Mathematics

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

AIMS Mathematics, 10(12): 28981–29011.
DOI: 10.3934/math.20251275
Received: 04 October 2025
Revised: 13 November 2025
Accepted: 24 November 2025
Published: 10 December 2025

Research article

Advancing artificial intelligence-enabled cybersecurity framework using ensemble deep representation learning for intelligent cybersecurity in cloud-edge-IoT environments

Khalid A. Alattas[*]

Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah, 23890, Saudi Arabia

* **Correspondence:** Email: kaalattas@uj.edu.sa.

Abstract: In the digital era, rapid developments in interconnected Internet of Things (IoT) devices and the increasing integration of edge computing have significantly reshaped the current landscape, allowing a large number of connected systems to manage data quickly and effectively with minimal latency. On the other hand, cloud computing provides a flexible digital infrastructure in which data and resources are distributed across multiple locations, enabling users to access them from numerous industrial settings via the internet. However, the extensive interconnectedness of IoT networks, combined with the inherently limited security of many devices, creates heightened risks that can threaten the vital operations of hospitals, cities, and organizations. Reinforcing the security features of IoT devices before they are used across diverse systems can help reduce the attack surface. Conventional security systems often fail or are poorly suited to the dynamic and shared nature of cloud environments, making them insufficient for cloud-based systems. Despite ongoing use and a surge in complex cyberattacks, cloud platforms have tackled their inherent security challenges and vulnerabilities in the past three years. The rapid advancement of deep learning in artificial intelligence has brought numerous benefits for addressing industrial security concerns in the cloud. This manuscript presents an Advancing Intelligent Cybersecurity through Ensemble Deep Representation Learning and Feature Dimensionality Reduction (AICEDRL-FDR) technique in cloud-edge-IoT environments. The AICEDRL-FDR technique aims to provide a reliable framework for proactive threat mitigation in next-generation digital infrastructures. The AICEDRL-FDR method uses data pre-processing stages—cleaning, normalization, and standardization—to improve dataset consistency and quality. The maximum relevance minimum redundancy (mRMR) technique is utilized for dimensionality reduction to reduce redundant and irrelevant features. Ensemble deep representation methods, such as deep convolutional autoencoder (DCAE), fuzzy deep belief network (FDBN), and temporal convolutional network (TCN), are applied to the attack detection procedure. A broad array of experimental studies was conducted to ensure the AICEDRL-FDR method achieves superior performance on the Edge-IIoT [1] and ToN-IoT [2] datasets. The comparison analysis of the AICEDRL-FDR method showed superior accuracy of 99.31% and 99.24% across diverse evaluation measures on the dual dataset.

Keywords: artificial intelligence; cybersecurity; internet of things; feature selection; deep learning
Mathematics Subject Classification: 68T07, 94A60

1. Introduction

The convergence of edge computing and artificial intelligence (AI) has fundamentally changed how information is processed, managed, and used across most industries. Traditionally, the data was sent to a central server for processing [3]. Edge computing shifts data handling closer to the source, reducing latency and enabling instant decisions [4]. This paradigm leverages the networking capabilities of computer resources at the network edge, where IoT devices generate a larger volume of data. Such as approach is essential for achieving faster information processing and response times, particularly in healthcare monitoring systems, industrial systems, and autonomous vehicles [5]. Edge computing mitigates the problems of centralized cloud servers, decentralizes information processing, and optimizes network resources. AI incorporation improves the ability of IoT networks in edge computing environments [6].

The world is increasingly dependent on technology. A large amount of information is collected and generated through the widespread adoption of technologies such as cloud computing and IoT [7]. While data is utilized to improve services and meet evolving business requirements, cyber threats remain a key challenge. A cyber threat is generally defined as a malicious and targeted attempt by a person or organization to breach another optimizes information system [8]. Denial-of-service (DoS), phishing, ransomware, malware, Man-in-the-middle, Zero-day exploits, social engineering, SQL injection, and insider threats. These cybercrimes or security incidents can affect individuals and organizations, cause disruptions, and result in massive financial losses [9]. Cloud environments have become increasingly significant as AI-based applications proliferate across industries such as autonomous systems, finance, and healthcare. Because cloud ecosystems are inherently susceptible to cybersecurity threats, this dependence creates consistency issues and increased complex security [10].

The rapid digitalization of industries and the increasing reliance on cloud and network infrastructure have shaped how organizations secure, store, and share data [11]. While these technical developments have enabled unparalleled scalability and efficiency, they have also introduced significant cybersecurity vulnerabilities. Cyber-attacks are no longer limited to simple malware or viruses; they now include complex, rapidly evolving threats [12]. Conventional security measures are struggling to keep pace with changing attacks, requiring stronger, real-world defense devices. AI has become a crucial component of next-generation cybersecurity solutions, offering advanced capabilities for detecting, analyzing, and responding to attacks in real time [13]. AI-powered systems use machine learning (ML) techniques to recognize patterns, identify anomalies, and detect previously unseen attacks. Unlike conventional approaches that rely on static, rule-driven methods, AI systems continuously learn from novel information, making them more effective against developing cyber-attacks [14]. Figure 1 illustrates the architecture of cloud-edge-IoT environments.

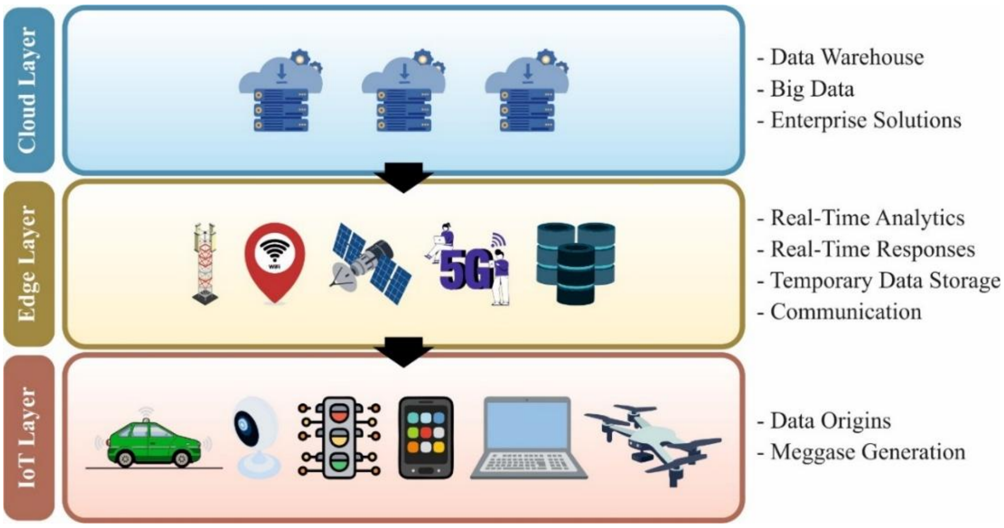

Figure 1. Architecture of cloud-edge-IoT environments.

This study presents a model called Advancing Intelligent Cybersecurity through Ensemble Deep Representation Learning and Feature Dimensionality Reduction (AICEDRL-FDR) in cloud-edge-IoT environments. The AICEDRL-FDR method uses data pre-processing stages—cleaning, normalization, and standardization—to improve dataset consistency and quality. The maximum relevance minimum redundancy (mRMR) technique is utilized for dimensionality reduction to reduce redundant and irrelevant features. Ensemble deep representation methods, such as deep convolutional autoencoder (DCAE), fuzzy deep belief network (FDBN), and temporal convolutional network (TCN), are applied to the attack detection procedure. A broad array of experimental studies is conducted to ensure the AICEDRL-FDR method achieves superior performance on the Edge-IIoT [1] and ToN-IoT [2] datasets. The key contributions of the AICEDRL-FDR method are listed below.

- The AICEDRL-FDR approach performs data pre-processing, including cleaning, normalization, and standardization, to enhance dataset quality and consistency. The process also improves data reliability and prepares it effectively for DL-based cybersecurity modelling. This process also contributes to more precise attack detection and overall system robustness in intelligent cybersecurity frameworks.
- The mRMR method is utilized for efficient dimensionality reduction. The model also ensures that only the most informative features are retained. The process also mitigates redundancy and removes attributes from the dataset. Additionally, the model improves learning efficiency and assists in faster, more accurate, and more reliable cybersecurity threat detection.
- Furthermore, an ensemble of deep representation learning models, namely DCAE, FDBN, and TCN, is employed to capture spatial, fuzzy, and temporal dependencies in cybersecurity data. The fusion model also improves the ability to learn intrinsic attack patterns and contextual relationships. It also contributes to achieving robust, adaptive, and highly accurate attack detection across dynamic cloud-edge-IoT environments.

- The novelty of the AICEDRL-FDR methodology lies in the ensemble of DL techniques that synergistically incorporate DCAE, FDBN, and TCN representations. Moreover, by integrating mRMR-based feature optimization, it ensures the selection of the most relevant and non-redundant features. This unique incorporation also enhances adaptability and intelligence in threat detection. Thus, an effective model is presented for dynamic and heterogeneous cloud-edge-IoT environments.

2. Literature review on cybersecurity approaches in cloud-edge-IoT environments

Farzaan et al. [15] introduced an innovative AI-based cyber incident response system specifically designed for cloud environments. In contrast with traditional techniques, the system uses innovative ML and AI methods to deliver scalable, precise, and seamlessly integrated results. This system has a significant impact on the field of AI-driven cybersecurity by demonstrating the effective integration of cloud infrastructure and AI techniques to address crucial gaps in cyber incident handling. Awan et al. [16] presented a new DL architecture, SecEdge, intended to improve the actual cybersecurity of mobile IoT platforms. This framework combines transformer-driven methods to effectively handle long-range dependencies, GNNs to model relational information, and federated learning to reduce latency and ensure information privacy. The customized learning mechanism continuously updates optimization parameters to counter evolving cyber-attacks. Alblehai [17] suggested an Intelligent Cybersecurity System Utilizing Self-Attention-driven DL and Metaheuristic Optimization Algorithm (ICSSADL-MHOA). This model is intended to enhance cybersecurity in IoT networks. At first, the information normalization phase uses min-max normalization to improve reliability, precision, and efficacy by organizing information into a consistent format. Moreover, the developed tuna swarm optimization (ITSO) method is applied for the FS procedure for detecting the appropriate attributes in the information. Dorothy et al. [13] addressed the flaws in conventional cloud attack intelligence systems, which are often based on static rules and signature-driven detection methods that are powerless to respond to evolving cyberattacks. These issues highlight the need for a more practical method, leading to the improvement of a new system that integrates an AI-based method. The work goes beyond static, rule-driven methods and proposes an AI-based approach to address deficiencies in current systems. Chaudhary et al. [18] introduced an innovative AI-driven method for improving cyber-attack detection and justification in cloud environments. By utilizing advanced anomaly detection and an ML approach, the proposed system continuously examines large volumes of data to identify and eliminate various threats dynamically. Among the main components are strong pre-processing, training on repetitive methods, automatic response mechanisms, threat detection, and ML method selection. Sathupadi et al. [19] addressed these limitations by proposing an edge-cloud hybrid architecture that utilizes edge tools for anomaly detection and cloud servers for complete failure prediction. A K-Neural Network (KNN) approach is used on edge tools to detect abnormalities in real time, reducing the need for continuous information transmission to the cloud. While the Long Short-Term Memory (LSTM) method in the cloud analyzes time-series data for failure prediction, it improves maintenance schedules and operational efficiency. Alrowais et al. [20] presented a novel Mayfly optimization (MFO) with a regularized extreme learning machine (RELM) technique, named MFO-RELM, for cybersecurity attack classification and recognition in an IoT platform. To accomplish this, the method pre-processes real-time IoT data into a meaningful format. Moreover, this technique performs classification and obtains pre-processed information.

Bhandari et al. [21] recommended a method for discovering malware threats, utilizing AI techniques to address diverse and released scenarios. The novel technique enables proactive tracking of network traffic to detect threats and malware on IoT platforms. Furthermore, the new method makes smart settings more aware and secure against potential upcoming attacks. Concurrent and performance testing of the Deep Neural Network (DNN) approach used in IoT devices is conducted to validate its applicability. Suryavanshi, Acharya, and Jadhav [22] developed a model by utilizing visual cryptography, Quality of Service (QoS)-aware optimization, and collaborative security frameworks. Aouedi and Piamrat [23] developed a scalable and energy-efficient solution for the Cloud-Edge-Internet of Things (CEI) continuum in industrial environments. It utilizes Hierarchical Federated Learning (HFL) to enable collaborative model training while conserving data locality and Spiking Neural Networks (SNN). Ali et al. [24] improved cybersecurity in AI-driven IoT-enabled smart cities by using advanced ML and DL methods. Xu and Xu [25] proposed a technique by employing Deep Q-Network (DQN) for optimal resource allocation, Deep Belief Network-Long Short-Term Memory (DBN-LSTM) technique for accurate patient health prediction, and Frog Leap Optimization (FLO) to improve model performance. Zang et al. [26] proposed HyperEye, a real-time system, by utilizing protocol-agnostic numerical features, protocol-specific text features, a cross-term fusion algorithm, and Genetic Algorithm-based Density-Based Spatial Clustering of Applications with Noise (GA-DBSCAN) optimization. Singh et al. [27] analyzed and detected botnet-based Distributed Denial of Service (DDoS) attacks in IoT networks by employing DL techniques. It also employs IoT traffic analysis, botnet behavior taxonomy, and comparative evaluation of DL-based detection methods. Saravanan and Santhosh [28] presented a Trustable Block Chain and Bandwidth Sensible-based Task Offloading (TBBS-TO) technique by employing the E-Poisson Enhanced Federated Trust (E-PEFT) consensus algorithm, Bi-directional Clustering Algorithm based on Local Density (BCALoD), and Multi-Agent Double Deep Q-Network (MA-DDQN) techniques. Sahi et al. [29] developed a lightweight model by using Fog-based Context Aware Feature Extraction using BranchyNET (FCAFE-BNET) approach. The model also utilizes early-exit Deep Neural Networks (DNNs) for faster inference, text-to-image data conversion for automated feature learning, and dynamic resource allocation across cloud and edge devices. Banse et al. [30] utilized AI-based evaluation, European Cybersecurity Certification Scheme (EUCS) compliance monitoring, and agile re-certification processes. Nagarjun and Rajkumar [31] introduced a method by utilizing DL and Blockchain (BC) technologies. The method also used Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) for multimodal feature analysis. Moreover, Quality of Service (QoS)-aware blockchain sidechains ensure transparency, immutability, and integrity of cloud data while enhancing real-time attack prediction. A detailed overview of DL-based attack detection is given in Table 1.

Table 1. Analysis of existing work in deep learning-based cybersecurity.

Authors	Purpose	Methodology	Datasets	Evaluations
Farzaan et al. [15]	To present an AI-driven cyber incident response method	ML	NSL-KDD, UNSW-NB15, and CIC-IDS-2017 Datasets	Accuracy of 90%, 75%, and 99%
Awan et al. [16]	To provide a comprehensive method for improving cybersecurity in mobile IoT settings	SecEdge, GNN	NSL-KDD, UNSW-NB15, and CICIDS2017 Datasets	Accuracy of 98.8%, 98.5%, and 98.7%

Continued on next page

Authors	Purpose	Methodology	Datasets	Evaluations
Alblehai [17]	To develop an extremely effective platform intended for IoT platforms	ICSSADL-MHOA, Min–Max, ITSO	ToN-IoT and Edge-IIoT Datasets	Accuracy of 99.37%
Dorothy et al. [13]	To present a model that delivers proactive protection measures against dynamic cyber-attacks	Random Forest and Isolation Forest Model	-	Accuracy of 95%
Chaudhary et al. [18]	To present an innovative AI-based method for improving cyber-attack recognition reduction in cloud environments	ML approaches	Multiple Datasets	Accuracy of 98.5%
Sathupadi et al. [19]	To present an edge-cloud hybrid platform for in-depth failure classification	KNN, LSTM	Multiple Sensor Datasets	Reduce latency by 35%
Alrowais et al. [20]	To achieve an effective recognition of cybersecurity attacks that are present in the IoT settings	MFO-RELM	N-BaIoT Datasets	Precision of 98.93%
Bhandari et al. [21]	To suggest a method for identifying malware threats utilizing AI approaches to encompass diverse and distributed scenarios	DNN	IoT-23 Datasets	Accuracy of 93%
Suryavanshi, Acharya, and Jadhav [22]	To improve intelligent cybersecurity and trust in cloud-edge-IoT industrial systems	Visual Cryptography, QoS-Aware Optimization, AI-Driven Trust Mechanisms	Edge-IIoT, ToN-IoT	Accuracy, Precision, Recall, F-Score
Aouedi and Piamrat [23]	To develop a scalable and energy-efficient CEI model	CEI, HFL, SNN	IoT Image Classification Data	Accuracy, Latency Reduction, Energy Efficiency
Ali et al. [24]	To strengthen cybersecurity in AI-driven IoT-enabled smart cities within advanced communication networks	AI, ML, DL	Real-World Smart City Data	Accuracy, Precision, Recall
Xu and Xu [25]	To enable real-time and accurate healthcare prediction with low latency	DQN, DBN-LSTM, FLO	IoT Healthcare Sensing Data	Accuracy, Makespan Reduction, Resource Utilization
Zang et al. [26]	To detect unknown encrypted malicious traffic in real-time	Cross-Term Fusion Algorithm, GA-DBSCAN Optimization	Open-World and Real-World Traffic Data	F1-Score 11.95% Improvement, Accuracy, Precision, Recall

Continued on next page

Authors	Purpose	Methodology	Datasets	Evaluations
Singh et al. [27]	To analyze and detect botnet-based DDoS attacks	DL	IoT Network Traffic Data	Accuracy, Detection Rate, F1-Score
Saravanan and Santhosh [28]	To enable efficient and trustable task offloading and resource allocation	TBBS-TO, E-PEFT, BCALoD, MA-DDQN	Commercial Blockchain Platform Data	Task Offloading Efficiency, Resource Utilization, Consensus Accuracy
Sahi et al. [29]	To develop a lightweight, adaptive IDS model	FCAFE-BNET, DNN, Text-to-Image Conversion, Dynamic Resource Allocation	NSL-KDD, UNSW-NB15, ToN-IoT, ADFA-LD	Accuracy, Inference Time Reduction, Detection Rate
Banse et al. [30]	To enable continuous and harmonized cybersecurity certification using AI-driven models	CaaS, EUCS Compliance Monitoring, AI	European cloud service environments	Certification Coverage, Compliance Rate, Trust Level
Nagarjun and Rajkumar [31]	To develop a robust anomaly detection method for cloud-based deployments by combining DL and BC technologies	Multimodal Feature Analysis, RNN, CNN, QoS-aware Blockchain Sidechains	Cloud System Logs	Precision: 98.5%, Accuracy: 99.4%, Recall: 98.3%, AUC: 99.2%

The existing studies encounter various limitations, namely high computational overhead, latency issues, and scalability challenges in dynamic networks. Many methods depend on centralized training or static rule-based detection, thus mitigating robustness against growing cyber-attacks. Lightweight and real-time detection for resource-constrained devices remains insufficiently addressed. Also, integration of multi-layered security, adaptive resource allocation, and trust evaluation mechanisms is still restricted in heterogeneous environments. A research gap remains in incorporating energy-efficient computation, low-latency anomaly detection, and collaborative AI-driven mechanisms with secure and transparent BC frameworks for highly dynamic IoT and cloud ecosystems. Additionally, adaptive optimization and automated feature selection for heterogeneous datasets have not been properly explored.

3. Proposed methodology

In this article, the AICEDRL-FDR method is proposed for cloud-edge IoT environments to provide a reliable framework for proactive threat mitigation in next-generation digital systems. To achieve this, the AICEDRL-FDR technique contains pre-processing, feature selection, and ensemble methods. Figure 2 represents the entire workflow of the AICEDRL-FDR technique.

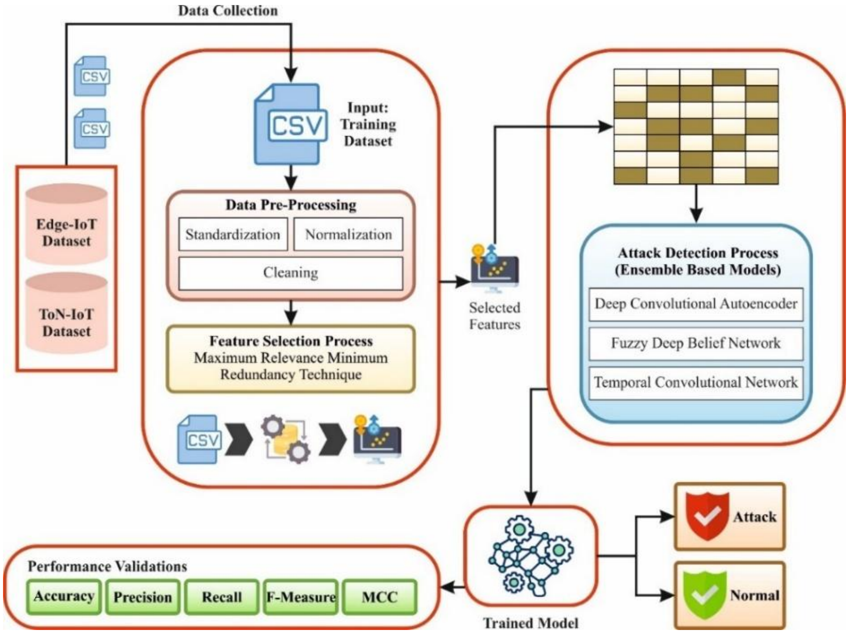

Figure 2. Entire workflow of the AICEDRL-FDR method.

3.1. Data pre-processing models

First, the AICEDRL-FDR method uses pre-processing steps, including normalization, standardization, and cleaning, to improve dataset quality and consistency. Data pre-processing is a vital stage that converts raw sensor data into a suitable format for effective identification [32]. The pre-processing procedure initiates with data normalization, transforming the data so that it has a standard deviation of 1 and a mean of 0, ensuring that features from diverse gadgets are comparable. The mathematical model of standardization is given in Eq (1).

$$x_{i,standardized} = \frac{x_i - \mu}{\sigma}. \tag{1}$$

Here, μ indicates the mean of feature values, σ signifies the SD of feature values, and x_i depicts the raw value of the feature. Normalization helps prevent features with wider ranges from overshadowing others during learning, ensuring that every feature contributes equally to the model.

Afterward, normalization scales the data to the particular range, usually [0,1], to guarantee that each feature is on the same scale. Standardization helps reduce bias introduced by the original scale of the data. The equation of normalization is presented in Eq (2).

$$x_{i,normalized} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}. \tag{2}$$

Here, $x_{\max}$ and $x_{\min}$ denote the maximal and minimal values of the feature, respectively.

The next step is data cleaning, which addresses issues such as missing values and noise. Missing values in IoT data are typically handled using mean imputation, which replaces missing values with the feature's mean. Mathematically, imputation is specified as in Eq (3).

$$x_{imputed} = mean(x). \tag{3}$$

Here, $mean(x)$ signifies the average of feature values. This approach upholds the integrity of the dataset and assures that missing values do not skew the identification.

To handle data noise, a moving average smoothing method is utilized. This model reduces short-term fluctuations and highlights longer-term trends by averaging data points within a window. The moving average is computed in Eq (4).

$$x_{smoothed} = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{4}$$

Here, N signifies the number of points in the smoothing window, and x_i represents every data point in the selected window, which is averaged to generate the smoothed value $x_{smoothed}$. This approach effectively filters out random noise, producing a clearer, more consistent signal from sensor data.

3.2. mRMR-based dimensionality reduction technique

For FS, the mRMR method is used to reduce irrelevant and redundant features [33]. This method demonstrates excellence in balancing maximum relevance to the target variable with minimum feature redundancy, thereby ensuring an optimal subset of informative attributes. The technique also preserves the original feature meaning and enhances interpretability, unlike other methods, namely principal component analysis (PCA) or mutual information. The robustness and computational efficiency of the model make it well-suited to high-dimensional cybersecurity datasets, thereby enhancing model accuracy and learning performance.

mRMR is a filtering technique for FS, that uses mutual information (MI) to measure dependencies between the class variable and the features, and among the features themselves. Its goal is to identify attributes, that maximize the information from the original dataset while using only a small subset of it. The optimal feature subset S is gradually built, beginning with only one attribute that has the highest MI with respect to the class variable $C\max_{f_i \in F} I(f_i, C)$. Until a predefined feature count is reached, this model iteratively adds an attribute to the subset.

$$\max_{f_i \in F-S}\left[\frac{I(f_{i;}\cdot C)}{\frac{1}{|S|}\Sigma_{f_j \in S} I(f_{i;}\cdot f_j)}\right]. \tag{5}$$

It should be noted that, in this paper, rather than utilizing the traditional mRMR as observed in Eq (5), we apply the variation provided in Eq (6). Another approach was to transform the denominator into an absolute connection; however, this would result in the redundancy properties being entirely gone, as it would link a pair of features through a single unrelated feature. Ultimately, a value of 1 was added to the denominator rather than, for example, values such as 0, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, or 0.8, as extensive experimentation proved that it performed better than the other values. For the new estimator, Eq (6) although it was not required. This ensured that the comparison was completed on equivalent terms, as the main aim was to estimate how the selection of the MI estimator affects the efficiency and reliability of mRMR. To properly investigate this, all parameters must be consistent.

$$\max_{f_i \in F-S}\left[\frac{I(f_{i;} \cdot C)}{1+\frac{1}{|S|}\Sigma_{f_j \in S} I(f_{i;} \cdot f_j)}\right]. \tag{6}$$

3.3. Ensemble deep representation methods

Additionally, ensemble deep representation methods such as DCAE, FDBN, and TCN are applied to the attack detection procedure. The fusion model is chosen for its complementary strengths: the DCAE effectively captures spatial patterns, FDBN models uncertainty with fuzzy logic, and the TCN shows effectiveness in learning temporal dependencies in sequential data. This integration provides a more comprehensive understanding of intrinsic attack behaviors than single-model approaches. Unlike conventional DL techniques, the ensemble improves detection accuracy, robustness, and adaptability in dynamic cloud-edge-IoT environments.

3.3.1. DCAE model

The DCAE utilizes deconvolutional and convolutional layers instead of fully connected layers, as in the DAE model [34]. Because of its use of CNN attributes, DCAE could also be suitable for image-processing applications. It achieves self-correction by isolating via a translation feature for latent features, local connections, and parameter sharing.

In this encoding method, convolutional layers are mapped to the internal layer to serve as a feature extractor. The concealed type of the n^{th} layer and the feature map are displayed below.

$$h_n = \sigma(x * WT_n + b_n). \tag{7}$$

Here, W represents the filter, b indicates the apposite n^{th} feature maps bias, σ represents a function of the activation (namely, sigmoid or ReLU), and $*$ means a 2D convolution procedure. This procedure was formerly used to transform the resulting features, whereas the deconvolutional layers implement the opposite task: rebuilding the latent model and restoring it to its original state.

$$y_n = \sigma\left(\sum_{n \in H} h_n * \widetilde{WT}_n + b\right). \tag{8}$$

The DCAE reduces reconstruction errors, thereby revealing latent representations within its internal layers. It selects the cross-entropy (logistic) loss function, which is consistent with research recommendations stating that systems trained with perceptive loss perform much better. The final layer often shows reduced performance, particularly in CNNs with deconvolutional layers. The backpropagation (BP) model, equivalent to that of classic networks, is applied to compute the error gradient.

$$E_1 = -\frac{1}{N}\sum_{n=1}^{N}(y_n \log \widehat{y_n} + (1-y_n)\log(1-\widehat{y_n})). \tag{9}$$

Here, the targeted pixel value of the image is represented by $\hat{y},$ and the reconstructed pixel value of the image is represented by $y.$

3.3.2. FDBN method

The FDBN method extends the basic perception of the processes involved in DBN learning [35]. Next, fuzzy set (FS) formulations are developed to improve the DL model. Each RBM is linked to a few biases and weights that define the energy of the visible layer (VL) and the hidden layer (HL). The step-by-step description of this model is clarified as follows:

Step-1: In a DBN, the inner RBMs are initialized randomly. The Bernoulli distribution is measured for an RBM.

Step-2: The RBM is pre-trained to pre-train the DBN using the contrastive divergence learning model. Here, the biases and weights are fuzzified.

Step-3: In the unsupervised method, the system is pre-trained utilizing the CD model.

Step-4: The RBM, linked to the linear map, is assigned to the final layer of the DBN.

Step-5: Lastly, the backpropagation model is employed to fine-tune the system. For an RBM, the function of energy formulation is assumed as shown below:

$$E(v,h;W,b,c) = -h^TWv - c^Tv - b^Th \tag{10}$$

$$= -\sum_j \sum_k W_{jk}h_jv_k - \sum_k c_kv_k - \sum_j c_jv_j. \tag{11}$$

Here, W denotes a weight, h and v are vectors representing the VL and HL, and h_j and v_k represent the jth and kth units. The biases associated with HL and VL are denoted by b and c, respectively.

For all pairs of VL and HL, the network likelihoods are set as:

$$p(v,h\colon W,b,c) = \frac{1}{\bar{\tilde{\tilde{Z}}}}e^{-E(v,h;W,b,c)}. \tag{12}$$

Here, $Z = \sum_{v,h} {}_{e^{-E(v,h;W,b,c)}}$ denotes the function of crisp partition, which totals over all probable HL and VL. The inferences employed are the conditional inference of both h given v or v provided h.

$$p(h|v) = \prod_j p\left(h_j|v\right). \tag{13}$$

For an individual distribution, $p(h_j|v)$, h_j represents randomly generated variables. The likelihood of h_j being equivalent to 1 is defined as

$$p\left(h_j = 1|v\right) = \sigma\left(b_j + W_jv\right). \tag{14}$$

Here, $\sigma(x) = \frac{1}{1+e^{-x}}$ represents the activation function of the sigmoidal.

Likewise, the conditional distribution for v assuming h is given below:

$$p(v|h) = \prod_k p\left(v_k|h\right). \tag{15}$$

$$p(v_k = 1|h) = \sigma(c_k + h^TW_k). \tag{16}$$

The inference projected equation is given below:

$$P(\vartheta_{u+1}=1)=E_{P(\vartheta_u)}(\sigma(W_u^T[\vartheta_u])+b_u). \tag{17}$$

The biases and weights are set at random with real numerals, which often leads to hesitation due to the numerous possible values.

This might delay the DBN's overall performance. FSs are categorized by their membership functions, which are termed grades. In an FS, if the membership grades are expressed as crisp values, the FS is called a normal or type-1 (T1) FS. An FS "F" is expressed in the mathematical formulation as below:

$$F=\{(x,\mu_F(x))|\forall x\in X\}, \tag{18}$$

or

$$F=\int_{\in X}\mu_F(x)/x. \tag{19}$$

While $\mu_F(x)$ denotes a membership grade of x for every $x\in X$, in reality, trapezoidal and triangular MFs are employed for perfecting the FSs. Given the uncertainty about the biases and weights, four diverse MFs are selected for examination.

Assume "w" represents the crisp weight at random, and m_L, and m_R represent the left and proper range of the MF. The membership value is denoted by μ, with $\mu\in[0,1]$. Therefore, W, b, and c are re-expressed as $\widetilde{W},\tilde{b}$, and $\tilde{c}$. These associations between VL and HL create a matrix.

Now, an initial RBM output is employed as the input to the following RBM, and novel weights are acquired as the transpose of preceding layer weights. The complete training ensues, and the weights are adjusted until the optimal weights are obtained.

$$\tilde{E}(v,,h:\widetilde{W},\tilde{b},\tilde{c})=-h^T\widetilde{W}v-\tilde{b}^Th-\tilde{c}^Tv. \tag{20}$$

Here, $\widetilde{W}$, $\tilde{b}$, and $\tilde{c}$ denote fuzzy weights and biases.

$$p(v,h:\widetilde{W},\tilde{b},\tilde{c})=\frac{1}{\tilde{Z}}e^{-\tilde{E}(v,h;\widetilde{W},\tilde{b},\tilde{c})}. \tag{21}$$

While, $\tilde{Z}=\sum_{v,h}e^{\tilde{E}(v,h;\widetilde{W},\tilde{b},\tilde{c})}$ is expressed as a function of fuzzy partition, which is the summary of each probable HL and VL vectors.

The conditional probability of HL and VL is stated as:

$$p(h_j=1|v)=\sigma(\tilde{b}_j+\widetilde{W}_jv). \tag{22}$$

$$p(v_k=1|h)=\sigma(\tilde{c}_k+h^T\widetilde{W}_k). \tag{23}$$

Here, $\tilde{\sigma}(\cdot)$ represents the function of fuzzy logistic with fuzzy arguments.

$$\tilde{P}(\vartheta_{u+1}=1)=\tilde{E}_{\tilde{P}(\vartheta_u)}(\tilde{\sigma}(\widetilde{W}_u^T[\vartheta_u])+\tilde{b}_u). \tag{24}$$

While θ_u denotes an associated dual variable at random. Besides, the conditional probabilities of the $(u+1)th$ node are evaluated for every probable integration of dual states. Figure 3 indicates the framework of the FDBN model.

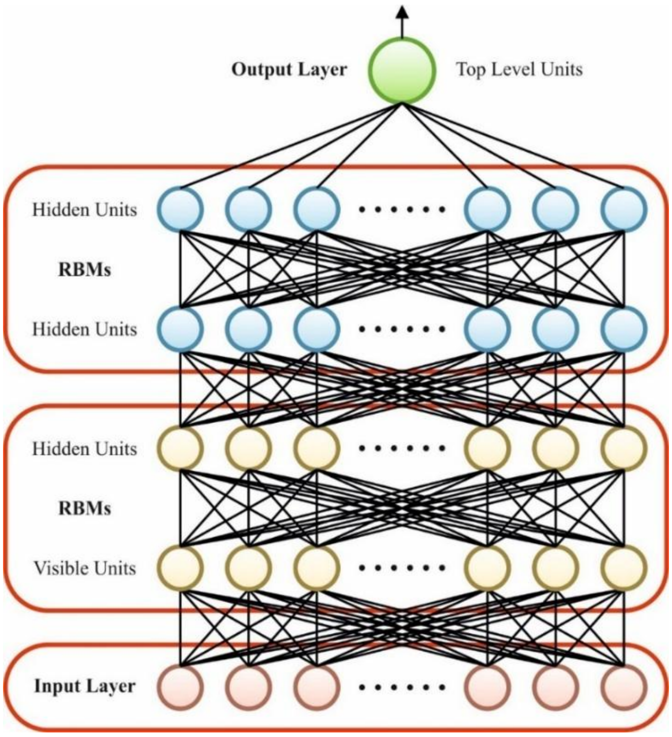

Figure 3. Framework of FDBN model.

Here, the basic perception of the contrastive divergence (CD) model used to train RBMs is described. Initially, the training vector is predetermined based on the RBM's VL. Next, the dual states of HLs are computed. The likelihood in Eq (14) was employed for the reconstruction procedure. Therefore, in fuzzy weights, an alteration between dual units i and j is described as below:

$$\Delta\tilde{w}_{ij} = \in (< v_i^n h_j^n > < v_i^{n+1} h_j^{n+1} >). \tag{25}$$

Here, $\in$ denotes the learning rate $n+1$, and n represents the reconstruction step.

In CD training, the HL h_0 from v_0 is computed first, followed by v_1. Therefore, h_0 is assessed using an inference. Nowadays, as an average probability is an inference output, v_1 and h_1 are computed appropriately. After training the CD, typical backpropagation is employed to complete the DBN. Now, in the learning stage, parameters such as biases and weights play a vital role.

3.3.3. TCN model

The TCN model is intended for time-series data processing [36]. Compared with conventional CNNs, TCNs use causal convolution to ensure that the output at any time step does not depend on future time steps, preserving the temporal sequence of the data. The receptive area of the TCN with diverse dilation factors is also represented.

$$y(t) = \sum_{k=0}^{K-1} f\,(k) \cdot x(t - d \cdot k). \tag{26}$$

Here, f depicts the convolutional kernel, $x(t)$ signifies the sequence of input, d represents the rate of dilation, K depicts the size of the kernel, and $y(t)$ refers to the sequence of output.

To alleviate overfitting and improve model performance, factors such as the learning rate, model architecture, and techniques such as activation functions, batch normalization, and dropout are often employed. Furthermore, to safeguard the steadiness of connections in deep TCN methods, a residual connection is presented.

Leaky ReLU (L-ReLU) is used as the activation function rather than conventional ReLU, as it retains a smaller slope on the negative axis, allowing a few negative values to move.

$$f(x) = \begin{cases} x, x > 0; \\ ax, x \leq 0. \end{cases} \tag{27}$$

Here, a represents the slope for the negative input area, and x refers to the value of the input. The "ax" part indicates that the input x is negative, and "a" refers to a smaller constant.

4. Performance assessment and metrics

In this section, the performance assessment of the AICEDRL-FDR methodology is verified under the Edge-IIoT [1] and ToN-IoT [2] datasets.

4.1. Result analysis on the edge-IIoT dataset

The Edge-IIoT dataset is a complete set of network event records intended for intrusion detection and cybersecurity studies in IIoT settings. It comprises 24,000 records, uniformly distributed across 12 event types, with 2,000 records per event type. The dataset includes both normal network activity and various cyberattack scenarios, including DDoS attacks (UDP, TCP, ICMP, and HTTP), SQL injection, password attacks, file upload attacks, backdoor intrusions, cross-site scripting (XSS), ransomware, and fingerprinting attempts. This balanced dataset enables practitioners and researchers to develop, test, and benchmark security methods for identifying a broad range of malicious actions in IIoT networks. The number of attributes is 63, but only 31 were selected.

Figure 4 presents the classifier result of the AICEDRL-FDR model on the Edge-IIoT dataset. Figure 4(a) shows the PR study, which represents maximal performance across all classes. Finally, Figure 4(b) presents the ROC analysis, showing that skilled outcomes have higher ROC values across various classes.

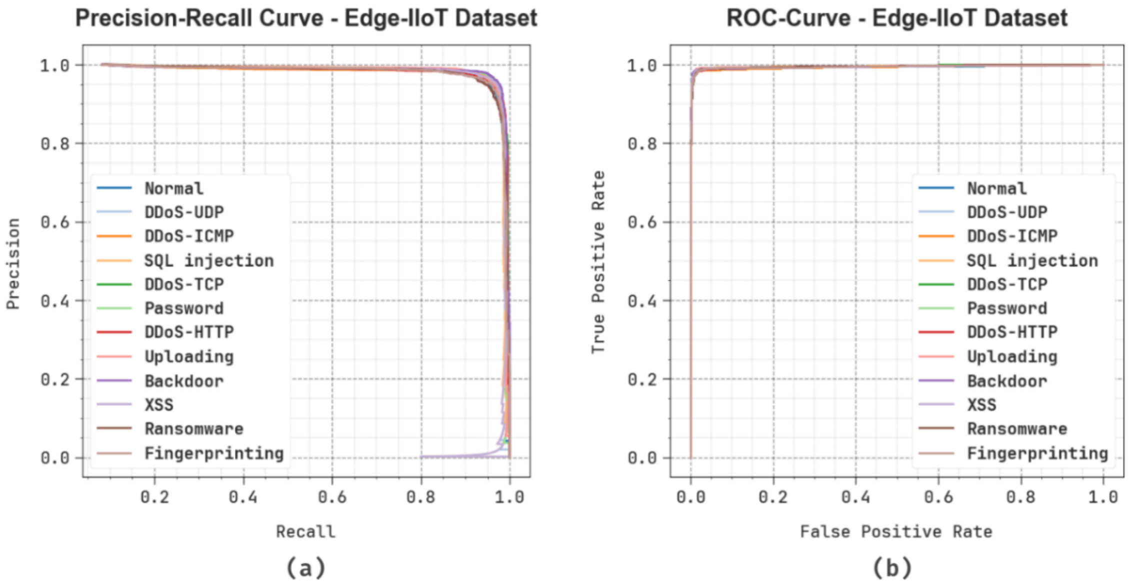

Figure 4. Edge-IIoT dataset: (a) PR curve, (b) ROC curve.

Table 2 and Figure 5 demonstrate the parameters of attack detection of the AICEDRL-FDR methodology on the Edge-IIoT dataset. Under 70%TRPHE, the AICEDRL-FDR model attains average $accur_y$, $preci_n$, $recal_l$, $F_{Measure}$, MCC, and Kappa of 99.31%, 95.89%, 95.89%, 95.89%, 95.52%, and 96.19%, respectively. Moreover, on 30%TSPHE, the AICEDRL-FDR approach achieves average $accur_y$, $preci_n$, $recal_l$, $F_{Measure}$, MCC, and Kappa of 99.30%, 95.81%, 95.81%, 95.80%, 95.42%, and 96.07%, respectively.

Table 2. Attack detection of the AICEDRL-FDR model on the Edge-IIoT dataset.

Class labels	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$	MCC	Kappa
TRPHE (70%)						
Normal	99.29	94.71	96.76	95.72	95.34	96.03
DDoS-UDP	99.29	95.54	96.08	95.81	95.42	96.12
DDoS-ICMP	99.23	96.15	94.57	95.35	94.94	95.61
SQL injection	99.33	95.64	96.51	96.08	95.71	96.51
DDoS-TCP	99.30	95.87	95.66	95.77	95.39	96.12
Password	99.46	96.67	96.60	96.64	96.34	97.01
DDoS-HTTP	99.30	95.73	95.87	95.80	95.41	96.14
Uploading	99.32	96.07	95.73	95.90	95.53	96.21
Backdoor	99.51	97.13	97.00	97.07	96.80	97.32
XSS	99.41	96.16	96.65	96.41	96.09	96.72
Ransomware	99.14	95.05	94.72	94.89	94.41	95.01
Fingerprinting	99.21	95.98	94.55	95.26	94.83	95.49
Average	99.31	95.89	95.89	95.89	95.52	96.19

Continued on next page

Class labels	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$	MCC	Kappa
TSPHE (30%)						
Normal	99.15	95.39	94.61	95.00	94.53	95.26
DDoS-UDP	99.31	95.47	95.80	95.64	95.26	95.97
DDoS-ICMP	99.36	96.80	95.51	96.15	95.80	96.45
SQL injection	99.42	95.65	97.00	96.32	96.01	96.55
DDoS-TCP	99.11	95.10	94.48	94.79	94.30	94.92
Password	99.40	96.18	97.22	96.69	96.37	96.87
DDoS-HTTP	99.26	95.48	95.64	95.56	95.16	95.70
Uploading	99.31	95.33	96.30	95.81	95.44	96.22
Backdoor	99.35	94.98	97.34	96.14	95.80	96.55
XSS	99.28	96.59	95.05	95.81	95.42	96.08
Ransomware	99.33	96.34	95.34	95.84	95.48	96.27
Fingerprinting	99.33	96.38	95.39	95.88	95.52	96.04
Average	99.30	95.81	95.81	95.80	95.42	96.07

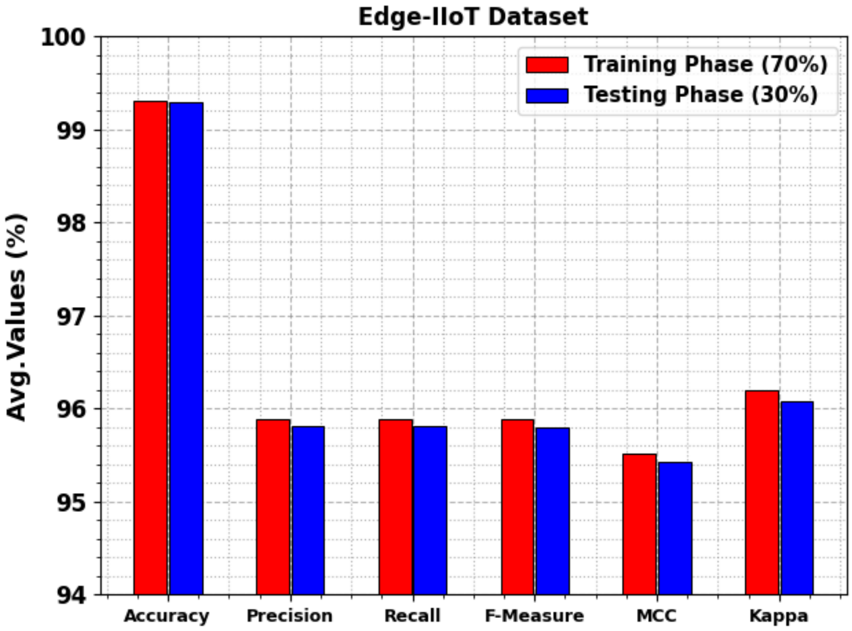

Figure 5. Average values of the AICEDRL-FDR model on the Edge-IIoT dataset.

Figure 6 displays the training (TRAN) and validation (VALD) $accu_y$ of the AICEDRL-FDR method on the Edge-IIoT dataset across 50 epochs. Both curves gradually rise and converge slowly, indicating that the model is successfully learning. The VALD $accu_y$ consistently remains greater than the TRAN $accu_y$, suggesting that the method is not overfitting and is simplifying well to the hidden data. The variations in $accu_y$ are predictable due to task complexity, but the overall upward tendency determines the robustness and stability of the method.

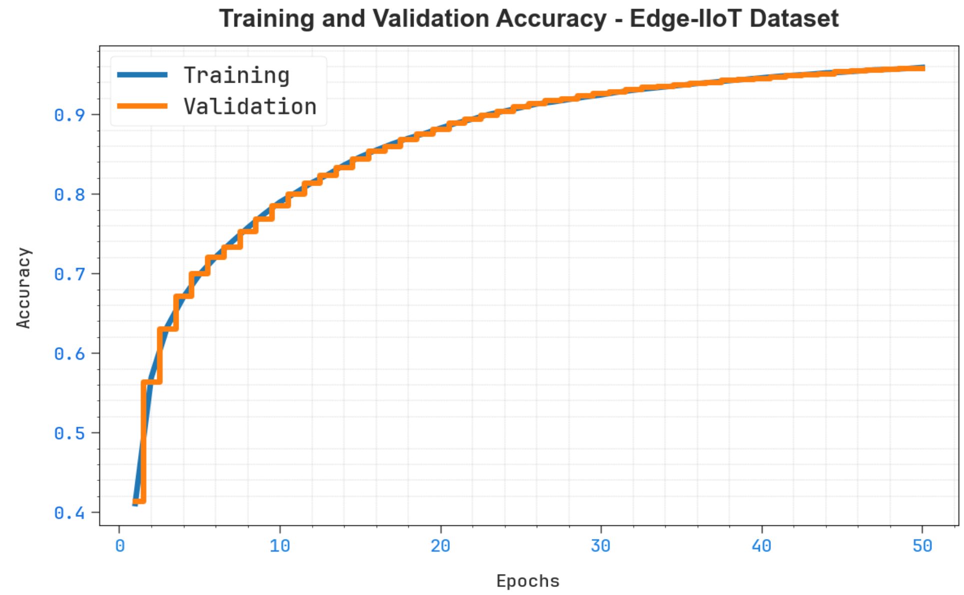

Figure 6. $Accur_y$ curve of the AICEDRL-FDR method on the Edge-IIoT dataset.

Figure 7 represents the TRAN and VALD loss of the AICEDRL-FDR model on the Edge-IIoT dataset across 50 epochs. Both curves show a steady downward trend, indicating that the method is successfully reducing the learning error. The VALD loss remains lower than the training loss throughout the maximum epochs, suggesting excellent generalization and no signs of overfitting. Any variations seen are becoming increasingly stable and reliable.

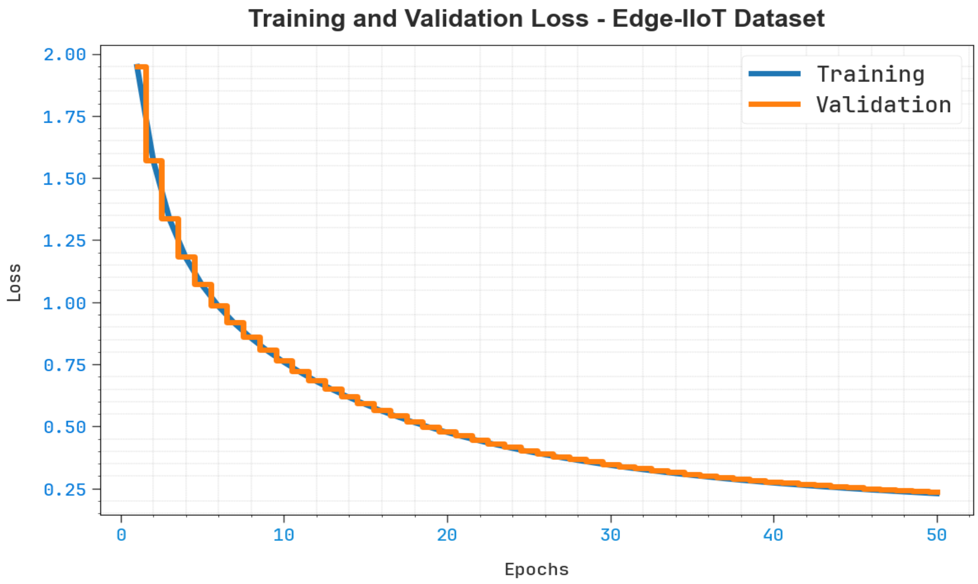

Figure 7. Loss curve of the AICEDRL-FDR method on the Edge-IIoT dataset.

Table 3 and Figure 8 provide a comparative outcome of the AICEDRL-FDR model on the Edge-IIoT dataset. The findings underscore that the variational autoencoders (VAEs), GB Machines, SVM Classifier, and Rule-Based IDS techniques yielded poor results. In the meantime, Autoencoder+LSTM, CNN-RNNs, and Bi-directional LSTM techniques achieved faster results. Additionally, the AICEDRL-FDR method provided a higher outcome, with maximum $accu_y$, $prec_n$, $recal_l$, and $F_{Measure}$ of 99.31%, 95.89%, 95.89%, and 95.89%, respectively.

Table 3. Comparative analysis of the AICEDRL-FDR technique on the Edge-IIoT dataset.

Edge-IIoT dataset				
Models	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$
VAE	91.61	91.69	91.61	91.60
GB machines	91.45	93.43	89.12	91.22
SVM classifier	91.61	91.68	91.61	91.61
Rule-based IDS	89.00	85.40	79.10	72.80
Autoencoder+LSTM	94.24	93.30	91.20	92.20
CNN-RNNs	96.00	93.21	94.95	92.89
Bi-directional LSTM	99.00	88.72	92.46	84.59
AICEDRL-FDR	99.31	95.89	95.89	95.89

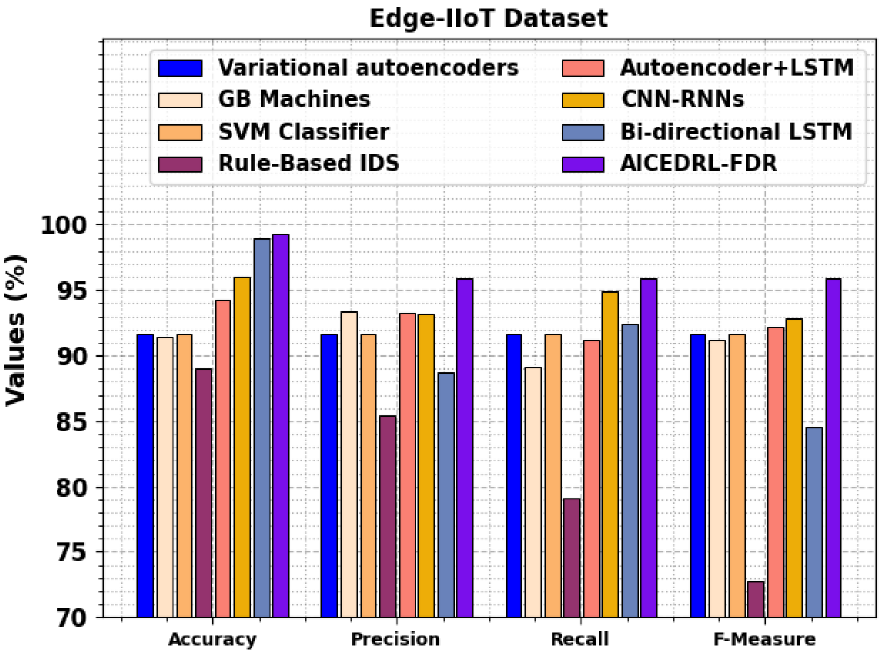

Figure 8. Comparative analysis of the AICEDRL-FDR method on the Edge-IIoT dataset.

Table 4 and Figure 9 present the computational time (CT) evaluation of the AICEDRL-FDR method relative to existing models under the Edge-IIoT dataset. These results indicate the average time each model takes to process and analyze the data from the Edge-IIoT dataset. The CT of the different methods was as follows: variational autoencoders (VAE), 4.28 seconds; GB Machines, 6.46 seconds; SVM Classifier, 8.47 seconds; Rule-Based IDS, 6.68 seconds; Autoencoder (AE) combined with LSTM, 5.21 seconds; CNN-RNNs, 8.54 seconds; and Bi-directional LSTM, 6.29 seconds. The AICEDRL-FDR model achieved the lowest CT at 2.24 seconds, illustrating its superior efficiency in handling cybersecurity tasks in cloud-edge-IoT environments.

Table 4. CT comparison of the AICEDRL-FDR method with existing models on the Edge-IIoT dataset.

Edge-IIoT dataset	
Models	CT (sec)
VAE	4.28
GB Machines	6.46
SVM Classifier	8.47
Rule-Based IDS	6.68
AE+LSTM	5.21
CNN-RNNs	8.54
Bi-directional LSTM	6.29
AICEDRL-FDR	2.24

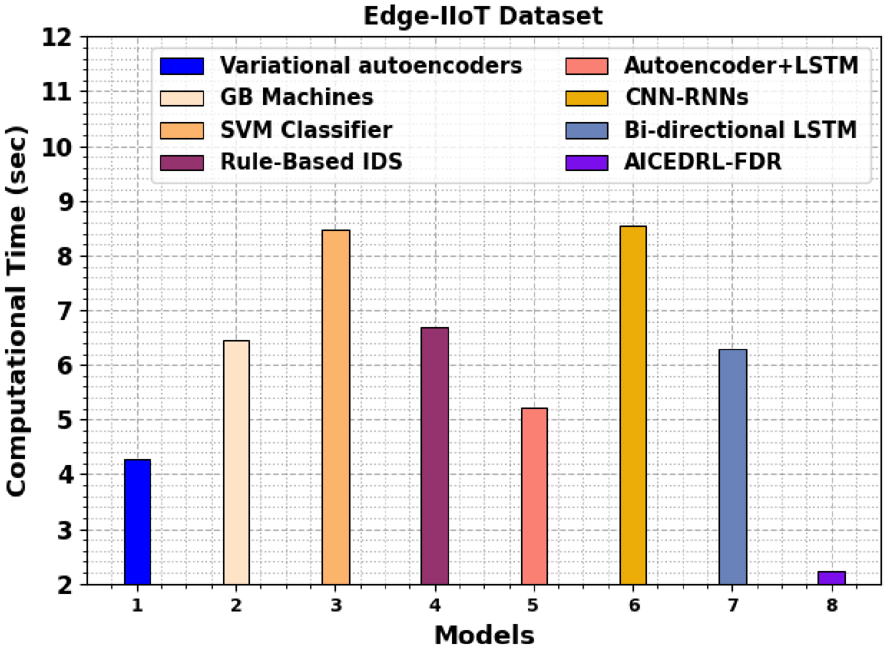

Figure 9. CT comparison of the AICEDRL-FDR method with existing models on the Edge-IIoT dataset.

Table 5 and Figure 10 present the error assessment of the AICEDRL-FDR method relative to existing techniques. The VAE model illustrated an $accu_y$ of 8.39%, $prec_n$ of 8.31%, $recal_l$ of 8.39%, and $F_{Measure}$ of 8.40%, highlighting consistent but moderate performance. GB Machines depicted an $accu_y$ of 8.55%, $prec_n$ of 6.57%, $recal_l$ of 10.88%, and $F_{Measure}$ of 8.78%, with slightly higher deviation in recall. SVM Classifier emulated VAE performance with an $accu_y$ of 8.39%, $prec_n$ of 8.32%, $recal_l$ of 8.39%, and $F_{Measure}$ of 8.39%. Rule-Based IDS presented the highest errors with an $accu_y$ of 11.00%, $prec_n$ of 14.60%, $recal_l$ of 20.90%, and $F_{Measure}$ of 27.20%, showing lower reliability. AE integrated with LSTM had moderate errors with an $accu_y$ of 5.76%, $prec_n$ of 6.70%, $recal_l$ of 8.80%, and $F_{Measure}$ of 7.80%. CNN-RNNs presented an accuracy of 4.00%.

Table 5. Error assessment of the AICEDRL-FDR methodology compared with existing techniques on the Edge-IIoT dataset.

Edge-IIoT dataset				
Methodology	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$
VAE	8.39	8.31	8.39	8.40
GB Machines	8.55	6.57	10.88	8.78
SVM Classifier	8.39	8.32	8.39	8.39
Rule-Based IDS	11.00	14.60	20.90	27.20
AE+LSTM	5.76	6.70	8.80	7.80
CNN-RNNs	4.00	6.79	5.05	7.11
Bi-directional LSTM	1.00	11.28	7.54	15.41
AICEDRL-FDR	0.69	4.11	4.11	4.11

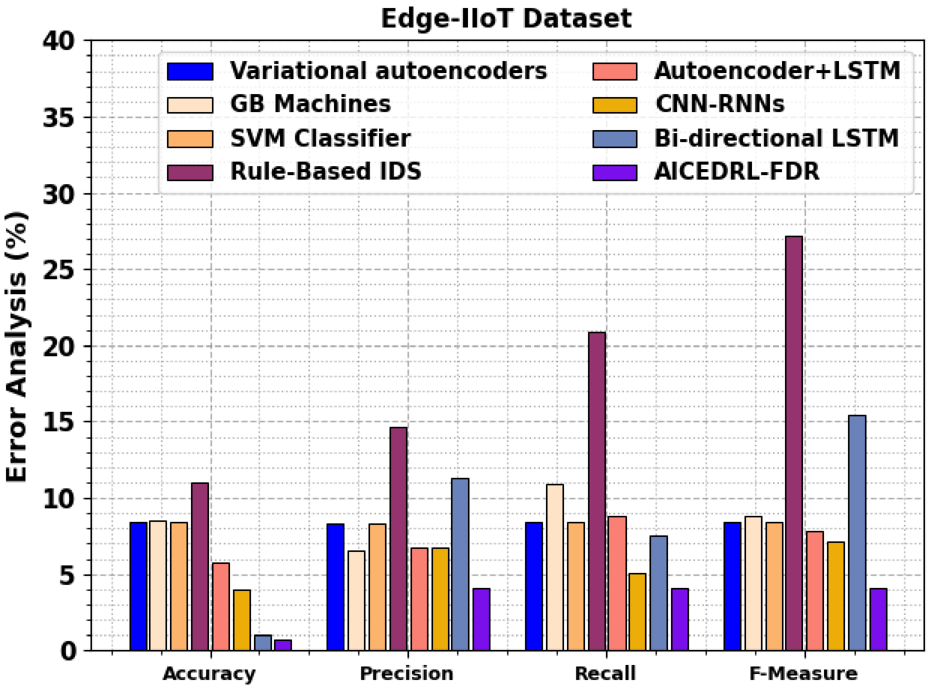

Figure 10. Error assessment of the AICEDRL-FDR methodology compared with existing techniques on the Edge-IIoT dataset.

4.2. Result analysis on the ToN-IoT dataset

The ToN-IoT dataset is a large-scale dataset intended for cybersecurity studies in IoT networks. It includes 119,957 instances, comprising both normal and malicious actions. The dataset consists of 78,369 standard instances and several threat types, such as MiTM (336 instances), DoS (5,440 instances), DDoS (5,987 instances), password attacks (6,016 instances), injection attacks (5,867 instances), XSS (5,951 instances), ransomware (5,976 instances), and backdoor attacks (6,015 instances). This dataset provides a comprehensive source for developing and evaluating intrusion detection methods and other security mechanisms for IoT settings. From the 42 features available, only 27 were selected.

Figure 11 presents the classifier outcomes of the AICEDRL-FDR method on the ToN-IoT dataset. Figure 11(a) illustrates the PR investigation, showing improved performance across each class. Figure 11(b) shows the ROC analysis, demonstrating efficient performance with higher ROC values across various classes.

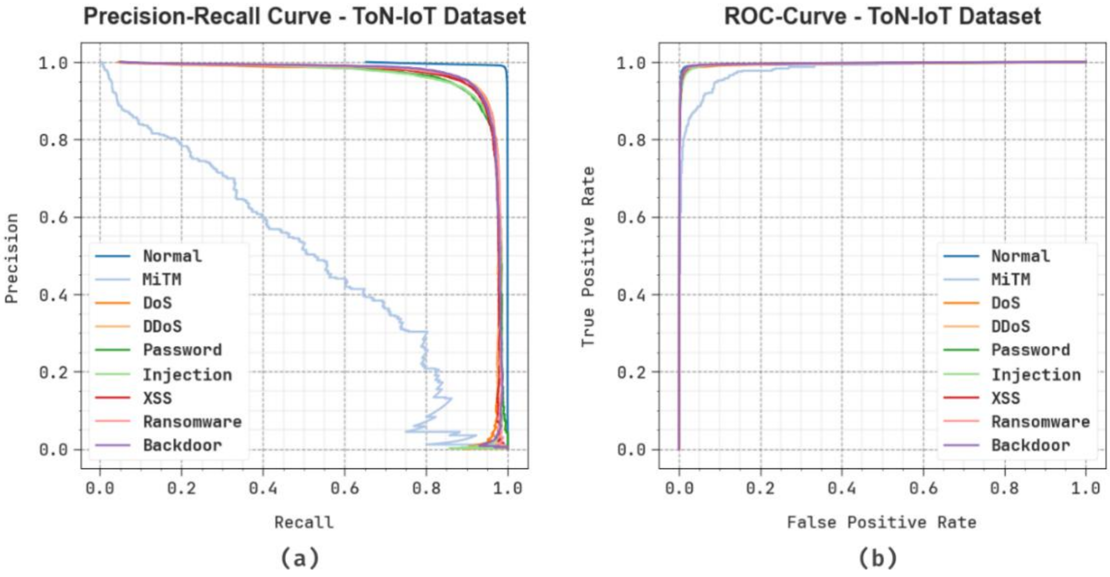

Figure 11. ToN-IoT dataset: (a) PR curve, (b) ROC curve.

Table 6 and Figure 12 portray the attack detection parameters of the AICEDRL-FDR approach on the ToN-IoT dataset. Under 70%TRPHE, the AICEDRL-FDR model achieves average $accur_y$, $preci_n$, $recal_l$,$F_{Measure}$, MCC, and Kappa of 99.18%, 91.97%, 85.31%, 87.16%, 87.25%, and 87.90%, respectively. Moreover, on 30%TSPHE, the suggested AICEDRL-FDR model achieves average $accur_y$, $preci_n$, $recal_l$, $F_{Measure}$, MCC, and Kappa of 99.24%, 91.31%, 87.22%, 88.74%, 88.43%, and 89.07%, respectively.

Table 6. Attack detection of the AICEDRL-FDR model on the ToN-IoT dataset.

Class labels	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$	MCC	Kappa
TRPHE (70%)						
Normal	98.29	98.27	99.12	98.70	96.23	96.88
MiTM	99.76	81.18	27.49	41.07	47.16	47.82
DoS	99.33	93.49	91.40	92.43	92.09	92.62
DDoS	99.27	92.80	92.65	92.73	92.34	93.10
Password	99.09	92.01	89.74	90.86	90.39	91.17
Injection	99.18	93.34	89.52	91.39	90.98	91.58
XSS	99.23	92.39	92.12	92.25	91.85	92.44
Ransomware	99.26	92.69	92.31	92.50	92.11	92.88
Backdoor	99.23	91.59	93.39	92.48	92.08	92.63
Average	99.18	91.97	85.31	87.16	87.25	87.90

Continued on next page

Class labels	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$	MCC	Kappa
TSPHE (30%)						
Normal	98.32	98.45	99.01	98.73	96.28	96.87
MiTM	99.82	72.34	40.00	51.52	53.72	54.25
DoS	99.37	93.13	93.24	93.19	92.85	93.57
DDoS	99.34	93.31	92.99	93.15	92.81	93.45
Password	99.15	92.56	90.14	91.34	90.90	91.57
Injection	99.23	93.71	90.31	91.97	91.59	92.34
XSS	99.24	92.47	91.95	92.21	91.81	92.48
Ransomware	99.36	93.96	93.28	93.62	93.29	93.96
Backdoor	99.31	91.85	94.11	92.96	92.61	93.15
Average	99.24	91.31	87.22	88.74	88.43	89.07

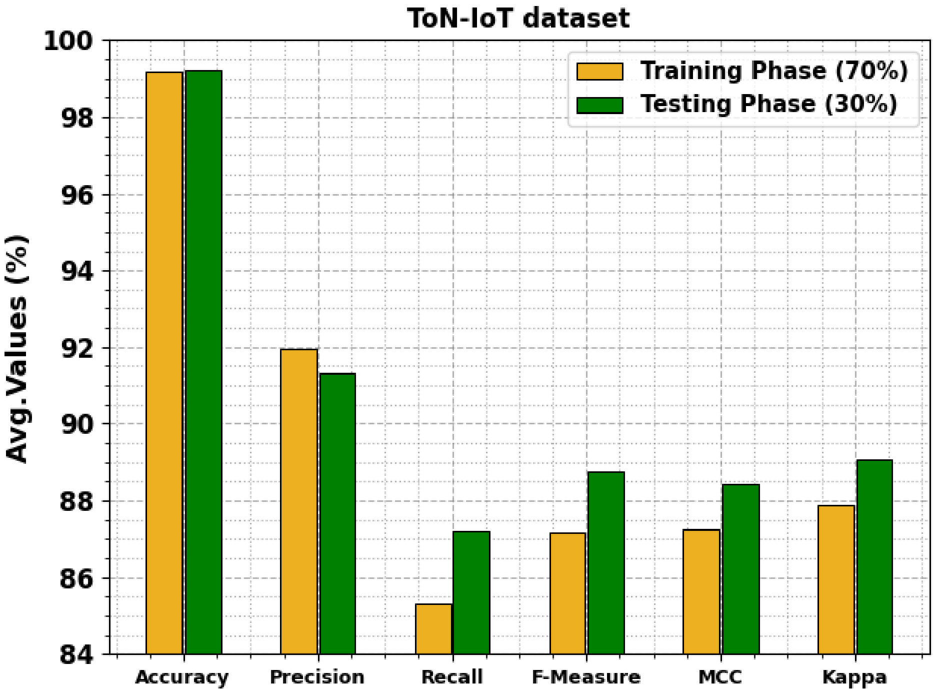

Figure 12. Average values of the AICEDRL-FDR model on the ToN-IoT dataset.

Figure 13 displays the TRAN and VALD $accur_y$ of the AICEDRL-FDR approach on the ToN-IoT dataset across 50 epochs. Both curves progressively rise and slowly converge, which illustrates that the method is successfully learning. The VALD $accur_y$ remains marginally better than the TRAN $accur_y$, suggesting that the technique is not overfitting and is simplifying well to the undetected data. The variations in $accur_y$ are assumed to be due to task complexity; nevertheless, the overall upward trend indicates strong performance and stability of the methodology.

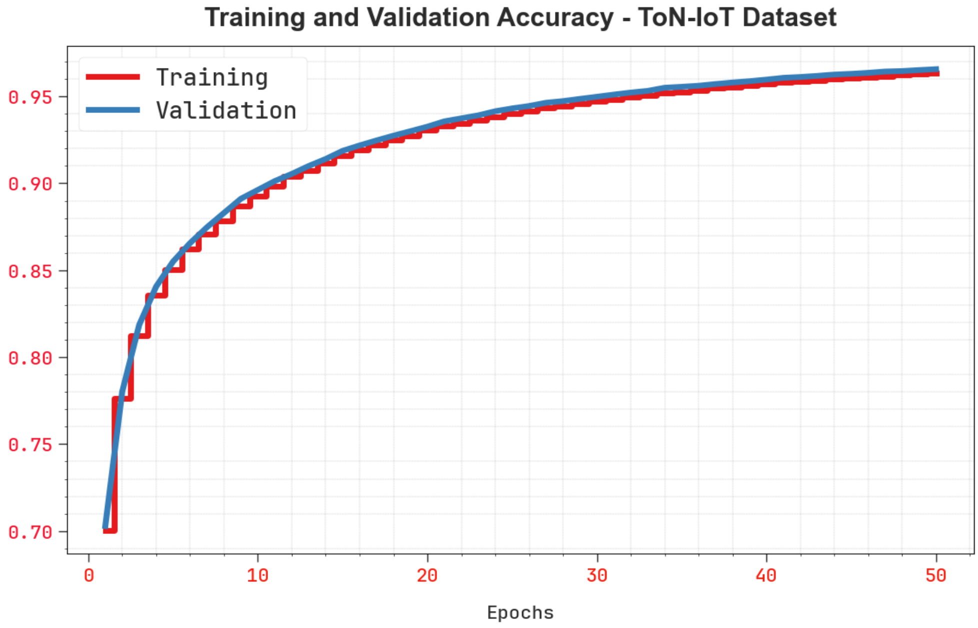

Figure 13. $Accur_y$ curve of the AICEDRL-FDR technique on the ToN-IoT dataset.

Figure 14 illustrates the TRAN and VALD loss of the AICEDRL-FDR method on the ToN-IoT dataset across 50 epochs. Both curves show a reliable downward trend, indicating that the technique is successfully reducing learning errors. The VALD loss remains worse than the training loss throughout most epochs, suggesting strong generalization and no signs of overfitting. Any variations that are detected are becoming progressively steady and reliable.

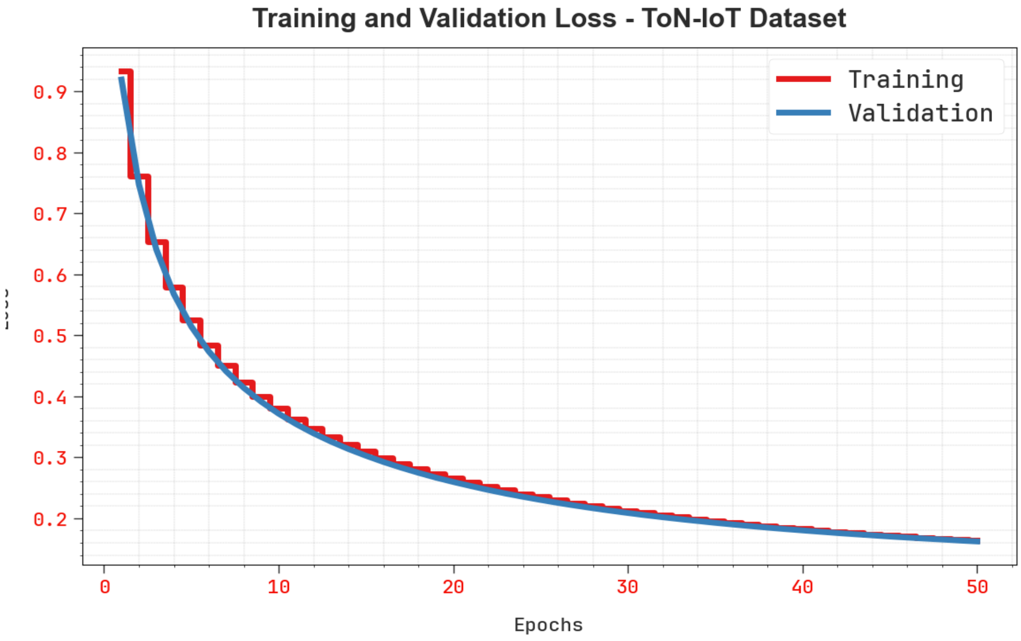

Figure 14. Loss curve of the AICEDRL-FDR technique on the ToN-IoT dataset.

Table 7 and Figure 15 illustrate the comparison between the AICEDRL-FDR technique on the ToN-IoT dataset with recent methodologies [37,38]. The empirical results show that the AICEDRL-FDR model has improved performance. According to $accur_y$, the AICEDRL-FDR model achieved the highest $accur_y$ at 99.24%. In contrast, the Feedforward NN, XGBoost, LR, Autoencoder Only, LSTM+CNN, Adversarial DBN-LSTM, and GA in SDN techniques achieved an $accur_y$ of 99.41%, 91.36%, 68.11%, 87.40%, 93.14%, 91.23%, and 80.00%, respectively.

Table 7. Comparative analysis of the AICEDRL-FDR technique on the ToN-IoT dataset.

ToN-IoT dataset				
Methods	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$
Feedforward NN	90.41	82.89	85.46	80.09
XGBoost	91.36	90.45	81.35	81.35
LR Model	68.11	68.13	68.11	68.10
Autoencoder Only	87.40	88.90	77.20	88.00
LSTM+CNN	93.14	90.20	81.90	83.00
Adversarial DBN-LSTM	91.23	83.75	85.21	83.75
GA in SDN	80.00	86.01	84.98	84.77
AICEDRL-FDR	99.24	91.31	87.22	88.74

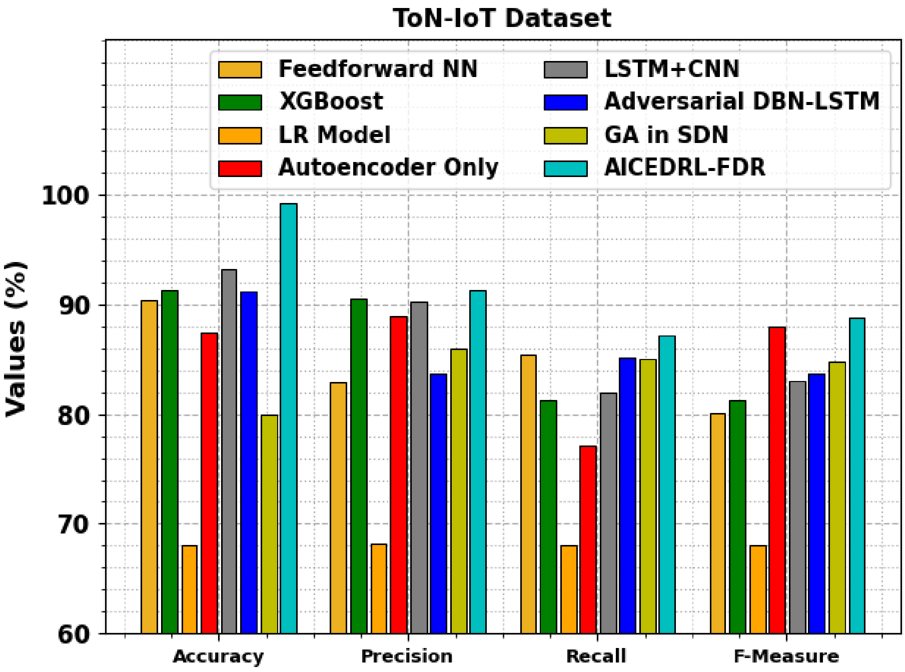

Figure 15. Comparative analysis of the AICEDRL-FDR technique on the ToN-IoT dataset.

Finally, the AICEDRL-FDR method achieved a greater $F_{Measure}$ of 88.74%. In contrast, the Feedforward NN, XGBoost, LR, Autoencoder Only, LSTM+CNN, Adversarial DBN-LSTM, and GA in SDN models have achieved $F_{Measure}$ values of 90.09%, 91.35%, 68.10%, 88.00%, 93.00%, 83.75%, and 84.77%, respectively. These results confirmed that the proposed model outperforms existing approaches.

Table 8 and Figure 16 present the computational time (CT) evaluation of the AICEDRL-FDR method relative to existing models. The Feedforward NN needed a CT of 6.34 seconds, XGBoost required 7.30 seconds, LR Model 7.33 required seconds, AE Only required 4.52 seconds, LSTM incorporated with CNN required 3.17 seconds, Adversarial DBN-LSTM required 3.81 seconds, GA in SDN required 5.75 seconds, and the AICEDRL-FDR model achieved the lowest CT of 1.08 seconds, highlighting its superior efficiency in handling cybersecurity tasks in cloud-edge-IoT environments.

Table 8. CT evaluation of the AICEDRL-FDR method compared with existing models on the ToN-IoT dataset.

ToN-IoT dataset	
Technique	CT (sec)
Feedforward NN	6.34
XGBoost	7.30
LR Model	7.33
AE Only	4.52
LSTM+CNN	3.17
Adversarial DBN-LSTM	3.81
GA in SDN	5.75
AICEDRL-FDR	1.08

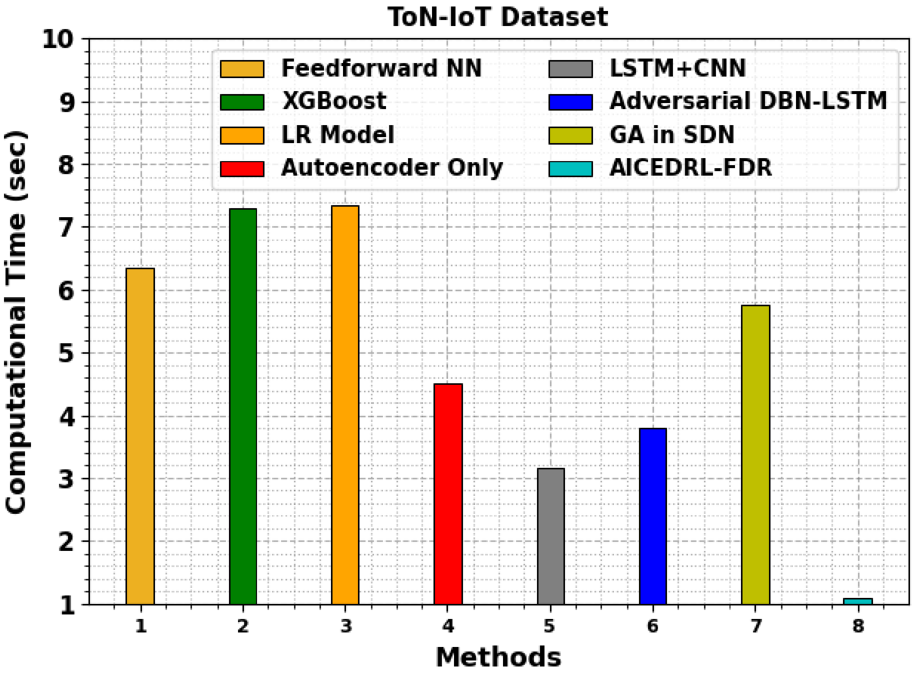

Figure 16. CT evaluation of the AICEDRL-FDR method compared with existing models on the ToN-IoT dataset.

Table 9 and Figure 17 present the error assessment of the AICEDRL-FDR methodology relative to existing techniques. The Feedforward NN illustrated moderate deviations with slightly higher error in recall, while XGBoost depicted more balanced performance across metrics. The LR Model had the highest deviations with an $accu_y$ of 31.89% and $F_{Measure}$ of 31.90%, indicating lower reliability. AE Only attained moderate deviations with an $F_{Measure}$ of 12.00%, whereas LSTM combined with CNN had relatively lower errors but modest overall performance. Adversarial DBN-LSTM depicted higher deviations in $prec_n$ at 16.25% and $recal_l$ at 14.79%. GA in SDN attained improved performance with $F_{Measure}$ of 15.23%. The AICEDRL-FDR model showed minimal overall deviation with an $accu_y$ of 0.76% and $F_{Measure}$ of 11.26%, demonstrating robust reliability and efficiency.

Table 9. Error assessment of the AICEDRL-FDR methodology with existing techniques on the ToN-IoT dataset.

ToN-IoT dataset				
Methodology	$Accur_y$	$Preci_n$	$Recal_l$	$F_{Measure}$
Feedforward NN	9.59	7.11	12.54	9.91
XGBoost	8.64	8.55	8.65	8.65
LR Model	31.89	31.87	31.89	31.90
Autoencoder Only	12.60	11.10	12.80	12.00
LSTM+CNN	6.86	5.80	8.10	7.00
Adversarial DBN-LSTM	8.77	16.25	14.79	16.25
GA in SDN	20.00	13.99	15.02	15.23
AICEDRL-FDR	0.76	8.69	12.78	11.26

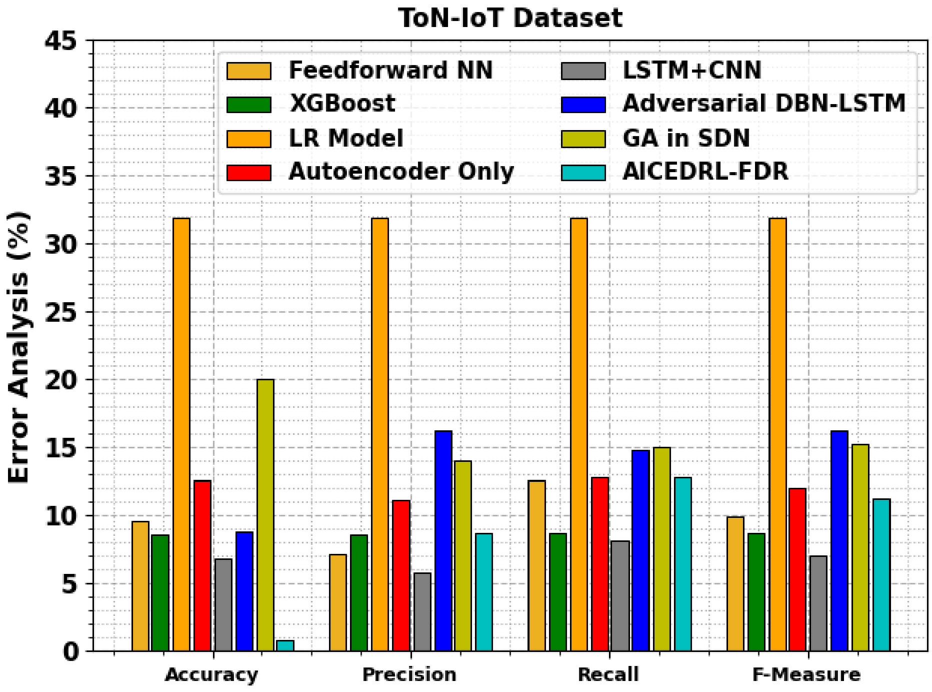

Figure 17. Error assessment of the AICEDRL-FDR methodology compared with existing techniques on the ToN-IoT dataset.

Table 10 presents the computational efficiency of the AICEDRL-FDR approach [39]. LeNet required 0.2860 GFLOPs, 3330 MB of GPU memory, and 4.68 seconds for inference, while CNN used 0.0968 GFLOPs, 4420 MB, and 4.90 seconds. ResNet and VGG required 456.7600 GFLOPs and 398.2900 GFLOPs, 3783 MB and 4553 MB of GPU memory, and inference times of 3.78 and 3.56 seconds, respectively. Moreover, NAS-Net achieved improved efficiency with 14.9763 GFLOPs, 3981 MB GPU usage, and 2.68 second of inference time. The AICEDRL-FDR model presented the highest efficiency with only 0.0176 GFLOPs, 589 MB GPU memory, and 1.00 seconds of inference time, making it highly appropriate for fast and resource-efficient cybersecurity deployment.

Table 10. Computational efficiency of the AICEDRL-FDR approach including FLOPs, GPU memory usage, and inference time.

Model	FLOPs (G)	GPU (M)	Inference time (sec)
LeNet	0.2860	3330	4.68
CNN	0.0968	4420	4.90
ResNet	456.7600	3783	3.78
VGG	398.2900	4553	3.56
NAS-Net	14.9763	3981	2.68
AICEDRL-FDR	0.0176	589	1.00

5. Conclusions

In this paper, the AICEDRL-FDR method is presented for cloud-edge IoT environments to provide a reliable framework for proactive threat mitigation in next-generation digital systems. First, the AICEDRL-FDR method uses pre-processing steps, including normalization, standardization, and cleaning, to improve dataset quality and consistency. For FS, the mRMR method is used to reduce irrelevant and redundant features. Additionally, ensemble deep representation methods such as DCAE, FDBN, and TCN are applied to the attack detection procedure. A broad array of experimental studies is conducted to ensure the enhanced performance of the AICEDRL-FDR technique on the Edge-IIoT [1] and ToN-IoT [2] datasets. A comparison analysis of the AICEDRL-FDR technique showed superior accuracy of 99.31% and 99.24% across diverse evaluation measures on both datasets. Some limitations include challenges posed by highly dynamic, real-world cyber threats that were not represented during training. The model's applicability may also be limited by a lack of detailed guidance on practical deployment in cloud-edge-IoT environments, as factors such as resource constraints, integration with legacy systems, and real-time monitoring are not fully addressed. Moreover, the model's scalability and ability to handle high-velocity, large-scale data streams remain unclear, which could affect its performance under heavy network traffic or in large, heterogeneous IoT networks. These gaps highlight the need for further analysis to assess robustness, optimize resource management, and ensure seamless real-world deployment. Future work should focus on optimizing computational overhead and mitigating latency for real-time attack detection. Incorporating adaptive learning mechanisms to handle zero-day attacks and continuously changing network behaviors is also recommended. The practical applicability and robustness may also be improved by computing the framework across diverse IoT devices and heterogeneous edge-cloud configurations.

Use of Generative-AI tools declaration

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

Acknowledgments

The author extends his appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number MoE-IF-UJ-R2-22-4100423-1.

Data availability statement

The data that support the findings of this study are openly available in the Kaggle repository at https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot and https://www.kaggle.com/datasets/dhoogla/cictoniot, reference numbers [1,2].

Conflict of interest

The author declares that he has no conflict of interest.

References

1. *Edge-IIoTset cyber security dataset of IoT & IioT*, 2021. Available from: https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot.
2. *CIC-ToN-IoT*, 2022. Available from: https://www.kaggle.com/datasets/dhoogla/cictoniot.
3. C. Gbaja, Next-generation edge computing: Leveraging Ai-driven Iot for autonomous, real-time decision making and cyber security, *Journal of Artificial Intelligence General Science*, **5** (2024), 357–371. https://doi.org/10.60087/jaigs.v5i1.204
4. S. K. Sundaramurthy, N. Ravichandran, A. C. Inaganti, R. Muppalaneni, AI-driven threat detection: leveraging machine learning for real-time cybersecurity in cloud environments, *Artificial Intelligence and Machine Learning Review*, **6** (2025), 23–43.
5. F. M. Rasel, B. Peter, AI-driven frameworks for enhancing cybersecurity in multi-cloud environments, *International Journal of Advanced Engineering Technologies and Innovations*, **1** (2025), 24–32. https://ijaeti.com/index.php/Journal/article/view/849
6. P. Prachi, AI-driven security mechanisms in IOT cloud solutions, *Smart Internet of Things*, **1** (2024), 260–264. https://doi.org/10.22105/siot.v1i4.137
7. S. Lad, Cybersecurity trends: Integrating ai to combat emerging threats in the cloud era, *Integrated Journal of Science and Technology*, **1** (2024), 12.
8. I. E. Kezron, Cybersecurity framework for securing cloud and AI-driven services in small and medium-sized businesses, *Journal of Tianjin University Science and Technology*, **58** (2025), 526–558.

9. R. Akbar, A. Zafer, Next-gen information security: AI-driven solutions for real-time cyber threat detection in cloud and network environments, *J. Cybersecur. Res.*, **12** (2024), 123–145. https://doi.org/10.13140/RG.2.2.11705.38245
10. M. Kumari, M. Gaikwad, S. A. Chavan, A secure IoT-edge architecture with data-driven AI techniques for early detection of cyber threats in healthcare, *Discov. Internet Things*, **5** (2025), 54. https://doi.org/10.1007/s43926-025-00147-z
11. M. M. Rashid, O. M. Yaseen, AI-driven cybersecurity measures for hybrid cloud environments: A framework for multi-cloud security management, *International Journal on Engineering Artificial Intelligence Management, Decision Support, and Policies*, **2** (2025), 30–39. https://doi.org/10.63503/j.ijaimd.2025.39
12. U. V. Menon, V. B. Kumaravelu, C. V. Kumar, A. Rammohan, S. Chinnadurai, R. Venkatesan, et al., AI-powered IoT: A survey on integrating artificial intelligence with IoT for enhanced security, efficiency, and smart applications, *IEEE Access*, **13** (2025), 50296–50339. https://doi.org/10.1109/ACCESS.2025.3551750
13. A. B. Dorothy, B. Madhavidevi, B. Nachiappan, G. Manikandan, P. K. Patjoshi, M. Sindhuja, AI-driven threat intelligence in cloud computing detecting and responding to cyber attacks, *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)*, Hassan, India, 2024, 1–6. https://doi.org/10.1109/IACIS61494.2024.10721888
14. G. D. Iyer, AI enhanced cybersecurity for cloud-IOT infrastructure in smart cities, *Metaversalize*, **2** (2025), 31–39. https://doi.org/10.22105/metaverse.v2i1.48
15. M. A. M. Farzaan, M. C. Ghanem, A. El-Hajjar, D. N. Ratnayake, AI-powered system for an efficient and effective cyber incidents detection and response in cloud environments, *IEEE Transactions on Machine Learning in Communications and Networking*, 3 (2025), 623–643. https://doi.org/10.1109/TMLCN.2025.3564912
16. K. A. Awan, I. U. Din, A. Almogren, A. Nawaz, M. Y. Khan, A. Altameem, SecEdge: A novel deep learning framework for real-time cybersecurity in mobile IoT environments, *Heliyon*, **11** (2025), e40874. https://doi.org/10.1016/j.heliyon.2024.e40874
17. F. Alblehai, Artificial intelligence-driven cybersecurity system for internet of things using self-attention deep learning and metaheuristic algorithms, *Sci. Rep.*, **15** (2025), 13215. https://doi.org/10.1038/s41598-025-98056-2
18. D. Chaudhary, S. K. Verma, V. M. Shrimal, R. Madala, R. Baliyan, M. Sathish, Ai-based methods to detect and counter cyber threats in cloud environments to strengthen cloud security, *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, Greater Noida, India, 2024, 1–6. https://doi.org/10.1109/ICEECT61758.2024.10739173
19. K. Sathupadi, S. Achar, S. V. Bhaskaran, N. Faruqui, M. Abdullah-Al-Wadud, J. Uddin, Edge-cloud synergy for AI-enhanced sensor network data: A real-time predictive maintenance framework, *Sensors*, **24** (2024), 7918. https://doi.org/10.3390/s24247918
20. F. M. Alrowais, S. Althahabi, S. S. Alotaibi, A. Mohamed, M. A. Hamza, R. Marzouk, Automated machine learning enabled cybersecurity threat detection in internet of things environment, *Comput. Syst. Sci. Eng.*, **45** (2023), 687–700. https://doi.org/10.32604/CSSE.2023.030188
21. G. Bhandari, A. Lyth, A. Shalaginov, T. M. Grønli, Distributed deep neural-network-based middleware for cyber-attacks detection in smart IoT ecosystem: A novel framework and performance evaluation approach, *Electronics*, **12** (2023), 298. https://doi.org/10.3390/electronics12020298

22. N. S. Suryavanshi, P. S. Acharya, A. N. Jadhav, A trust-security-resource optimization framework for cloud-edge-IoT collaboration in industrial applications, *IJSAT-International Journal on Science and Technology*, **16** (2025), 1626. https://doi.org/10.71097/IJSAT.v16.i1.1626
23. O. Aouedi, K. Piamrat, Toward a scalable and energy-efficient framework for industrial cloud-edge-IoT continuum, *IEEE Internet of Things Magazine*, **7** (2024), 14–20. https://doi.org/10.1109/IOTM.001.2300229
24. J. Ali, S. K. Singh, W. W. Jiang, A. M. Alenezi, M. Islam, Y. I. Daradkeh, et al., A deep dive into cybersecurity solutions for AI-driven IoT-enabled smart cities in advanced communication networks, *Comput. Commun.*, **229** (2025), 108000. https://doi.org/10.1016/j.comcom.2024.108000
25. G. Y. Xu, M. D. Xu, An effective prediction of resource using machine learning in edge environments for the smart healthcare industry, *J. Grid Computing*, **22** (2024), 54. https://doi.org/10.1007/s10723-024-09768-0
26. X. D. Zang, Z. L. Zheng, H. S. Zheng, X. Liu, M. K. Khan, W. W. Jiang, HyperEye: A lightweight features fusion model for unknown encrypted malware traffic detection, *IEEE T. Consum. Electr.*, **71** (2025), 5079–5089. https://doi.org/10.1109/TCE.2025.3558353
27. N. J. Singh, N. Hoque, K. R. Singh, D. K. Bhattacharyya, Botnet-based IoT network traffic analysis using deep learning, *Secur. Privacy*, **7** (2024), e355. https://doi.org/10.1002/spy2.355
28. K. Saravanan, R. Santhosh, TBBS-TO: Trustable blockchain and bandwidth sensible-based task offloading and resource allocation in cloud-IoT network, *Journal of Cybersecurity and Information Management*, **15** (2025), 133–150. https://doi.org/10.54216/JCIM.150111
29. M. Sahi, N. Auluck, A. Azim, M. A. Maruf, Dynamic hierarchical intrusion detection task offloading in IoT edge networks, *Software: Practice and Experience*, **54** (2024), 2249–2271. https://doi.org/10.1002/spe.3338
30. C. Banse, B. Fanta, J. Alonso, C. Martinez, EMERALD: Evidence management for continuous certification as a service in the cloud, 2025, arXiv:2502.07330. https://doi.org/10.48550/arXiv.2502.07330
31. A. V. Nagarjun, S. Rajkumar, Design of an anomaly detection framework for delay and privacy-aware blockchain-based cloud deployments, *IEEE Access*, **12** (2024), 84843–84862. https://doi.org/10.1109/ACCESS.2024.3414998
32. G. Logeswari, J. D. Roselind, K. Tamilarasi, V. Nivethitha, A comprehensive approach to intrusion detection in IoT environments using hybrid feature selection and multi-stage classification techniques, *IEEE Access*, **13** (2025), 24970–24987. https://doi.org/10.1109/ACCESS.2025.3532895
33. N. Papaioannou, G. Myllis, A. Tsimpiris, V. Vrana, The role of mutual information estimator choice in feature selection: An empirical study on mRMR, *Information*, **16** (2025), 724. https://doi.org/10.3390/info16090724
34. A. Ponraj, P. Nagaraj, D. Balakrishnan, P. N. Srinivasu, J. Shafi, W. Kim, et al., A multi-patch-based deep learning model with VGG19 for breast cancer classifications in the pathology images, *Digit. Health*, **11** (2025). 20552076241313161. https://doi.org/10.1177/20552076241313161
35. A. K. Shukla, P. K. Muhuri, Deep belief network with fuzzy parameters and its membership function sensitivity analysis, *Neurocomputing*, **614** (2025), 128716. https://doi.org/10.1016/j.neucom.2024.128716

36. J. Y. Bai, W. Zhu, S. H. Liu, C. H. Ye, P. Zheng, X. C. Wang, A temporal convolutional network-bidirectional long short-term memory (TCN-BiLSTM) prediction model for temporal faults in industrial equipment, *Appl. Sci.*, **15** (2025), 1702. https://doi.org/10.3390/app15041702
37. M. J. C. S. Reis, AI-driven anomaly detection for securing IoT devices in 5G-enabled smart cities, *Electronics*, **14** (2025), 2492. https://doi.org/10.3390/electronics14122492
38. A. H. Salem, S. M. Azzam, O. E. Emam, A. A. Abohany, Advancing cybersecurity: A comprehensive review of AI-driven detection techniques, *J. Big Data*, **11** (2024), 105. https://doi.org/10.1186/s40537-024-00957-y
39. R. Lyu, M. S. He, Y. Zhang, L. Jin, X. L. Wang, Network intrusion detection based on an efficient neural architecture search, Symmetry, 13 (2021), 1453. https://doi.org/10.3390/sym13081453