Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(5): 11998–12031.
DOI: 10.3934/math.2025544
Received: 02 February 2025
Revised: 13 April 2025
Accepted: 22 April 2025
Published: 23 May 2025

Research article

An attention mechanism based recurrent neural network with dimensionality reduction model for cyber threat detection in IoT environment

Randa Allafi*

Department of Computer Science, College of Science, Northern Border University, Arar, Saudi Arabia

* **Correspondence:** Email: Randa.allafi@nbu.edu.sa.

Abstract: Besides the developing threat of cyberattacks, cybersecurity is one of the most significant Internet of Things (IoT) regions. While the IoT has formed a novel model where a network of devices and machines is efficient in collaborating and communicating with each other, it is a novel process invention in enterprises. The role of cybersecurity is to mitigate risks for institutions and users by safeguarding data confidentiality across networks. The growing tools and technologies for cybersecurity improve safety in IoT systems. AI is helpful in improving cybersecurity by providing real-time information for faster threat detection, rapid responses, and smarter decisions. Moreover, integrating blockchain (BC) with IoT illustrates promise but encounters threats like performance issues, security vulnerabilities, and scalability limits. Still, BC plays a key part in protecting low-energy IoT devices. In this study, I proposed a novel approach using an Attention Mechanism-Based Recurrent Neural Network and Dimensionality Reduction for Cyber Threat Detection (AMRNN-DRCTD) model. The main goal of the proposed AMRNN-DRCTD model was to enhance the detection system for cyberattacks in IoT networks. I considered possible security breaches in BC and their influence on network processes. At the initial stage, the data normalization applied zero-mean normalization to alter data into a consistent setup. The feature selection process employed the chaotic and terminal strategy-based butterfly optimization algorithm (CTBOA). Furthermore, the proposed AMRNN-DRCTD model utilized the hybrid convolutional neural network and bi-directional long short-term memory with an attention mechanism (CNN-BiLSTM-AM) technique for the classification process. Finally, the Honey Badger Algorithm (HBA)-based hyperparameter selection range was accomplished to optimize the detection outcomes of the CNN-BiLSTM-AM technique. The experimental evaluation of the AMRNN-DRCTD methodology was examined under the BoT-IoT dataset. The performance validation of the AMRNN-DRCTD methodology highlighted a superior accuracy output of 99.28% over existing approaches.

Keywords: cyber threat detection; IoT; honey badger algorithm; recurrent neural network; attention mechanism; blockchain
Mathematics Subject Classification: 00A69

1. Introduction

The IoT incorporates the physical and digital universe into different methods, offering significant business opportunities for multiple segments like tourism, energy, and industry. It generates a novel model in which a network of devices and machines can collaborate and communicate with one another to drive novel procedures [1]. Nevertheless, the IoT is fragile, relating to several security concerns that are frequently highly required because of its complicated context and huge tool counts that give flaws regarding sources. It is considered the network management of home appliances, devices, and IoT vehicles. It is challenging because of the dynamic relationship between actors, devices, and resource constraints, including software, sensors, hardware, and connectivity that permits them to connect, exchange, and gather data [2]. IoT fundamentals are the smart factory that understands diverse components: process, person, technological ecosystem, and intellectual object. The IoT holds conventional internet connectivity to similarly classical non-physical gadgets like electric tools and cars. The IoT is also sturdily connected to manufacturing to yield higher-quality products at lower cost by collecting cloud computing (CC), big data analytics (BDA), and Industrial IoT (IIoT) comprising robots [3]. BC technology has been extensively utilized to preserve data security in decentralized systems, like the combined IoT and edge computing, IIoT, medical IoT, and Internet of Vehicles (IoV), by offering a traceable, tamper-resistant, and transparent data management structure. Regardless of significant potential, numerous problems structure the vast combination of IoT and BC. One of the major challenges is the discrepancy between standard BC compromise protocols and large-scale IoT networks with restricted calculating sources [4]. As the prevalence of IoT technology penetrates to add a smart grid's infrastructure, more cyber-threat risks are continuously developing. Primarily, the number of possible threat points through the system is enormous, and once a particular gadget is compromised, the whole grid becomes susceptible to cyber threats [5]. Cybersecurity involves protecting software, data, and electronics and the processes by which methods are acquired.

Generally, security goals include privacy, which relates to information suitable for revealing to unauthorized gadgets or individuals to be destroyed or modified. Consequently, owing to the countless IoT-based connected gadgets, society is becoming more susceptible to cyber-threats like denial-of-service threats by insiders and hackers, such as denying direct access to gadgets and more [6]. Technology is growing more central in everyday life, meaning that cybersecurity and cybercrime gadgets advance concurrently throughout the manufacturing area required to invest in cybersecurity countermeasures. In contrast, novel technologies are developed for IoT cybersecurity management [7]. Some analyses were directed to address the security tasks and issues of IoT and CC utilizing a trivial authentication process and the secure data searching and sharing of the cloud-based IoT. Consequently, such cyber threats need to be addressed for safe IoT usage. Thus, considerable efforts have been made to handle the security concerns related to the IoT technique over recent years. Numerous novel cybersecurity technologies have been advanced to couple the areas of deep learning (DL) and machine learning (ML) with cybersecurity [8]. DL and ML-based methods perform better with substantial data sizes and are flexible to diverse threat scenarios. The rapid growth and widespread adoption of IoT systems across various industries have made them increasingly attractive targets for cyber threats. These systems are characterized by their complexity and the large volume of data they generate, effectively creating threats in detecting and reducing security risks [9]. Conventional safety measures often fall short due to IoT networks' dynamic and decentralized nature, which requires real-time threat detection and adaptive responses. To address these challenges, innovative approaches are required to improve the accuracy and efficiency of threat detection while managing the high-dimensional data typical of IoT environments. By employing advanced models such as recurrent neural networks (RNNs) integrated with attention mechanisms (AMs), it is possible to improve cybersecurity resilience in IoT systems, enabling more precise detection of complex threats in real-time. This approach not only improves the ability of the model to concentrate on critical features but also reduces the computational burden, making it appropriate for resource-constrained IoT devices [10].

I propose a novel approach using Attention Mechanism-Based Recurrent Neural Network and Dimensionality Reduction for Cyber Threat Detection (AMRNN-DRCTD) model. The main goal of the proposed AMRNN-DRCTD model is to enhance the detection system for cyberattacks in IoT networks. I consider possible security breaches in BC and their influence on network processes. At the initial stage, the data normalization applies zero-mean normalization to alter the data into a consistent setup. The feature selection process employs the chaotic and terminal strategy-based butterfly optimization algorithm (CTBOA). Furthermore, the proposed AMRNN-DRCTD model utilizes the hybrid convolutional neural network and bi-directional long short-term memory with attention mechanism (CNN-BiLSTM-AM) technique for the classification process. Finally, the Honey Badger Algorithm (HBA)-based hyperparameter selection range is used to optimize the detection outcomes of the CNN-BiLSTM-AM technique. The experimental evaluation of the AMRNN-DRCTD methodology is examined under the BoT-IoT dataset. The key contribution of the AMRNN-DRCTD methodology is listed below.

- The AMRNN-DRCTD model applies zero-mean normalization to pre-process the data, centring the features around zero. This assists in standardizing the input data and mitigates bias in training. By ensuring consistent feature scaling, it improves the overall performance and stability of the model.
- The AMRNN-DRCTD approach employs the CTBOA method for feature selection, effectively detecting the most relevant features. This approach enhances the technique's capability to concentrate on crucial data while reducing noise, significantly improving the model's predictive accuracy.
- The AMRNN-DRCTD methodology utilizes a hybrid approach integrating CNN, BiLSTM, and AM for classification. This incorporation allows the model to effectively capture spatial and temporal dependencies in the data, resulting in more accurate and robust classification performance.
- The AMRNN-DRCTD method implements the HBA model for hyperparameter tuning, improving its optimization process. This technique enhances the model's performance by adjusting key parameters for better detection outcomes. The HBA method's capability to efficiently explore the hyperparameter space results in a more robust and accurate model.
- The AMRNN-DRCTD model incorporates a unique combination of advanced techniques:

CTBOA for feature selection, hybrid CNN-BiLSTM-AM for classification, and HBA for tuning. This synergy ensures effective feature extraction, robust classification, and optimal hyperparameter settings. The novelty is in the seamless integration of these techniques, improving the model's accuracy and performance for accurate detection.

- The structure of the article is as follows: In Section 2, I provide a review of the literature. In Section 3, I describe the proposed method. In Section 4, I present the evaluation of the results. In Section 5, I provide the conclusions.

2. Literature survey

Gelenbe and Nakip [11] developed a model for assessing the security of an n gadget, or IP address, IoT system by concurrently recognizing each compromised IP address and IoT gadget. This utilizes a particular Random NN structure formed by dual mutually connected sub-networks that complement one another in a recurrent framework named the Associated RNN (ARNN). For every IP address or n device in the IoT system, dual separate ARNN neurons recommend opposite views: Not compromised or compromised. Mirzaaxmedov [12] introduced an inclusive systematic review of existing works, discovering the multiple attacks and challenges that threaten IoT cybersecurity. Projected solutions and frameworks are also deliberated. Moreover, this paper investigates developing trends and recognizes gaps in current knowledge. A unique aspect of this investigation is its in-depth examination of ML models to mitigate and identify IoT risks. Zeng et al. [13] propose a conceptual model for AI-enabled anomaly detection in smart city IoT networks, grounded in the Complex Adaptive Systems (CAS) theory, Technology Acceptance Model (TAM), and Theory of Planned Behavior (TPB). Kaliyaperumal et al. [14] aimed to exploit unsupervised learning for training recognition methods for countering these attacks effectively. The developed approaches employ basic autoencoder (bAEs) for the reduction of dimensions and contain a three-phase recognition method: Deep autoencoder (dAE) attack detection and one-class support vector machine (OCSVM), together with density-based spatial clustering of applications with noise (DBSCAN) for threat clustering. Yakubu [15] proposed a comprehensive solution for reducing IoT cybersecurity threats using a multi-layered security approach, device-level security, authentication and management policies, and utilizing AI, BC, and CC models to improve overall network security and resilience. Nagarjuna Pitty et al. [16] compared the implementation of diverse ML models that are SVM, deep neural network (DNN), random forest (RF), and k-nearest neighbour (KNN) to address the recognition of anomaly and IoT security development. The existing investigation intended to enhance the present works on the effective security of IoT methods with sufficient cybersecurity extents. Adewuyi [17] inspected the connection of IoT, data analytics, and cyber security, offering an in-depth investigation of the susceptibilities inherent in IoT gadgets and the pioneering security solutions established for addressing these concerns over data-driven models. Furthermore, I discover developing trends and upcoming directions in protecting smart ecosystems, providing useful visions into how incorporating these fields can generate strong technological infrastructure.

Prasad et al. [18] developed cybersecurity via an Attention-based stacked autoencoder with a POA for detecting and mitigating Attacks (CASAE-POADMA). After min-max normalization, the Greylag Goose Optimization (GGO) approach is utilized for the feature selection procedure. The ASAE model is applied to mitigate and detect threats. Finally, the hyper-parameter tuning is implemented using the POA model. Markkandeyan et al. [19] improved cyber threat detection in IoT environments using a hybrid DL strategy. This approach integrates Improved Particle Swarm Optimization (IPSO) for efficient optimization, Enhanced Long Short-Term Memory (E-LSTM) for detecting suspicious actions, and Adaptive TensorFlow Deep Neural Network (ATFDNN) for accurate detection of malware-infected programs and software piracy. Dhanvijay and Kamble [20] proposed an Ensemble of Deep Learning Models with Prediction Scoring-based Optimized Feature Sets (EDLM-PSOFS) technique. This approach utilizes Correlation-Adaptive LASSO Regression (CALR) for robust feature extraction, Global Attention Long Short-Term Memory networks (GA-LSTMs) for capturing temporal patterns, and integrates the Exploit Prediction Scoring System (EPSS) to improve model interpretability and reduce false positives. Misra et al. [21] guided students and young researchers in computing disciplines on converting their project work into quality publications by choosing suitable topics, conducting proper reviews, and presenting results effectively. Adeniyi et al. [22] explored the integration of BC technology with Green Computing (GC) to address security and privacy challenges in multi-tenancy cloud environments. By utilizing Ganache and MetaMask. Guo et al. [23] proposed a novel power grid load forecasting model that integrates Convolutional Neural Networks (CNN), LSTM, Multi-Head Self-Attention (MHSA), Global AM (GAM), and Channel AM (CAM) to capture spatial and temporal features effectively. Zhang et al. [24] presented a data security intrusion detection system (IDS) that integrates the Mamba and ECANet models, utilizing an end-to-end learning approach for effective feature extraction, AM-based optimization, and robust performance in practical applications. Wu et al. [25] introduced GraphKAN by utilizing a Graph Attention Network (GAT) to dynamically allocate node weights, integrating Kolmogorov–Arnold Network (KAN) with multi-head attention mechanisms for enhanced feature extraction and employing parameterized B-splines to enhance the nonlinear expression of global features. Sana et al. [26] introduced a novel IDS for IoT environments, employing supervised ML methods, LSTM, and vision transformers (ViT) optimized using Bayesian optimization (BO) to improve detection performance. Mancy and Naith [27] proposed SwinIoT, an effective anomaly detection framework by utilizing Swin Transformer (ST), incorporating hierarchical and windowed attention mechanisms, custom attention models, and real-time optimization. Huang et al. [28] presented an FL-based approach incorporating CNN, AM, and variational autoencoders (VAEs) for improved network intrusion detection in the Industrial IoT (IIoT) while ensuring data privacy protection. Table 1 illustrates the summary of the literature review on cybersecurity and intrusion detection models in IoT systems.

Table 1. Summary of the literature review on cybersecurity and intrusion detection models in IoT systems.

Ref.	Techniques	Metrics	Findings
[11]	ARNN, Ground Truth Data, Offline And Online Learning, Interconnected Neurons, Incremental Learning	Accuracy, TNR, TPR, F1 Score, Recall, and Precision.	ARNN outperforms existing methods in attack detection.
[12]	Systematic Review, ML, AI, Cybersecurity Frameworks, IoT Risk Mitigation	Threat Identification, Privacy Concerns, Attack Detection, Security Solutions	Privacy and cybercrimes are major IoT safety concerns, and AI holds potential for future enhancements.
[13]	AI-enabled Anomaly Detection, CAS, TAM, TPB, User Engagement	Standard Datasets	Emphasizes the requirement for user engagement and continuous education to improve smart city cybersecurity resilience.
[14]	Unsupervised Learning, bAEs, OCSVM, dAE, DBSCAN	Precision, Recall, Specificity, F-Measure, Accuracy, False Negative Rate (FNR), Prevention Rate, Priority-based Blocking Rate, and Success Rate	The proposed model outperforms in accuracy and scalability, effectually detecting novel attacks and addressing imbalanced training data.
[15]	Multi-layered Security Approach, Device-Level Security, Authentication and Management Policy, AI, BC, CC	Threat Classification, Security Measures Evaluation, Authentication Process Review	The study stresses the requirement for robust security and collaboration to address IoT cybersecurity threats and ensure secure growth.
[16]	KNN, SVM, RF, DNN	Accuracy	DNN exhibited the highest accuracy, RF balanced accuracy and time, while KNN had the lowest accuracy and higher FPR.
[17]	Data Analytics, Cybersecurity Solutions, IoT Security	IoT Device Vulnerabilities, Data-driven Security, Emerging Trends	Advanced data analytics can enhance IoT security and resilience against cyber threats.
[18]	Min-Max, CASAE-POADMA, GGO	Accuracy, Precision, Recall, F1-Score, AUC-Score	The proposed method achieved superior accuracy in attack detection and mitigation for IoT networks.
[19]	ATFDNN, IPSO, E-LSTM	Accuracy, False Alarm Ratio, Training Time, Detection Rate	The hybrid DL strategy outperformed conventional methods in detecting malware and software piracy in IoT environments.
[20]	EDLM-PSOFS, CALR, GA-LSTMs, EPSS	Accuracy, Precision, Recall, F1-Score, Running Time, False Positive Rate, RMSE, MAPE, R-Squared	The presented approach improves IoT IDS by reducing false positives and improving detection of unknown threats.

Continued on next page

Ref.	Techniques	Metrics	Findings
[21]	Project Topic Selection, Writing Style Guide, Paper Structuring, Review Paper Writing	Publication Quality, Presentation Techniques, Writing Accuracy	The study guides students and researchers on converting projects into quality publications through topic selection, literature review, and structured writing.
[22]	BC, GC, Multi-Tenancy Security, Ganache and MetaMask	Security Enhancements, Privacy Improvements, Cloud Tenant Isolation	BC improves security and privacy in multi-tenant cloud environments.
[23]	CNNs, LSTMs, MHSA, GAM, CAM	Load Forecasting Accuracy, Dimensionality Reduction, Feature Extraction Efficiency	The proposed model improves load forecasting accuracy, enhancing smart grid scheduling.
[24]	Mamba Model, ECANet, AM, End-to-End Learning	Detection Accuracy, False Alarm Rate	The proposed method enhances detection accuracy by approximately 5% over conventional methods.
[25]	GAT, KAN, Multi-Head Attention Mechanisms, Parameterized B-Splines	Accuracy, Precision, Recall, F1-Score, FNR	The GraphKAN model improves detection accuracy by outperforming advanced models in smart grid security.
[26]	Tree-based SVM, Ensemble Methods, NN, LSTM, ViT, BO	Accuracy, F1-Score, and AUC	ViT outperformed LSTM with perfect training accuracy in all metrics.
[27]	ST, Hierarchical Attention, Windowed Attention, Custom Attention Models, Real-time Optimization	Accuracy, Precision, Recall, F1-Score, TPR, FPR, AP, mAP	SwinIoT outperformed existing models in anomaly detection.
[28]	FL, CNN, AM, VAE	Accuracy, Precision, Recall, F1-Score, FPR	The FL model improves accuracy, precision, and reduces FPR while ensuring data privacy.

While various approaches are proposed for improving IoT security, several limitations exist. Many models, comprising ARNN, GGO, and IPSO, depend heavily on large-scale datasets for effective training, which can be challenging in real-world applications where local data is limited. Additionally, existing methods often fail to address data privacy, class imbalance, and heterogeneous device environments. While models like CASAE-POADMA and EDLM-PSOFS concentrate on optimization, there is a lack of models that simultaneously offer high performance, real-time detection, and robust privacy protection in decentralized environments. Moreover, most IDSs fail to fully capture non-stationary, nonlinear, and multivariate data complexities in IoT settings. There is also a requirement for comprehensive models that incorporate effective data privacy solutions while attaining superior detection accuracy across IoT systems.

3. Methodology

I propose a novel approach using the AMRNN-DRCTD model. The main goal of the proposed AMRNN-DRCTD model is to enhance the detection system for cyberattacks in IoT networks. This study considers possible security breaches in BC and their influence on network processes. The AMRNN-DRCTD approach accomplishes that through zero-mean normalization, dimensionality reduction using CTBOA, a hybrid classification process, and HBA parameter tuning. Figure 1 illustrates the workflow of the AMRNN-DRCTD approach.

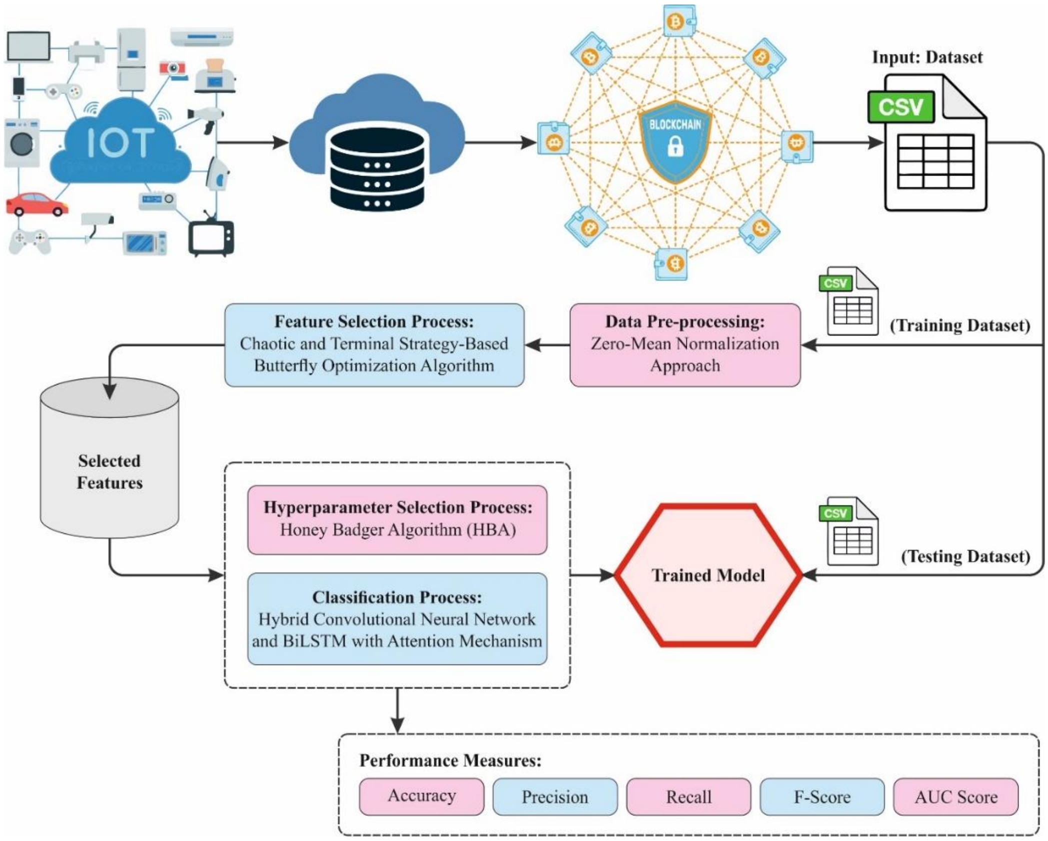

IOT
BLOCKCHAIN
Input: Dataset
CSV
(Training Dataset)
Feature Selection Process:
Chaotic and Terminal Strategy-Based Butterfly Optimization Algorithm
Data Pre-processing:
Zero-Mean Normalization Approach
Selected Features
Hyperparameter Selection Process:
Honey Badger Algorithm (HBA)
Classification Process:
Hybrid Convolutional Neural Network and BiLSTM with Attention Mechanism
Trained Model
(Testing Dataset)
Performance Measures:
Accuracy
Precision
Recall
F-Score
AUC Score

Figure 1. Workflow of the AMRNN-DRCTD model.

3.1. Zero-Mean normalization

At the initial stage, the data normalization applies zero-mean normalization to transform data into a consistent format [29]. This model is chosen for this model because it centers the data around zero, which assists in eliminating any bias caused by varying feature scales. By subtracting the mean of each feature, the model ensures that all input features have a consistent scale, enhancing the efficiency and stability of the learning process. This normalization technique prevents the dominance of features with larger magnitudes and accelerates convergence during training. Compared to other normalization techniques, such as min-max scaling, zero-mean normalization is less sensitive to outliers and can better handle data with a broader range. It is specifically efficient in neural networks, where the input data's distribution significantly affects the model's performance. This approach improves the model's generalization capability, making it more effective for accurate predictions across datasets.

Here, I accept the zero-mean normalization approach to pre-process the data input into the method. The mean value of the data gained while the variance is one, and processing is 0, which may implicitly prevent the impact of exceptions and maximum values. The code of the processing is as shown:

$$\chi = \frac{x_0-\mu}{\sigma}. \tag{1}$$

Here, x represents data processed, μ refers to the sample mean, and σ signifies sample standard deviation.

3.2. Dimensionality reduction process

The CTBOA is employed in the feature selection process [30]. This model is chosen because it effectively balances exploration and exploitation in the search space. Unlike conventional optimization methods, CTBOA utilizes chaotic sequences and terminal strategies to improve its search efficiency, mitigating the likelihood of getting trapped in local optima. This makes it highly appropriate for high-dimensional datasets where detecting the most relevant features is critical. The algorithm's population-based approach allows it to evaluate multiple feature subsets simultaneously, speeding up the selection process. Compared to other methods like genetic algorithms (GAs) or particle swarm optimization (PSO), CTBOA presents a more robust and dynamic search mechanism, resulting in more accurate feature selection. Its flexibility and ability to adapt to complex problem spaces make it ideal for improving the model's performance in feature-rich environments. Figure 2 depicts the working flow of the CTBOA method.

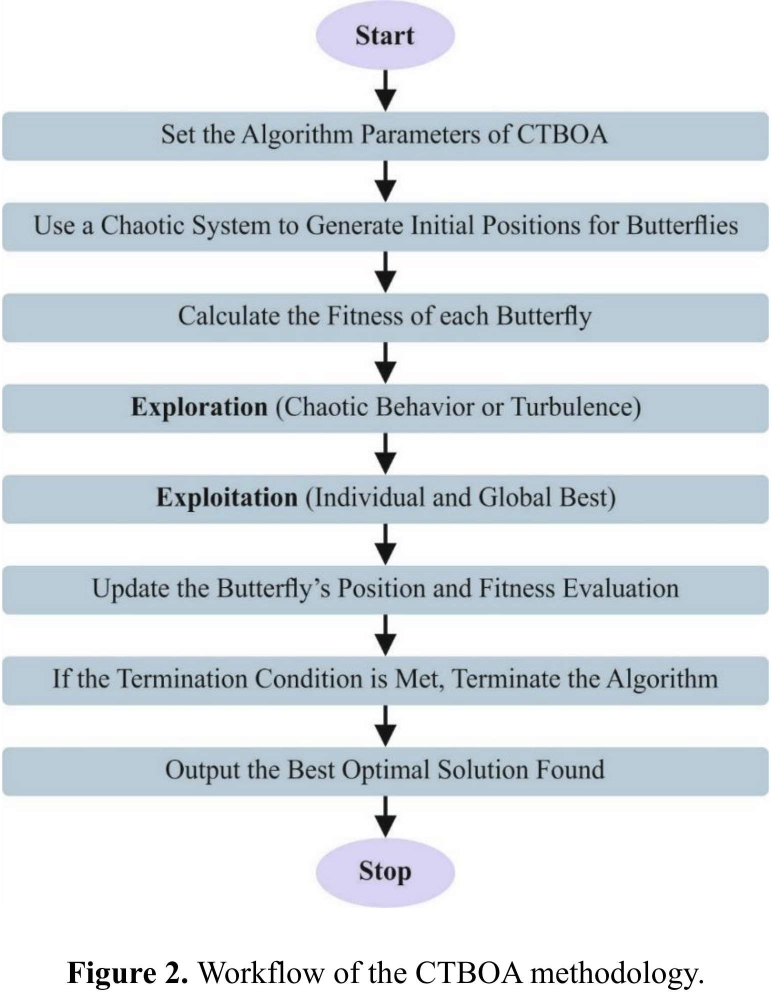

Start
Set the Algorithm Parameters of CTBOA
Use a Chaotic System to Generate Initial Positions for Butterflies
Calculate the Fitness of each Butterfly
Exploration (Chaotic Behavior or Turbulence)
Exploitation (Individual and Global Best)
Update the Butterfly's Position and Fitness Evaluation
If the Termination Condition is Met, Terminate the Algorithm
Output the Best Optimal Solution Found
Stop

Figure 2. Workflow of the CTBOA methodology.

BOA is a novel intellectual optimization model. The basic principle of BOA originates from the butterflies' searching behavior. Butterflies attract buddies to release fragrances and move towards regions with greater concentrations of fragrances. Every individual butterfly has a position that signifies a feasible solution. Regarding the initialization stage, individual butterflies inside the population are arbitrarily distributed as per Eq (2).

$$\overrightarrow{X_\iota} = \overrightarrow{LB} + r(1, Dim).* \left(\overrightarrow{UB} - \overrightarrow{LB}\right). \tag{2}$$

$\left[\overrightarrow{}In\ particular, UB\ and\ \overrightarrow{LB}\right]$ are the solution area's upper and lower searching boundaries. Dim represents the dimension of the problem, and the variable r specifies a random number in the range of $[0,1]$.

In BOA, the fragrance factor assesses the movement direction of individual butterflies to a considerable extent. Every butterfly changes its position according to the strength of its fragrance and the fragrance in its neighborhood. Regions with sophisticated fragrance strength depict better solution areas. Fragrance factor f_i is designed in Eq (3).

$$f_i = c \cdot I^a. \tag{3}$$

In this case, c specifies the sensory modality, generally 0.01. I represent the stimulant intensity dependent on the optimization fitness objectives. a is a power exponent formulated by Eq (4).

$$a = 0.1 + 0.2\left(\frac{t}{T}\right). \tag{4}$$

Here, T and t signify the existing iteration and the highest iterations, the BOA contains dual major stages, such as local and global search. These two stages are controlled by a switching probability p that assigns a value of 0.8. While $r < p$, the individual will perform a global search, which is given in Eq (11).

$$\overrightarrow{X_{i-BOA}^{t+1}} = \overrightarrow{X_i^t} + \left(r^2 \cdot \overrightarrow{gbest} - \overrightarrow{X_i^t}\right) \cdot f_i. \tag{5}$$

Here, $\overrightarrow{X_i^t}$ is the solution vector of the i-th butterfly at $t$he t-th iteration, and $\overrightarrow{gbest}$ specifies the optimum solution determined for each solution in the existing phase.

Otherwise, the individual accomplishes a local search, which is given in Eq (6).

$$\overrightarrow{X_{i-BOA}^{t+1}} = \overrightarrow{X_i^t} + \left(r^2 \cdot \overrightarrow{X_j^t} - \overrightarrow{X_k^t}\right) \cdot f_i. \tag{6}$$

$\overrightarrow{X_k^t}$ and $\overrightarrow{X_j^t}$ are arbitrarily selected from the solution space as the k-th and j-th butterflies.

1) Improved fragrance factor: The stimulant intensity inside the BOA is controlled by the fitness of the optimization objective. Assuming that the optimum values of diverse optimization concerns can vary extensively, the fragrance factor evaluated by Eq (5) is prone to fluctuation, and there might be a condition where the butterfly fragrance is absent, resulting in the BOA ceasing to upgrade the butterfly's location. Thus, it requires enhancing the butterfly's fragrance factor. Moreover, the normalized data can speed up the solution; the enhanced favor factor, depending upon this concept, is intended as Eq (7).

$$I = 1 - \frac{Fitness\left(\overrightarrow{X_i^t}\right) - gbestvalue}{worstvalue - gbestvalue}. \tag{7}$$

Here, $worstvalue$ and $gbestvalue$ are the fitness values related to the worst and optimal locations of the butterfly population, correspondingly, as per Eqs (3) and (7), enhancing the butterfly fragrance factor will not cause a lack of fragrance, and BOA will upgrade the butterfly location based on the novel fragrance factor.

2) Chaotic Learning Strategy: The chaotic learning approach includes chaotic and learning stages, and the factors are given.

(a) Chaotic phase. Chaos is generally created in nonlinear structures. It can improve the model's global searching capability and enhance problem-solving accuracy. Tent chaotic perturbation makes the chaotic variable through the Tent chaotic mapping. Subsequently, it is presented in the problem's solution area to be considered and chaotically perturbs the individual. The Tent chaotic mapping is given in Eq (8).

$$z_{i+1} = (2z_i)mod1 + r \cdot \frac{1}{N_T}. \tag{8}$$

Here, N_T represents the chaotic particle counts. To measure the Tent mapping, Eq (8) is enhanced, and the upgraded Tent mapping is given in Eq (9). The advanced Tent chaotic has a wider mapping range $[-1,1]$ that is beneficial to upgrading the model's global searching ability.

$$z_{i+1}^{new} = sgn(0.5 - r) \cdot \left((2z_i)mod1 + r \cdot \frac{1}{N_T}\right). \tag{9}$$

Here, sgn specifies the sign function that regulates the direction of interference.

(b) Learning phase. In a population, $\overrightarrow{gbest}$ is presumed to increase the population's average value to a certain degree based on the population size, and the equation for the position of the mean is given in Eq (10).

$$X_{\vec{m}} = \left(\frac{1}{N}\sum_{i=1}^{N}\overrightarrow{X_{\iota,1}}, \frac{1}{N}\sum_{i=1}^{N}\overrightarrow{X_{\iota,2}}, \cdots \frac{1}{N}\sum_{i=1}^{N}\overrightarrow{X_{\iota,D\iota m}}\right). \tag{10}$$

Here, $\overrightarrow{x_{\iota,J}}$ specifies the j-th dimension of the i-th butterfly.

Describe the process of learning the average location of the $\overrightarrow{gbest}$ of the population as the learning stage is given in Eq (11).

$$X_{learning} \rightarrow = z_{i+1}^{new} \cdot r(1, Dim).* \left(\overrightarrow{gbest} - X_m\right). \tag{11}$$

The chaotic learning approach is utilized for the butterfly location upgrade with Eq (12).

$$\overrightarrow{X_{\iota-CL}^{t+1}} = \overrightarrow{X_\iota^t} + X_{learning} \rightarrow. \tag{12}$$

The chaotic learning approach creates the average value learned to the finest individual from diverse directions, enhancing the local and global searching capability and assisting the model in improving problem-solving precision.

3) Final elimination strategy: The final elimination approach inserts the population diversity to eliminate adverse individuals and arbitrarily initializes the positions of the five poorest individuals in every iteration, depending on Eq (13).

$$\overrightarrow{X_{s[(N-4):N]}} = \overrightarrow{LB} + r(1, Dim).* \left(\overrightarrow{UB} - \overrightarrow{LB}\right). \tag{13}$$

Here, s specifies the ordinal number attained to order the fitness from the finest to the worst. N represents the size of the population.

The fitness function (FF) utilized in the CTBOA model is planned to balance the number of nominated features in every solution (least) and the accuracy of classification (greatest) that is attained by employing these chosen features. Equation (14) signifies the FF for assessing the solutions.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|}. \tag{14}$$

Here, $\gamma_R(D)$ signifies the classification rate of error. $|R|$ denotes the cardinality of the chosen subset, and $|C|$ refers to the total number of features, and α and β are dual parameters that correspond to the significance of classifier quality and sub-set length. ∈ [1,0] and $\beta = 1 - \alpha$.

3.3. Hybrid classification process

In addition, the proposed AMRNN-DRCTD model performs a hybrid CNN-BiLSTM-AM technique for the classification process [31]. This model is chosen due to its capability to capture both spatial and temporal dependencies in data. The CNN component outperforms automatically extracting spatial features, making it ideal for image or signal processing tasks. The BiLSTM layer improves this by capturing long-range temporal dependencies from past and future contexts, which is crucial for time-series or sequential data. The AM model additionally refines the process by enabling the model to concentrate on the most critical features, enhancing accuracy and mitigating noise. This integration allows the model to outperform conventional methods that rely on either spatial or temporal processing alone. The hybrid approach is more effective in handling complex datasets, giving a better balance of feature extraction, temporal modeling, and adaptive focus. As a result, the proposed model performs better in classification tasks than standalone CNNs, LSTMs, or conventional ML techniques. Figure 3 depicts the infrastructure of CNN-BiLSTM-AM.

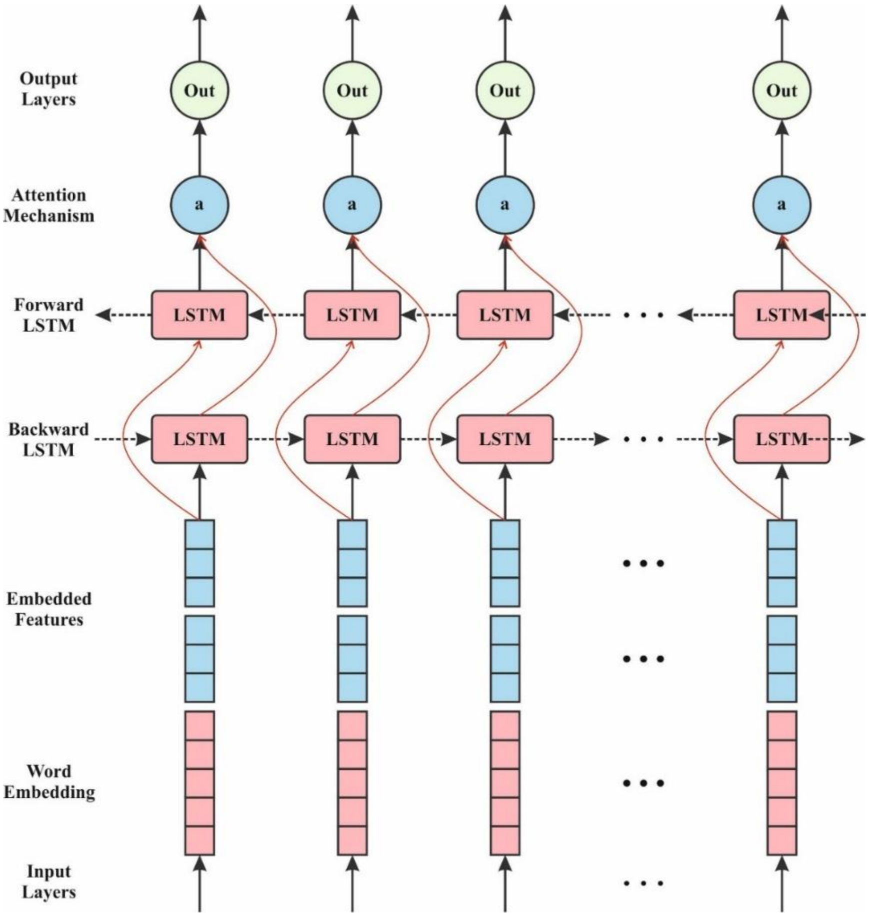


Figure 3. Structure of CNN-BiLSTM-AM.

The primary objective of the CNN is to remove the main characteristics from input data. A normal CNN structure comprises numerous layers, such as the pooling, convolution, input, fully connected (FC), and activation layers. The particular equation utilized in the ID-CNN system design is

$$x_j^l = f\left(\sum_{i=1}^{m} x_j^{l-1} * k_{ij}^l + b_j^l\right) \tag{15}$$

$$x_j^l = f\left(down\left(x_j^{l-1} + b_j^l\right)\right) \tag{16}$$

$$h_{w,b}(x) = f(w^T x + b). \tag{17}$$

During Eq (15), x_j^l denotes jth feature mapping of the lth layer, $f(*)$ denotes the activation function, and this study applies the activation function of ReLU, x_j^{l-1} refers to several jth feature mapping in the lth layer. m represents input feature map counts, k_{ij}^l signifies trainable convolutional kernel, and b denotes a biased term. During Eq (16), down denotes the down-sampling function of pooling. During Eq (17), $h(x)$ signifies output, and w means the weighting matrix of the convolution layer. In this paper, 1D-CNN methods were applied to remove the local spatial characteristics. The spatial characteristics removed by CNN denote the local architectural data associated with the spatial locations within the information, like waveform and edge models. This capability to remove local and global spatial connections enables CNN to identify composite designs and frameworks inside the data successfully.

An LSTM system is a development model for RNN's drawbacks. In contrast to RNN, LSTM presents different gating mechanisms. This architectural development efficiently resolves the fact that RNN has shorter-term memories and the gradient explosion and disappearance problem, thus improving the model's capability to recollect longer-term determined states. It mainly controls the process of the complete system processes gate, input gate, and output gate. The operational standards of LSTM follow the succeeding equations:

$$f_t = \sigma\left(M_f[h_{t-1}, x_t] + D_f\right) \tag{18}$$

$$i_t = \sigma\left(M_j[h_{t-1}, x_t] + D_i\right) \tag{19}$$

$$\tilde{C}_t = \tanh(M_c[h_{t-1}, x_t] + D_c) \tag{20}$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \tag{21}$$

$$O_t = \sigma(M_{\mathrm{o}}[h_{t-1}, x_t] + D_{\mathrm{o}}) \tag{22}$$

$$h_t = O_t \tanh(C). \tag{23}$$

During the above-mentioned equations, x_i refers to the present input, f_t denotes forget gate output, and i_t stands for input gate output. h_{t-1} and h_t represent the present and past moments' outputs. σ signifies the activation function of the sigmoid. M_f, M_i, M_c, and M_{o} symbolize the weighting of all gate components. D_f, D_i, D_c, and D_0 characteristics biased terms of all gates. C_{t-1} and C_t indicate storage component conditions of the new and past instants, and C_t embodies the candidate's memory cell state at the instant. O_t denotes the output gate's vector output. tanh means activation function.

During this LSTM system, the communication of time series data is one-way, from before and after. Bi-LSTM constructs on LSTM by adding the backward LSTM calculation, making a dual-layer architecture for bi-directional transmission. The forward LSTM component handles the input information in the direction of forward, whereas the reverse LSTM component seizures and handles data from the direction of reverse. The last output is gained by linear superposition with particular weightings. This bi-directional component architecture reflects the forward and backward data of the time series. It can contribute most of the data, thus successfully enhancing the forecast precision of time series.

During DL, the AM can select significant data akin to human vision. Its essential nature is mapping relationships amongst key and query values. The self-attention (SA) mechanism is a different kind of AM, which allows the method to concentrate mainly on various portions of the sequence of input while assessing the relationship among input components, thus enhancing the method's precision. During the SA mechanism, the input sequence experiences various linear conversions to make the value, query, and key matrices. Formerly, the similarities among them are computed utilizing the scaled dot-product and standardized by the function of Softmax to gain the weighted coefficients. If the query vector Q is related to the key vector K, their dot product becomes more significant, resulting in a rise in the equivalent weighted coefficients. At last, these weightings are applied to implement weighted sums of the vector value V, producing the attention vector demonstration.

$$Attention\ (Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{24}$$

Here, d_k denotes vector dimension in K or Q.

Cross-attention is the method that concentrates on the significance of the relationships amongst dissimilar sequences and was extensively applied in sequence-to-sequence and multi-modal tasks. This method dynamically fine-tunes the attention all sequences provide to others by computing the similarities among the Q vector of a single sequence and the V and K vectors of other sequences, thus enabling efficient data fusion. Unlike SA, where $V, Q,$ and K stem from a similar sequence, cross-attention utilizes Q from a single sequence and V and K from others. This effectively removes either temporal or spatial data from global data.

During earlier studies, several investigators have used both CNN and Bi-LSTM for time series classification. CNN efficiently protects the spatial content of the data, mainly concentrating on local spatial characteristics while avoiding time-based relations. On the other hand, Bi-LSTM is calculated to take temporal characteristics. However, its capability for modeling spatial features is comparatively low. These dual models deprive the method of fully comprehending the comprehensive information data in extracting the features. Nevertheless, afterwards, CNN removes characteristics from the novel data, and the data resolution is decreased, leading to an intrusion into the temporal data characteristics of the information.

3.4. Parameter tuning using HBA

Finally, the HBA-based hyperparameter selection range is achieved to optimize the detection outcomes of the CNN-BiLSTM-AM model [32]. This model is chosen for its ability to efficiently optimize the parameters of the model through a robust search mechanism inspired by the adaptability and exploration strategies of honey badgers. The HBA method incorporates exploration and exploitation techniques, allowing it to avoid local optima and find globally optimal solutions for hyperparameter tuning. This is advantageous when dealing with complex models where conventional optimization methods may face difficulty in effectively exploring a vast search space. Compared to grid or random search algorithms, HBA gives a more dynamic and intelligent approach to hyperparameter selection, significantly improving convergence speed and model performance. Its versatility makes it appropriate for various ML models, ensuring that the chosen parameters enhance detection accuracy and mitigate overfitting. Overall, HBA provides a more effective and adaptive solution for fine-tuning models than more conventional hyperparameter optimization techniques. Figure 4 specifies the steps involved in the HBA approach.

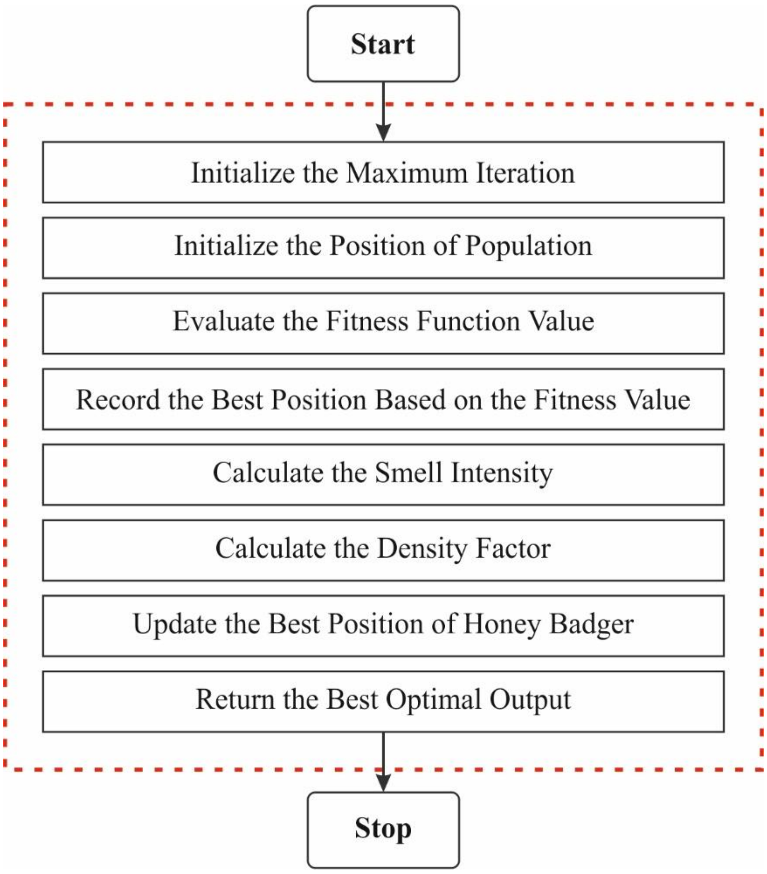

Start
Initialize the Maximum Iteration
Initialize the Position of Population
Evaluate the Fitness Function Value
Record the Best Position Based on the Fitness Value
Calculate the Smell Intensity
Calculate the Density Factor
Update the Best Position of Honey Badger
Return the Best Optimal Output
Stop

Figure 4. The process involved in the HBA model.

The HBA is a population-centered meta-heuristic optimizer model influenced by the dynamic search execution of HBs using honey and digging-seeking methods. HBA has accumulated extensive attention and is used in different fields. The significant popularity of HBA in the scientific community derives from its directness, ease of use, effective computation duration, accelerated convergence speed, higher efficiency, and ability to deal with various categories of optimizer problems, differentiating it from the famous optimizer model proposed. These animals utilize two major senses, digging and scent, to recognize food resources. The HBA accepts an architecture similar to other meta-heuristics. It uses a set of possible solutions for a provided optimizer obstacle, honing them through several methods, namely randomization and adjacency to the global optimal problem.

(a) Initialization stage

The HB optimization algorithm typically starts with the population's initialization. Assume that Nop refers to total honey badger counts. The position of h_{th} HB is initialized as in Eq (25).

$$Y_h = lower + (upper - lower) \times r_1, h = 1, 2, 3, \dots, Nop. \tag{25}$$

Here, Y_h describes the location of h_{th} the HB, upper, and lower describe the upper and lower limits of the searching area. While r_1 describes an arbitrarily selected value inside the range $(0,1)$. In the evaluation stage, the target function of all agents is gained by utilizing Eq (26).

$$f_h = f_{obj}(y_h). \tag{26}$$

(b) Digging stage (Exploration)

The way particular HBs model their food using their sense of smell can help us determine the direction of the digging phase. During the digging phase, the density factor, intensity operator, and trend modifier are decided in cooperation with the location-updated equation.

(1) Digging stage position upgrade

The HB's intense capability to identify smell enables it to determine prey in a hard situation and make accurate, objective recognition results. This stage, recognized as the digging stage, is attained utilizing Eq (27).

$$y_{new} = y_{prey} + F \times \beta \times IF \times y_{prey} + F \times r_2 \times \alpha \times d_h \\ \times |\cos(2\pi r_3) \times [1 - \cos(2\pi r_4)]|. \tag{27}$$

Here, y_{prey} characterizes the optimal location (prey location), β describes a constant value≥ 1, $and\, d_h$ epitomizes changes among the optimal position y_{prey} and h_{th} honey badger. r_2, r_3, r_4 Characterize different randomly generated numbers in the interval of $[1,\ 0]$, and F serves as a flag, fine-tuning the exploration route to assist the model in leaving the local best region. IF represents the intensity variable.

(2) Describing intensity factor

The intensity variable demonstrates the HB's controllable range to its prey, which is generally affected by the prey's capacity and caution for counter-reconnaissance.

$$IF = r_5 \times \frac{s}{4\pi d_h^2}. \tag{28}$$

$$s = (y_h - y_{h+1})^2. \tag{29}$$

$$d_h = y_{prey} - y_h. \tag{30}$$

Here, d_h denotes length dividing the h_{th} HB and its objective, S describes the source of strength, r_5 refers to the value generated randomly in the range $[0,1]$, and y_{prey} characterizes the prey location, the highest position thus far gained.

(3) Trend modifier (F) description

This mechanism enables the HB to change the search tendency and search the solution area entirely. It is usually described as Eq (31).

$$F = \begin{cases} 1, \ if\ r_6 \leq 0.5 \\ \quad -1\ else \end{cases}. \tag{31}$$

(4) Density variable (α) description

The density variable (α) is the constant variable, which modifies with time (a repetitive process). Its objective is to ensure the change in the initial stage of the iterative procedure step towards the future progress phase, as delineated in Eq (32).

$$\alpha = c \times \left(\frac{-curlt}{\text{Max}\, lt}\right). \tag{32}$$

Here, C characterizes a static number ≥ 1 but is selected as 2, and $curIr$ describes the recent iteration, $Ma\chi Ir$ indicates the total iteration counts.

$(c.)$ Honey stage (Exploitation)

The HB and honey guide bird are examples of mutually valuable cooperation. The main work of the honeyguide bird is to point the HB in the food direction. Equation (33), which is applied to define this phase, is recognized as the $Honey\ stage$.

$$y_{new} = y_{prey} + F \times r_7 \times \alpha \times d_h. \qquad (33)$$

Here, r_7 characterizes a value generated at random in the interval of $[0,1]$, y_{prey} characterizes the optimal position so far determined, and y_h denotes h_{th} HB's following location. α and F are gained utilizing Eqs (31) and (30) correspondingly. It is observable from Eq (32), in which an HB utilizes distance information d_h to start hunts in the proximity to the top position (prey position) y_{best} thus far discovered. Searching is measured by changes in foraging behavior over time (α). In addition, an HB may discover disruption F. Algorithm 1 demonstrates the HBA model.

Algorithm 1: HBA Technique

1. **Initialization:**

- **Initialize Population:** Generate a random population of honey badgers (candidate solutions).
    - Let the number of honey badgers be N.
    - Each honey badger represents a solution in the search space.
- **Set Parameters:**
    - Set the maximum number of iterations T.
    - Set other parameters like step size, exploration range, etc.

2. **Fitness Evaluation:**
    - Compute the fitness of each honey badger by applying the fitness function to each solution (this depends on the problem being solved, e.g., for optimization, it could be the objective function).
3. **Foraging Process (Exploration and Exploitation):**

    - **Exploration:**

        - Honey badgers utilize an exploration process to search for food over a large area. In the algorithm, this can be represented as a large random search within the problem space.
        - Each honey badger explores by moving arbitrarily within the search space.

    - **Exploitation:**

        - Once a food source is found, honey badgers exploit the area, concentrating their search on promising regions. This can be represented by adjusting the search process to focus more on the best candidate solutions found.

    - During every iteration, each badger will adjust its position using a combination of **local search** (focused on areas where it has previously found solutions) and **global search** (searching randomly in the broader space).

The update mechanism for every honey badger could be represented as:

$$x_i(t+1) = x_i(t) + \alpha \cdot \left(best\ solution\ so\ far - \ x_i(t)\right) + \beta \cdot random\ perturbation$$

where:

- $x_i(t)$ is the position of the i-th honey badger at time t.
- α controls the exploration factor (larger values mean more exploration).
- β controls the exploitation factor (larger values mean more exploitation).
- The random perturbation ensures the algorithm continues searching the space in a diverse manner.

4. **Updating the Best Solution:**
   - After each iteration, update the global optimum solution by checking the fitness values of all the honey badgers.
   - If a better solution is found by any honey badger, update the best solution.
5. **Termination:**
   - The approach stops when a predefined termination condition is met, such as:
     - Reaching the maximum number of iterations.
     - Achieving a solution with a fitness value above a certain threshold.
   - The best solution found during the execution of the algorithm is returned.

Fitness selection is a substantial aspect that influences the performance of HBA. The hyperparameter range procedure contains the solution-encoded system for estimating the efficiency of the candidate solution. The HBA considers accuracy to be the foremost standard for projecting the FF. The mathematical computation is given below:

$$Fitness \ = \ \max(P) \tag{34}$$

$$P = \frac{TP}{TP+FP}. \tag{35}$$

Here, TP and FP illustrate the true and false positive values.

4. Performance validation

The experimental analysis of the AMRNN-DRCTD model is examined under the BoT-IoT dataset [33]. It has 2056 samples below five classes, as depicted in Table 2. The total number of features is 32, but only 25 are selected in this dataset. The AMRNN-DRCTD technique is simulated by employing Python 3.6.5 tool on PC i5-8600k, 250GB SSD, GeForce 1050Ti 4GB, 16GB RAM, and 1TB HDD. The parameter settings are provided in the following: learning rate: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and batch size: 5.

Table 2. Details of the BoT-IoT dataset.

BoT-IoT Dataset	
Class Labels	**No. of Samples**
"DDoS"	500
"DoS"	500
"Recon"	500
"Theft"	79
"Normal"	477
Total Samples	**2056**

Figure 5 establishes the classifier results of the AMRNN-DRCTD approach for the BoT-IoT dataset. Figure 5(a),(b) demonstrates the confusion matrices with correct recognition and classification of 5 class labels below 70%TRPH and 30%TSPH. Figure 5(c) exhibits the PR values, indicating superior performance over all classes. This is followed by Figure 5(d), which exemplifies the ROC values, establishing capable outcomes with better ROC analysis for different class labels.

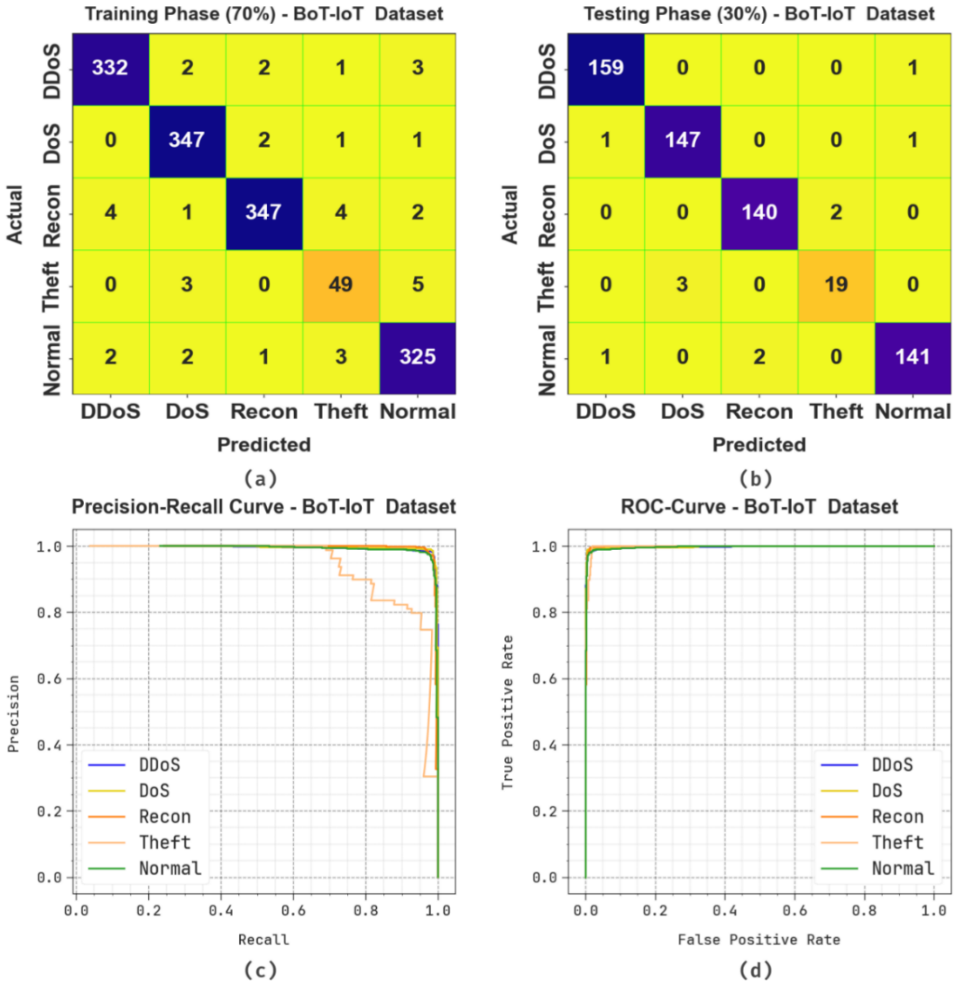


Figure 5. BoT-IoT dataset (a-b) 70%TRPH and 30%TSPH of the confusion matrix, and (c-d) curves of PR and ROC.

Table 3 and Figure 6, the attack detection of AMRNN-DRCTD technique on the BoT-IoT dataset under 70%TRPH and 30%TSPH is established. The results showed that the AMRNN-DRCTD methodology has effectively detected all class labels. Based on 70%TRPH, the AMRNN-DRCTD approach achieves an average $accu_y$ of 98.92%, $prec_n$ of 95.15%, $reca_l$ of 95.40%, F_{score} of 95.27%, and AUC_{score} of 97.36%. Furthermore, with 30%TSPH, the AMRNN-DRCTD system realizes an average $accu_y$ of 99.29%, $prec_n$ of 96.89%, $reca_l$ of 96.18%, F_{score} of 96.52%, and $AUC_{score}$of 97.86%.

Table 3. Attack detection of the AMRNN-DRCTD technique on the BoT-IoT dataset.

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	AUC_{score}
TRPH (70%)					
DDoS	99.03	98.22	97.65	97.94	98.55
DoS	99.17	97.75	98.86	98.30	99.06
Recon	98.89	98.58	96.93	97.75	98.23
Theft	98.82	84.48	85.96	85.22	92.66
Normal	98.68	96.73	97.60	97.16	98.30
Average	**98.92**	**95.15**	**95.40**	**95.27**	**97.36**
TSPH (30%)					
DDoS	99.51	98.76	99.38	99.07	99.47
DoS	99.19	98.00	98.66	98.33	99.01
Recon	99.35	98.59	98.59	98.59	99.09
Theft	99.19	90.48	86.36	88.37	93.01
Normal	99.19	98.60	97.92	98.26	98.75
Average	**99.29**	**96.89**	**96.18**	**96.52**	**97.86**

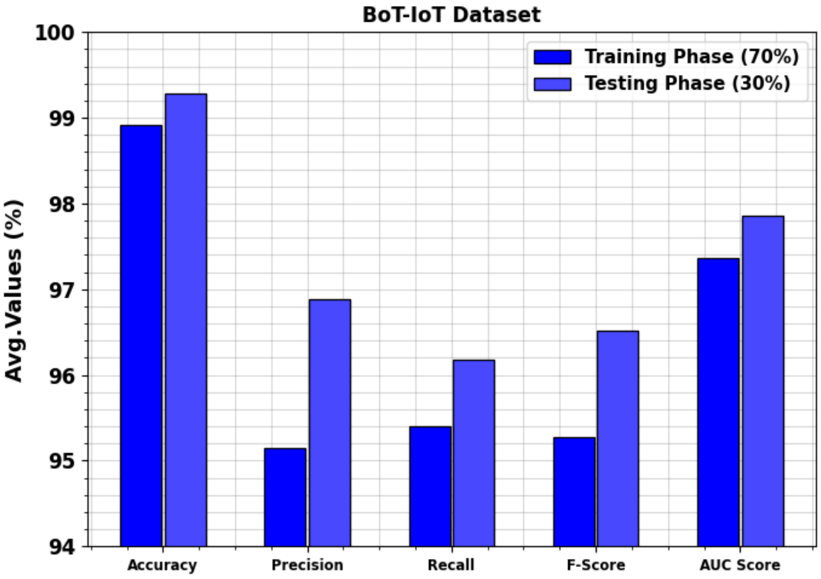


Figure 6. Average of the AMRNN-DRCTD technique on the BoT-IoT dataset.

Figure 7 illustrates the training (TRA) $accu_y$ and validation (VAL) $accu_y$ analysis of the AMRNN-DRCTD methodology on the BoT-IoT dataset. The $accu_y$ analysis is computed within the range of 0-25 epochs. The figure highlights that the TRA and VAL $accu_y$ analysis exhibited an increasing trend, which informed the capacity of the AMRNN-DRCTD technique and resulted in superior outcomes across multiple iterations. Furthermore, the TRA and VAL $accu_y$ leftovers closer across the epochs, which indicates inferior overfitting and exhibits maximal performance of the AMRNN-DRCTD method, assuring reliable prediction on hidden samples.

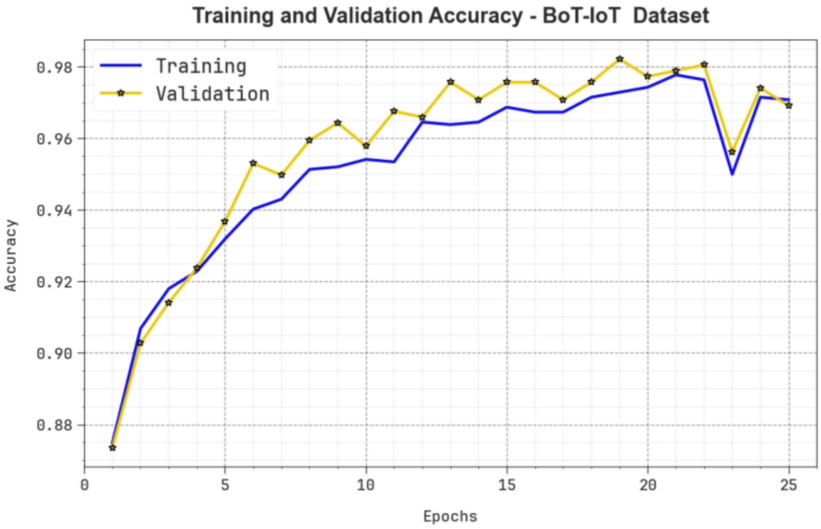


Figure 7. $Accu_y$ curve of the AMRNN-DRCTD technique on the BoT-IoT dataset.

In Figure 8, the TRA loss (TRALOS) and VAL loss (VALLOS) curves of the AMRNN-DRCTD technique on the BoT-IoT dataset are exhibited. The loss values are calculated across an interval of 0−25 epochs. The TRALOS and VALLOS values establish a diminishing tendency, informing the capacity of the AMRNN-DRCTD method to balance a trade-off between generalization and data fitting.

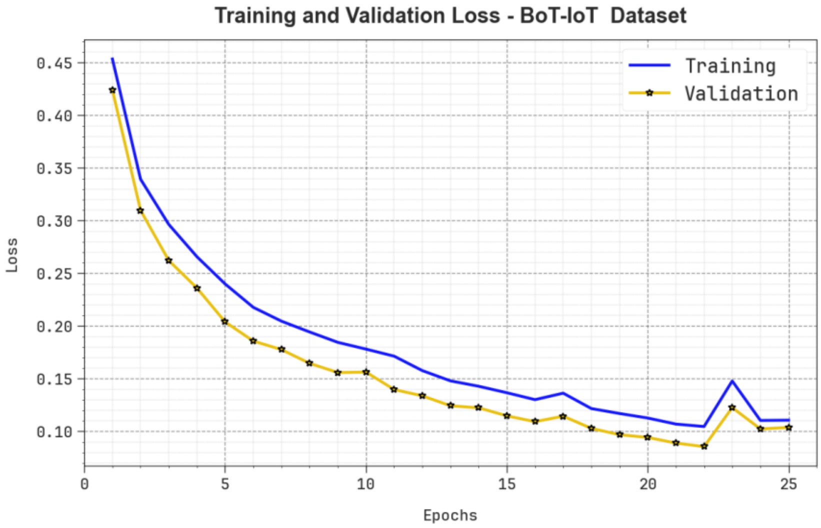


Figure 8. Loss analysis of the AMRNN-DRCTD methodology on the BoT-IoT dataset.

Table 4 and Figure 9 present the comparative study of the AMRNN-DRCTD approach on the BoT-IoT dataset with existing methods [19,35−38]. The table values stated that the proposed AMRNN-DRCTD has achieved effective performance. The existing techniques like IPSO, E-LSTM, ATFDNN, AE-multilyer perceptron (AE-MLP), XGBoost, Random Forest (RF), Decision Tree (DT), SVM, and KNN methods have attained the worst performance. Furthermore, the H3SC-DLIDS methodology was somewhat closer results. Besides, the AMRNN-DRCTD technique obtained better values of $accu_y$ of 99.29%, $prec_n$ of 96.89%, $reca_l$ of 96.18%, and F_{score} of 96.52%.

Table 4. Comparative outcomes of the AMRNN-DRCTD method with the existing techniques under the BoT-IoT dataset [19,35−38].

BoT-IoT Dataset				
Classifiers	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
AMRNN-DRCTD	99.29	96.89	96.18	96.52
IPSO	97.85	94.92	92.79	95.85
E-LSTM	97.58	95.69	94.30	95.29
ATFDNN	95.79	92.99	93.21	93.84
H3SC-DLIDS	99.07	96.68	95.70	96.17
AE-MLP	98.21	95.94	93.33	95.16
XGBoost	97.12	94.30	92.15	95.08
RF	97.03	95.00	93.72	94.60
DT	95.24	92.45	92.54	93.29
SVM Classifier	97.80	93.07	93.10	93.82
KNN Algorithm	99.00	93.70	93.76	94.53

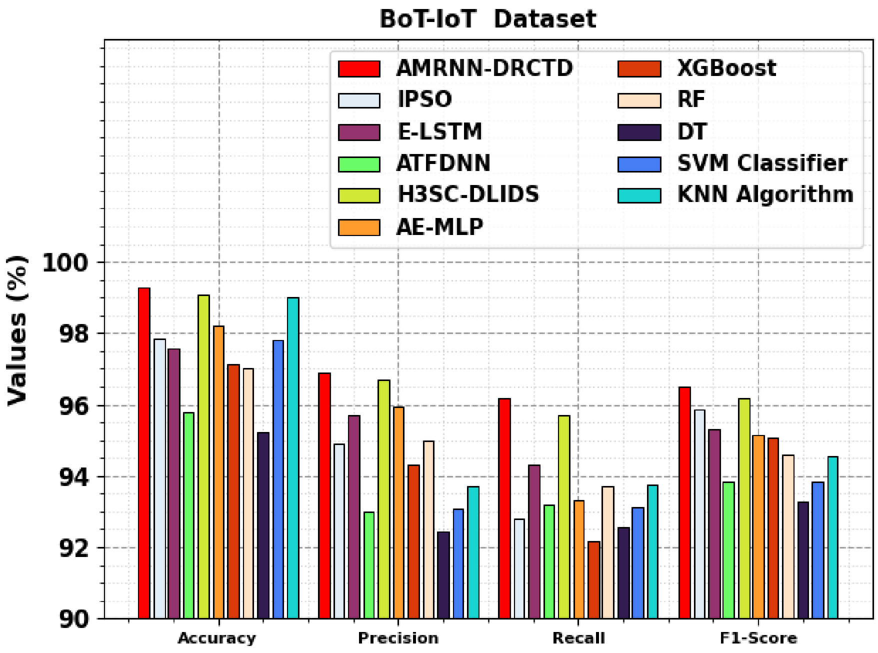


Figure 9. Comparative analysis of the AMRNN-DRCTD method with the existing techniques under the BoT-IoT dataset.

Table 5 and Figure 10 illustrate the computational time (CT) analysis of the AMRNN-DRCTD approach with the existing models. The times range from 4.55 seconds for the AMRNN-DRCTD model to 14.86 seconds for AE-MLP. IPSO and E-LSTM techniques take 13.40 seconds and 11.50 seconds, respectively, while ATFDNN and H3SC-DLIDS have CTs of 14.46 and 5.61 seconds. Other techniques such as XGBoost, RF, and DT have times of 10.56, 14.60, and 8.02 seconds, respectively. The SVM classifier takes 6.80 seconds, and the KNN method has a CT of 14.21 seconds. These varying times reflect the efficiency and computational demands of each classifier when applied to the dataset.

Table 5. CT evaluation of AMRNN-DRCTD approach with the existing models under the BoT-IoT dataset.

BoT-IoT Dataset	
Classifiers	**CT (sec)**
AMRNN-DRCTD	4.55
IPSO	13.40
E-LSTM	11.50
ATFDNN	14.46
H3SC-DLIDS	5.61
AE-MLP	14.86
XGBoost	10.56
RF	14.60
DT	8.02
SVM Classifier	6.80
KNN Algorithm	14.21

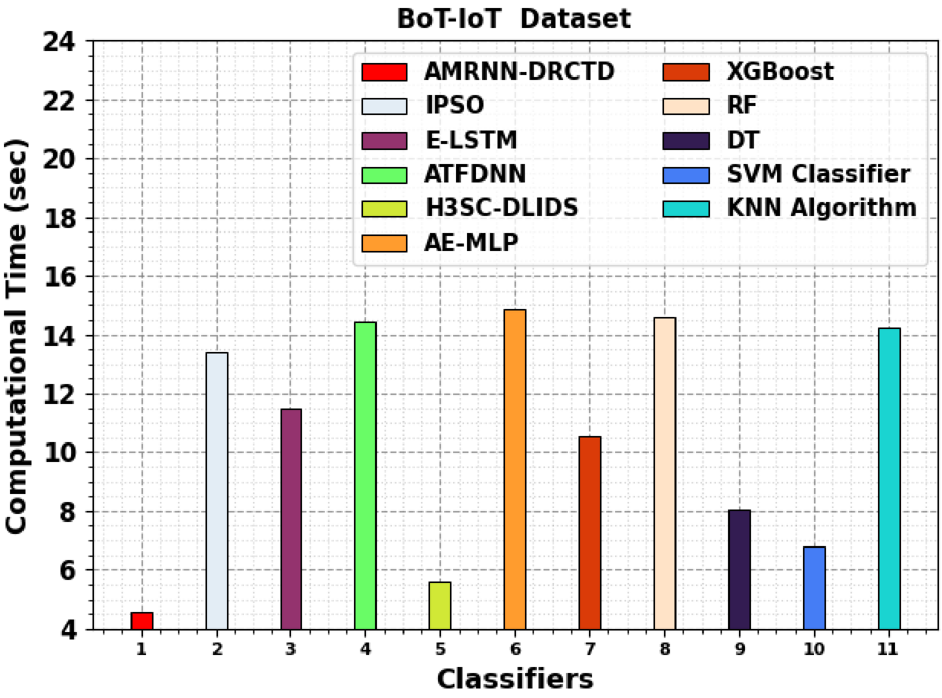


Figure 10. CT evaluation of AMRNN-DRCTD approach with the existing models under the BoT-IoT dataset.

Table 6 and Figure 11 describe the ablation study of the AMRNN-DRCTD model. The BoT-IoT dataset illustrates the performance of various techniques based on $accu_y$, $prec_n$, $reca_l$, and F_{score}. The AMRNN-DRCTD technique achieves the highest $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.29, 96.89, 96.18, and 96.52, respectively. CNN-BiLSTM-AM follows with an $accu_y$ of 98.51, $prec_n$ of 96.15, $reca_l$ of 95.43, and an F_{score} of 95.86. HBA performs slightly lower with an $accu_y$ of 97.74, $prec_n$ of 95.65, $reca_l$ of 94.79, and an F_{score} of 95.19. CTBOA has the lowest performance among the listed techniques, with an $accu_y$ of 97.12, $prec_n$ of 95.11, $reca_l$ of 94.06, and an F_{score} of 94.43. Furthermore, these results emphasize the varying levels of efficiency across the diverse techniques in detecting intrusions in IoT environments.

Table 6. Result analysis of the ablation study of the AMRNN-DRCTD model under the BoT-IoT dataset.

BoT-IoT Dataset				
Technique	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
AMRNN-DRCTD	99.29	96.89	96.18	96.52
CNN-BiLSTM-AM	98.51	96.15	95.43	95.86
HBA	97.74	95.65	94.79	95.19
CTBOA	97.12	95.11	94.06	94.43

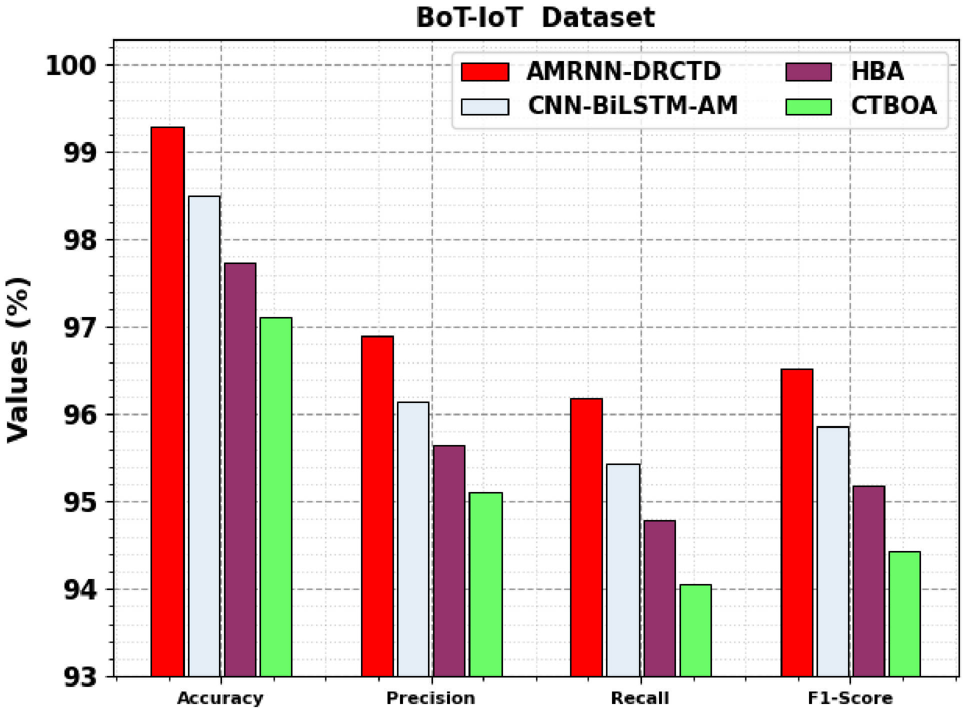


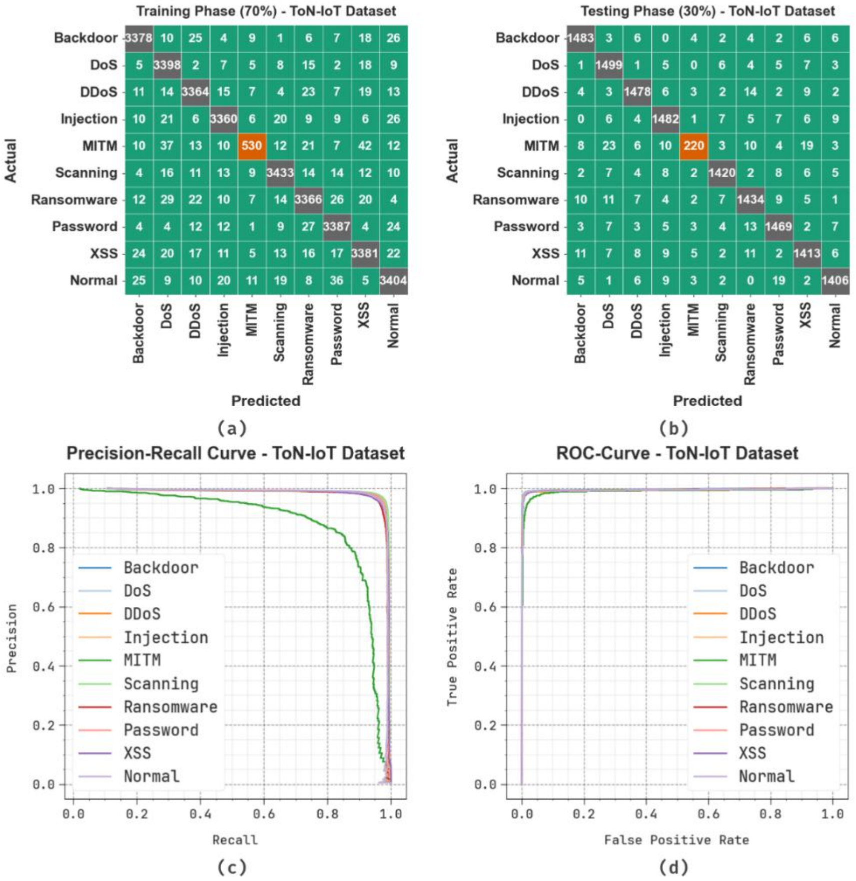
Figure 11. Result analysis of the ablation study of the AMRNN-DRCTD model under the BoT-IoT dataset.

Also, the performance results of the AMRNN-DRCTD technique are inspected below the ToN-IoT dataset [34]. It contains 46000 records under 10 classes, as depicted in Table 7. The total number of features is 42, but only 31 have been selected.

Table 7. Details of the ToN-IoT dataset.

ToN-IoT Dataset	
Type of Event	**Total Data Record**
"Backdoor"	5000
"DoS"	5000
"DDoS"	5000
"Injection"	5000
"MITM"	1000
"Scanning"	5000
"Ransomware"	5000
"Password"	5000
"XSS"	5000
"Normal"	5000
Total	**46000**

Figure 12 exhibits the classifier results of the AMRNN-DRCTD method on the ToN-IoT dataset. Figure 12(a),(b) illustrates the confusion matrices with perfect recognition and classification of each class label below 70%TRPH and 30%TSPH. Figure 12(c) demonstrates the PR values, specifying maximum outcomes over every class label. Followed by this, Figure 12(d) demonstrates the ROC graph, indicating proficient outcomes with high ROC analysis for dissimilar classes.



Training Phase (70%) - ToN-IoT Dataset

Actual \ Predicted	Backdoor	DoS	DDoS	Injection	MITM	Scanning	Ransomware	Password	XSS	Normal
Backdoor	3378	10	25	4	9	1	6	7	18	26
DoS	5	3398	2	7	5	8	15	2	18	9
DDoS	11	14	3364	15	7	4	23	7	19	13
Injection	10	21	6	3360	6	20	9	9	6	26
MITM	10	37	13	10	530	12	21	7	42	12
Scanning	4	16	11	13	9	3433	14	14	12	10
Ransomware	12	29	22	10	7	14	3366	26	20	4
Password	4	4	12	12	1	9	27	3387	4	24
XSS	24	20	17	11	5	13	16	17	3381	22
Normal	25	9	10	20	11	19	8	36	5	3404

(a)

Testing Phase (30%) - ToN-IoT Dataset

Actual \ Predicted	Backdoor	DoS	DDoS	Injection	MITM	Scanning	Ransomware	Password	XSS	Normal
Backdoor	1483	3	6	0	4	2	4	2	6	6
DoS	1	1499	1	5	0	6	4	5	7	3
DDoS	4	3	1478	6	3	2	14	2	9	2
Injection	0	6	4	1482	1	7	5	7	6	9
MITM	8	23	6	10	220	3	10	4	19	3
Scanning	2	7	4	8	2	1420	2	8	6	5
Ransomware	10	11	7	4	2	7	1434	9	5	1
Password	3	7	3	5	3	4	13	1469	2	7
XSS	11	7	8	9	5	2	11	2	1413	6
Normal	5	1	6	9	3	2	0	19	2	1406

(b)

Precision-Recall Curve - ToN-IoT Dataset

(c)

ROC-Curve - ToN-IoT Dataset

(d)

Figure 12. ToN-IoT dataset (a-b) 70%TRPH and 30%TSPH of the confusion matrix, and (c-d) curves of PR and ROC.

Table 8 and Figure 13 demonstrate the AMRNN-DRCTD approach's attack detection on the ToN-IoT dataset below 70%TRPH and 30%TSPH. The results showed that the AMRNN-DRCTD technique has effectively detected all class labels. Based on 70%TRPH, the AMRNN-DRCTD methodology achieves an average $accu_y$ of 99.26%, $prec_n$ of 95.74%, $reca_l$ of 94.68%, F_{score} of 95.16%, and AUC_{score} of 97.13%. Furthermore, with 30%TSPH, the AMRNN-DRCTD system reaches an average $accu_y$ of 99.28%, $prec_n$ of 95.92%, $reca_l$ of 94.45%, F_{score} of 95.08%, and AUC_{score} of 97.02%.

Table 8. Attack detection of the AMRNN-DRCTD technique on the ToN-IoT dataset.

Class Labels	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	AUC_{score}
TRPH (70%)					
Backdoor	99.34	96.99	96.96	96.97	98.30
DoS	99.28	95.50	97.95	96.71	98.70
DDoS	99.28	96.61	96.75	96.68	98.17
Injection	99.33	97.05	96.75	96.90	98.20
MITM	99.30	89.83	76.37	82.55	88.09
Scanning	99.37	97.17	97.09	97.13	98.37
Ransomware	99.12	96.03	95.90	95.97	97.71
Password	99.31	96.44	97.22	96.83	98.39
XSS	99.10	95.91	95.89	95.90	97.69
Normal	99.10	95.89	95.97	95.93	97.73
Average	**99.26**	**95.74**	**94.68**	**95.16**	**97.13**
TSPH (30%)					
Backdoor	99.44	97.12	97.82	97.47	98.73
DoS	99.28	95.66	97.91	96.77	98.68
DDoS	99.35	97.05	97.05	97.05	98.34
Injection	99.27	96.36	97.05	96.70	98.30
MITM	99.21	90.53	71.90	80.15	85.86
Scanning	99.43	97.59	96.99	97.29	98.36
Ransomware	99.14	95.79	96.24	96.02	97.86
Password	99.24	96.20	96.90	96.55	98.21
XSS	99.11	95.80	95.86	95.83	97.68
Normal	99.36	97.10	96.77	96.93	98.21
Average	**99.28**	**95.92**	**94.45**	**95.08**	**97.02**

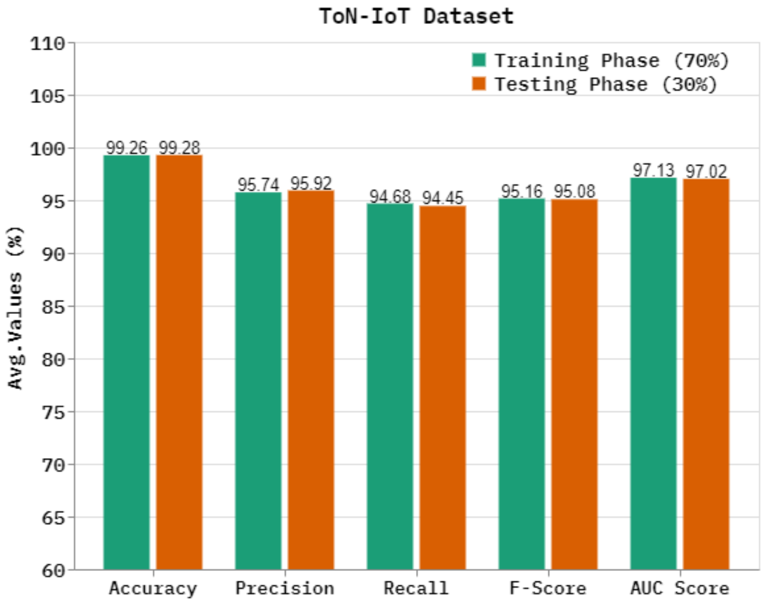


Figure 13. Average of the AMRNN-DRCTD technique on the ToN-IoT dataset.

Figure 14 illustrates the TRA $accu_y$ and VAL $accu_y$ analysis of the AMRNN-DRCTD technique on the ToN-IoT dataset. The $accu_y$ analysis is computed within the range of 0–25 epochs. The figure highlights that the TRA and VAL $accu_y$ analysis displays an increasing tendency, which informs the capacity of the AMRNN-DRCTD methodology with maximum outcomes across multiple iterations. In addition, the TRA and VAL $accu_y$ remnants are closer across the epochs, which indicates inferior overfitting and displays improved performance of the AMRNN-DRCTD approach, ensuring reliable prediction on unnoticed samples.

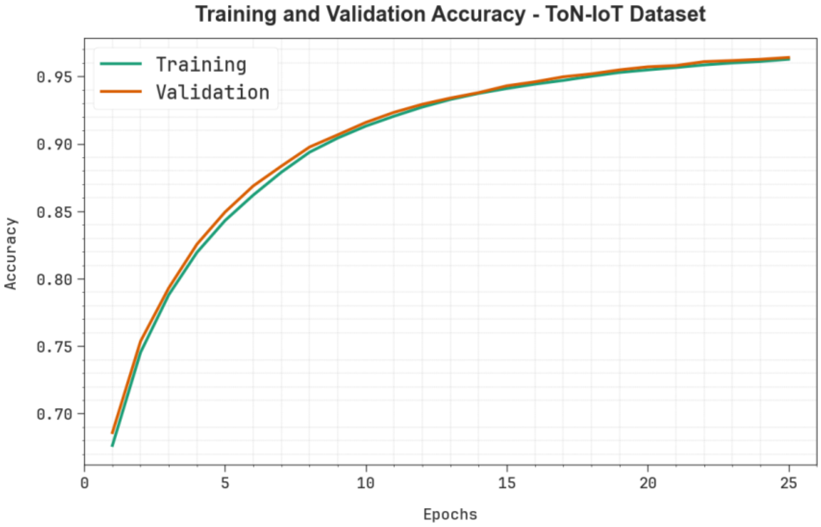


Figure 14. $Accu_y$ analysis of the AMRNN-DRCTD technique over the ToN-IoT dataset.

Figure 15 shows the TRALOS and VALLOS curves of the AMRNN-DRCTD technique on the ToN-IoT dataset. The loss values are calculated over an interval of 0–25 epochs. The TRALOS and VALLOS values exemplify a reducing trend, informing the capacity of the AMRNN-DRCTD method to balance a trade-off between data fitting and simplification.

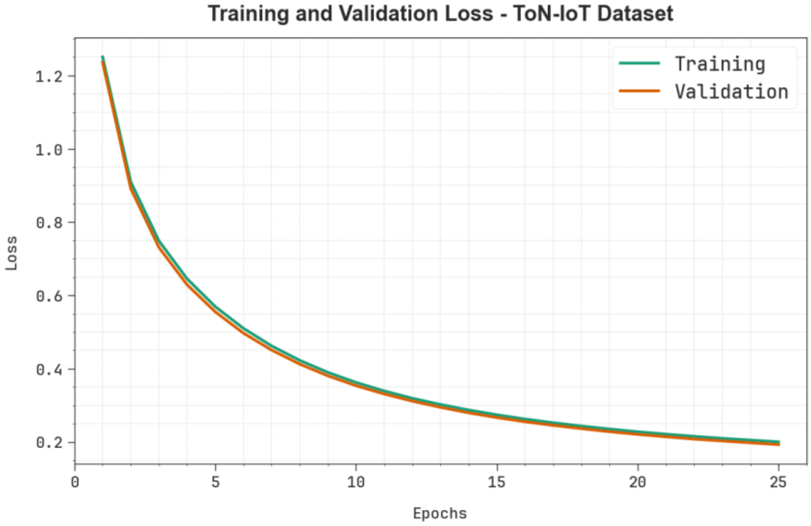


Figure 15. Loss graph of the AMRNN-DRCTD technique on the ToN-IoT dataset.

Table 9 and Figure 16 denote the comparative results of the AMRNN-DRCTD approach on the ToN-IoT dataset with existing systems [20,35−38]. The table values stated that the proposed AMRNN-DRCTD approach has achieved successful performance. The existing techniques like Naïve Bayes (NB), XGBoost, DenseNet, Inception Time, E-GraQhSAGE, Anomal-E, and NEGSC methods had the the worst performance. Furthermore, the AMRNN-DRCTD model accomplished better values of $accu_y$ of 99.28%, $prec_n$ of 95.92%, $reca_l$ of 94.45%, and F_{score} of 95.08%.

Table 9. Comparative analysis of the AMRNN-DRCTD technique under the ToN-IoT dataset [20,35−38].

ToN-IoT Dataset				
Technique	$\boldsymbol{Accu_y}$	$\boldsymbol{Prec_n}$	$\boldsymbol{Reca_l}$	$\boldsymbol{F_{score}}$
AMRNN-DRCTD	99.28	95.92	94.45	95.08
EDLM-PSOFS	98.63	94.00	91.09	93.69
CALR	95.44	93.33	93.13	93.23
EPSS	96.22	91.48	93.37	92.27
NB	96.96	92.07	90.05	91.08
XGBoost Method	98.54	91.03	91.69	90.05
DenseNet Model	98.76	93.39	90.93	91.86
Inception Time	98.48	93.27	90.30	92.90
E-GraQhSAGE	94.92	92.69	92.53	92.66
Anomal-E	95.56	90.86	92.66	91.59
NEGSC Model	94.71	91.97	92.46	92.27

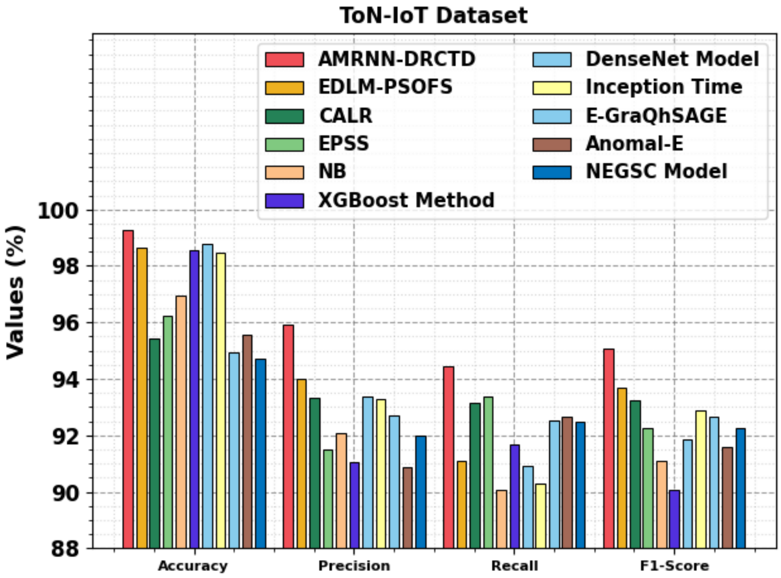


Figure 16. Comparative analysis of the AMRNN-DRCTD technique under the ToN-IoT dataset.

Table 10 and Figure 17 demonstrate the CT analysis of the AMRNN-DRCTD approach compared to existing methods. The ToN-IoT dataset presents the performance of various techniques based on their CT in seconds. The AMRNN-DRCTD technique exhibits the fastest CT at 6.99 seconds, followed by EDLM-PSOFS at 9.67 seconds. CALR and EPSS take significantly longer at 21.69 and 21.95 seconds, respectively. Other techniques like the NB and XGBoost methods have CTs of 19.40 and 12.81 seconds, respectively. The DenseNet model and Inception Time have CTs of 14.31 and 11.29 seconds, respectively. E-GraQhSAGE and Anomal-E take 19.96 and 17.32 seconds, while the NEGSC model takes the longest CT at 24.18 seconds. These results highlight the varying efficiency of the diverse techniques in processing and classifying IoT data.

Table 10. CT evaluation of AMRNN-DRCTD approach with existing methods under ToN-IoT dataset.

ToN-IoT Dataset	
Technique	**CT (sec)**
AMRNN-DRCTD	6.99
EDLM-PSOFS	9.67
CALR	21.69
EPSS	21.95
NB	19.40
XGBoost Method	12.81
DenseNet Model	14.31
Inception Time	11.29
E-GraQhSAGE	19.96
Anomal-E	17.32
NEGSC Model	24.18

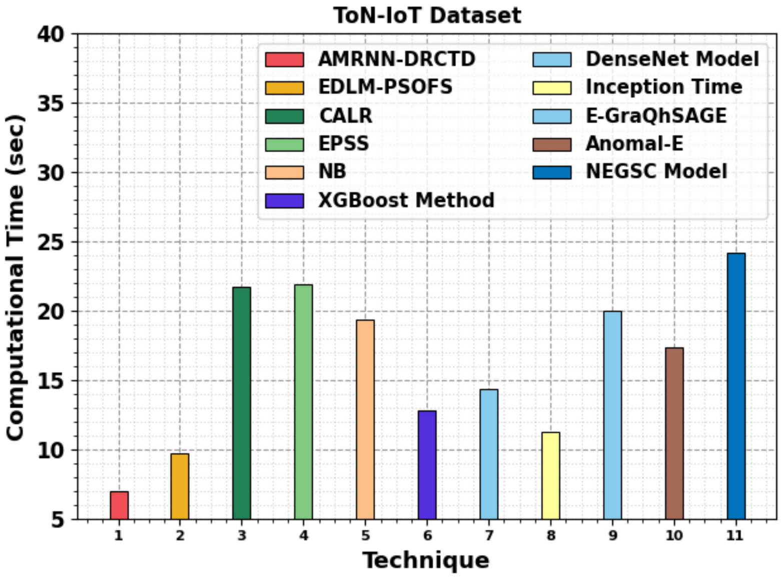


Figure 17. CT evaluation of the AMRNN-DRCTD approach with existing methods under the ToN-IoT dataset.

The ablation study of the AMRNN-DRCTD methodology is depicted in Table 11 and Figure 18. The ToN-IoT dataset presents the performance of various techniques based on $accu_y$, $prec_n$, $reca_l$, and F_{score}. The AMRNN-DRCTD technique achieves the highest $accu_y$ at 99.28%, with $prec_n$, $reca_l$, and F_{score} values of 95.92%, 94.45%, and 95.08%, respectively. The CNN-BiLSTM-AM technique follows with an $accu_y$ of 98.56%, $prec_n$ of 95.22%, $reca_l$ of 93.83%, and an F_{score} of 94.48%. The HBA technique depicts an $accu_y$ of 97.77%, with $prec_n$, $reca_l$, and F_{score} values of 94.66%, 93.16%, and 93.75%, respectively. The CTBOA technique attains an $accu_y$ of 97.02%, $prec_n$ of 94.15%, $reca_l$ of 92.59%, and an F_{score} of 93.03%. These results highlight the efficiency of the AMRNN-DRCTD technique, followed by the other models in terms of classification performance.

Table 11. Result analysis of the ablation study of the AMRNN-DRCTD methodology under the ToN-IoT dataset.

ToN-IoT Dataset				
Technique	$\boldsymbol{Accu_y}$	$\boldsymbol{Prec_n}$	$\boldsymbol{Reca_l}$	$\boldsymbol{F_{score}}$
AMRNN-DRCTD	99.28	95.92	94.45	95.08
CNN-BiLSTM-AM	98.56	95.22	93.83	94.48
HBA	97.77	94.66	93.16	93.75
CTBOA	97.02	94.15	92.59	93.03

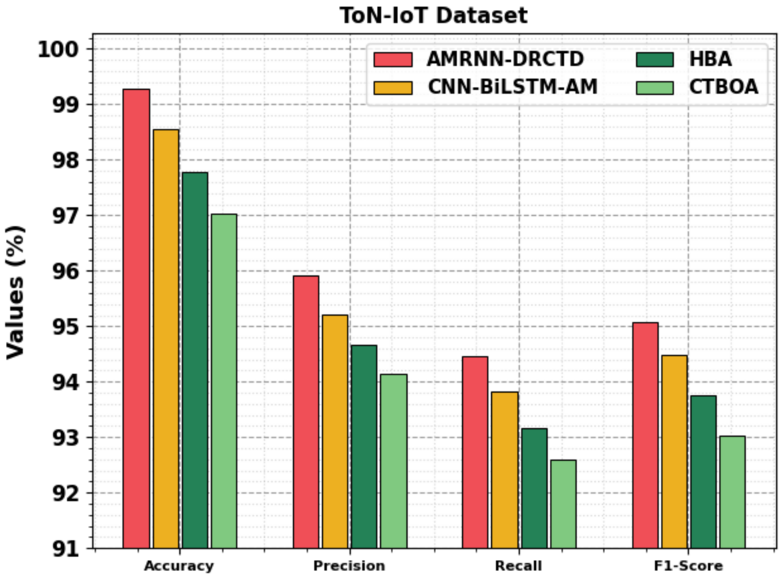


Figure 18. Result analysis of the ablation study of the AMRNN-DRCTD methodology under the ToN-IoT dataset.

5. Conclusions

In this study, a novel approach using the AMRNN-DRCTD model is proposed. The main goal of the proposed AMRNN-DRCTD model is to enhance the detection system for cyberattacks in IoT. I consider possible security breaches in BC and their influence on network processes. The AMRNN-DRCTD approach has zero-mean normalization, dimensionality reduction using CTBOA, a hybrid classification process, and parameter tuning using HBA to accomplish that. At the initial stage, the data normalization applies zero-mean normalization to transform data into a consistent format. The CTBOA is employed for the feature selection process. In addition, the proposed AMRNN-DRCTD approach performs a hybrid CNN-BiLSTM-AM technique for the classification process. Finally, the HBA-based hyperparameter selection range is achieved to optimize the detection outcomes of the CNN-BiLSTM-AM technique. The experimental evaluation of the AMRNN-DRCTD methodology is examined under the BoT-IoT dataset. The performance validation of the AMRNN-DRCTD methodology highlighted a superior accuracy output of 99.28% over existing approaches. The limitations of the AMRNN-DRCTD methodology comprise the dependency on specific network conditions and the challenge of handling highly unbalanced datasets, which can affect the overall performance of the detection system. Additionally, I primarily concentrate on a specific subset of IoT environments, restricting the model's generalizability to other application areas with diverse network topologies and resource constraints. The scalability of the proposed system in large-scale IoT networks remains a concern, as the computational resources needed for real-time threat detection may increase significantly. Moreover, while the approach exhibits promising results in controlled environments, the impact of environmental noise and external factors on system performance needs additional investigation. Future works should explore integrating more diverse datasets, improved anomaly detection techniques, and strategies to enhance the robustness and scalability of IoT security systems. Furthermore, integrating federated learning and edge computing may improve privacy and performance outcomes in distributed IoT networks.

Use of Generative-AI tools declaration

The author declares he has not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The author extends his appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number “NBU-FPEJ-2025-170-03”.

Conflict of interest

The author declares no conflicts of interest in this paper.

Data availability statement

The data supporting this study’s findings are openly available at https://research.unsw.edu.au/projects/bot-iot-dataset and https://research.unsw.edu.au/projects/toniot-datasets, reference numbers [33,34].

References

1. S. Mishra, The impact of AI-based cyber security on the banking and financial sectors, *J. Cybersecurity Inform. Manage.*, **14** (2024). https://doi.org/10.54216/JCIM.140101
2. S. F. Rabooki, B. Li, F. G. Febrinanto, C. Peng, E. Naghizade, F. Han, et al., GraphDART: Graph distillation for efficient advanced persistent threat detection, *arXiv preprin*, 2025. https://doi.org/10.48550/arXiv.2501.02796
3. Y. M. Tashtoush, D. A. Darweesh, G. Husari, O. A. Darwish, Y. Darwish, L. B. Issa, et al, Agile approaches for cybersecurity systems, IoT, and intelligent transportation, *IEEE Access*, **10** (2021), 1360−1375. https://doi.org/10.1109/ACCESS.2021.3136861
4. A. Abdullah, R. Hamad, M. Abdulrahman, H. Moala, S. Elkhediri, CyberSecurity: A review of internet of things (IoT) security issues, challenges and techniques, In: *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, IEEE, 2019, 1−6. https://doi.org/10.1109/CAIS.2019.8769560
5. M. Roopak, G. Y. Tian, J. Chambers, Deep learning models for cyber security in IoT networks, In: *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*, IEEE, 2019. https://doi.org/10.1109/CCWC.2019.8666588
6. R. O. Andrade, S. G. Yoo, L. Tello-Oquendo, I. Ortiz-Garcés, A comprehensive study of the IoT cybersecurity in smart cities, *IEEE Access*, **8** (2020), 228922−228941. https://doi.org/10.1109/ACCESS.2020.3046442
7. M. Kuzlu, C. Fair, O. Guler, Role of artificial intelligence in the Internet of Things (IoT) cybersecurity, *DIOT*, **1** (2021), 7. https://doi.org/10.1007/s43926-020-00001-4
8. K. Kimani, V. Oduol, K. Langat, Cyber security challenges for IoT-based smart grid networks, *Int. J. Crit. Infr. Prot.*, **25** (2019), 36−49. https://doi.org/10.1016/j.ijcip.2019.01.001

9. I. Lee, Internet of Things (IoT) cybersecurity: Literature review and IoT cyber risk management, *Future Internet*, **12** (2020), 157. https://doi.org/10.3390/fi12090157
10. A. Maseleno, Design of optimal machine learning based cybersecurity intrusion detection systems, *J. Cybersecurity Inform. Manage.*, **1** (2019).
11. E. Gelenbe, M. Nakip, IoT network cybersecurity assessment with the associated random neural network, *IEEE Access*, 2023. https://doi.org/10.1109/ACCESS.2023.3297977
12. D. M. Mirzaaxmedov, Cybersecurity risk analysis in the IoT: A systematic review, *Econ. Soc.*, **7** (2024), 145−151.
13. H. Zeng, M. Yunis, A. Khalil, N. Mirza, Towards a conceptual framework for AI-driven anomaly detection in smart city IoT networks for enhanced cybersecurity, *J. Innov. Knowl.*, **9** (2024), 100601. https://doi.org/10.1016/j.jik.2024.100601
14. P. Kaliyaperumal, S. Periyasamy, M. Thirumalaisamy, B. Balusamy, F. Benedetto, A novel hybrid unsupervised learning approach for enhanced cybersecurity in the IoT, *Future Internet*, **16** (2024), 253. https://doi.org/10.3390/fi16070253
15. A. L. Yakubu, Cybersecurity in the Internet of Things: Securing the connected world, *Fac. Nat. Appl. Sci. J. Comput. Appl.*, **2** (2024), 100−104.
16. D. R. N. Pitty, V. Jain, M. Tamilselvam, D. Haripriya, S. Bansal, Cybersecurity challenges in the era of the Internet of Things (IoT): Developing robust frameworks for securing connected devices, *Libr. Prog. Int.*, **44** (2024), 5644−5653.
17. A. Adewuyi, A. A. Oladele, P. U. Enyiorji, O. O. Ajayi, T. E. Tsambatare, K. Oloke, et al., The convergence of cybersecurity, Internet of Things (IoT), and data analytics: Safeguarding smart ecosystems, *World J. Adv. Res. Rev.*, **23** (2024), 379−394. https://doi.org/10.30574/wjarr.2024.23.1.1993
18. K. S. Prasad, E. L. Lydia, M. V. Rajesh, K. Radhika, J. V. N. Ramesh, N. Neelima, et al., Augmenting cybersecurity through attention based stacked autoencoder with optimization algorithm for detection and mitigation of attacks on IoT assisted networks, *Sci. Rep.*, **14** (2024), 30833. https://doi.org/10.1038/s41598-024-81162-y
19. S. Markkandeyan, A. D. Ananth, M. Rajakumaran, R. G. Gokila, R. Venkatesan, B. Lakshmi, Novel hybrid deep learning based cyber security threat detection model with optimization algorithm, *Cyber Secur. Appl.*, **3** (2025), 100075. https://doi.org/10.1016/j.csa.2024.100075
20. D. M. Dhanvijay, M. M. Dhanvijay, V. H. Kamble, Cyber intrusion detection using ensemble of deep learning with prediction scoring based optimized feature sets for IoT networks, *Cyber Secur. Appl.*, **3** (2025), 100088. https://doi.org/10.1016/j.csa.2025.100088
21. S. Misra, *A step by step guide for choosing project topics and writing research papers in ICT related disciplines*, In: Misra, S., Muhammad-Bello, B. (eds) Information and Communication Technology and Applications. ICTA 2020, Communications in Computer and Information Science, Springer, Cham, **3** (2021), 727−744. https://doi.org/10.1007/978-3-030-69143-1_55
22. E. A. Adeniyi, R. O. Ogundokun, S. Misra, J. B. Awotunde, K. M. Abiodun, *Enhanced security and privacy issue in multi-tenant environment of green computing using blockchain technology*, In: Blockchain applications in the smart era, Springer, Cham, 2022, 65−83. https://doi.org/10.1007/978-3-030-89546-4_4
23. W. Guo, S. Liu, L. Weng, X. Liang, Power grid load forecasting using a CNN-LSTM network based on a multi-modal attention mechanism, *Appl. Sci.*, **15** (2025), 2435. https://doi.org/10.3390/app15052435

24. H. Zhang, D. Zhu, Y. Gan, S. Xiong, End-to-end learning-based study on the Mamba-ECANet model for data security intrusion detection, *J. Inform. Technol. Policy*, 2024, 1–17. https://doi.org/10.62836/jitp.v1i1.219
25. Y. Wu, Z. Zang, X. Zou, W. Luo, N. Bai, Y. Xiang, et al., Graph attention and Kolmogorov–Arnold network based smart grids intrusion detection, *Sci. Rep.*, **15** (2025), 8648. https://doi.org/10.1038/s41598-025-88054-9
26. L. Sana, M. M. Nazir, J. Yang, L. Hussain, Y. L. Chen, C. S. Ku, et al., Securing the IoT cyber environment: Enhancing intrusion anomaly detection with vision transformers, *IEEE Access,* 2024. https://doi.org/10.1109/ACCESS.2024.3404778
27. H. Mancy, Q. H. Naith, SwinIoT: A hierarchical transformer-based framework for behavioral anomaly detection in IoT-Driven smart cities, *IEEE Access*, 2025. https://doi.org/10.1109/ACCESS.2025.3551207
28. J. Huang, Z. Chen, A. Z. Liu, H. Zhang, H. X. Long, Improved intrusion detection based on hybrid deep learning models and federated learning, *Sensors*, **24** (2024), 4002. https://doi.org/10.3390/s24124002
29. C. Su, J. Huang, S. Dong, Y. He, J. Li, L. Hu, et al., Transformer-gate recurrent unit-based hourly purified natural gas prediction algorithm, *Processes*, **13** (2025), 116. https://doi.org/10.3390/pr13010116
30. X. Ru, An improved butterfly optimization algorithm for numerical optimization and parameter identification of photovoltaic model, *Eng. Let.*, 2025.
31. M. Lin, Y. Luo, S. Chen, Z. Qiu, Z. Dai, Low-voltage biological electric shock fault diagnosis based on the attention mechanism fusion parallel convolutional neural network/bidirectional long short-term memory model, *Mathematics*, **12** (2024), 3984. https://doi.org/10.3390/math12243984
32. U. I. Maijeddah, M. Abdullahi, I. H. Hassan, A hybrid transfer learning model with optimized SVM using honey badger optimization algorithm for multi-class lung cancer classification, *Sci. World J.*, **19** (2024), 977–986. https://doi.org/10.4314/swj.v19i4.10
33. https://research.unsw.edu.au/projects/bot-iot-dataset
34. https://research.unsw.edu.au/projects/toniot-datasets
35. I. Tareq, B. M. Elbagoury, S. El-Regaily, E. S. M. El-Horbaty, Analysis of ToN-IoT, UNW-NB15, and Edge-IIoT datasets using DL in cybersecurity for IoT, *Appl. Sci.*, **12** (2022), 9572. https://doi.org/10.3390/app12199572
36. I. Katib, M. Ragab, Blockchain-assisted hybrid Harris Hawks optimization based deep DDoS attack detection in the IoT environment, *Mathematics*, **11** (2023), 1887. https://doi.org/10.3390/math11081887
37. M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, H. Jabeen, S. A. Haider, et al., Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSW-NB15 and Bot-IoT data-sets, *IEEE Access*, **10** (2021), 2269–2283. https://doi.org/10.1109/ACCESS.2021.3137201
38. C. Yang, L. Wu, J. Xu, Y. Ren, B. Tian, Z. Wei, Graph learning framework for data link anomaly detection, *IEEE Access*, 2024. https://doi.org/10.1109/ACCESS.2024.3445533