Improving Network Security using Tunicate Swarm Algorithm with Stacked Deep Learning Model on IoT Environment

Abedallah Z. Abualkishik *, Rasha Almajed

American University in the Emirates, Dubai, UAE

Emails: abedallah.abualkishik@aue.ae, rasha.almajed@aue.ae

Abstract

The Internet of Things (IoT) represents important security vulnerabilities, increasing difficulties in cyberattacks. Attackers employ these vulnerabilities to establish distributed denial-of-service (DDoS) attacks, compromising availability and causing financial losses to digital platforms. Newly, numerous Machine Learning (ML) and Deep Learning (DL) approaches have been presented for the identification of botnet attacks in IoT networks. By analyzing the patterns of communication and behavior of IoT devices, DL algorithms will be differentiated between malicious and normal activity, therefore supporting the earlier detection and avoidance of botnet attacks. This is essential to protect the integrity and security of IoT systems that can be increasingly vulnerable to botnet-driven attacks because of their limited security measures and often large-scale applications. In this aspect, this study designs an innovative tunicate swarm algorithm with stacked deep learning for botnet detection (TSASDL-BD) technique for IoT platforms. The purpose of the TSASDL-BD technique is to recognize the botnets and achieve maximum network security. In the TSASDL-BD technique, the TSA is applied for the effectual feature selection process, which aids in reducing the dimensionality problem. For botnet detection, the TSASDL-BD technique makes use of the stacked long short-term memory gated recurrent unit (SLSTM-GRU) model. Finally, the artificial humming algorithm (AHA) can be used for the optimal selection of the hyperparameter values of the SLSTM+GRU system. The outcome analysis of the TSASDL-BD method on the benchmark database takes place. The extensive outcomes stated that the TSASDL-BD approach gains maximum detection results over other algorithms with respect of different measures

Keywords: Internet of Things; Intrusion Detection System; Denial-of-Service; Artificial Humming Algorithm; Feature selection

1. Introduction

The development of IoT devices has carried numerous advantages to rural, home and industrial areas. These devices can be improved computational efficiency, permitting highly advanced functionalities and applications [1]. But, it also implies that there is a superior requirement for security actions to avoid these devices from being employed due to attack vulnerabilities, and utilized in botnets. Botnets are a variety of devices that execute orchestrated attacks in servers and network services, aiming to cause inaccessibility [2]. It is formed by affected endpoints like Internet of Things (IoTs) devices, wireless routers, computers and mobile phones. Recently, constructed with a concentration on ease of use, without being involved in confirming security development. While affected IoT devices typically serve for attacking huge businesses, banks and government services [3].

Intrusion Detection Systems (IDS) utilize different techniques to identify intrusions [4]. The first technique is signature-based detection, which includes observing for known intrusions in the form of signatures, rules or patterns [5]. This technique can only be efficient against known attacks and cannot identify zero-day or unknown attacks. Zero-day attacks employ old vulnerabilities or new vulnerabilities variously, creating recognition by signatures ineffectual for detection [6]. The second approach is anomaly-based detection, which contains identifying abnormal behavior with the comparison of normal or predicted behavior. This algorithm can identify an extensive of malicious intrusions; but, it can be also limit the possible of false positives [7]. Anomaly-based approaches are categorized into statistical techniques, machine learning (ML) algorithms and other approaches

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

depending on data mining and game theory models. Botnet attack detection in IoT networks is developed as a classification issue. In binary classification, all samples at a network traffic packet can be categorized as both malicious and benign depending on specific predetermined features.

Alternatively, a certain class of botnet attack has been detected in multi-class classification [8]. Consequently, Artificial Intelligence (AI) methods could be attained excellent effectiveness to handle classification methods in diverse application fields comprising voltage stability evaluation of power systems amongst several others. Particularly, different ML and Deep Learning (DL) are established for classifying network traffic data in IoT environments [9]. The architectures learn the selective features of malicious and benign traffic employing distinct models namely Long Short-Term Memory (LSTM), Support Vector Machine (SVM), Gated Recurrent Unit (GRU), Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs) and Random Forest (RF). For detailed knowledge, wide-ranging evaluations and studies under the application of DL and ML in intrusion detection [10].

This study designs an innovative tunicate swarm algorithm with stacked deep learning for botnet detection (TSASDL-BD) method for IoT infrastructure. The purpose of the TSASDL-BD technique is to recognize the botnets and achieve maximum network security. In the TSASDL-BD technique, the TSA is applied for the effectual feature selection process, which aids in reducing the dimensionality problem. For botnet detection, the TSASDL-BD technique makes use of the stacked long short-term memory+gated recurrent unit (SLSTM-GRU) model. Finally, the artificial humming algorithm (AHA) can be used for the optimal choice of the hyperparameter values of the SLSTM+GRU algorithm. The outcome analysis of the TSASDL-BD system under the benchmark dataset takes place.

2. Related works

Al-Fawa'reh et al. [11] presented MalBoT-DRL, a robust malware botnet detector employing deep reinforcement learning (DRL). This method combines damped improvemetnal statistics with an attention-gaining method, incorporation cannot be widely discovered in the study. This combination assists MalBoT-DRL to dynamically modify for varying malware patterns within IoT platforms. In [12], the authors developed an integrated Mutual Information (MI) assisted FS algorithm with ML techniques. The N-BaIoT standard database has been employed that comprises multi-class and binary classifications. The FS technique integrates the MI algorithm, Principal Component Analysis (PCA) and ANOVA f-test at a finely-granulated identification. In the classification stage, numerous ensembles and specific classifiers are employed. Soe et al. [13] introduced an ML-assisted botnet attack detection architecture with a sequential identification model. An effectual FS technique has been modified for implementing a lightweight identification algorithm with superior effectiveness. The overall identification efficiency obtains higher accuracy employing 3 various ML approaches, comprising J48-DT, NB, and ANNs.

In [14], the authors developed a botnet detection architecture utilizing the barnacles mating optimization with ML (BND-BMOML) method. This introduced algorithm primarily follows a data standardization model. The BMO technique was exploited for FS. The Elman neural network (ENN) approach has been implemented for botnet detection. Secondarily, the architecture implements a chicken swarm optimizer (CSO) technique for the tuning appraoch. Almuqren et al. [15] implemented a Hybrid Meta-heuristic with ML-based Botnet Detection (HMMLB-BND) system. The modified firefly optimizer (MFFO) algorithm was employed for extraction. A hybrid CNN-quasi-RNN (QRNN) algorithm was also utilized. To optimize the hyperparameter tuning method, the chaotic butterfly optimizer algorithm (CBOA) was implemented. Abu Al-Haija and Al-Dala'ien [16] suggested an ELM termed ELBA-IoT that models deep features of IoT environment as well as exploits ensemble learning for detecting anomalous network traffic. Additionally, this IoT-assisted botnet detection method represents the assessment of 3 diverse ML approaches, which consist of DT methods such as RUSBoosted, AdaBoosted, and bagged.

In [17], a hybrid ML technique was developed to identify botnet attacks in network traffic. The presented technique integrates a 3 ML approaches like rule-based methods, k-means, and DT. The effectiveness confirmation of this developed analysis was accompanied by applying the CTU-13 database and feature extraction in the BMO for network traffic flow. Pokhrel et al. [18] projected a new architecture employing the ML technique. Numerous ML algorithms namely the NB method, Multi-layer Perception-ANN (MLP-ANN), and KNN have been employed. Feature engineering and SMOTE are also incorporated with ML methods.

3. The proposed model

In this article, we design an innovative TSASDL-BD algorithm for an IoT platform. The purpose of the TSASDL-BD method is to recognize the botnets and achieve maximum network security. This comprises main processes

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

like TSA-based FS, SLSTM-GRU-based classification, and AHA-assisted hyperparameter tuning. Fig. 1 illustrates the workflow of TSASDL-BD system

A. Feature selection using TSA

At this phase, the TSA can be applied for the effectual FS process, which aids in reducing the dimensionality problem. TSA refers to an innovative bio-derived meta-heuristic model [19]. The design appeals to stimulation from the tunicate swarm's movements of intelligent cooperative foraging and jet propulsion. Early accent relies on inhibiting clashes between survey units in the propulsion procedure. The effort aligns with the optimum neighboring way that ends in merging towards the leading search agents. To determine new location and stop clashes among SAs, Vector $\vec{A}$ is calculated using the following Eq. (1).

$$\overrightarrow{AC} = \frac{\overrightarrow{GV}}{\overrightarrow{SF}} \tag{1}$$

where $\overrightarrow{GV}$ represents the force of gravity that is planned as developed by Kaur et al. The calculation of social forces $\overrightarrow{SF}$ amongst SAs executed by utilizing the following Eq. (2).

$$\overrightarrow{SF} = [\mathrm{Min}_s + r_3 \times (\mathrm{Max}_s - Min)] \tag{2}$$

where Min_s and Max_s signify primary and subsequent velocities of social interaction, correspondingly.

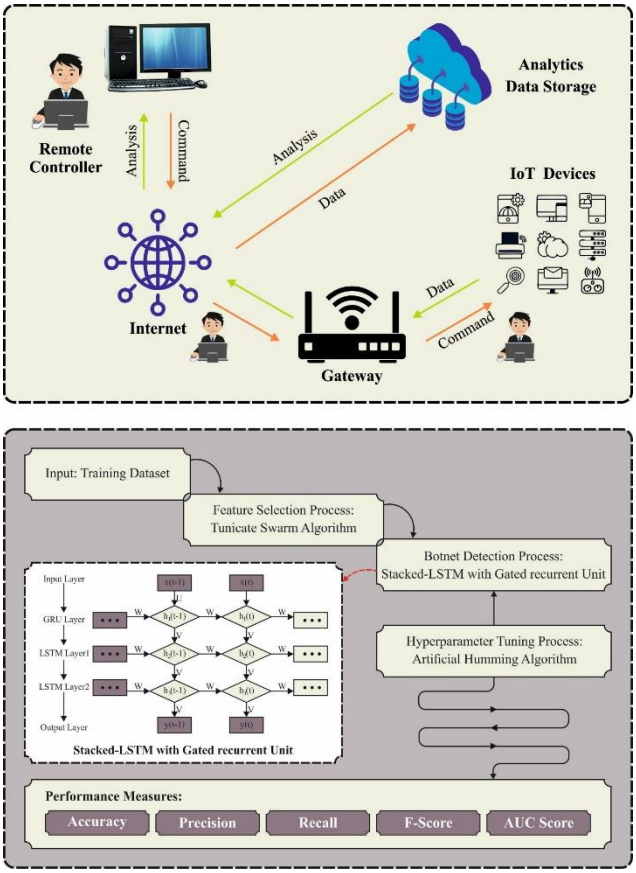

Figure 1: Workflow of TSASDL-BD approach

Succeeding to clash avoidance, the following equation is used exactly to clear this occurrence.

$$\overrightarrow{Dst} = \left|Pos_{FS} - rand_1 \times \overrightarrow{Pos_T}(x)\right| \tag{3}$$

$\overrightarrow{Dst}$ is the distance vector among the place of food source (P_{FS}) and $(\overrightarrow{P_T})$ is the location of tunicate SA. x indicates the present round, r_4 represents a arbitrary number ranging from 0 to 1 and || is employed to complete the value.

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

$\overrightarrow{Dst}$ denotes the distance vector spanning amid the place of the food sources (Pos) and $\left(\overrightarrow{Pos_T}\right)$ denotes a place of the tunicate search agent. Variable x indicates the existing round, $rand_1$ represents a random value of 0 to 1.

To prompt this occurrence accurately, subsequent expression displays the optimum solution as upgraded location $\left(\overrightarrow{P_T}(x)\right)$ of tunicate SAs to the food source.

For an indication of this occurrence mathematically, the below-mentioned mathematical formula validates the ideal solution as revised place $\overrightarrow{Pos_T}(x)$ of tunicate SAs to the food source.

$$\overrightarrow{Pos_T}(x) = Pos_{FS} + \vec{A} \times \vec{D}\ if r_4 \geq 0.5 \quad (4)$$

$$\overrightarrow{Pos_T}(x) = Pos_{FS} - \vec{A} \times \vec{D}\ if r_4 \leq 0.5 \quad (5)$$

The tunicate establishes cooperative behavior, an idea categorized by an organization of other SAs locations with those of greater SAs. The equation of tunicate's collective behavior can be shown as given:

$$\overrightarrow{Pos_T}(x+1) = \left[\left\{\overrightarrow{Pos_T}(x) + \overrightarrow{Pos_T}(x+1)\right\}\right]/(2+r) \quad (6)$$

The fitness function (FF) applied in the TSA model is developed to have a balance amongst the no. of preferred features in every solution (minimum) and the classification accurateness (maximum) acquired through these certain features, Eq. (7) means the FF to determine solutions.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \quad (7)$$

$|R|$ is the cardinality of the preferred subsets, $\gamma_R(D)$ denotes the classification error rate, and $|C|$ represents the total no. of features in the database, α and β can be a 2 factors for the significance of subset length and classification quality. $\beta = 1 - \alpha$ and $\in$ [1,0].

B. Botnet detection of SLTM-GRU model

For botnet detection, the TSASDL-BD technique creates usage of the SLSTM-GRU model. The two-stage stacked LSTM (SLSTM) and GRU are compared with the standard architectures [20]. These models are discussed below.

To capture dependency at varying time scales whereas overwhelming the shortcomings of the LSTM-NN. GRU is simulated by the LSTM-developed model that can be simple for implementation and computation. Here, the input and output gates could be integrated to create a single gate called an upgrade gate. The upgrade gate is interposed among the candidate activation and previous activation, which determines how the unit upgrades its contents. GRU exploits the memory module instead of the hidden unit that confirms the gradient doesn't explode or vanish after several iterations. It can be closely linked to the reset unit that acts as an internal state of the GRU. The reset gate determines which data to either use or ignore the prior unseen state.

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \quad (8)$$

$$Z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \quad (9)$$

$$h_t = (1 - z_t) o h_{t-1} + z_t o \sigma_h\left(\sigma_g(W_h x_t + U_h(r_t o h_{t-1}) + b_h)\right) \quad (10)$$

Here, x_t and h_t denote the input and output vector, r_t and $z_t$shows the reset and update vector correspondingly, σ_g & σ_h indicate the sigmoid function and hyperbolic tangent, W, U and b refer to the parameter vectors and matrices.

The SLSTM depends on the LSTM-NN model. All the output layers of LSTM method performs as an input for succeeding blocks in the layer, thus providing the architecture with the ability for capturing the time series model and combine the learning view of prior layers whereas generating a high level of the concluding outcomes. Stacking LSTM in NN is used for enhancing the predictable performance although allowing the higher levels of temporal characteristics. It also assists in gradually making a high range of representation of the input dataset. Stacking helps to maximize the memory sizes of the system, even though the stacked system needs a larger dataset for training. Owing to less convergence rate of training errors, stacked NN suffers from degradation problems, though the error is varied from gradient vanishing problems.

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

The presented method mines the learning features of the LSTM and GRU methods because of the capability of capturing the features of these systems in a single framework. This output acts as an input of GRU architecture for processing the time-series data. The produced outcome of the GRU algorithm shows the outcome of hybrid SLSTM and GRU models.

C. Hyperparameter tuning using AHA technique

In conclusion, the AHA was implemented for the optimal choice of the hyperparameter values of the SLSTM+GRU model. The main inspiration for AHA comes from the foraging behaviors, memory capacity, and fight skills of hummingbirds [21]. The three mathematical formulae are applied to stimulate the foraging strategies (migrating, guided and territorial foraging,) of the hummingbird. A population of n humming birds are initialized arbitrarily and positioned under n food sources employing $x_i = L + r \cdot (U - L)$ but, the location of the i^{th} food supply is x_i, the upper and lower limits of a d-dimensional issue represents L and U correspondingly, $i = 1, \dots, n$, and r denotes random integer between zero and one:

$$VT_{i,j} = \begin{cases} 0, & i \neq j \\ null, & i = j' \end{cases} \quad (11)$$

Here, $i = 1, \dots, n$ and $j = 1, \dots, n$. $null$ represents the food supply where the hummingbird takes the food. In Eq. (11), 0 implies i^{th} hummingbird has visited j^{th} food sources at the existing iteration. The behavior of hummingbird was searched the food supply and maximum nectar refilling rate has been modelled during the guided foraging. A hummingbird fly to the food supply by executing axial, omnidirectional, and diagonal fight abilities. This can be described using the following equation:

$$v_i(t+1) = x_{i,trg}(t) + \alpha \cdot D \cdot \left(x_i(t) - x_{i,trg}(t)\right) \quad (12)$$

In Eq. (12), the candidate food supply position is $v_i(t+1)$, the guided factor subject towards the standard distribution through standard deviation 1 is α and mean 0, the i^{th} food supply place in time t is $x(t)$, the targeted food supply that i^{th} hummingbird to visit is $x_{i,trg}(t)$. The location of i^{th} food supply is updated using Eq. (13)

$$x_i(t+1) = \begin{cases} x_i(t), & f(x_i(t)) \leq f(v_i(t+1)) \\ v_i(t+1), & f(x_i(t)) > f(v_i(t+1)) \end{cases} \quad (13)$$

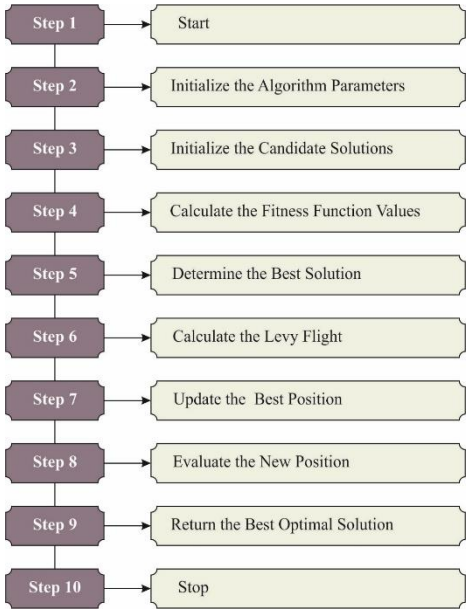

Figure 2: Steps involved in AHA

Where the FF value is represented by f. After visiting the target food supply, a hummingbird search for the new food supply:

$$v_i(t+1) = x_i(t) + b \cdot D \cdot x_i(t) \quad (14)$$

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

In Eq. (14), the territorial factor exposed to the normal distribution with a SD of 1 is b and mean of 0. The visit table is updated once the territorial foraging is performed. Also, the hummingbird migrates towards the food supply that is far away when a visited place endures from an abundant food source and it could be precisely modeled as given below:

$$x_{wrst}(t+1) = L + r \cdot (U - L) \tag{15}$$

In Eq. (15), the food supply with the poorer nectar refilling rate refers to x_{wrst}. The migrating behavior assists the AHA to prevent local stagnation. Fig. 2 describes the steps presented in AHA.

The AHA algorithm derives an FF to obtain enriched classification effectiveness. It finds a positive integers to indicate the higher effectiveness of the candidate outcomes. The minimizing classification error rate has been measured as the FF, as shown in Eq. (16).

$$fitness(x_i) = ClassifierErrorRate(x_i)$$
$$= \frac{No.of\ misclassified\ instances}{Total\ no.of\ instances} * 100 \tag{16}$$

4. Performance validation

The experimental validation of the TSASDL-BD technique can be examined employing the botnet database, encompassing 20689 instances at two classes as defined in Table 1.

Table 1: Database details

Classes	No. of Samples
Botnet	2554
Normal	18135
Total Samples	20689

In Fig. 3, exhibits the confusion matrices produced by the TSASDL-BD system with different epochs. The obtained outcome shows the TSASDL-BD approach is successful identification of the normal and botnet samples at 2 classes.

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

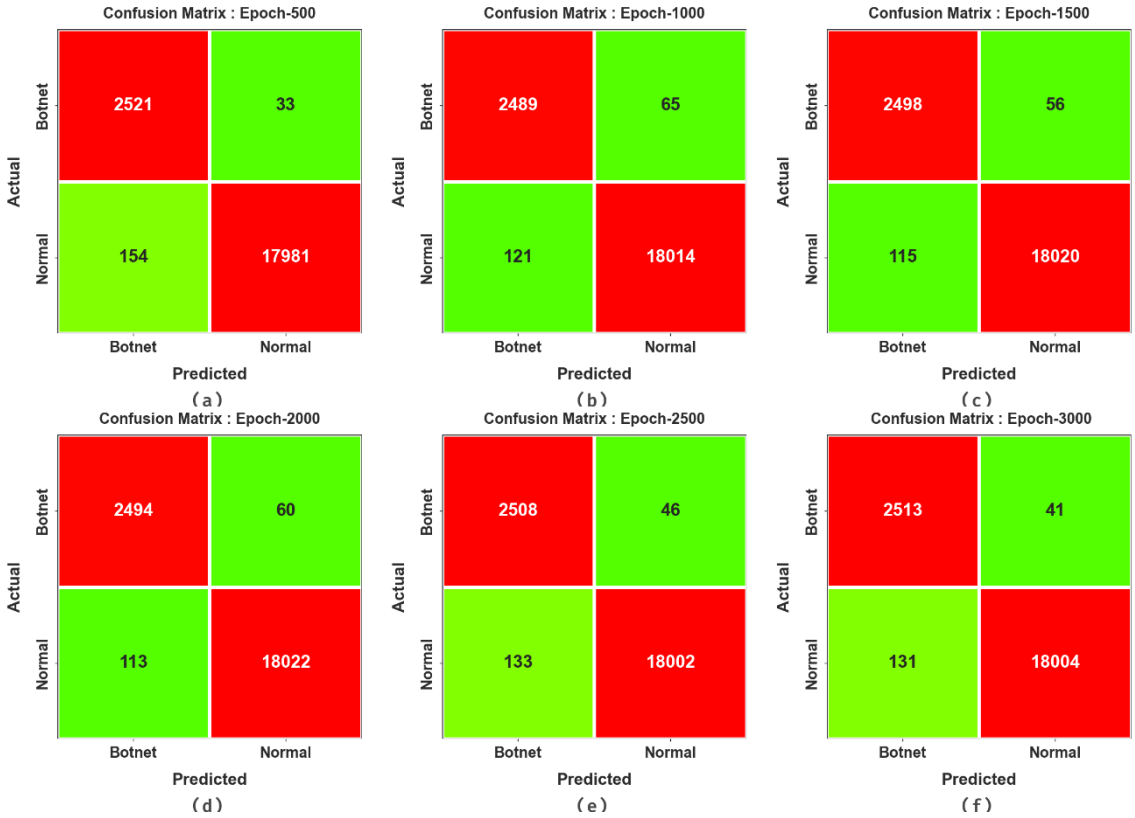

Figure 3: Confusion matrices of TSASDL-BD approach (a-f) Epochs 500-3000

Table 2 reports a detailed botnet detection result of the TSASDL-BD technique. In Fig. 4, an average results of the TSASDL-BD method in terms of $accu_y$ and $prec_n$ are given. The results imply that the TSASDL-BD system gains increased performance. On 500 epochs, the TSASDL-BD technique offers $an\ accu_y$ of 98.93% and $prec_n$ of 97.03%. Additionally, based on 2000 epochs, the TSASDL-BD system gives an $accu_y$ of 98.51% and $prec_n$ of 97.67%. Besides, on 3000 epochs, the TSASDL-BD algorithm provided $accu_y$ of 99.17% and $prec_n$ of 97.41%.

In Fig. 5, an average analysis of the TSASDL-BD system with respect to $reca_l$, F_{score}, and AUC_{score} are provided. The simulated outcome represents that the TSASDL-BD technique achieves boosted performance. According to 500 epochs, the TSASDL-BD system offers $reca_l$ of 98.93%, $F_{score}$of 97.95%, and AUC_{score} of 98.93%. Moreover, with 2000 epochs, the TSASDL-BD methodology gives $reca_l$ of 98.51%, $F_{score}$of 98.09%, and AUC_{score} of 98.51%. Lastly, based on 3000 epochs, the TSASDL-BD approach provides $reca_l$ of 98.84%, $F_{score}$of 98.11%, and AUC_{score} of 98.84% respectively.

Table 2: Botnet detection outcome of TSASDL-BD approach under various epochs

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	AUC_{score}
Epoch - 500					
Botnet	98.71	94.24	98.71	96.42	98.93
Normal	99.15	99.82	99.15	99.48	98.93
Average	98.93	97.03	98.93	97.95	98.93
Epoch - 1000					
Botnet	97.45	95.36	97.45	96.40	98.39
Normal	99.33	99.64	99.33	99.49	98.39
Average	98.39	97.50	98.39	97.94	98.39
Epoch - 1500					
Botnet	97.81	95.60	97.81	96.69	98.59

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

Normal	99.37	99.69	99.37	99.53	98.59
Average	98.59	97.64	98.59	98.11	98.59
Epoch - 2000					
Botnet	97.65	95.67	97.65	96.65	98.51
Normal	99.38	99.67	99.38	99.52	98.51
Average	98.51	97.67	98.51	98.09	98.51
Epoch - 2500					
Botnet	98.20	94.96	98.20	96.55	98.73
Normal	99.27	99.75	99.27	99.51	98.73
Average	98.73	97.35	98.73	98.03	98.73
Epoch - 3000					
Botnet	99.17	95.05	98.39	96.69	98.84
Normal	99.17	99.77	99.28	99.52	98.84
Average	99.17	97.41	98.84	98.11	98.84

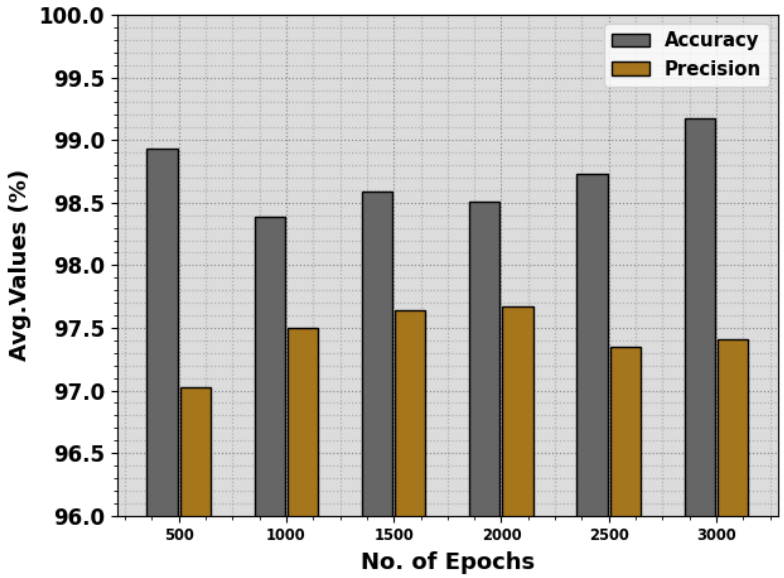

Figure 4: Average $accu_y$ and $prec_n$ of TSASDL-BD approach under various epochs

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

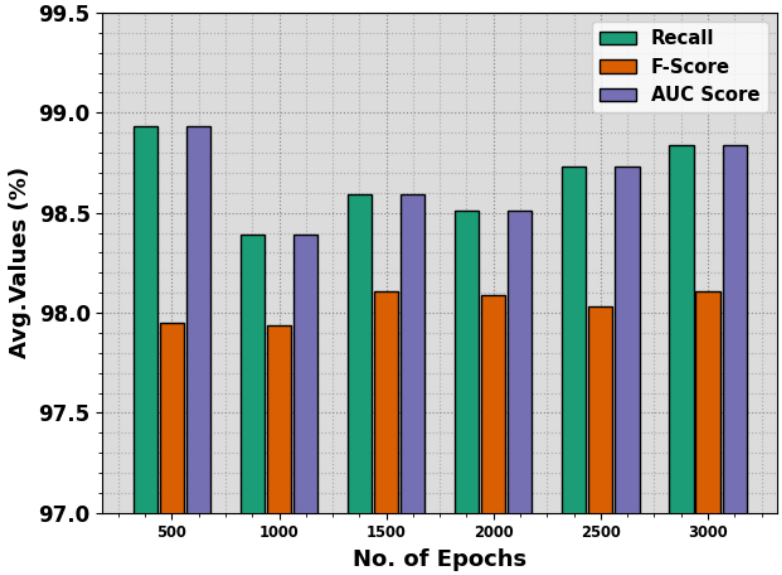

Figure 5: Average $reca_l$, F_{score}, and AUC_{score} of TSASDL-BD approach under various epochs

To evaluate the effectiveness of the TSASDL-BD approach at different epochs, we have created accuracy curves for the testing (TES) and training (TRA) phases, as shown in Fig. 6. The Two curves offer valued insights toward the learning development and the potential of model for generalization. Although raising the epochs count, a visible enhancement in both TES and TRA accuracy curves develops obviously. This heightening indicates the ability of the model for higher recognization patterns along with the TRA and TES databases.

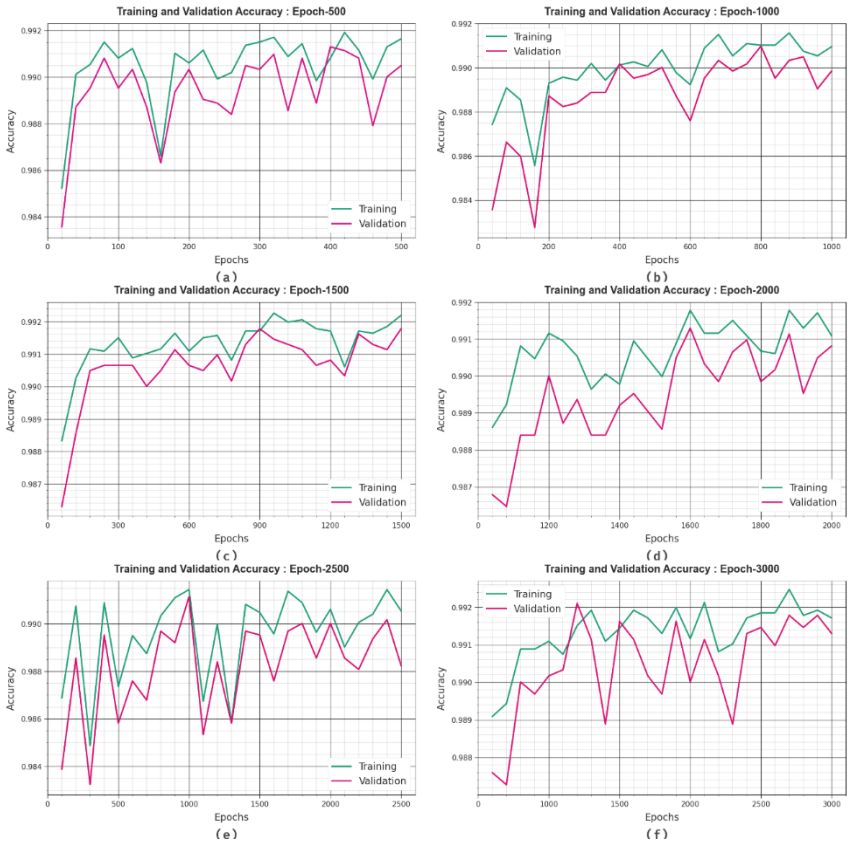

Figure 6: $Accu_y$ curve of TSASDL-BD approach (a-f) Epochs 500-3000

Fig. 6 similarly illustrates an overview of the TSASDL-BD algorithm at diverse epochs, and the model's loss values during the TRA process. The reduction trends for TRA loss over epochs represents that the method

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

perpetually improves the weights for decreasing the predictable errors with the TRA and TES databases. This loss deliberates the model fitted to the TRA data. Significantly, the TRA and TES loss constantly reduces, showing the proficient learning patterns existing in these two databases. Besides, it displays the variation of the model for lessening the variances amongst the predictive and original TRA labels.

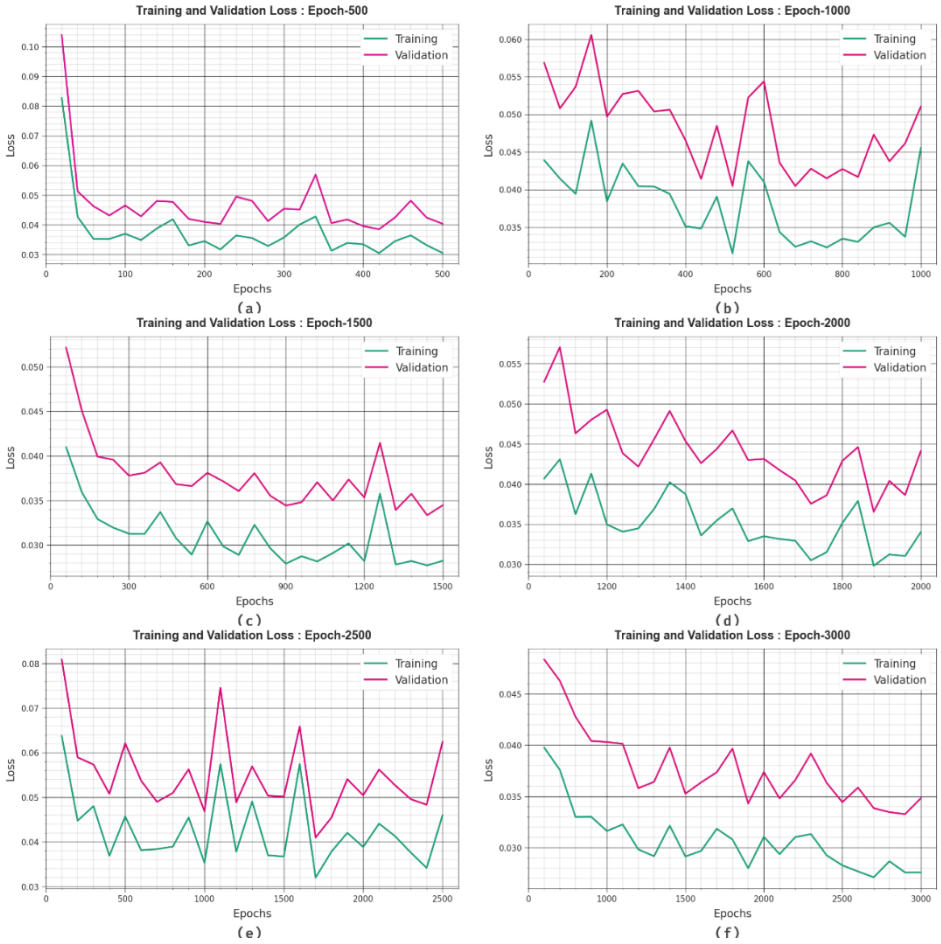

Figure 7: Loss curve of TSASDL-BD approach (a-f) Epochs 500-3000

The precision-recall curve of the TSASDL-BD system with numerous epochs, the model plots precision against recall showing that our model gets higher precision-recall values through each class as exhibited in Fig. 8. This graph demonstrates the capability of the model to identify diverse classes, especially surpassing in appropriately detecting positive samples but reducing false positives.

Fig. 9 likewise contains ROC curves of the TSASDL-BD algorithm on several epochs that showcase the model's proficiency for differentiating among class labels. These curves offer respected insights into the trade-off among FPR and TPR in distinct classification thresholds and epochs. This emphasizes the model's accurate predicted performance through different classes, additionally highlighting its classification abilities.

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

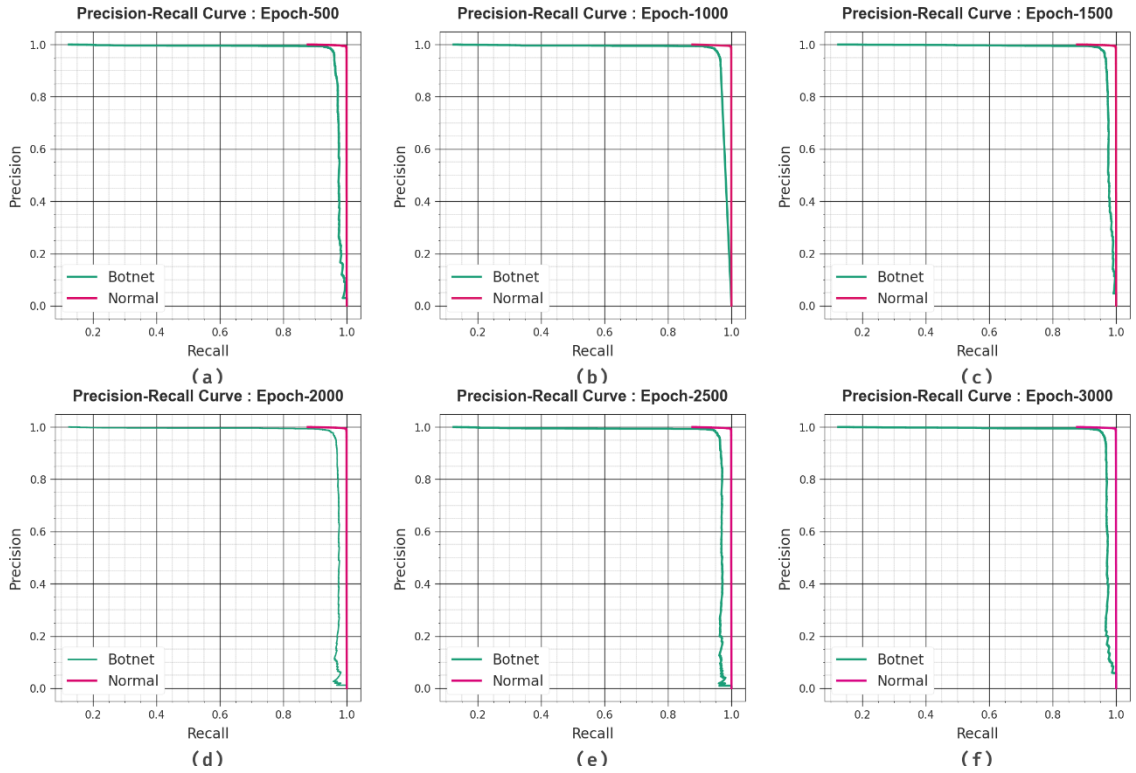

Figure 8. PR curve of TSASDL-BD approach (a-f) Epochs 500-3000

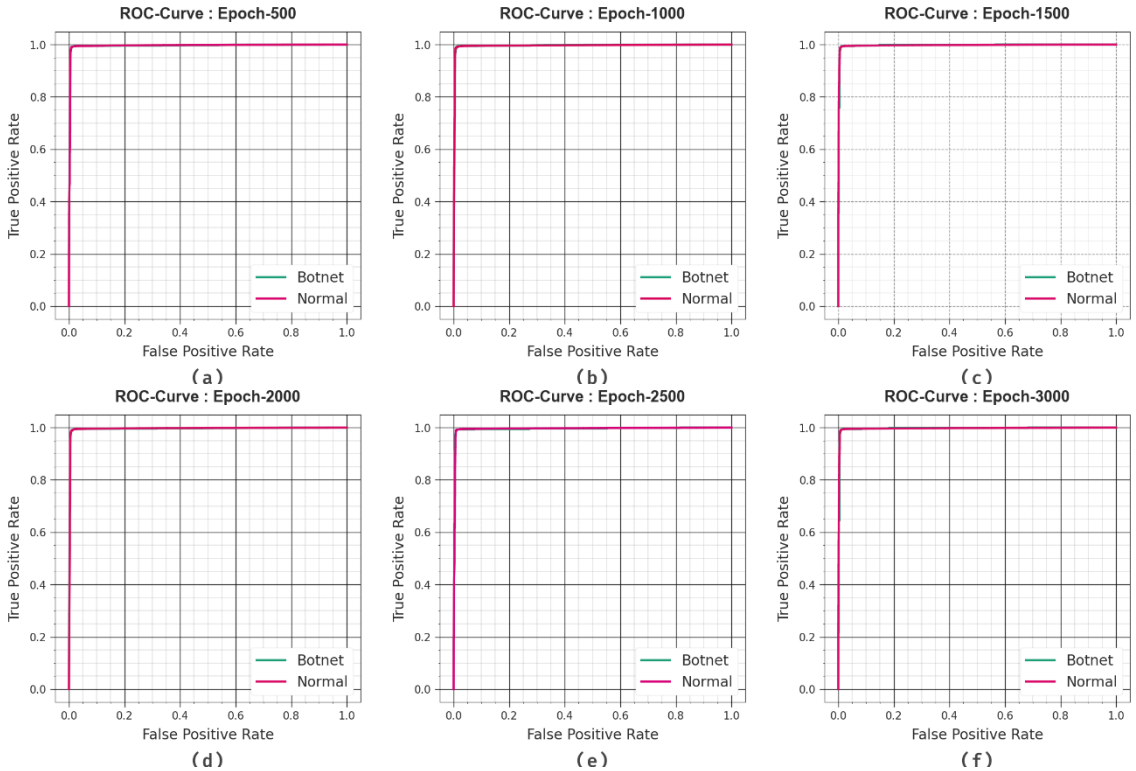

Figure 9: ROC curve of TSASDL-BD approach (a-f) Epochs 500-3000

An extensive comparative botnet detection results of the TSASDL-BD algorithm is examined in Table 3 [22]. Fig. 10 reports extensive results of the TSASDL-BD method with respect of $accu_y$. Based on $accu_y$, the TSASDL-BD technique offers an increasing $accu_y$ of 99.17% while the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

based, and FL-ANN algorithms obtain decreasing $accu_y$ values of 99.04%, 94.45%, 94.95%, 97.83%, 92.83%, and 98.87%, respectively.

Table 3: Comparison analysis of the TSASDL-BD approach with other algorithms

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
TSASDL-BD	99.17	97.41	98.84	98.11
BDC-RSODL	99.04	96.86	98.11	97.94
P2P-BDS	94.45	95.55	96.60	94.60
MTC-CNN	94.95	95.81	97.69	96.11
DT	97.83	94.86	95.87	95.59
Host-based	92.83	95.29	96.79	96.52
FL-ANN	98.87	96.22	97.79	97.02

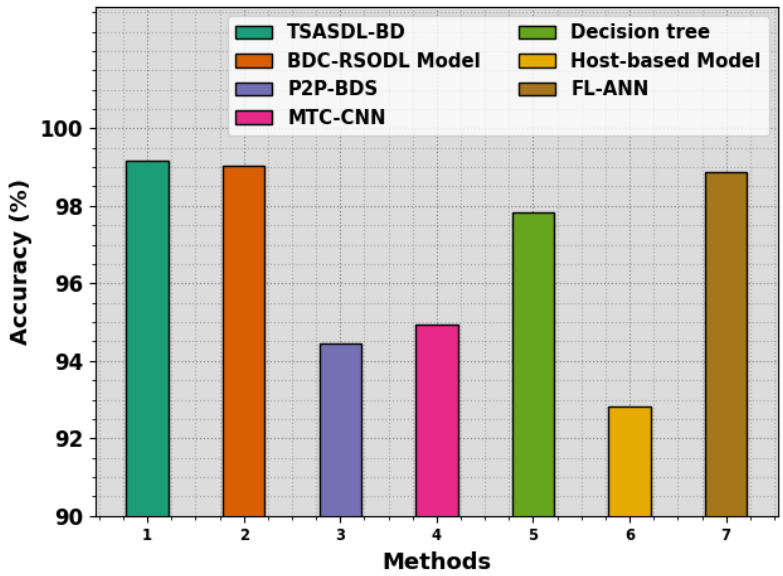

Figure 10: $Accu_y$ outcome of TSASDL-BD model with other systems

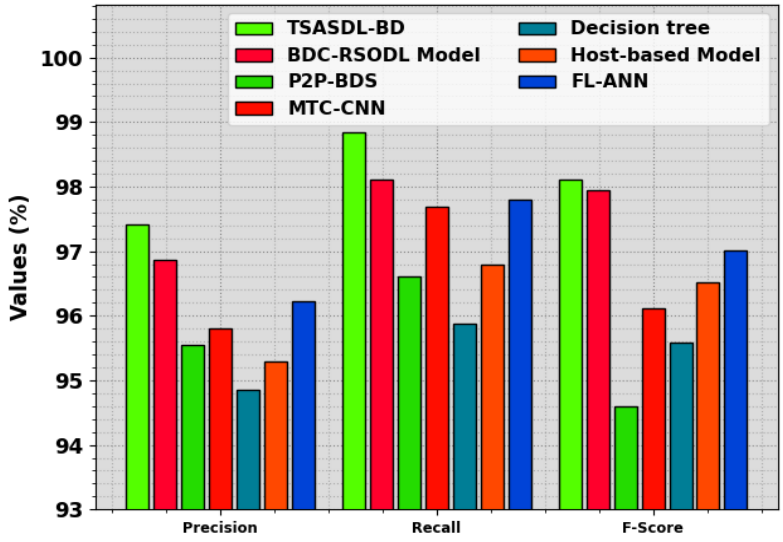

Figure 11: Comparative results of the TSASDL-BD model with other algorithms

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

Fig. 11 shows an extensive analysis of the TSASDL-BD methodology with respect to $prec_n$, $reca_l$, and F_{score}. According to $prec_n$, the TSASDL-BD system gives an improved $prec_n$ of 97.41% whereas the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based, and FL-ANN algorithms get reducing $prec_n$ values of 96.86%, 95.55%, 95.81%, 94.86%, 95.29%, and 96.22%, individually. Moreover, with $reca_l$, the TSASDL-BD system get raises $reca_l$ of 98.84% but, the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based, and FL-ANN methods acquire lesser $reca_l$ values of 98.11%, 96.60%, 97.69%, 95.87%, 96.79%, and 97.79%, correspondingly. Besides, with F_{score}, the TSASDL-BD approach offers improving F_{score} of 98.11% then, the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based, and FL-ANN techniques get minimizing F_{score} values of 97.94%, 94.60%, 96.11%, 95.59%, 96.52%, and 97.02%, respectively. Thus, the TSASDL-BD technique can be applied to the automated CC detection process.

5. Conclusion

In this study, we design an innovative TSASDL-BD methodology for the IoT environment. The purpose of the TSASDL-BD technique is to recognize the botnets and achieve maximum network security. In the TSASDL-BD technique, the TSA is applied for the effectual feature selection process, which aids in reducing the dimensionality problem. In botnet detection, the TSASDL-BD technique makes usage of the SLSTM-GRU model. Finally, the AHA can be used for the optimal selection of the hyperparameter values of the SLSTM+GRU algorithm. The performance analysis of the TSASDL-BD system with the benchmark database takes place. The extensive outcomes stated that the TSASDL-BD algorithm gains maximum detection results over other systems with respect to different measures.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] AL-Akhras, M., Alshunaybir, A., Omar, H. and Alhazmi, S., 2023. Botnet attacks detection in IoT environment using machine learning techniques. International Journal of Data and Network Science, 7(4), pp.1683-1706.

[2] Waqas, M., Kumar, K., Laghari, A.A., Saeed, U., Rind, M.M., Shaikh, A.A., Hussain, F., Rai, A. and Qazi, A.Q., 2022. Botnet attack detection in Internet of Things devices over cloud environment via machine learning. Concurrency and Computation: Practice and Experience, 34(4), p.e6662.

[3] Rajit Nair, Unraveling the Decision-making Process Interpretable Deep Learning IDS for Transportation Network Security, Journal of Cybersecurity and Information Management , Vol. 12 , No. 2 , (2023) : 69-82 (Doi : https://doi.org/10.54216/JCIM.120205)

[4] Habibi, O., Chemmakha, M. and Lazaar, M., 2023. Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection. Engineering Applications of Artificial Intelligence, 118, p.105669.

[5] Alissa, K., Alyas, T., Zafar, K., Abbas, Q., Tabassum, N. and Sakib, S., 2022. Botnet attack detection in IoT using machine learning. Computational Intelligence and Neuroscience, 2022.

[6] Sudhakar and Kumar, S., 2023, April. ABBDIoT: Anomaly-Based Botnet Detection Using Machine Learning Model in the Internet of Things Network. In International Conference on IoT, Intelligent Computing and Security: Select Proceedings of IICS 2021 (pp. 235-245). Singapore: Springer Nature Singapore.

[7] Ali Kadhim Nsaif, Securing Pervasive Computing Networks: Enhancing Network Security via Network Virtualization in Wireless Communications Infrastructure, ournal of Intelligent Systems and Internet of Things, Vol. 12 , No. 2 , (2024) : 75-88 (Doi : https://doi.org/10.54216/JISIoT.120206)

[8] Alani, M.M., 2022. BotStop: Packet-based efficient and explainable IoT botnet detection using machine learning. Computer Communications, 193, pp.53-62.

[9] Li, Y., Zhu, M., Luo, X., Yin, L. and Fu, Y., 2023. A privacy-preserving botnet detection approach in a largescale cooperative IoT environment. Neural Computing and Applications, 35(19), pp.13725-13737.

[10] Mahmoud M. Ismail, Ahmed A. Metwaly, Enhancing Wireless Ad-Hoc Network Security by Mitigating Distributed Denial-of-Service (DDoS) Attacks, International Journal of Wireless and Ad Hoc Communication , Vol. 8 , No. 2 , (2024) : 46-52 (Doi : https://doi.org/10.54216/IJWAC.080205)

[11] Al-Fawa'reh, M., Abu-Khalaf, J., Szewczyk, P. and Kang, J.J., 2023. MalBoT-DRL: Malware Botnet Detection Using Deep Reinforcement Learning in IoT Networks. IEEE Internet of Things Journal.

[12] Al-Sarem, M., Saeed, F., Alkhammash, E.H. and Alghamdi, N.S., 2021. An aggregated mutual information-based feature selection with machine learning methods for enhancing IoT botnet attack detection. Sensors, 22(1), p.185.

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024

[13] Soe, Y.N., Feng, Y., Santosa, P.I., Hartanto, R. and Sakurai, K., 2020. Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors, 20(16), p.4372.
[14] S. Alrayes, F., Maray, M., Gaddah, A., Yafoz, A., Alsini, R., Alghushairy, O., Mohsen, H. and Motwakel, A., 2022. Modeling of Botnet Detection Using Barnacles Mating Optimizer with Machine Learning Model for Internet of Things Environment. Electronics, 11(20), p.3411.
[15] Almuqren, L., Alqahtani, H., Aljameel, S.S., Salama, A.S., Yaseen, I. and Alneil, A.A., 2023. Hybrid Metaheuristics with Machine Learning based Botnet Detection in Cloud-Assisted Internet of Things Environment. IEEE Access.
[16] Abu Al-Haija, Q. and Al-Dala'ien, M.A., 2022. ELBA-IoT: an ensemble learning model for botnet attack detection in IoT networks. Journal of Sensor and Actuator Networks, 11(1), p.18.
[17] Zaheer, A., Tahir, S., Almufareh, M.F. and Hamid, B., 2023, March. A Hybrid Model for Botnet Detection Using Machine Learning. In 2023 International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-8). IEEE.
[18] Pokhrel, S., Abbas, R. and Aryal, B., 2021. IoT security: botnet detection in IoT using machine learning. arXiv preprint arXiv:2104.02231.
[19] Khan, S., Singh, Y.V., Yadav, P.S., Sharma, V., Lin, C.C. and Jung, K.H., 2023. An Intelligent Bio-Inspired Autonomous Surveillance System Using Underwater Sensor Networks. Sensors, 23(18), p.7839.
[20] Muhammad, A.U., Yahaya, A.S., Kamal, S.M., Adam, J.M., Muhammad, W.I. and Elsafi, A., 2020, October. A hybrid deep stacked LSTM and GRU for water price prediction. In 2020 2nd International Conference on Computer and Information Sciences (ICCIS) (pp. 1-6). IEEE.
[21] Ekinci, S., Izci, D. and Yilmaz, M., 2023. Simulated Annealing-aided Artificial Hummingbird Optimizer for Infinite Impulse Response System Identification. IEEE Access.
[22] Alshahrani, S.M., Alrayes, F.S., Alqahtani, H., Alzahrani, J.S., Maray, M., Alazwari, S., Shamseldin, M.A. and Al Duhayyim, M., 2023. IoT-Cloud-Assisted Botnet Detection Using Rat Swarm Optimizer with Deep Learning. Computers, Materials & Continua, 74(2).

DOI: https://doi.org/10.54216/IJWAC.080207
Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024