full-time series data. However, there is a requirement to employ full-time series data harshly parameters present functions efficacy [3]. In addition, early identification permits enhanced IoT Botnet response suggestions. As an outcome, it reduces injuries that are affected by probable assaults. The dynamic analysis method surveys how malware relates to its atmospheres when executed [4]. The use of botnets and bot malware supports other dangerous online actions like distributed denial of service assaults, click scams, and spam and virus distribution. IoT Botnet development contains propagation and extensive scan stage [5]. If it is viable to diagnose and distinguish bots beforehand, they initiate a definite assault, namely DDoS; IoT Botnet recognition solutions have a harsher effect. So, it is vital to classify dangerous activities of IoT Botnet modules as much as possible.

A botnet attack is one of the severe attacks recognized for spreading quickly among devices linked to the Internet [6]. There are chief gaps in prior techniques for discovering suitable and effective mechanisms to defend IoT devices from botnet assaults [7]. An IDS is the only dominant solution for dealing with botnet attacks. It utilizes artificial intelligence (AI) to discover novel botnet attack designs. An IDS is separated into dual kinds such as misuse and anomaly models [8]. These types are highly based on being signature-based. Many IDSs, like Suricata and Snort, are obtainable. AI techniques are employed to identify IoT attacks with further assured recognition. AI techniques can discover alterations in networks and approaches to attacks. This was one of the high tasks tackled by security solutions to handle IoT attacks [9]. Generally, hackers make slight variations in preceding attacks that security solutions cannot identify. Numerous researchers employ AI methods to prevent threats to the IoT environment by examining network traffic. DL and ML models are built into security systems to discover such assaults proficiently. DL is one of the AI developments used in real time to handle complex nonlinear data [10]. A deep recurrent neural network (DRNN) is executed to detect botnet assaults from IoT devices.

This article designs a gorilla troops optimizer with a DL-enabled botnet attack detection and classification (GTODL-BADC) technique. The GTODL-BADC technique follows feature selection (FS) with optimal DL-based classification for accomplishing security in the IoT environment. For data preprocessing, the min-max data normalization approach is primarily used. The GTODL-BADC technique uses the GTO algorithm to select features and elect optimal feature subsets. Moreover, multi-head attention-based long short-term memory (MHA-LSTM) methodology is applied for botnet detection. Finally, the tree seed algorithm (TSA) can select the optimum hyperparameter for the MHA-LSTM technique. The experimental validation of the GTODL-BADC technique can be tested on a benchmark dataset.

## 2 Related Works

In [11], an innovative lightweight and generic NIDS with a 2-phase architecture was designed. This technique initially developed 21 statistical features, and depending on these features, a model has been devised according to an AE for filtering. Next, a new technique was developed to convert packet length sequences like a 3-channel RGB image for detection dependent upon a lightweight CNN technique. Hezam et al. [12] developed a DL method that includes 3 DL methods, such as CNN, LSTM-RNN, and RNN, to combat DDoS attack-targeted IoT environments. The methods are examined by applying an N-BaIoT database, which could be gathered by affecting nine IoT devices with two major serious DDoS botnets such as Mirai and BASHLITE. Haq et al. [13] designed two innovative architectures namely Deep Neural Network (DNN), DNNBoT1 and DNNBoT2, for identifying and categorizing

botnet attacks, namely BASHLITE and Mirai. The application of PCA has been accomplished to feature extraction. The system could be presented depending on rigorous hyperparameter tuning with GridsearchCV. Khan et al. [14] considered a lightweight and robust DL method. This technique was to exhibit the scalability and attack detection effectiveness employed for training as well as testing. Besides, the developed Hybrid system was related to a benchmark Artificial Neural Network (ANN) model. In [15], the federated DL (FDL) technique was developed for zero-day botnet attack detection. An optimum DNN method was utilized for classification. A method parameter supports remote controls of the self-sufficient DNN architecture training at numerous IoT-edge devices. However, the federated averaging (FedAvg) technique could be exploited to combine local model updates. A global DNN was generated, followed by a count of communication iterations among the IoT-edge devices and architecture parameter server. Hasan et al. [16] planned a hybrid intelligent DL approach for protecting the IIoT environment from dangerous and difficult multi-variant Botnet attacks. The developed method was severely analyzed with a new database, normal and comprehensive efficiency assessment metrics, and standard DL methods. Also, cross-validation of these outcomes was further executed to exhibit overall effectiveness.

In [17], an ARP spoofing identification method was developed by applying an explainable DL method such as ARP-PROBE for IoT networks. This introduced algorithm depends on features removed in network packets for identifying ARP spoofing rapidly and efficiently employing an FS and extraction model, which recognizes and chooses the extremely significant features. In [18], cooperative game theory incorporating three methods, namely LSTM, AE, and SVM, has been implemented to recognize IoT botnet attacks. The developed methods depend upon the efficient FS through cooperative game theory and shapely values under a database collected at a 5 IoT device attacked with botnets and employing AE, LSTM, and SVM for recognizing IoT Botnet traffic. Nazir et al. [19] aimed to detect effectual ML and DL models for IoT Botnet recognition by evaluating standard datasets, metrics, and preprocessing models. In [20], a novel model by joining collaborative threat intelligence and blockchain (BC) technology with ML methods, this model also utilizes Random Forest (RF), Decision Tree (DT) classifier, Ensemble, CNN, and LSTM methods. Abualigah et al. [21] proposed a novel IPDOA model, which enhances the search procedure of the Prairie Dog Optimization Algorithm (PDOA) by integrating the initial upgrading mechanism of the Dwarf Mongoose Optimization Algorithm (DMOA). Sangaiah et al. [22] incorporated linear correlation feature selection techniques utilizing INTERACT and MLP, suggesting the uninterrupted employment of data balancing approaches. Javadpour et al. [23] proposed a novel distributed multi-agent IDPS (DMAIDPS) model, where learning agents execute a six-step recognition procedure for classifying network behaviour. Several DL methods comprising CNN, LSTM-RNN, and DNN, along with a rigorous hyperparameter tuning process, are utilized for detecting IoT botnets, while federated DL models and fusion intellectual DL methods improve cybersecurity in IIoT environments by accentuating effectual FS and model optimization.

## 3 The Proposed Method

This article designs a novel GTODL-BADC technique. The technique follows FS with optimal DL-based classification for accomplishing security in the IoT environment. It comprises four main processes: min-max data normalization, GTO-based feature subset selection, MHA-LSTM-based classification, and TSA-based hyperparameter tuning. Fig. 1 illustrates the entire flow of the technique.

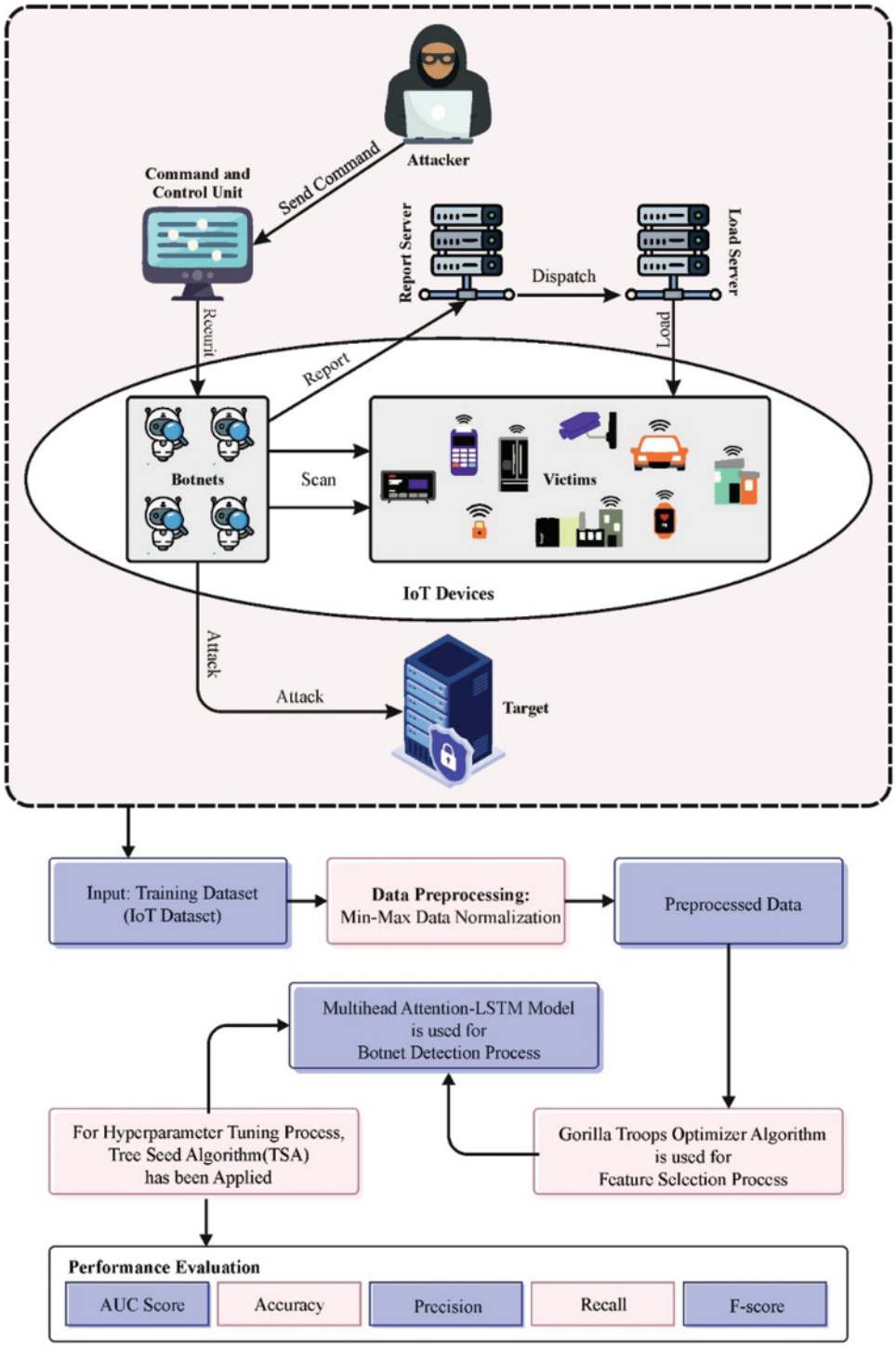


**Figure 1:** Overall flow of GTODL-BADC technique

### *3.1 Data Normalization*

For data preprocessing, the min-max data normalization approach is primarily used. Min-max normalization is a critical preprocessing method applied in botnet recognition in IoT atmospheres to normalize and measure sensor data features [24]. This technique changes raw sensor values into a shared range, naturally among 0 and 1, by deducting the least value and separating by range (difference amid maximal and minimal values). This normalization safeguards that numerous sensor readings with dissimilar measures are carried to an even scale, permitting actual comparison and analysis. In botnet recognition, this standardized data becomes helpful for training ML methods. By simplifying consistent feature representation through various IoT devices, min-max normalization donates to creating robust methods to classify abnormal patterns related to potential botnet actions through multiple devices and sensors.

### *3.2 GTO-Based FS*

The GTODL-BADC technique uses the GTO model to elect optimal feature subsets at this stage. The GTO model relies entirely on many separate performances of gorillas that are arithmetically replicated. Five behaviours are taken in this state to improve gorillas' behaviour, such as 3 for the exploration and 2 for the exploitation phases [25]. These actions include migration to a weird area, migration to other gorillas, travel near a definite spot, challenges for adult females and conducting silverback. The 2 phases signify the mentioned planned choices which are separated into exploitation as well as exploration phase demonstrated in the following sub-sections.

Exploration stage. In this stage, three distinct behaviors are explained: the 1st one is to manifest GTO exploration, whereas 2nd tactic signifies migrant behaviour to other gorillas. Besides, 3rd plan goals at cheering GTO's abilities in defining countless computing spaces denotes movement near a definite spot. Eq. (11) signifies three behaviours arithmetically, where action to unknown endpoint approach in this equation. Suppose an arbitrary number ($rn$) exceeds a factor ($Fr$). Also, migrants to other gorillas or migrants near a definite spot are cautiously chosen if an arbitrary number is equal or more than 50 percent.

$$GtX\,(Itn+1) = \begin{cases} LB + rn_1 \times (UB - LB), Fr > rn \\ Z \times X(Itn) \times Q + X_r(Itn) \times (rn_2 - D \times (1 - Itn/MxItn)), 0.5 \le rn, \\ X\,(Itn) + (X\,(Itn) - GoX_r\,(t)) \times rn_3 - \left(X\,(Itn) - GoX_r\,(Itn) \times Q^2\right), 0.5 > rn \end{cases} \tag{1}$$

$$D = cos\,(2 \times rn_4) + 1 \tag{2}$$

$$Q = D \times \left(1 - \frac{Itn}{MxItn}\right) \tag{3}$$

$$Z = [-(D \times (1 - Itn/MxItn)), D \times (1 - Itn/MxItn)] \tag{4}$$

Whereas $rn, rn_1, rn_2, rn_3$, and $rn_4$ demonstrate random values amongst [0, 1], while $X\,(Itn)$ and $GtX(Itn + l)$ state complete and future routes of the gorilla's location. Random movable variables $X_r$ and $GtXr$ determine a gorilla's present group and its potential location. The factor ($Fr$) must be in the range [0:1] and describes the prospect of deciding on a travelling technique to an anxious site. UB denotes minimum bound, and LB signifies maximum bound. The variables D and Q are defined accurately by Eqs. (1) and (4). A maximal and current iteration amount is categorized by ($Itn$) and

($MxItn$). Also, representation (Z) is $[-(D\times(1-Itn/MxItn)), D\times(1-Itn/MxItn)]$, while representation (s) means arbitrary values between [−1:1].

Exploitation phase: 2 strategies are projected in this phase when factor $D \times (1 - Itn/MxItn)$ is equated with variable (Y). These two performances led silverback to compete with adult females. When the value of Y equals or is less than the value of $D \times (1 - Itn/MxItn)$, 1st one is defined, and then the method of silverback selected can guide others to food sources. This approach is signified mathematically in Eq. (15), which is mentioned below:

$$GtX\,(Itn+1) = Q \times R\,(Itn) \times (X\,(Itn) - X_{sb}) + X\,(Itn) \tag{5}$$

$$R\,(Itn) = \left(\left|\left(\frac{1}{NG}\right)\sum_{i=1}^{NG} GtX_i\,(Itn)\right|^{2Q}\right)^{\left(\frac{1}{2Q}\right)} \tag{6}$$

Whereas $NG$ is gorillas' populace; $X_{sb}$ specifies silverback; $X(Itn)$ means the gorilla location vector; and $GtX_i\,(Itn)$ indicates gorilla position 0 in iteration $Itn$.

If the value of $y$ is more than the term $D \times (1 - Itn/MxItn)$, the approach of competing for adult females is nominated. It signified arithmetically Eq. (17), which is given below:

$$GX\,(Itn) = X_{sb} - (X_{sb} \times L - X\,(Itn) \times L) \times A \tag{7}$$

$$L = 2 \times rn_5 - 1 \tag{8}$$

$$A = \beta \times E, E = \begin{cases} NG_1 rn \geq 0.5 \\ NG_2 rn < 0.5 \end{cases} \tag{9}$$

where $L$ denotes the force of impact; $rn_5$ represents an arbitrary amount from [0:1]; $\beta$ represents the pre-optimization value that is definite and set to 3; factor ($A$) vector denotes ferocity level in battle; and $E$ is used as a follower for violence effectiveness.

If the fitness value of $GtX(Itr)$ is less than $X(Itn)$, then the $GtX(Itn)$ solution will substitute $X(Itn)$.

Enhanced GTO combining tangent flight approach. An improved GTO (IGTO) includes this section's Tangent Flight Strategy (TFS). Cauchy calculated below, and its tangent function is similar to TFS:

$$f = tan\left(pp \times \frac{\pi}{2}\right) \tag{10}$$

$$pp = randn\,(1, Dim) \tag{11}$$

Meanwhile, $pp$ denotes an evenly distributed random amount with values in the interval [0, 1], and $Dim$ denotes the number of dimensions in the function. This process is proficient in competently penetrating search space. This function is periodic and never breaks the balance between exploitation as well as exploration. TFS is added to Eq. (15) by the recommended IGTO technique. The separation between the gorilla and silverback will narrow as an outcome of this alteration, radically decreasing the final step size and enhancing the principal value. This model is explained scientifically below:

$$GtX\ (Itn+1) = \left(\frac{\tan\left(\pi \times \frac{2pp-1}{2}\right)}{100}\right) \times Q \times R(Itn) \times (X\ (Itn) - X_{sb}) + X\ (Itn) \tag{12}$$

The fitness function (FF) reflects classification accuracy and the number of nominated features. It increases classification accuracy and reduces the set size of the nominated feature. So, FF is employed to estimate individual solutions as given in Eq. (13).

$$Fitness = \alpha * ErrorRate + (1-\alpha) * \frac{\#SF}{\#All\_F} \tag{13}$$

Whereas *ErrorRate* denotes the classification error rate employing a particular feature. *ErrorRate* is intended as a percentage of improper categorized to the amount of classification prepared, conveyed as a value amid 0 and 1 (*ErrorRate* is the complement of classification accuracy), $\#SF$ denotes the number of selected features, and $\#All\_F$ signifies the total quantity of attributes in original datasets. $\alpha$ employed to control the significance of classification quality and subset length. In the tests, $\alpha$ is set to 0.9.

### *3.3 Botnet Detection Using MHA-LSTM*

For the classification process, the MHA-LSTM model can be applied. LSTM is comprised of a memory unit called a cell $\left(\tilde{C}_t\right)$,the update gate ($u_t$), forget gate ($f_t$), input gate ($i_t$), and output gate ($o_t$) [26]. Using the above gates, it is possible to obtain, keep, or write data from or into a cell. Correspondingly, assume $W$ and $b$, $X(f, i, c, o)$ as the controlling gates of present input $X_t$ prior output $o_t$, weight matrix, and bias.

$$f_t = S_{igmoid}\left(W_f^o\left[h_{t-1}, X_t\right] + b_f\right) \tag{14}$$

Eq. (14) signifies the entry-wise multiplication of prior data and present input that relies on the existing values of the forget gate. Zero and non-zero values of the forget gate imply throwing away and passing the data individually. At the same time, input implements data and keeps it in a memory unit. Next, the input gate ($i_f$) decides on the sigmoid function what data to be transmitted and forgotten from the storage unit. The input gate generates a near-zero output to avoid cell updates from novel data input.

$$i_t = S_{igmoid}\left(W_i^o\left[h_{t-1}, X_t\right] + b_i\right) \tag{15}$$

$$\tilde{C}_t = \tanh\left(W_C^O\left[h_{t-1}, X_t\right] + b_C\right) \tag{16}$$

Lastly, the new memory unit is combined with the output gate to determine the existing value of LSTM, in which the output gate exploits sigmoid activation to elect which condition in the existing cell serves as an outcome and the novel memory unit exploits tanh to allocate output value.

$$o_t = S_{igmoid}\left(W_o^O\left[h_{t-1}, X_t\right] + b_o\right) \tag{17}$$

$$h_t = o_t^o \tanh\left(C_t\right) \tag{18}$$

Among the difficulties experienced in this field is the capability to address tasks with longer-term dependency. LSTM is an effective model for forecasting accurate time series. Despite addressing challenges such as gradient expansion and vanishing problems, LSTM is widely adopted for applications that heavily depend on prior information.

MHA-LSTM is a refined neural network structure that integrates the strength of multi-head attention mechanisms and LSTM. In this hybrid method, multi-head attention is combined into the LSTM framework to increase the network's capability to capture longer-range needs and instantly appear to dissimilar portions of the input sequence. Fig. 2 depicts the infrastructure of MHA-LSTM.

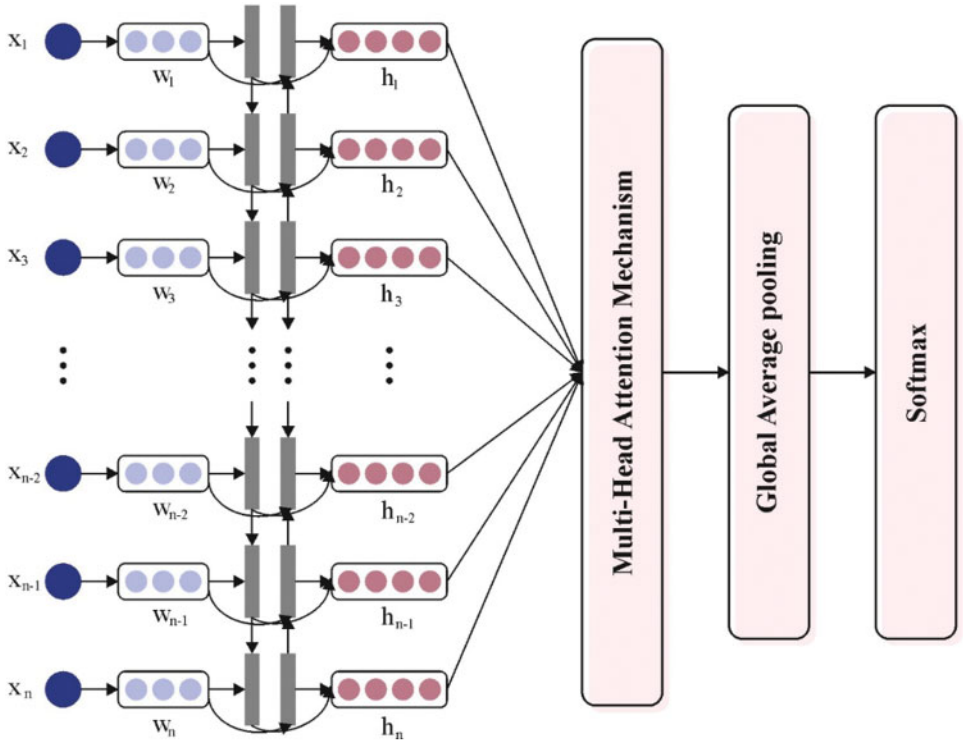


**Figure 2:** Architecture of MHA-LSTM

The multi-head attention mechanism permits the method to concentrate on dissimilar places within the input sequence in parallel, allowing it to capture difficult relationships and dependencies more efficiently. This is mainly beneficial for challenges involving sequential data where definite elements have varying levels of significance at dissimilar time steps.

### *3.4 Hyperparameter Tuning*

Finally, the TSA can be applied to optimize the hyperparameter selection of the MHA-LSTM model. The TSA is simulated by nature, as presented by Kıran in 2015 [27]. TSA has designed the connection of positions of seeds and trees from searching space. An optimum tree in population or arbitrarily elected tree position was utilized for all seed productions. An essential parameter of the TSA technique is the ST control parameter. This parameter ensures a variety of seed production. This variety has been recognized by employing the formulas in Eqs. (19) and (20). Once the arbitrarily elected number is lesser than the ST parameter value, the 1st formula is utilized, and once it is greater, the 2nd formula is employed.

$$S_{i,j} = T_{i,j} + \alpha_{i,j} \times \left(B_j - T_{r,j}\right) \tag{19}$$

$$S_{i,j} = T_{i,j} + \alpha_{i,j} \times \left(T_{i,j} - T_{r,j}\right) \tag{20}$$

Meanwhile, $S_{i,j}$ signifies the seeds produced. $T_{i,j}$ denotes the tree of a specific size. $\alpha_{i,j}$ stands for the random number created among [−1, 1]. $B_j$ implies the best tree. $T_{r,j}$ indicates the tree arbitrarily elected from the population. In the early searching space, the primary population (tree places) was stated to have feasible performance in optimizer issues, which was achieved utilizing Eq. (21).

$$T_{i,j} = L_{j,\min} + r_{i,j}\left(H_{j,\max} - L_{j,\min}\right) \tag{21}$$

$L_{j,\min}$, and $H_{j,\max}$ denote the low and upper bounds of searching space, respectively. $r_{i,j}$ signifies the arbitrarily created value between zero and one. To select the optimum performance in the population, the function $f$ is determined, which is utilized in Eq. (22).

$$B = \min f\left(\vec{T}_i\right) i = 1, 2, \ldots, N \tag{22}$$

In this case, $N$ denotes the trees from the population. At this point, trees are 1st planted from the searching space (a); after that, seed production is applied to all the trees (b), and lastly seed chosen is implemented (c). The pseudocode of TSA is provided in Algorithm 1.

**Algorithm 1:** Pseudocode of TSA

Step1: The initialization of the algorithm
Arbitrarily created tree places on the $D$-dimension searching space utilizing Eq. (3). Estimate the tree places by the fitness function.
Choose the optimum place utilizing Eq. (22).
Step2: Searching with seeds
For every tree
Choose the amount of seeds created for this tree.
For each seed
FOR every size
IF (rand < $ST$)
Upgrade this size utilizing Eq. (19).
ELSE
Upgrade this size utilizing Eq. (20).
END IF
END FOR
END FOR
Choose the optimum seed and examine it with a tree.
If the seed place is superior to the tree place, the seed alternates for this tree.
END FOR
Step3: Chosen better performance
Choose the best performance of the population.
If the new optimum performance is superior to the preceding optimum solution, the new better performance will replace the preceding better performance.
Step4: Testing the end situation

Fitness selection is a considerable factor influencing the performance of TSA. The hyperparameter selection procedure includes a solution encoding method to estimate the effectiveness of candidate solutions. In this work, TSA reflects accuracy as the main principle for designing FF, as expressed below:

$$Fitness = \max\ (P) \tag{23}$$

$$P = \frac{TP}{TP + FP} \tag{24}$$

From the above mentioned expression, $TP$ and $FP$ signify true positive and false positive values, respectively.

## 4 Result Analysis and Discussion

This section examines the performance of the GTODL-BADC technique under the Bot-IoT Database [28]. It includes 900 samples and two classes, as represented in Table 1.

**Table 1:** Details of the dataset

| Classes | No. of instances |
|---|---|
| Botnet | 450 |
| Normal | 450 |
| **Total instances** | **900** |

Fig. 3 displays the confusion matrices accomplished by the GTODL-BADC method under 80:20 and 70:30 of the training phase (TRPH)/testing phase (TSPH). The attained outcomes indicate proficient recognition under two classes.

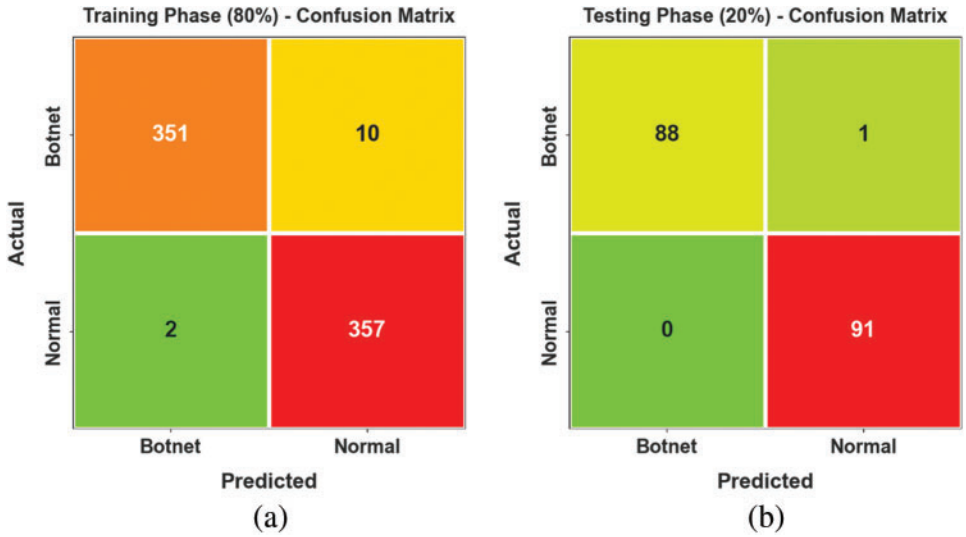


**Figure 3:** (Continued)

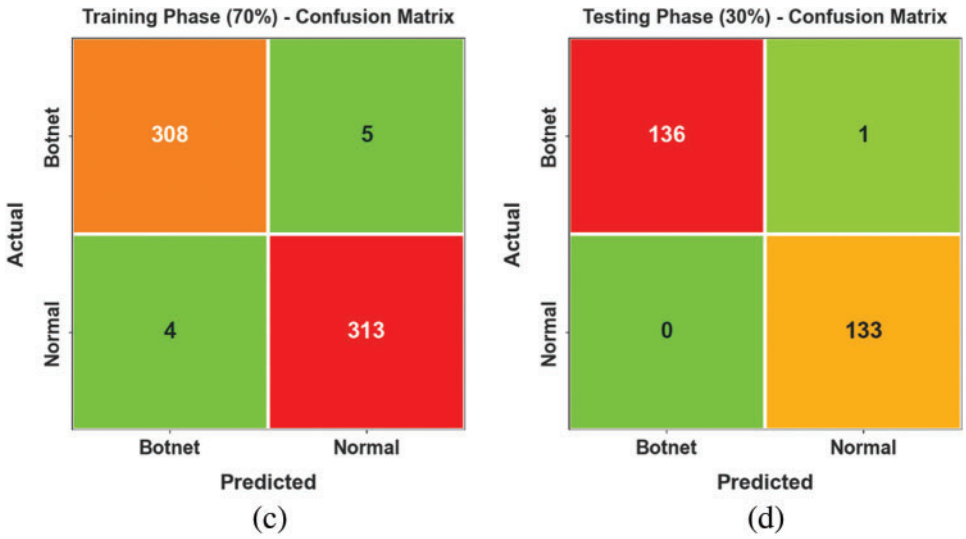


**Figure 3:** Confusion matrices of GTODL-BADC model (a, b) 80:20 TRPH/TSPH and (c, d) 70:30 of TRPH/TSPH

In Table 2 and Fig. 4, the botnet recognition analysis of the GTODL-BADC technique can be illustrated on 80:20 of TRPH/TSPH. The results depict that the GTODL-BADC technique achieves effectual botnet detection results. With 80% of TRPH, the GTODL-BADC technique gains average $accu_y$, $prec_n$, $reca_l$, $F_{score}$, and $AUC_{score}$ values of 98.34%, 98.35%, 98.34%, 98.33%, and 98.34%. Additionally, with 30% of TSPH, the GTODL-BADC methodology gets average $accu_y$, $prec_n$, $reca_l$, $F_{score}$, and $AUC_{score}$ values of 99.44%, 99.46%, 99.44%, 99.44%, and 99.44%, correspondingly.

**Table 2:** Botnet recognition outcomes of the GTODL-BADC approach on 80:20 TRPH/TSPH

| Classes | $Accu_y$ | $Prec_n$ | $Reca_l$ | $F_{score}$ | $AUC_{score}$ |
|---|---|---|---|---|---|
| **TRPH (80%)** | | | | | |
| Botnet | 97.23 | 99.43 | 97.23 | 98.32 | 98.34 |
| Normal | 99.44 | 97.28 | 99.44 | 98.35 | 98.34 |
| **Average** | **98.34** | **98.35** | **98.34** | **98.33** | **98.34** |
| **TSPH (20%)** | | | | | |
| Botnet | 98.88 | 100.00 | 98.88 | 99.44 | 99.44 |
| Normal | 100.00 | 98.91 | 100.00 | 99.45 | 99.44 |
| **Average** | **99.44** | **99.46** | **99.44** | **99.44** | **99.44** |

Table 3 and Fig. 5 show the botnet recognition analysis of the GTODL-BADC technique under 70:30 of TRPH/TSPH. The acquired outcomes show that the GTODL-BADC technique gets successful botnet detection outcomes. According to 70% of TRPH, the GTODL-BADC technique achieves average $accu_y$, $prec_n$, $reca_l$, $F_{score}$, and $AUC_{score}$ values of 98.57%, 98.57%, 98.57%, 98.57%, and 98.57%. Besides, on 30% of TSPH, the GTODL-BADC methodology gives average $accu_y$, $prec_n$, $reca_l$, $F_{score}$, and $AUC_{score}$ values of 99.64%, 99.63%, 99.64%, 99.63%, and 99.64%, respectively.

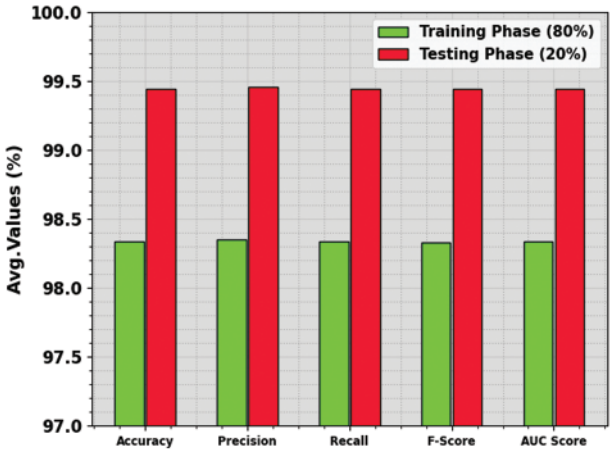


**Figure 4:** Average outcomes of the GTODL-BADC approach on 80:20 TRPH/TSPH

**Table 3:** Botnet recognition analysis of the GTODL-BADC approach on 70:30 TRPH/TSPH

| Classes | $Accu_y$ | $Prec_n$ | $Reca_l$ | $F_{score}$ | $AUC_{score}$ |
|---|---|---|---|---|---|
| **TRPH (70%)** | | | | | |
| Botnet | 98.40 | 98.72 | 98.40 | 98.56 | 98.57 |
| Normal | 98.74 | 98.43 | 98.74 | 98.58 | 98.57 |
| **Average** | **98.57** | **98.57** | **98.57** | **98.57** | **98.57** |
| **TSPH (30%)** | | | | | |
| Botnet | 99.27 | 100.00 | 99.27 | 99.63 | 99.64 |
| Normal | 100.00 | 99.25 | 100.00 | 99.63 | 99.64 |
| **Average** | **99.64** | **99.63** | **99.64** | **99.63** | **99.64** |

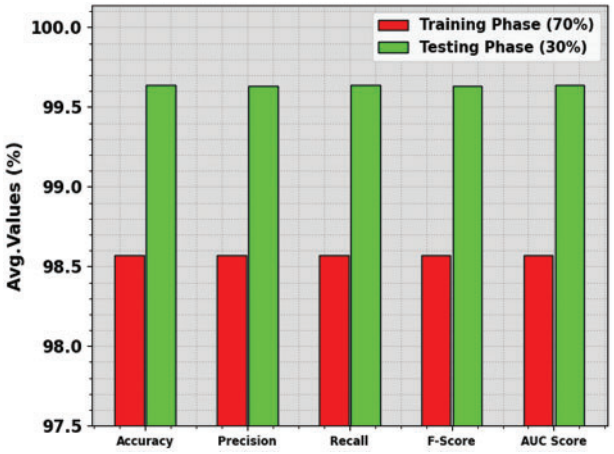


**Figure 5:** Average outcome of the GTODL-BADC approach with 70:30 TRPH/TSPH

The $accu_y$ curves for training (TR) and validation (VL) displayed in Fig. 6 for the GTODL-BADC technique with 70:30 TRPH/TSPH provide valued insights into its efficiency with numerous epochs. Precisely, it can reliably enhance in both TR and TS $accu_y$ to refining epochs, showing the model's ability to recognize and learn patterns in these data of TR and TS. The upgrade trends in TS $accu_y$ display the model's adaptability for the dataset of TR and its capabilities for producing correct predictions on unseen data, underscoring supreme generalization proficiencies.

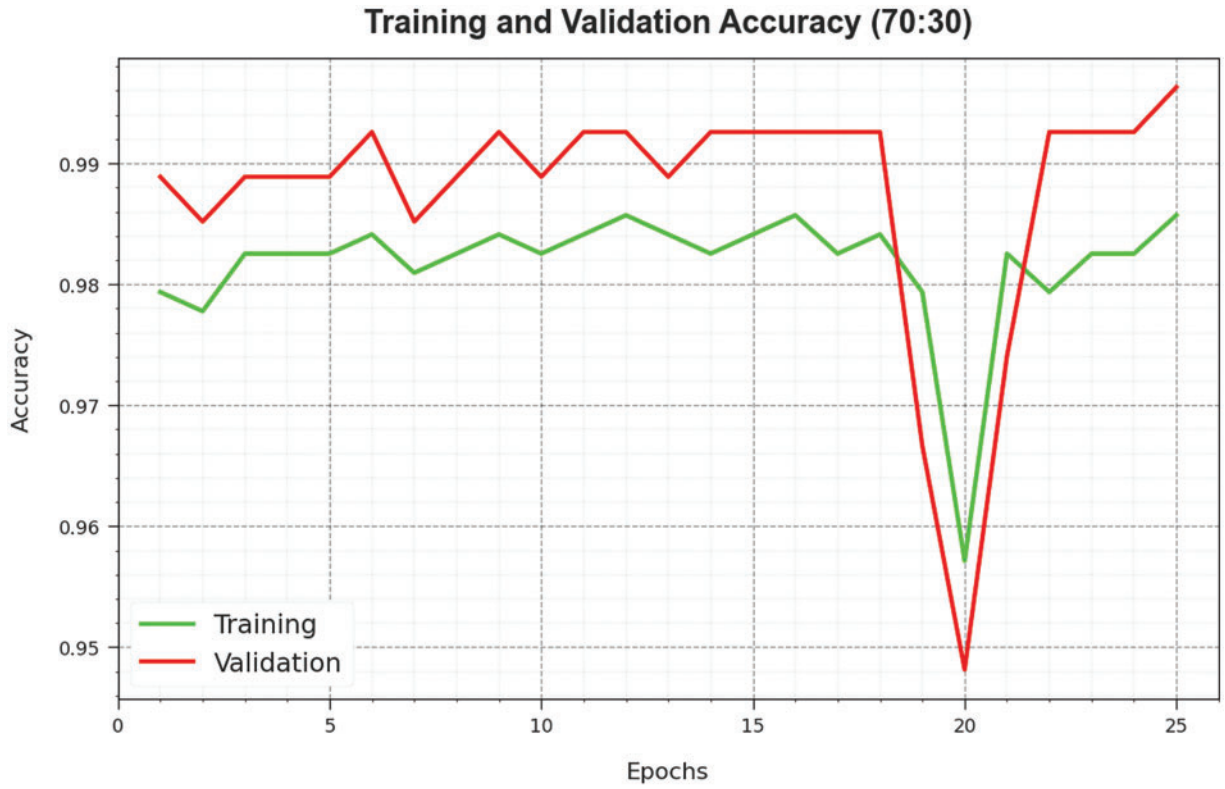


**Figure 6:** ***Accu***$_y$ curve of the GTODL-BADC approach with 70:30 TRPH/TSPH

Fig. 7 specifies a wide-ranging overview of the TR and TS loss values to the GTODL-BADC methodology with 70:30 TRPH/TSPH in diverse epochs. This TR loss constantly lessened as the model grew in weight to reduce classification errors with these datasets. These loss curves considerably indicate the model's alignment with the TR database, underscoring proficiencies for capturing patterns. The continuous parameters are modified in the GTODL-BADC technique to minimize discrepancies between actual and predicted TR labels.

As regards the PR curve shown in Fig. 8, the findings confirm that the GTODL-BADC technique on 70:30 TRPH/TSPH reliably achieves boosted PR values in every class. These outcomes underscore the model's efficient capacity for discerning among many classes, emphasizing its effectiveness in recognizing class labels precisely.

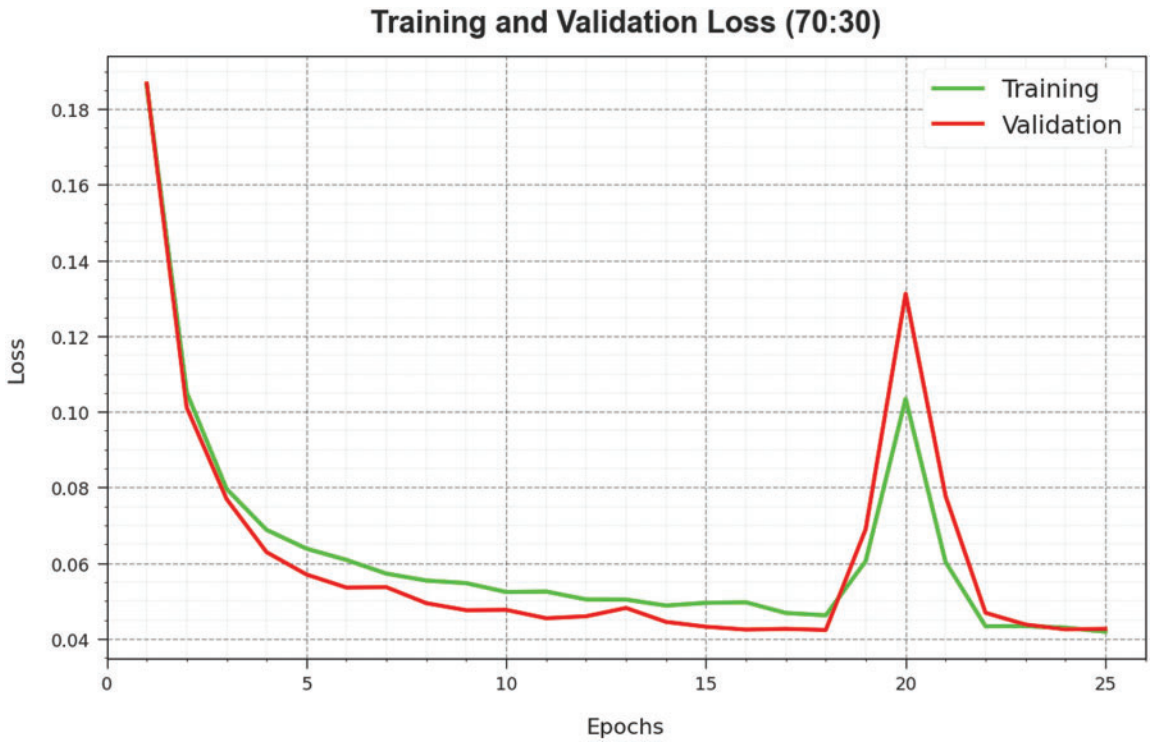


**Figure 7:** Loss curve of the GTODL-BADC approach on 70:30 TRPH/TSPH

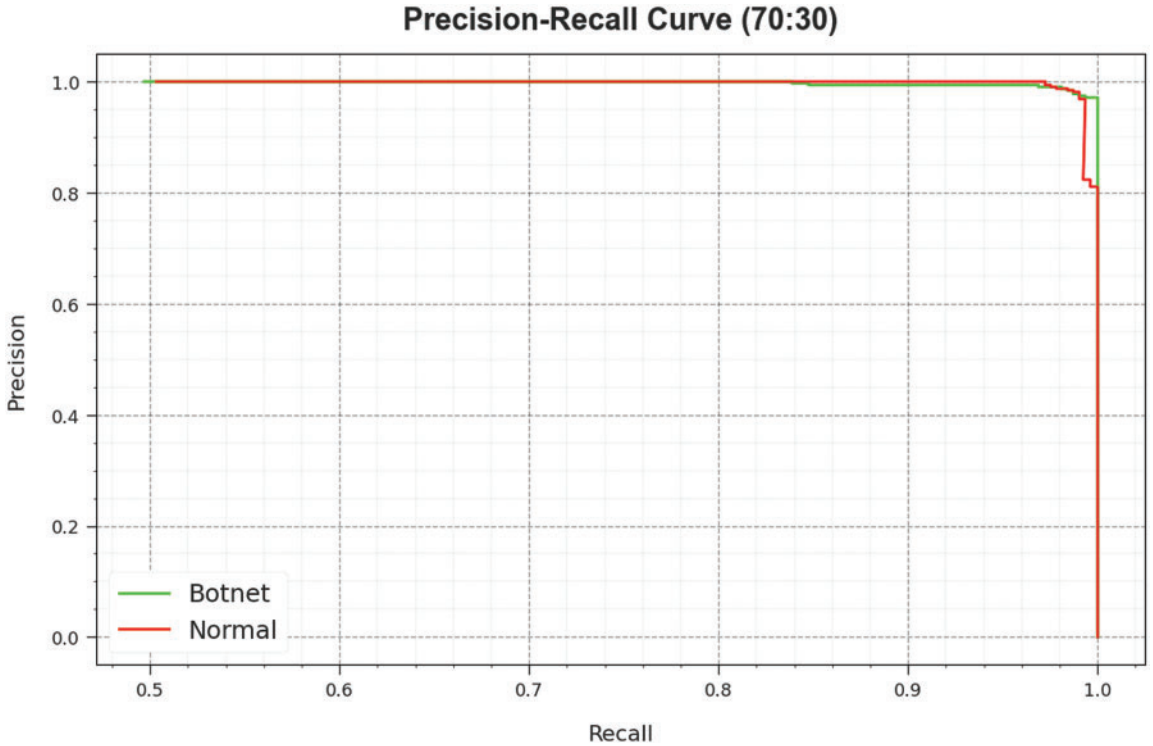


**Figure 8:** PR curve of the GTODL-BADC approach under 70:30 TRPH/TSPH

Additionally, Fig. 9 reveals ROC curves generated by the GTODL-BADC technique with 70:30 TRPH/TSPH, signifying its proficiency in differentiating amongst classes. These curves give valued insights into how the trade-off between FPR and TPR varied at diverse classification epochs and thresholds. The acquired outcomes emphasize the model's exact classification efficiency in diverse class labels, emphasizing its effectiveness in addressing several classification challenges.

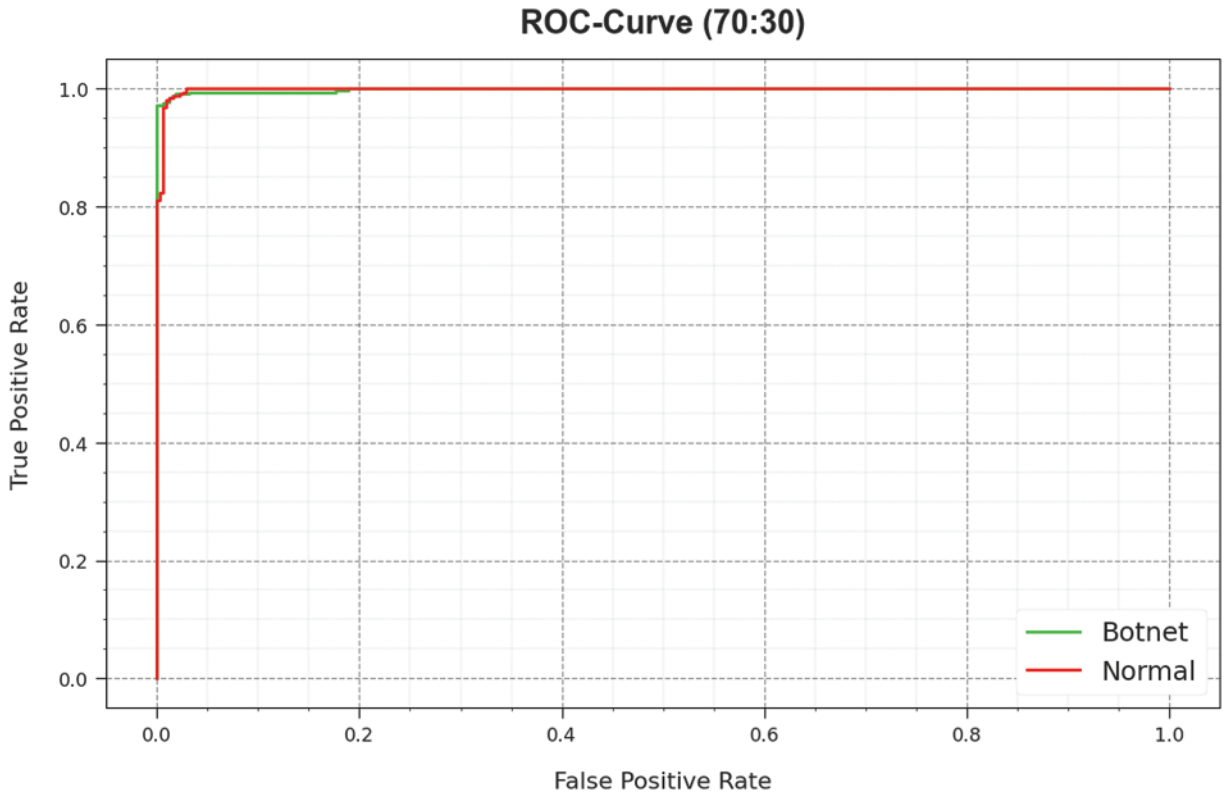


**Figure 9:** ROC curve of the GTODL-BADC approach with 70:30 TRPH/TSPH

Table 4 demonstrates a comparison analysis of the GTODL-BADC technique with recent approaches [29]. In Fig. 10, a brief analysis of the GTODL-BADC technique in terms of $accu_y$. The results indicate that the GTODL-BADC technique boosts performance. Based on $accu_y$, the GTODL-BADC technique gains an increased $accu_y$ of 99.64%, whereas the BNTCB-POADL, BDCRSO-DL, P2PBDS, MTCCNN, DT, host, and FLANN models obtain decreased $accu_y$ values of 99.54%, 99.15%, 94.52%, 95.04%, 97.91%, 92.95%, and 98.96%, respectively.

**Table 4:** Comparison outcome of the GTODL-BADC approach with other models

| Methods | $Accu_y$ | $Prec_n$ | $Reca_l$ | $F_{score}$ |
|---|---|---|---|---|
| GTODL-BADC | 99.64 | 99.63 | 99.64 | 99.63 |
| BNTCB-POADL | 99.54 | 99.41 | 99.51 | 99.48 |
| BDCRSO-DL | 99.15 | 96.93 | 99.21 | 98.04 |
| P2PBDS | 94.52 | 95.65 | 96.71 | 94.68 |

(Continued)

**Table 4 (continued)**

| Methods | $Accu_y$ | $Prec_n$ | $Reca_l$ | $F_{score}$ |
|---|---|---|---|---|
| MTCCNN | 95.04 | 95.89 | 97.81 | 96.20 |
| DT | 97.91 | 94.97 | 95.97 | 95.69 |
| Host-based model | 92.95 | 95.36 | 96.86 | 96.61 |
| FLANN | 98.96 | 96.34 | 97.89 | 97.12 |

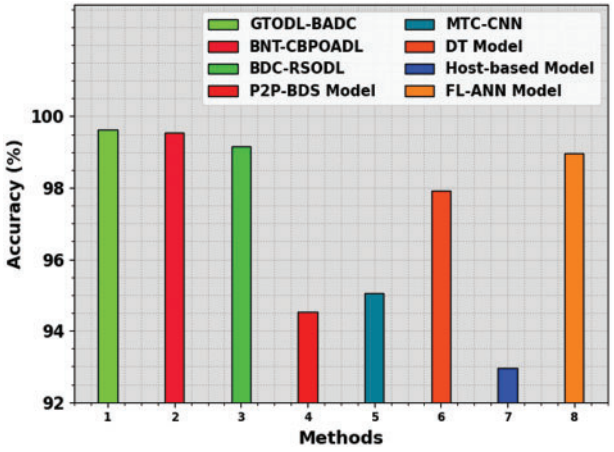


**Figure 10:** $Accu_y$ analysis of the GTODL-BADC approach with other models

In Fig. 11, a comprehensive analysis of the GTODL-BADC technique concerning $prec_n$, $reca_l$, and $F_{score}$, the accomplished outcomes show the GTODL-BADC technique achieved increased performance. According to $prec_n$, the GTODL-BADC technique gets improved $prec_n$ of 99.63% whereas the BNTCB-POADL, BDCRSO-DL, P2PBDS, MTCCNN, DT, host, and FLANN methodologies acquire diminished $prec_n$ values of 99.41%, 96.93%, 95.65%, 95.89%, 94.97%, 95.36%, and 96.34%. Based on $reca_l$, the GTODL-BADC method gives an increased $reca_l$ of 99.64%, but the BNTCB-POADL, BDCRSO-DL, P2PBDS, MTCCNN, DT, host, and FLANN methodologies get lessened $reca_l$ values of 99.51%, 99.21%, 96.71%, 97.81%, 95.97%, 96.86%, and 97.89%. Also, on $F_{score}$, the GTODL-BADC technique offers an increased $F_{score}$ of 99.63%, however, the BNTCB-POADL, BDCRSO-DL, P2PBDS, MTCCNN, DT, host, and FLANN methodologies obtain reduced $F_{score}$ values of 99.48%, 98.04%, 94.68%, 96.20%, 95.69%, 96.61%, and 97.12%, respectively.

These achieved results ensured the accurate and automated botnet detection results of the GTODL-BADC technique.

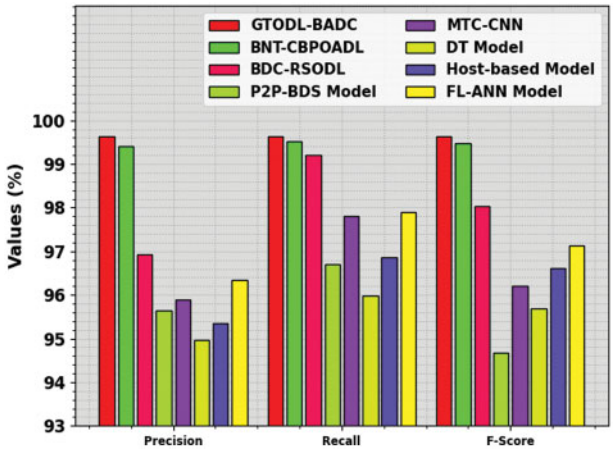


**Figure 11:** Comparative analysis of the GTODL-BADC approach with other models

## 5 Conclusion

In this article, a novel GTODL-BADC methodology is presented. The GTODL-BADC methodology follows FS with optimal DL-based classification for accomplishing security in an IoT environment. For data preprocessing, the min-max data normalization approach is primarily used. The GTODL-BADC technique uses the GTO algorithm to select features and elect optimal feature subsets. Moreover, the MHA-LSTM-based classification model can be applied for botnet detection. Finally, TSA can be used to select the optimum hyperparameter for the MHA-LSTM technique. The experimental validation of the GTODL-BADC technique was tested on a benchmark dataset. The simulation results highlighted that the GTODL-BADC technique demonstrates promising performance in the botnet detection process. The GTODL-BADC approach may comprise scalability threats with large-scale IoT utilization and the requirement for additional analysis across various IoT environments. Future studies may explore incorporating further security layers and improving real-time threat response abilities to reduce growing botnet outbreaks effectually.

**Acknowledgement:** None.

**Funding Statement:** None.

**Availability of Data and Materials:** Data sharing does not apply to this article as no dataset were generated during the current study.

**Conflicts of Interest:** The author declares that he has no conflict of interest.

## References

[1] S. Alrayes *et al.*, "Modelling of botnet detection using barnacles mating optimizer with machine learning model for internet of things environment," *Electronics*, vol. 11, no. 20, pp. 3411, 2022. doi: 10.3390/electronics11203411.

[2] W. Ding, M. Abdel-Basset, and R. Mohamed, "DeepAK-IoT: An effective deep learning model for cyberattack detection in IoT networks," *Inf. Sci.*, vol. 634, no. 3, pp. 157–171, 2023. doi: 10.1016/j.ins.2023.03.052.

[3] R. Sudhakar and S. Kumar, "ABBDIoT: Anomaly-based botnet detection using machine learning model in the internet of things network," in *Int. Conf. IoT, Intell. Comput. Secur.: Sel. Proc. IICS 2021*, Singapore, Springer Nature Singapore, 2023, pp. 235–245.

[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. doi: 10.1109/ACCESS.2022.3204001.

[5] M. Sabir, J. Ahmad, and D. M. Alghazzawi, "A lightweight deep autoencoder scheme for cyberattack detection in the internet of things," *Comput. Syst. Sci. Eng.*, vol. 46, no. 1, pp. 57–72, 2023. doi: 10.32604/csse.2023.034277.

[6] I. Apostol, M. Preda, C. Nila, and I. Bica, "IoT botnet anomaly detection using unsupervised deep learning," *Electronics*, vol. 10, no. 16, pp. 1876, 2021. doi: 10.3390/electronics10161876.

[7] Y. Masoudi-Sobhanzadeh and S. Emami-Moghaddam, "A realtime IoT-based botnet detection method using a novel two-step feature selection technique and the support vector machine classifier," *Comput. Netw.*, vol. 217, pp. 109365, 2022. doi: 10.1016/j.comnet.2022.109365.

[8] M. Cui, K. Wang, X. Ding, Z. Xu, X. Wang and P. Shi, "Multi-view stable feature selection with adaptive optimization of view weights," *Knowl.-Based Syst.*, vol. 299, no. 10, pp. 111970, 2024. doi: 10.1016/j.knosys.2024.111970.

[9] M. M. Alani, "BotStop: Packet-based efficient and explainable IoT botnet detection using machine learning," *Comput. Commun.*, vol. 193, no. 7, pp. 53–62, 2022. doi: 10.1016/j.comcom.2022.06.039.

[10] G. Kirubavathi and U. K. Sridevi, "Detection of IoT botnet using machine learning and deep learning techniques," 2023. doi: 10.21203/rs.3.rs-2630988/v1.

[11] C. Wei, G. Xie, and Z. Diao, "A lightweight deep learning framework for botnet detecting at the IoT edge," *Comput. Secur.*, vol. 129, pp. 103195, 2023.

[12] A. A. Hezam, S. A. Mostafa, A. A. Ramli, H. Mahdin, and B. A. Khalaf, "Deep learning approach for detecting botnet attacks in IoT environment of multiple and heterogeneous sensors," in *Int. Conf. Adv. Cyber Secur.*, Singapore, Springer Singapore, Jul. 28–31 2021, 2021, pp. 317–328.

[13] M. A. Haq and M. A. R. Khan, "DNNBoT: Deep neural network-based botnet detection and classification," *Comput. Mater. Contin.*, vol. 71, no. 1, pp. 1729–1750, 2022. doi: 10.32604/cmc.2022.020938.

[14] S. Khan and A. B. Mailewa, "Discover botnets in IoT sensor networks: A lightweight deep learning framework with hybrid self-organizing maps," *Microprocess. Microsyst.*, vol. 97, pp. 104753, 2023.

[15] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3930–3944, 2021.

[16] T. Hasan *et al.*, "Securing industrial Internet of Things against botnet attacks using hybrid deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, pp. 5, 2022.

[17] M. M. Alani, A. I. Awad, and E. Barka, "ARP-PROBE: An ARP spoofing detector for Internet of Things networks using explainable deep learning," *Internet of Things*, vol. 23, pp. 100861, 2023.

[18] M. Asadi, "Detecting IoT botnets based on the combination of cooperative game theory with deep and machine learning approaches," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 12, pp. 5547–5561, 2022. doi: 10.1007/s12652-021-03185-x.

[19] A. Nazir *et al.*, "Advancing IoT security: A systematic review of machine learning approaches for the detection of IoT botnets," *J. King Saud Univ.—Comput. Inf. Sci.*, vol. 35, no. 10, pp. 101820, 2023. doi: 10.1016/j.jksuci.2023.101820.

[20] A. Nazir *et al.*, "Collaborative threat intelligence: Enhancing IoT security through blockchain and machine learning integration," *J. King Saud Univ.—Comput. Inf. Sci.*, vol. 36, no. 2, pp. 101939, 2024. doi: 10.1016/j.jksuci.2024.101939.

[21] L. Abualigah *et al.*, "Improved prairie dog optimization algorithm by dwarf mongoose optimization algorithm for optimization problems," *Multimed. Tools Appl.*, vol. 83, no. 11, pp. 32613–32653, 2024. doi: 10.1007/s11042-023-16890-w.

[22] A. K. Sangaiah, A. Javadpour, and P. Pinto, "Towards data security assessments using an IDS security model for cyber-physical smart cities," *Inf. Sci.*, vol. 648, no. 1, pp. 119530, 2023. doi: 10.1016/j.ins.2023.119530.

[23] A. Javadpour, P. Pinto, D. Ja'fari, and W. Zhang, "DMAIDPS: A distributed multi-agent intrusion detection and prevention system for cloud IoT environments," *Cluster Comput.*, vol. 26, no. 1, pp. 367–384, 2023. doi: 10.1007/s10586-022-03621-3.

[24] N. A. Hikal and M. M. Elgayar, "Enhancing IoT botnets attack detection using machine learning-IDS and ensemble data preprocessing technique," in *Internet of Things—Appl. Future: Proc. ITAF 2019*, Singapore, Springer Singapore, 2020, pp. 89–102.

[25] A. Shaheen, R. El-Sehiemy, A. El-Fergany, and A. Ginidi, "Fuel-cell parameter estimation based on improved gorilla troops technique," *Sci. Rep.*, vol. 13, no. 1, pp. 8685, 2023. doi: 10.1038/s41598-023-35581-y.

[26] F. Shahid *et al.*, "1D convolutional LSTM-based wind power prediction integrated with PkNN data imputation technique," *J. King Saud Univ.—Comput. Inf. Sci.*, vol. 35, no. 10, pp. 101816, 2023. doi: 10.1016/j.jksuci.2023.101816.

[27] M. Beşkirli and M. S. Kiran, "Optimization of butterworth and bessel filter parameters with improved tree-seed algorithm," *Biomimetics*, vol. 8, no. 7, pp. 540, 2023. doi: 10.3390/biomimetics8070540.

[28] The Bot-IoT Dataset. Accessed: Jan. 12, 2024. [Online]. Available: https://research.unsw.edu.au/projects/bot-iot-dataset

[29] F. Alrowais, M. M. Eltahir, S. S. Aljameel, R. Marzouk, G. P. Mohammed and A. S. Salama, "Modelling of botnet detection using chaotic binary pelican optimization algorithm with deep learning on internet of things environment," *IEEE Access*, vol. 11, pp. 130618–130626, 2023. doi: 10.1109/ACCESS.2023.3332690.