International Journal of Wireless and Ad Hoc Communication

♦ ISSN: 2692-4056 Vol: 08, No .02, 2024 ♦ PP. 35–41

Article DOI: https://doi.org/10.54216/IJWAC.080207

FULL LENGTH ARTICLE

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

Abedallah Z. Abualkishik[1,*] **Rasha Almajed**[1]

[1] American University in the Emirates, Dubai, UAE

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

Received: September 14, 2023 Revised: December 22, 2023 Accepted: May 22, 2024 ⋆ Corresponding author

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. Recently, 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 can differentiate between malicious and normal activity, therefore supporting earlier detection and avoidance of botnet attacks. This is essential to protect the integrity and security of IoT systems that are 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. In the TSASDL-BD technique, the TSA is applied for effective feature selection, reducing the dimensionality problem. For botnet detection, the TSASDL-BD technique uses a stacked long short-term memory gated recurrent unit (SLSTM-GRU) model. Finally, the artificial humming algorithm (AHA) is used for optimal selection of the hyperparameter values of the SLSTM-GRU system. The extensive outcomes state that the TSASDL-BD approach gains maximum detection results over other algorithms with respect to 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]. However, 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 collections of compromised devices that execute orchestrated attacks against servers and network services, aiming to cause inaccessibility [2]. They are formed by affected endpoints such as Internet of Things devices, wireless routers, computers, and mobile phones. Recently, several IoT devices have been constructed with a concentration on ease of use, without being involved in confirming security development. 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 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

© 2024The Author(s). Published by ASPG. This article is licensed under CC BY 4.0.

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 by comparison with normal or predicted behavior. This algorithm can identify an extensive set of malicious intrusions, but it can also increase the possibility of false positives [7]. Anomaly-based approaches are categorized into statistical techniques, machine learning algorithms, and other approaches 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 in network traffic packets can be categorized as malicious or benign depending on specific predetermined features. Alternatively, a certain class of botnet attack can be detected in multi-class classification [8]. Consequently, Artificial Intelligence (AI) methods can attain excellent effectiveness to handle classification methods in diverse application fields. Particularly, different ML and DL architectures 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, including 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) [10].

This study designs an innovative tunicate swarm algorithm with stacked deep learning for botnet detection (TSASDL-BD) method for IoT platforms. The purpose of the TSASDL-BD method is to recognize botnets and achieve maximum network security. The TSA is applied for the feature selection process, which aids in reducing the dimensionality problem. The TSASDL-BD technique also uses the stacked long short-term memory gated recurrent unit (SLSTM-GRU) model. Finally, the artificial humming algorithm (AHA) is used for the optimal choice of the hyperparameter values of the SLSTM-GRU algorithm.

2. RELATED WORKS

Al-Fawa'reh et al. [11] presented MalBoT-DRL, a robust malware botnet detector employing deep reinforcement learning. This method combines damped incremental statistics with an attention-gaining method, assisting the detector to dynamically modify for varying malware patterns within IoT platforms. Al-Sarem et al. [12] developed an integrated Mutual Information (MI) assisted feature-selection algorithm with ML techniques. The N-BaIoT standard database was employed and comprises multi-class and binary classifications. The FS technique integrates MI, Principal Component Analysis (PCA), and ANOVA f-test at a finely granulated identification stage.

Soe et al. [13] introduced an ML-assisted botnet attack detection architecture with a sequential identification model. An effective FS technique was modified for implementing a lightweight identification algorithm with superior effectiveness. Alrayes et al. [14] developed a botnet detection architecture utilizing barnacles mating optimization with ML (BND-BMOML). The BMO technique was exploited for FS, and an Elman neural network (ENN) approach was implemented for botnet detection. Almuqren et al. [15] implemented a hybrid meta-heuristic with ML-based botnet detection system, using a modified firefly optimizer for extraction, a hybrid CNN-quasi-RNN algorithm, and a chaotic butterfly optimizer for hyperparameter tuning.

Abu Al-Haija and Al-Dala'ien [16] suggested ELBA-IoT, an ensemble learning model for botnet attack detection in IoT networks. Zaheer et al. [17] developed a hybrid ML technique that integrates rule-based methods, k-means, and decision trees. Pokhrel et al. [18] projected a new architecture employing ML techniques including naive Bayes, multilayer perceptron artificial neural networks, and K-nearest neighbors, while incorporating feature engineering and SMOTE.

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 botnets and achieve maximum network security. This comprises three main processes: TSA-based feature selection, SLSTM-GRU-based classification, and AHA-assisted hyperparameter tuning. Fig. 1 illustrates the workflow of the TSASDL-BD system.

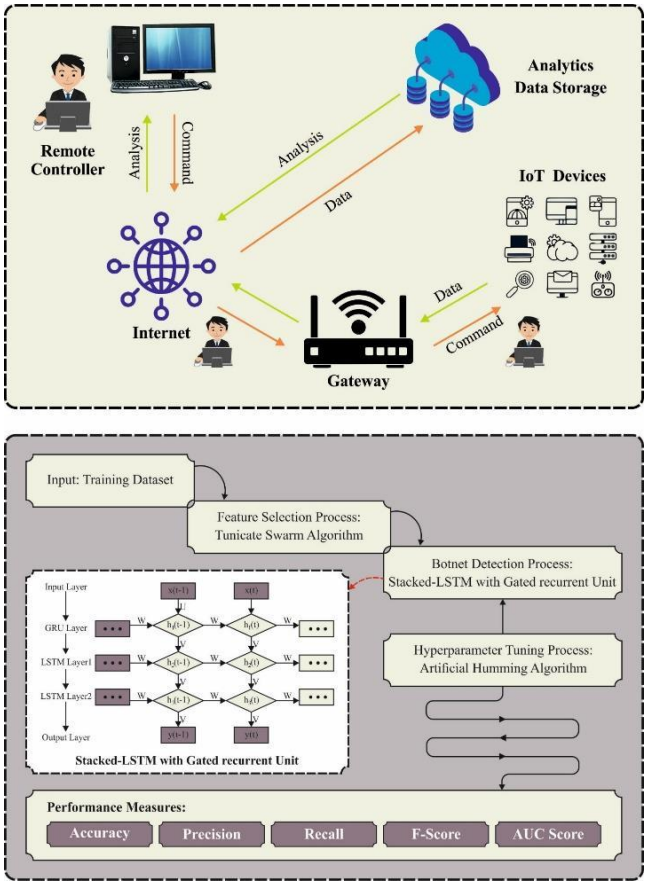


Figure 1. Workflow of TSASDL-BD approach.

3.1 Feature Selection Using TSA

At this phase, the TSA is applied for the effective feature selection 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. To determine a new location and stop clashes among search agents, vector $\overrightarrow{AC}$ is calculated

using Eq. (1):

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

The calculation of social forces among search agents is executed by utilizing Eq. (2):

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

where Min_s and Max_s signify primary and subsequent velocities of social interaction, respectively. Succeeding to clash avoidance, the following equation is used:

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

where $\overrightarrow{Dst}$ is the distance vector between the place of the food source and the location of the tunicate search agent. The tunicate agent location is updated as:

$$\overrightarrow{Pos_T}(x) = \begin{cases} Pos_{FS} + \overrightarrow{AC} \times \overrightarrow{Dst}, & rand_2 \geq 0.5, \\ Pos_{FS} - \overrightarrow{AC} \times \overrightarrow{Dst}, & rand_2 < 0.5. \end{cases} \tag{4}$$

3.2 Botnet Detection Using SLSTM-GRU

The TSASDL-BD technique makes use of the SLSTM-GRU model for botnet detection. The GRU model uses reset and update gates to control the information flow. The core GRU computations are represented as:

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z), \tag{5}$$
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r), \tag{6}$$
$$\tilde{h}_t = \sigma_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h), \tag{7}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \tag{8}$$

Here, x_t and h_t denote the input and output vectors; r_t and z_t show the reset and update vectors, respectively; σ_g and σ_h indicate the sigmoid function and hyperbolic tangent; and W, U, and b refer to parameter vectors and matrices.

The SLSTM depends on the LSTM neural network model. Each output layer of the LSTM method performs as an input for succeeding blocks in the layer, providing the architecture with the ability to capture time-series models and combine the learning view of prior layers while generating a high-level final outcome. Stacking LSTM in neural networks enhances predictive performance while allowing higher levels of temporal characteristics. 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. The output acts as an input to the GRU architecture for processing time-series data [20].

3.3 Hyperparameter Tuning Using AHA Technique

In conclusion, the AHA is implemented for the optimal choice of hyperparameter values of the SLSTM-GRU model. The main inspiration for AHA comes from foraging behaviors, memory capacity, and fight skills of hummingbirds [21]. A population of n hummingbirds is initialized arbitrarily and positioned under n food sources using $x_i = L + r \cdot (U - L)$.

The visit table is initialized as:

$$VT_{i,j} = \begin{cases} 0, & i \neq j, \\ null, & i = j. \end{cases} \tag{9}$$

The guided foraging behavior is described using:

$$v_i(t+1) = x_{i,trg}(t) + \alpha \cdot D \cdot (x_i(t) - x_{i,trg}(t)). \tag{10}$$

The food-supply location is updated using:

$$x_i(t+1) = \begin{cases} x_i(t), & f(x_i(t)) \leq f(v_i(t+1)), \\ v_i(t+1), & f(x_i(t)) > f(v_i(t+1)). \end{cases} \tag{11}$$

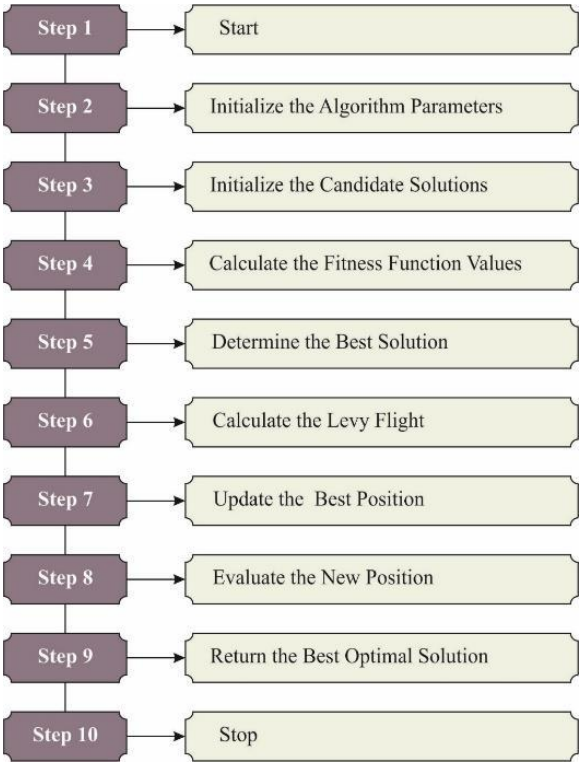


Figure 2. Steps involved in AHA.

After visiting the target food supply, a hummingbird searches for a new food supply:

$$v_i(t+1) = x_i(t) + b \cdot D \cdot x_i(t). \tag{12}$$

The hummingbird migrates toward the food supply that is far away when a visited place endures from an abundant food source:

$$x_{worst}(t+1) = L + r \cdot (U - L). \tag{13}$$

The AHA algorithm derives a fitness function (FF) to obtain enriched classification effectiveness. The minimizing classification error rate is measured as:

$$\begin{aligned} fitness(x_i) &= ClassifierErrorRate(x_i) \\ &= \frac{\text{No. of misclassified instances}}{\text{Total no. of instances}} \times 100. \end{aligned} \tag{14}$$

4. PERFORMANCE VALIDATION

The experimental validation of the TSASDL-BD technique is examined employing the botnet database, encompassing

20,689 instances at two classes, as defined in Table 1.

Table 1. Database details.

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

Fig. 3 exhibits the confusion matrices produced by the TSASDL-BD system with different epochs. The obtained outcome shows that the TSASDL-BD approach successfully identifies normal and botnet samples in two classes.

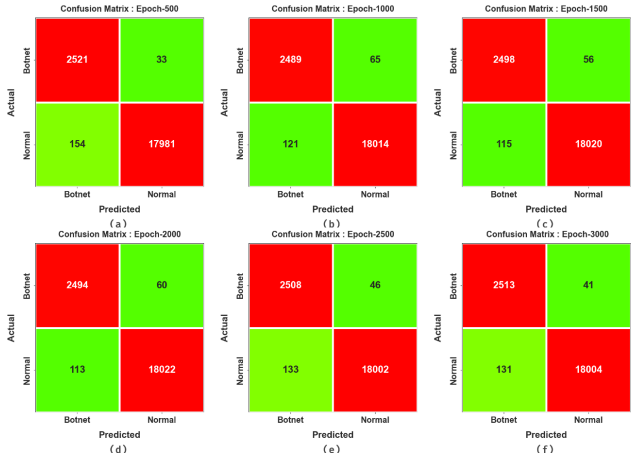


Figure 3. Confusion matrices of TSASDL-BD approach: (a–f) epochs 500–3000.

Table 2 reports a detailed botnet detection result of the TSASDL-BD technique. The results imply that the TSASDL-BD system gains increased performance. At 500 epochs, the TSASDL-BD technique offers an accuracy of 98.93% and precision of 97.03%. At 2000 epochs, the TSASDL-BD system gives an accuracy of 98.51% and precision of 97.67%. At 3000 epochs, the TSASDL-BD algorithm provides accuracy of 99.17% and precision of 97.41%.

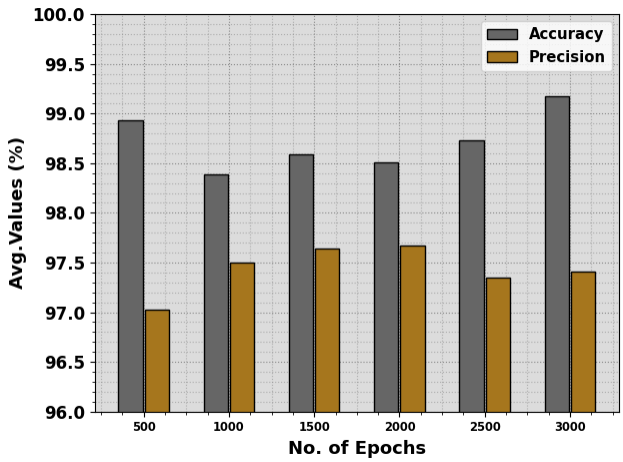


Figure 4. Average accuracy and precision of TSASDL-BD approach under various epochs.

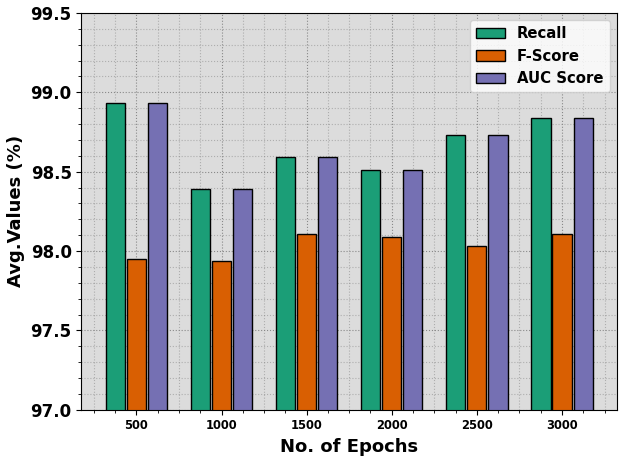


Figure 5. Average recall, F-score, and AUC-score of TSASDL-BD approach under various epochs.

To evaluate the effectiveness of the TSASDL-BD approach at different epochs, accuracy curves for the testing and training phases are shown in Fig. 6. Although the epoch count increases, a visible enhancement in both testing and training accuracy curves develops. This indicates the ability of the model to recognize patterns from the training and testing databases.

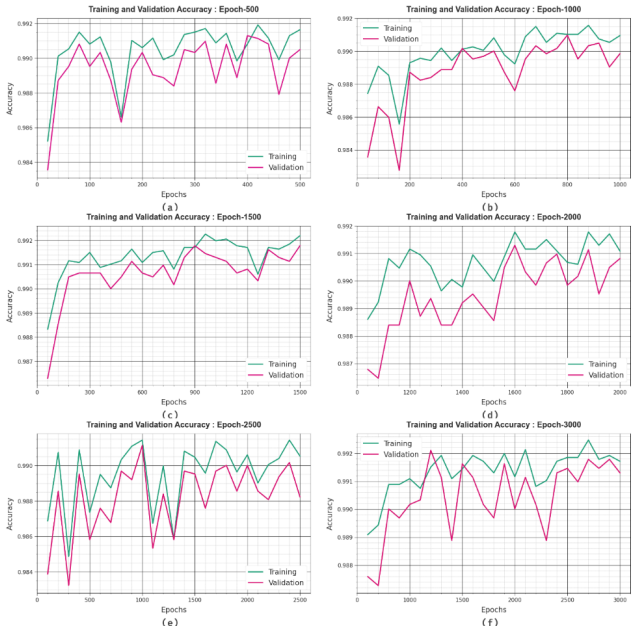


Figure 6. Accuracy curve of TSASDL-BD approach: (a–f) epochs 500–3000.

Fig. 7 illustrates an overview of the TSASDL-BD algorithm at diverse epochs and the model's loss values during the training process. The reducing trends for training loss over epochs represent that the method perpetually improves the weights for decreasing prediction errors with the training and testing databases.

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

Epoch	Class	Accuy	Precn	Recall	Fscore	AUCscore
500	Botnet	98.71	94.24	98.71	96.42	98.93
500	Normal	99.15	99.82	99.15	99.48	98.93
500	Average	98.93	97.03	98.93	97.95	98.93
1000	Botnet	97.45	95.36	97.45	96.40	98.39
1000	Normal	99.33	99.64	99.33	99.49	98.39
1000	Average	98.39	97.50	98.39	97.94	98.39
1500	Botnet	97.81	95.60	97.81	96.69	98.59
1500	Normal	99.37	99.69	99.37	99.53	98.59
1500	Average	98.59	97.64	98.59	98.11	98.59
2000	Botnet	97.65	95.67	97.65	96.65	98.51
2000	Normal	99.38	99.67	99.38	99.52	98.51
2000	Average	98.51	97.67	98.51	98.09	98.51
2500	Botnet	98.20	94.96	98.20	96.55	98.73
2500	Normal	99.27	99.75	99.27	99.51	98.73
2500	Average	98.73	97.35	98.73	98.03	98.73
3000	Botnet	99.17	95.05	98.39	96.69	98.84
3000	Normal	99.17	99.77	99.28	99.52	98.84
3000	Average	99.17	97.41	98.84	98.11	98.84

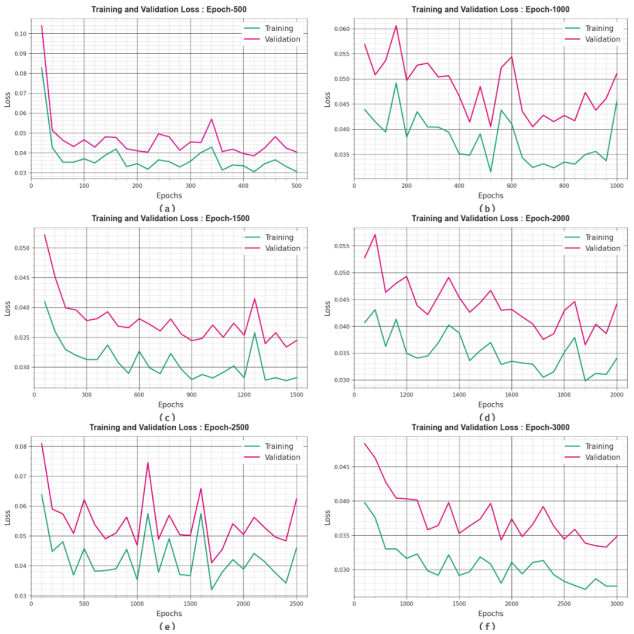


Figure 7. Loss curve of TSASDL-BD approach: (a–f) epochs 500–3000.

The precision-recall curve of the TSASDL-BD system with numerous epochs demonstrates the trade-off between precision and recall for diverse thresholds. The receiver operating characteristic curve likewise assesses the performance of the TSASDL-BD technique over multiple epochs.

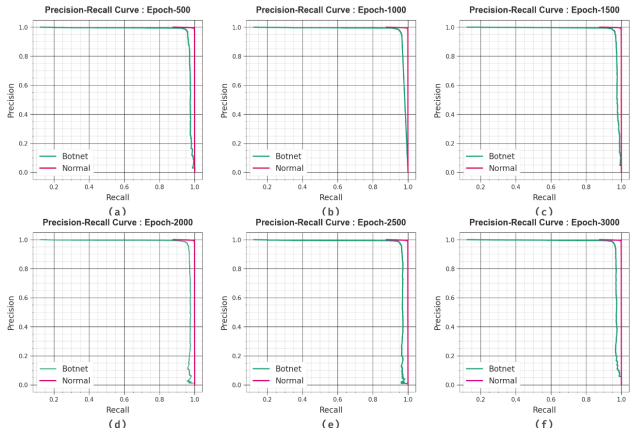


Figure 8. PR curve of TSASDL-BD approach: (a–f) epochs 500–3000.

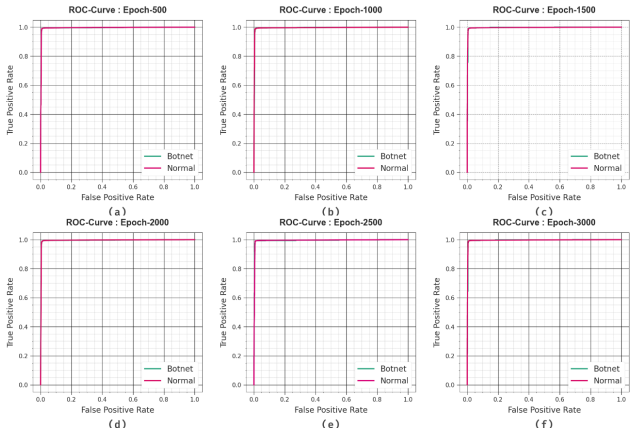


Figure 9. ROC curve of TSASDL-BD approach: (a–f) epochs 500–3000.

An extensive comparative botnet detection result of the TSASDL-BD algorithm is examined in Table 3. Fig. 10

and Fig. 11 show comparative results with other algorithms. The TSASDL-BD system obtains a maximum accuracy of 99.17%, while BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based, and FL-ANN algorithms obtain lower accuracy values of 99.04%, 94.45%, 94.95%, 97.83%, 92.83%, and 98.87%, respectively [22].

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

Methods	Accuy	Precn	Recall	Fscore
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

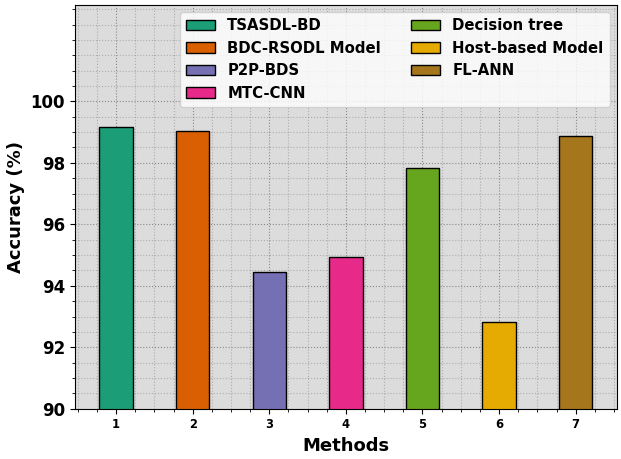


Figure 10. Accuracy outcome of TSASDL-BD model with other systems.

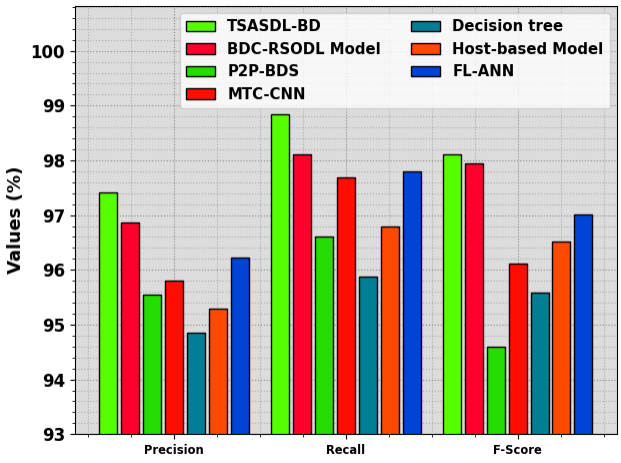


Figure 11. Comparative results of the TSASDL-BD model with other algorithms.

Fig. 11 shows an extensive analysis of the TSASDL-BD methodology with respect to precision, recall, and F-score. According to precision, the TSASDL-BD system gives an improved value of 97.41%, whereas the BDC-RSODL, P2P-BDS, MTC-CNN, DT, host-based, and FL-ANN algorithms get reducing values of 96.86%, 95.55%, 95.81%, 94.86%, 95.29%, and 96.22%, respectively. Moreover, with recall, the TSASDL-BD system raises recall to 98.84%, while the compared methods acquire lesser recall values. Besides, with F-score, the TSASDL-BD approach offers an improved value of 98.11%. Thus, the TSASDL-BD technique can be applied to automated command-and-control detection.

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 botnets and achieve maximum network security. In the TSASDL-BD technique, the TSA is applied for the effective feature selection process, which aids in reducing the dimensionality problem. In botnet detection, the TSASDL-BD technique makes use of the SLSTM-GRU model. Finally, the AHA is 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 state 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] M. AL-Akhras, A. Alshunaybir, H. Omar, and S. Alhazmi, "Botnet attacks detection in iot environment using machine learning techniques," *International Journal of Data and Network Science*, vol. 7, no. 4, pp. 1683–1706, 2023.

[2] M. Waqas, K. Kumar, A. A. Laghari, U. Saeed, M. M. Rind, A. A. Shaikh, F. Hussain, A. Rai, and A. Q. Qazi, "Botnet attack detection in internet of things devices over cloud environment via machine learning," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, p. e6662, 2022.

[3] R. Nair, "Unraveling the decision-making process interpretable deep learning ids for transportation network security," *Journal of Cybersecurity and Information Management*, vol. 12, no. 2, pp. 69–82, 2023.

[4] O. Habibi, M. Chemmakha, and M. Lazaar, "Imbalanced tabular data modelization using ctgan and machine learning to improve iot botnet attacks detection," *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105669, 2023.

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

[6] Sudhakar and S. Kumar, "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*. Springer Nature Singapore, 2023, pp. 235–245.

[7] A. K. Nsaif, "Securing pervasive computing networks: Enhancing network security via network virtualization in wireless communications infrastructure," *Journal of Intelligent Systems and Internet of Things*, vol. 12, no. 2, pp. 75–88, 2024.

[8] M. M. Alani, "Botstop: Packet-based efficient and explainable iot botnet detection using machine learning," *Computer Communications*, vol. 193, pp. 53–62, 2022.

[9] Y. Li, M. Zhu, X. Luo, L. Yin, and Y. Fu, "A privacy-preserving botnet detection approach in a largescale cooperative iot environment," *Neural Computing and Applications*, vol. 35, no. 19, pp. 13 725–13 737, 2023.

[10] M. M. Ismail and A. 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, pp. 46–52, 2024.

[11] M. Al-Fawa'reh, J. Abu-Khalaf, P. Szewczyk, and J. J. Kang, "Malbot-drl: Malware botnet detection using deep reinforcement learning in iot networks," *IEEE Internet of Things Journal*, 2023.

[12] 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, 2021.

[13] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine learning-based iot-botnet attack detection with sequential architecture," *Sensors*, vol. 20, no. 16, p. 4372, 2020.

[14] 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, 2022.

[15] L. Almuqren, H. Alqahtani, S. S. Aljameel, A. S. Salama, I. Yaseen, and A. A. Alneil, "Hybrid metaheuristics with machine learning based botnet detection in cloud-assisted internet of things environment," *IEEE Access*, 2023.

[16] Q. Abu Al-Haija and M. A. Al-Dala'ien, "Elba-iot: an ensemble learning model for botnet attack detection in iot networks," *Journal of Sensor and Actuator Networks*, vol. 11, no. 1, p. 18, 2022.

[17] A. Zaheer, S. Tahir, M. F. Almufareh, and B. Hamid, "A hybrid model for botnet detection using machine learning," in *2023 International Conference on Business Analytics for Technology and Security (ICBATS)*. IEEE, 2023, pp. 1–8.

[18] S. Pokhrel, R. Abbas, and B. Aryal, "Iot security: botnet detection in iot using machine learning," 2021, arXiv preprint arXiv:2104.02231.

[19] S. Khan, Y. V. Singh, P. S. Yadav, V. Sharma, C. C. Lin, and K. H. Jung, "An intelligent bio-inspired autonomous surveillance system using underwater sensor networks," *Sensors*, vol. 23, no. 18, p. 7839, 2023.

[20] A. U. Muhammad, A. S. Yahaya, S. M. Kamal, J. M. Adam, W. I. Muhammad, and A. Elsafi, "A hybrid deep stacked lstm and gru for water price prediction," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*. IEEE, 2020, pp. 1–6.

[21] S. Ekinci, D. Izci, and M. Yilmaz, "Simulated annealing-aided artificial hummingbird optimizer for infinite impulse response system identification," *IEEE Access*, 2023.

[22] 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," *Computers, Materials & Continua*, vol. 74, no. 2, 2023.