

Article

Piranha Foraging Optimization Algorithm with Deep Learning Enabled Fault Detection in Blockchain-Assisted Sustainable IoT Environment

Haitham Assiri

Department of Computer Science, College of Engineering and Computer Sciences, Jazan University,
Jazan 45142, Saudi Arabia; habusallam@jazanu.edu.sa

Abstract: As the acceptance of Internet of Things (IoT) systems quickens, guaranteeing their sustainability and reliability poses an important challenge. Faults in IoT systems can result in resource inefficiency, high energy consumption, reduced security, and operational downtime, obstructing sustainability goals. Thus, blockchain (BC) technology, known for its decentralized and distributed characteristics, can offer significant solutions in IoT networks. BC technology provides several benefits, such as traceability, immutability, confidentiality, tamper proofing, data integrity, and privacy, without utilizing a third party. Recently, several consensus algorithms, including ripple, proof of stake (PoS), proof of work (PoW), and practical Byzantine fault tolerance (PBFT), have been developed to enhance BC efficiency. Combining fault detection algorithms and BC technology can result in a more reliable and secure IoT environment. Thus, this study presents a sustainable BC-Driven Edge Verification with a Consensus Approach-enabled Optimal Deep Learning (BCEVCA-ODL) approach for fault recognition in sustainable IoT environments. The proposed BCEVCA-ODL technique incorporates the merits of the BC, IoT, and DL techniques to enhance IoT networks' security, trustworthiness, and efficacy. IoT devices have a substantial level of decentralized decision-making capacity in BC technology to achieve a consensus on the accomplishment of intrablock transactions. A stacked sparse autoencoder (SSAE) model is employed to detect faults in IoT networks. Lastly, the Piranha Foraging Optimization Algorithm (PFOA) approach is used for optimum hyperparameter tuning of the SSAE approach, which assists in enhancing the fault recognition rate. A wide range of simulations was accomplished to highlight the efficacy of the BCEVCA-ODL technique. The BCEVCA-ODL technique achieved a superior FDA value of 100% at a fault probability of 0.00, outperforming the other evaluated methods. The proposed work highlights the significance of embedding sustainability into IoT systems, underlining how advanced fault detection can provide environmental and operational benefits. The experimental outcomes pave the way for greener IoT technologies that support global sustainability initiatives.

Keywords: blockchain; sustainability; Internet of Things; deep learning; hyperparameter tuning; consensus algorithm



Academic Editor: Yang (Jack) Lu

Received: 6 November 2024

Revised: 6 January 2025

Accepted: 9 January 2025

Published: 7 February 2025

Citation: Assiri, H. Piranha Foraging Optimization Algorithm with Deep Learning Enabled Fault Detection in Blockchain-Assisted Sustainable IoT Environment. *Sustainability* **2025**, *17*, 1362. <https://doi.org/10.3390/su17041362>

Copyright: © 2025 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fast-developing technologies that depend on the IoT are achieving many considerations due to advancements in wireless communications and sensor technologies [1]. The effect of this growth can be comprehended in different domains, like medicine, diverse business sectors, industries, advanced smart sectors, and so on. The advanced development of IoT technologies provides transformative potential to enhance quality of life and

operational efficiency over various domains. But the sustainability of these systems is frequently challenged by operational faults, which impact the security of IoT systems. Security is the main issue in these domains, as in them, several heterogeneous devices are interconnected [2]. Recently, several IoT attacks have threatened its recognition. Presently, the most common attacks are denial of service (DoS), spyware, malware, zero-day, and phishing attacks, among others. These attacks can be external or internal, passive or active. At a minimum, all layers of the IoT endure at least one of these kinds of attacks [3]. DoS attacks that avoid access to the network affect each layer of the IoT. However, privacy and security can be two significant problems for IoT nodes, while transactions with edge computing nodes persist because of trust connections. BC technology is an intrinsic security technology designed to offer confidentiality, privacy, security, and trust for IoT networks [4]. This technology makes decisions to resolve numerous issues present in standard dispersed database systems. The hack-proof category of BC technology gives security characteristics for data formed by IoT devices [5].

The latest advanced technology, BCs, which are unreliable and immutable public ledgers, are a correct solution for handling this trust absence difficulty [6]. The key management system and distributed consensus method in BCs support exchanging resources and facilities and execute transactions whose administration is cryptographically confirmable. Transactions are protected because they are irreversible and trackable. Recently, BC technology has rapidly emerged, particularly in IoT platforms [7]. A popular BC app site, Ethereum, employs a closely Turing-complete language in its BC that implies that some program codes developed by users can be performed under BCs in an autonomous and automated way [8]. Autonomy has been attained through smart contracts permitted by BCs. Embedded contract clauses are automatically performed while a specific state is fulfilled. Established as an Ethereum fork, Monax Industry offers an adaptive “BC as a service,” permitting users to describe arbitrary procedures for their protocols by creating smart contracts, which could be implemented with a private BC network. Combining BC with IoT networks confirms authentic data establishment [9]. Executing the BC method allows services to function in a decentralized way, but there is no requirement for third parties to be engaged as intermediates. The BC network does not involve numerous non-colluding parties, making it more reliable for IoT networks [10].

This study presents a novel BC-Driven Edge Verification with a Consensus Approach-enabled Optimal Deep Learning (BCEVCA-ODL) approach for IoT fault recognition. This study presents a fault detection mechanism specifically developed for sustainable IoT environments with a focus on sustainability. The proposed model exploits a cutting-edge deep learning approach for analyzing real-time data streams, enabling fault detection and fault management. With the prioritization of fault detection, the system supports global sustainability goals. The proposed BCEVCA-ODL technique incorporates the merits of BC, IoT, and DL techniques to enhance IoT networks’ security, trustworthiness, and efficacy. IoT devices have a substantial level of decentralized decision-making capacity in BC technology to achieve a consensus on the accomplishment of intrablock transactions. A stacked sparse autoencoder (SSAE) model was employed to detect faults in IoT networks. Lastly, the Piranha Foraging Optimization Algorithm (PFOA) approach was used for the optimum hyperparameter tuning of the SSAE approach, which assisted in enhancing the fault recognition rate. A wide range of simulations was accomplished to highlight the efficacy of the BCEVCA-ODL technique. In short, the key contributions of this study are listed as follows.

- The BCEVCA-ODL approach incorporates BC technology, IoTs, and DL models to improve IoT networks’ security, trustworthiness, and performance. Utilizing BC’s decentralized consensus, IoT’s extensive connectivity, and DL’s advanced analytics

presents a robust solution for enhanced network management. This incorporation confirms more reliable and secure IoT operations, addressing key threats in modern network environments.

- The BCEVCA-ODL model utilizes BC technology to ease consensus among IoT devices for intrablock transactions through decentralized decision making. This methodology improves the reliability and safety of the model by ensuring transparent and tamper-proof transaction verification. By incorporating BC, the method effectually addresses vulnerabilities and enhances the comprehensive robustness of the IoT network.
- The BCEVCA-ODL technique employs the SSAE method to improve fault detection accuracy in IoT networks, utilizing advanced deep learning (DL) models for greater diagnostic capabilities. This methodology enhances the detection of anomalies and system faults by employing the SSAE model's ability to learn and represent intrinsic data patterns. Hence, it presents more precise and reliable fault detection than conventional techniques.
- The BCEVCA-ODL approach employs the PFOA technique to fine-tune the hyperparameters of the SSAE method, crucially enhancing fault recognition rates and the model's overall performance. By optimizing hyperparameters with PFOA, the technique attains more precise and effective fault detection. This custom-made optimization improves the SSAE model's efficiency in diagnosing network issues.
- The BCEVCA-ODL methodology uniquely combines BC, IoT, and DL to optimize fault detection and improve the model's reliability. This novel incorporation gives a synergistic solution that advances IoT network management and safety. The novelty is in integrating these technologies to create a robust framework for enhanced network performance and fault mitigation.

2. Related Works

Kumar et al. [11] developed a BC-orchestrated DL named BDSDT. Primarily, an innovative, scalable BC method was introduced. Secondarily, BDSDT incorporates off-chain storage in the InterPlanetary File System (IPFS). The authenticated data were also employed to develop a DL method. The final integrates Deep Sparse-AE (DSAE) with Bidirectional Long Short-Term Memory (BiLSTM). In [12], an edge-terminal collaborative extraction task processing model was created. The architecture forms a delay-and-throughput-enabled BC technology and employs the asynchronous advantage actor-critic (A3C) method for cooperatively enhancing block size and interval configuration, transmission power distribution, and offloading solution. Kumar et al. [13] developed a BC and DL-based secure data processing technique. This model employs the smart contract-assisted PBFT consensus method. An intrusion detection system (IDS) is dependent upon a fusion approach of Sparse AE-based Attention Bidirectional GRU (SAE-ABI-GRU), which was implemented by inspecting the connection load behavior of the MEC-assisted Road-Side Unit (RSU) server. In [14], a BC-based intelligent edge cooperation system (BIECS) was developed. By leveraging BC technology, this analysis makes trust between edge nodes and utilizes an incentive method for resource exchange between multi-structure providers. The architecture initially develops an enhanced Long Short-Term Memory (LSTM) method and, next, chooses edge nodes to perform offloaded processes and manage the respective BC method relevant to all task implementation. Alqaralleh et al. [15] implemented DL using a BC-related secure image communication and detection framework. Firstly, ECC was exploited, and the optimal key generation of ECC occurred via a combination of the grasshopper with fruit fly optimizer (GO-FFO) technique. Secondly, the neighborhood indexing sequence (NIS) using burrow wheeler transform (BWT), named NIS-BWT, was enforced. Lastly, a deep belief network (DBN) was employed. Zhang et al. [16] projected

a BC and AI-enabled secure cloud-edge-end collaboration PloT (BASE-PloT) approach. The adaptability of three common BCs with PloT could be examined, and a few everyday application scenes of BASE-PloT were exhibited. Therefore, a BC-empowered federated deep-A3C task offloading architecture was developed. The short-period queuing delay optimizer was decoupled by utilizing the Lyapunov optimizer and coupling among the long-range security limitations.

In [17], a model that incorporates a BC with deep RNN (DRNN) and edge computing was presented. The original RF signals of drones with numerous flight modes were remotely identified and then gathered at the cloud server to train a DRNN method. BC exploited this developed model to protect data transmission and data integrity. The DRNN method could be assessed on an open database named DroneRF. Sharadq et al. [18] introduced a HybridChain-IDS model, which utilizes BC-based intrusion detection with NIK-512 hashing, Cheetah Optimizer Algorithm (COA) scheduling, ResCapsNet, and k-nearest neighbor (k-NN) for attack prediction. However, scalability, latency, and the complexity of advanced techniques like DSAE with BiLSTM pose challenges. Hu et al. [19] reviewed point machine fault and anomaly detection methods, analyzed their practical application challenges, and proposed research directions for developing an intelligent fault detection system. Alserhani [20] proposed a trust management system for IoT devices using RSA encryption, Self-Adaptive Tasmanian Devil Optimization (SA_TDO) for key generation, SHA3-512 for data integrity, and IDS with deep neural network (DNN) optimized by Archimedes Optimization Algorithm (AOA) for threat detection, enhancing IoT network security. Alyoubi et al. [21] presented the Capuchin Search Algorithm with a DL-based Data Edge Verification model (CSADL-DEVM) technique, with CSA, DL, and hyperparameter tuning for data edge verification and fault detection in BC-assisted IoT platforms using an Elman Recurrent Neural Network (ERNN). Kokila and Reddy [22] introduced BlockDLO, an IoT security approach integrating BC and DL, featuring network localization, page rank-based clustering, shared-chain and deep distributed file systems, Ethereum smart contracts, route optimization, and DL for malicious data detection. Sekhar and Aruna [23] evaluated the performance and security of algorithms in IoT edge computing environments, focusing on preserving device privacy and improving effectiveness in large-scale network systems. Alamro et al. [24] introduced the BHS-ALOHD technique, which integrates BC, Ant Lion Optimizer (ALO) for feature selection, and Hybrid DL models such as CNN-LSTM for intrusion detection, with the Flower Pollination Algorithm (FPA) used for optimal hyperparameter tuning.

Kulandaivelu, Rajappan, and Murugasamy [25] proposed BGJOA-DLSMTD, a secure medical data transmission and diagnosis approach using a Golden Jackal Optimization Algorithm (GJOA) for encryption, BC for data storage, and DL models such as Bayesian Optimization, DBN, and CapsNet for image classification and feature extraction. Perumal et al. [26] presented an Enhanced Metaheuristics with DL Model for BC-Assisted Cybersecurity Solution (EMDLM-BCCS), with data preprocessing, attack detection via Extreme Learning Machine (ELM), and optimization through Elite-Oppositional Grasshopper Optimization Algorithm (EGOA) models. Khan et al. [27] proposed a BC-based approach with DL techniques and a real-time message content validation scheme using decentralized BC to address single points of failure and ensure data integrity. Swamynathan et al. [28] proposed a BC-based solution for VANETs, integrating DL (FA_ECCN) to predict driver behavior and Binary Fire Hawks Optimization (BFHO) for optimized routing with trust values. It utilizes IPFS for BC storage and DPBFT for validation, improving traffic security and routing efficiency. Katib and Ragab [29] presented the H3SC-DLIDS approach, integrating Harris Hawks Optimization (HHO) and sine cosine algorithms (SCAs) for feature selection using the LSTM autoencoder for intrusion detection, aimed at identifying DDoS

attacks in a BC-supported IoT environment. Zuo [30] surveyed the integration of AI and BC, exploring how BC improves AI functionality, how AI assists BC, and their mutual benefits in securing and enhancing various domains comprising IoT. Mohammed et al. [31] presented the “Pattern-Proof Malware Validation” (PoPMV) methodology for BC in ICPS, utilizing LSTM and reinforcement learning to improve security, processing speed, and attack detection while optimizing ICPS functionality. Kumar et al. [32] proposed a secure IoT-enabled Smart Grid network incorporating Digital Twin, SDN, DL, and BC to enhance attack detection, communication security, and real-time services in the Smart Grid environment. Ranjan and Kumar [33] proposed a hybrid encryption approach for secure medical data transmission, utilizing IoT sensors, BC, and DL techniques such as LSTM and convolutional neural network (CNN) with optimized key generation (SI-LA) to ensure data privacy, security, and efficiency.

The existing studies on BC-assisted IoT and healthcare systems often encounter scalability, latency, and complexity challenges, particularly in integrating DL models with BC for data security and fault detection. Additionally, while various optimization techniques like genetic algorithms, DBNs, and RNNs have been proposed, these methods need more effectual integration for real-time intrusion detection and data integrity validation in dynamic IoT environments. Moreover, integrating advanced encryption techniques comprising hybrid models faces issues with computational overhead and resource constraints in resource-limited IoT devices. Lastly, the current models often need to pay more attention to the interoperability of BC systems across diverse IoT platforms and the efficiency of hybrid approaches in large-scale deployments. A key research gap is the need for techniques that balance security, real-time intrusion detection, and minimal computational overhead in BC-assisted IoT and healthcare systems, particularly in large-scale, resource-constrained environments.

3. The Proposed Model

This study introduces a novel BCEVCA-ODL approach for IoT fault detection in a sustainable IoT environment. The proposed BCEVCA-ODL model incorporates the merits of BC, IoT, and DL methods to improve the IoT network’s security, trustworthiness, and effectualness. Figure 1 represents the flow of the BCEVCA-ODL approach. The proposed model offers ongoing discourse in sustainability by designing a scalable and effective model to detect faults in sustainable IoT networks, highlighting the alignment of technological advancements with environmental stewardship. It underlines the significance of embedding sustainable practices into IoT development to ensure a harmonious balance between innovation and ecological preservation.

3.1. IoT Network and Consensus Algorithm

As a distribution technology, BC is secure and transparent, works irrespective of a central control body, and mainly deals with data storage and management [34]. The choice to incorporate BC technology, IoTs, and DL methods is justified due to their complementary merits in addressing IoT network threats. BC’s decentralized consensus mechanism improves security and trustworthiness by averting tampering and confirming transparent transaction verification. IoT’s extensive connectivity eases real-time data collection and device interaction, which is significant for effectual network management. DL techniques provide advanced analytics, enabling precise fault detection and predictive maintenance through intrinsic pattern detection. This integration employs the merits of every technology to create a robust, secure, and efficient solution for managing and safeguarding modern IoT networks. Figure 2 illustrates the structure of BC technology.

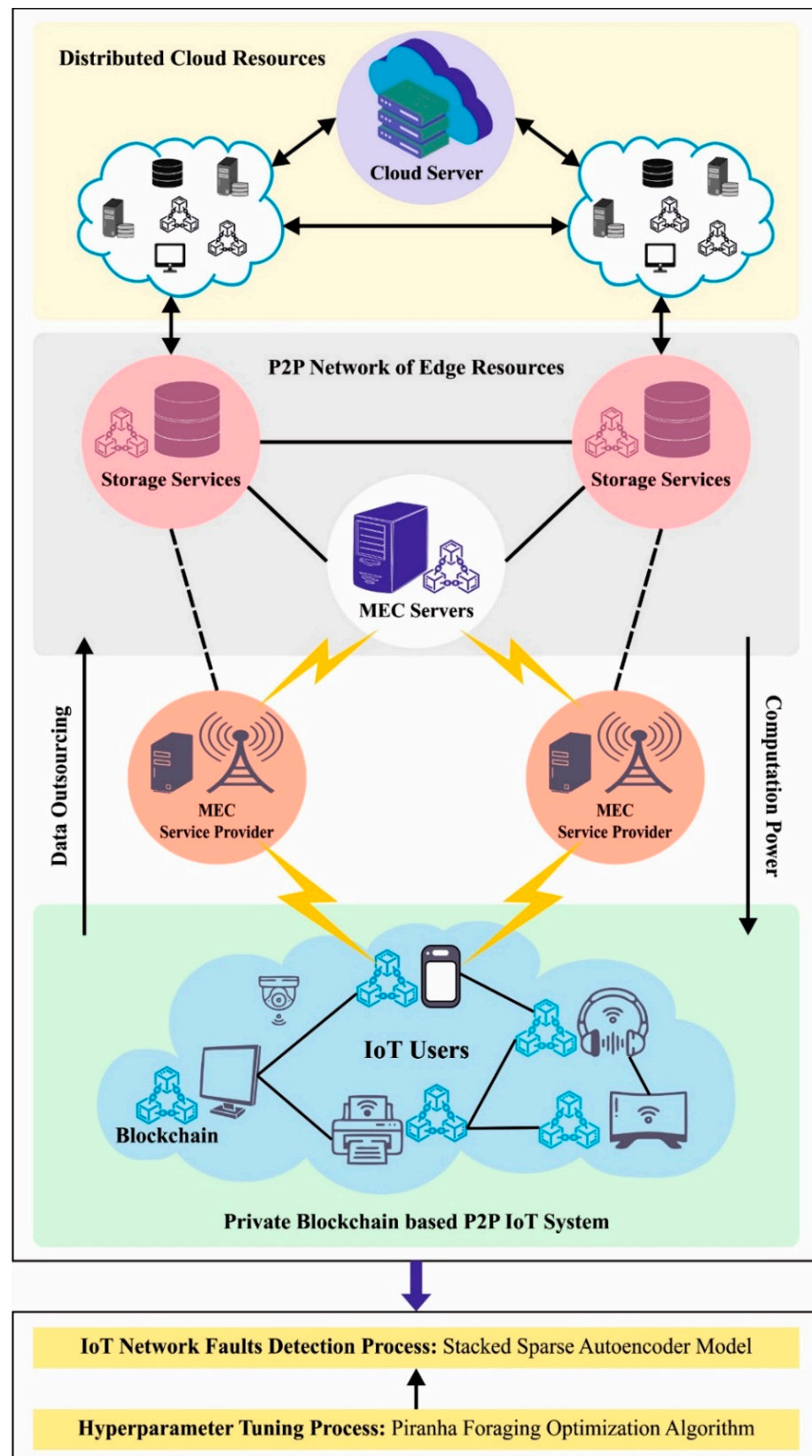


Figure 1. The overall workflow of the BCEVCA-ODL approach for IoT fault detection combines BC, IoTs, and DL techniques to enhance IoT networks' security, trustworthiness, and effectiveness. The figure depicts the key stages of the model, including data collection, fault detection, and the implementation of security measures for reliable IoT performance.

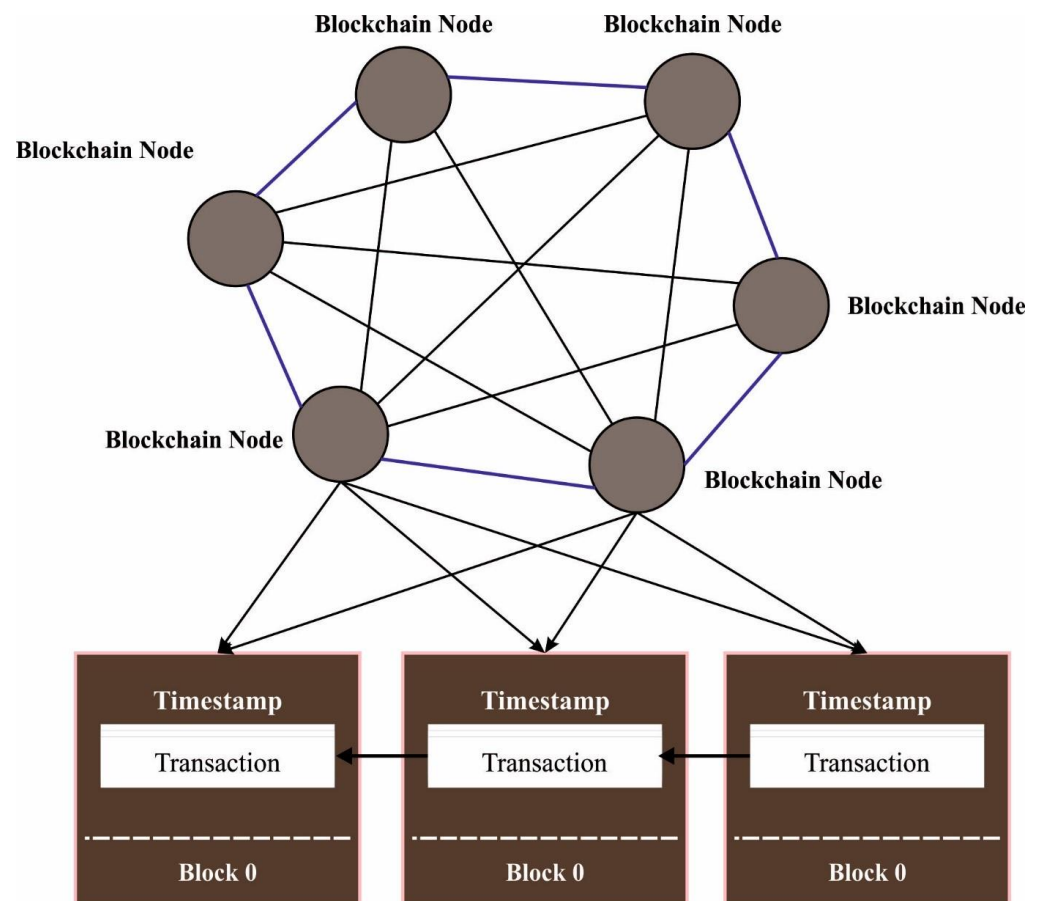


Figure 2. The architecture of BC technology illustrates the components involved in the decentralized system. The figure depicts multiple BC nodes responsible for processing and validating transactions. Each node maintains a timestamp and records transactions later grouped into blocks. Block 0 represents the initial block in the BC, with subsequent blocks linked to it to form a secure, immutable chain of transactions. This architecture ensures transparency, security, and reliability within the BC network.

Unlike classical techniques, BC enables peer-to-peer transmission of digital resources without requiring an intermediary. The consensus algorithm has transitioned from the centralized system, where the central system or administrator can invalidate or validate transactions, including database management and banking systems. The administrator is the invalid or valid manager in this kind of system. In decentralized systems like BC, the absence of an administrator requires an additional protocol for validation and verification. The intermediary function is moved to the periphery participating pair in the chain infrastructure. It is a decentralized system since the peers do not know each other. The consensus algorithm establishes the process to confirm, validate, and verify transactions. Next, the transaction is recorded in an extensive distributed directory, which creates block records. Lastly, a consensus protocol is implemented. Therefore, verification, immutable, validation, and consensus recording lead to the security and trust of the BC.

Like PoS, the DPoS is a decentralized BC consensus mechanism that optimizes energy usage for efficient mining. In DPoS, stakeholders elect delegates to mine new blocks and validate transactions, with witnesses chosen by currency holders to perform tasks such as validating transactions and finding the block nonce. Delegates set key parameters like block size, mining power, and witness incentives, while stakeholders retain decision-making authority. The DPoS implementation allows stakeholders to vote for delegates based on their holdings, ensuring transparency and security. Delegates selected for their performance

are incentivized to complete tasks efficiently, increasing their re-election chances. This adaptive selection process provides a secure, efficient, decentralized network, promoting fairness and accountability within the BC ecosystem.

3.2. SSAE-Based Detection Model

The SSAE model is employed to detect faults in the IoT network [35]. The SSAE method is particularly appropriate for fault detection in IoT networks because it can learn complex patterns and anomalies in high-dimensional data. The architecture of the SSAE model, which stacks various autoencoders, allows it to effectually extract hierarchical features and reconstruct data, making it efficient in detecting deviations from normal behavior. Its sparse representation assists in focusing on critical features while ignoring noise, which is significant for detecting subtle faults. Moreover, the unsupervised learning capability of the SSAE model enables it to detect previously unseen anomalies without needing labelled data, making it highly adaptable to the dynamic behavior of IoT environments. This model improves fault detection accuracy and reliability, which is significant for maintaining optimal network performance. Many SAEs create SSAE neural networks via end-to-end connections. The preceding layer of the sparse self-encoder result is exploited as an input at the subsequent level to gain a complex feature symbol of input data. Figure 3 exhibits the architecture of SSAE.

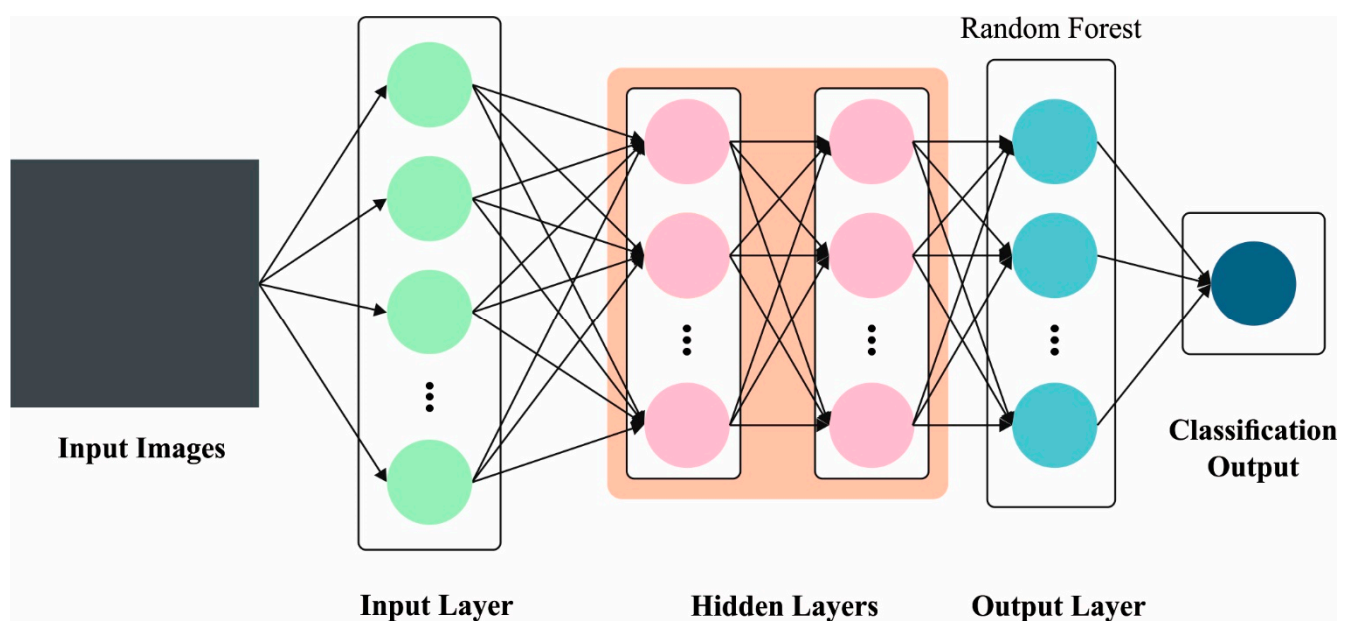


Figure 3. The SSAE approach’s structure illustrates the model’s key components. The process begins with input images sent to the system for analysis and processed by an RF model for feature extraction. The processed data are then passed through the autoencoder’s input layer, followed by multiple hidden layers that perform additional feature learning and transformation. Finally, the output layer gives the classification output, depicting the model’s final decision based on the learned features. This structure enables effective feature learning and classification.

The greedy layer-wise pretraining approach is mainly applied to train all SSAE layers to acquire access to the SSAE model’s link weight and bias value. Then, the error backpropagation technique is employed to perfect SSAE until the consequences of error functions among input and output data fulfil a predictable need in order to obtain the finest parameter method.

The error function $J_{sparse}(W, b)$:

$$\begin{aligned} & \frac{\partial}{\partial w_{ij}^r} J_{sparse}(W, b) \\ &= \frac{1}{2n_r} \sum_{r=1}^{n_r} \frac{\partial}{\partial w_{ij}^r} J_{sparse}(W, b, X(n), Y(n)) + \lambda w_{ij}^r \end{aligned} \quad (1)$$

$$\begin{aligned} & \frac{\partial}{\partial b^r} J_{sparse}(W, b) \\ &= \frac{1}{2n_r} \sum_{r=1}^{n_r} \frac{\partial}{\partial b^r} J_{sparse}(W, b, X(n), Y(n)) \end{aligned} \quad (2)$$

In Equation (1), the gradient of the sparse cost function J_{sparse} concerning the weight w_{ij}^r is computed. The term $\frac{1}{2n_r} \sum_{r=1}^{n_r} \frac{\partial}{\partial w_{ij}^r} J_{sparse}(W, b, X(n), Y(n))$ depicts the average gradient over n_r samples. The second term, w_{ij}^r , is a regularization term that penalizes large weights, helping to enforce sparsity and avoid overfitting by introducing a penalty proportional to the weight size, where λ is the regularization coefficient. Likewise, in Equation (2), the gradient of the sparse cost function J_{sparse} concerning the bias b^r is computed. The first term $\frac{1}{2n_r} \sum_{r=1}^{n_r} \frac{\partial}{\partial b^r} J_{sparse}(W, b, X(n), Y(n))$ illustrates the average gradient over n_r samples. This term helps adjust the model's biases during training to minimize the cost function and improve the model's accuracy. Therefore, the upgrading procedure of bias and weight is mentioned below:

$$w_{ij}^k = w_{ij}^k - \eta \frac{\partial}{\partial w_{ij}^k} J(W, b) \quad (3)$$

$$b^r = b^r - \eta \frac{\partial}{\partial b^r} J(W, b) \quad (4)$$

In Equations (3) and (4), the weight w_{ij}^k and bias b^r are updated using gradient descent. The weight update rule (Equation (3)) adjusts w_{ij}^k by subtracting the learning rate η times the gradient of the cost function $J(W, b)$ concerning w_{ij}^k . In contrast, the bias update rule (Equation (4)) similarly adjusts b^r by subtracting η times the gradient of the cost function relating to b^r . These updates iteratively minimize the cost function, improving the model's performance.

In the SSAE model, the training data consist of high-dimensional input vectors that are given into multiple model layers for unsupervised pretraining. The input data are preprocessed to normalize and scale the features, ensuring uniformity across the data. The model configuration comprises a deep architecture with several stacked autoencoder layers, each trained using greedy layer-wise pretraining, followed by fine-tuning using error backpropagation. The dimensional characteristics of the input data, such as the number of features and samples, play a significant role in determining the optimal weight and bias updates during the training process. The learning rate η is adapted for diverse parameters and scarce features to prevent overfitting and ensure effectual convergence.

3.3. Hyperparameter Tuning

Finally, the PFOA approach is utilized to optimize the hyperparameter tuning of the SSAE model [36]. The PFOA model stands out for hyperparameter tuning due to its biologically inspired approach, replicating piranhas' foraging behavior to explore the search space effectively. Unlike conventional techniques such as grid or random search, the PFOA model balances exploration and exploitation, enhancing the likelihood of finding optimal hyperparameters. Its swarm-based nature allows for parallel search and adaptation, making it effectual for complex, high-dimensional spaces. Furthermore, PFOA's dynamic adjustment of search strategies improves convergence speed and solution quality, giving a

robust alternative to other optimization methods. This adaptability and efficiency make it a compelling choice for tuning hyperparameters in diverse machine learning models. The input to the PFOA model for hyperparameter tuning comprises the set of hyperparameters to be optimized, the performance metrics of the approach being tuned, and the search space constraints. The PFOA model employs these inputs to explore and exploit potential hyperparameter settings, aiming to improve the model's performance by optimizing these parameters based on the defined evaluation criteria. Figure 4 depicts the steps involved in the PFOA model.

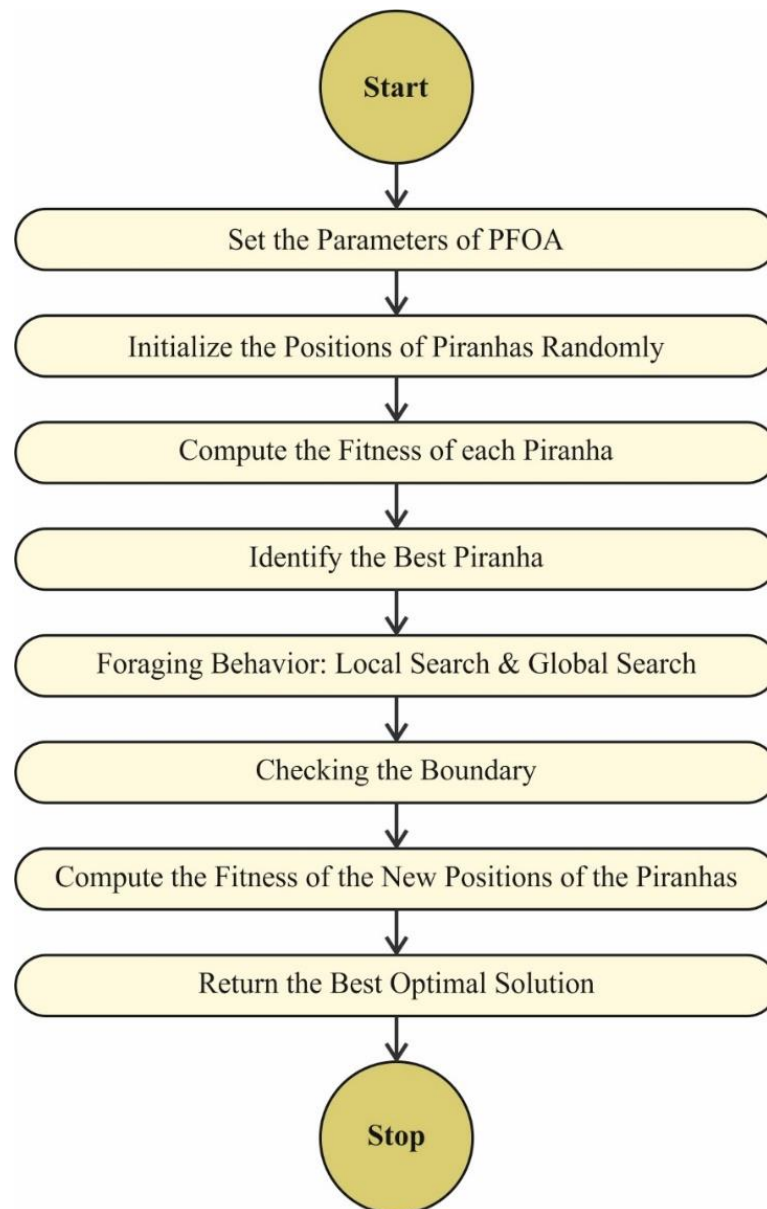


Figure 4. The steps involved in the PFOA model. The process begins with the initialization phase, where the parameters of the PFOA are set, and the positions of the piranhas are arbitrarily initialized. Next, the fitness of each piranha is computed to analyze their performance. The best-performing piranha is then identified. The model proceeds with the foraging behavior, comprising local and global search strategies to explore potential solutions. Boundary conditions are checked to ensure the positions remain valid. Afterwards, the fitness of the new positions of the piranhas is computed. The process continues iteratively, aiming to find the optimum optimal outcome. The algorithm stops once the optimal solution is detected and returned.

The scavenging foraging, localized group, and bloodthirsty cluster attacks are the three different patterns of the PFOA approach, and they mainly comprise initialization, evaluation, and parameter and agent location updating:

$$x = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1D} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2D} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{(n-1)1} & x_{(n-1)2} & x_{(n-1)3} & \cdots & x_{(n-1)D} \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{nD} \end{bmatrix}$$

The position vector $x_i = [x_{i1}x_{i2} \cdots x_{iD}]$ represents an individual piranha.

STEP 1: Initialization

The location vector of every individual was initialized based on the following equation:

$$x_i = lb_i + \beta_1 \times (ub_i - lb_i) \quad (5)$$

In Equation (5), the position of individuals in the candidate solution of a piranha is x_i . The search agent's upper and lower limitations are ub_i and lb_i , respectively. A random number within $[0, 1]$ is β_1 .

STEP 2: Describe the F predation intensity parameter

Based on F_i , the F predation intensity parameter, and d_i , the distance between the prey and piranhas, a piranha is very sensitive to blood detection. The range of blood smell agrees with the inverse square law. Piranhas swim toward the area of higher blood smell, and the greater the blood concentration, the quicker their movement will be, yielding Equation (6).

$$\begin{aligned} F_i &= \beta_2 \times \frac{Z_i}{4\pi d_i^2} \\ d_i &= x_{prey} - x_i \\ Z_i &= [x_i(t) - x_{i+1}(t)]^2 \end{aligned} \quad (6)$$

Here, the blood smell for the position of i^{th} individuals is F_i , Z can represent the source intensity, Z_i indicates the source concentration observed by the i^{th} search agents that will be changed in real time, d_i symbolizes the distance between the location of i^{th} individuals and the prey (optimum solution), and a random integer within $[0, 1]$ is β_2 .

STEP 3: Nonlinear parametric control strategies

This stage effectively controls the time-variant randomized process and avoids early population convergence while ensuring a silky and smooth transformation between exploitation and exploration. In the initial and middle phases of PFOA, a significant value of S helps the searching agent implement auditory-based global search and prevent trapping in local optima; however, PFOA converges fast as S changes in the later stage.

$$S = C \cdot \cos \left[\frac{\pi}{2} \otimes \left(\frac{t}{\text{Max_iter}} \right) \right]^4 \quad (7)$$

In Equation (7), the maximum iteration number is Max_iter , and the constant C is set to 5 and $\otimes$ represents the product of the value and parameter.

STEP 4: Reverse escape search strategy

This stage uses fag E for changing the direction of the population search, which prevents getting stuck in the local region and moving the search toward a new area to improve the solution. This repeatedly occurs in the search procedure and provides the search agent with better possibilities to rigorously and carefully explore the search region, as follows:

$$E = \begin{cases} 1 & \beta_3 \leq 0.5 \\ -1 & \beta_3 > 0.5 \end{cases} \quad (8)$$

In Equation (8), a random integer within $[0, 1]$ is represented by β_3 .

STEP 5: Update the formula of the proxy position

When hungry, the individual attacks prey, and each time prey strays into their territory, a splash of water is formed, and the intensely hearing piranha collects these signals and assembles quickly to encircle and attach the prey, with the agent closer toward the target being the first to attack, as follows:

$$x_i(t+1) = \gamma_1 \sum_{k=1}^{pc} \frac{L_k(t) - x_i(t)}{pc} - x_{prey}(t) \quad (9)$$

In Equation (9), the updated location of the search agent is $x_i(t+1)$, pc denotes a random number within $[6, \text{Search Agents_no}/2]$, and Search Agents_no symbolizes the overall amount of agents. The fraction of local population attack that makes the first attack is $L_k(t)$, where $L \in X$ and X represent the number of random piranhas. The existing agent's location is $x_i(t)$; the place of the optimum agent in the prior iteration indicates $x_{prey}(t)$, a uniformly distributed random integer within $[-2, 2]$ is γ_1 .

Piranha has a strong awareness and unique taste for blood concentration. Once the prey is injured and bleeding, it attracts distant piranhas toward the region with high blood absorption for an aggressive attack. The higher the blood smell, the quicker it swims. This phase dramatically depends on the blood smell F_i , the distance d_i between its prey and the piranhas, and the nonlinear cosine factor S . A change in the direction of piranha movement enables it to search for better prey locations and prevent local optimum.

$$x_i(t+1) = \gamma_1 \cdot e^{V^2} \cdot x_{prey}(t) + G \cdot x_{prey}(t) \cdot E \cdot F_i + E \cdot \beta_4 \cdot S \cdot F_i \quad (10)$$

In Equation (10), the updated location of the search agent is $x_i(t+1)$. The uniformly distributed random numbers γ_1 and γ_2 are within the ranges $[2, 2]$ and $[-1/2, 1/2]$, respectively, influencing the agent's global and local search behavior. β_4 is a random integer within the range. The coefficient G represents the foraging capability of the piranha, controlling how effectively it can locate prey. The term $\gamma_1 \cdot e^{V^2}$ introduces a uniform dispersion, allowing for dynamic tradeoffs and adjustments between global exploration (larger search space) and local exploitation (focused search area). The position of the prey at time t , denoted by $x_{prey}(t)$, influences the agent's movement. E is a parameter controlling the movement direction, while F_i represents a factor related to the i^{th} agent's fitness or position. Finally, S is a scaling factor affecting the movement's magnitude during the foraging process.

While foraging, piranhas randomly escape from their habitat in turbid watersheds due to their poor vision during the night and swim in their territory as follows.

$$x_{i(r+1)} = \frac{1}{2} [e^{\gamma_2} * x_{C_1}(t) - E * x_i(t)] \quad (11)$$

In Equation (11), the updated location of the search agent is $x_i(t+1)$. The uniformly distributed random number γ_2 is within the range $[-1, 1]$, affecting the randomness of the agent's movement. E is a parameter that changes the movement direction, making the agent's movement more or less random based on its current position. $x_{C_1}(t)$ depicts the location of a randomly chosen agent, denoted as C1C_1C1, selected from the piranha group. The agent C_1 is different from the i^{th} agent, ensuring the diversity of interactions (i.e., $C_1 \neq i$). The current position of the i^{th} agent at time t is $x_i(t)$, and the term e^{γ_2} provides

a scaling effect to adjust the movement based on the randomness of γ_2 . This equation models how piranhas adjust their position while foraging, considering their poor vision and tendency to escape their habitat in turbid waters.

$$SR(i) = \frac{fitness_{Max} - fitness(i)}{fitness_{Max} - fitness_{Min}} \quad (12)$$

$$x_i(t+1) = x_{prey}(t) + \frac{1}{2} \{ [x_{C_1}(t) - E \cdot x_{C_2}(t)] - [x_{C_2}(t) - E \cdot x_{C_3}(t)] \} \quad (13)$$

In Equation (12), the agent's survival rate (SR) is evaluated to maintain population diversity, where SR is computed by comparing the fitness of the current agent to the maximum and minimum fitness values. If SR is less than or equal to 1/4, indicating a lower SR, the offspring population is regenerated to ensure the diversity and vitality of the population. The fitness values used in the calculation are normalized between the maximum and minimum fitness values, enhancing the stability of the population. In Equation (13), the updated position of the search agent, $x_i(t+1)$, is determined by the location of the optimum agent from the previous iteration, $x_{prey}(t)$, along with a directional change parameter E . Parameter E influences the search behavior by modifying the movement direction, allowing the agent to explore new regions. The positions of agents $x_{C_1}(t)$, $x_{C_2}(t)$, and $x_{C_3}(t)$ are chosen randomly from the piranha population, ensuring that $C_1 \neq C_2 \neq C_3$. These agents contribute to the exploration process by updating the position based on their relative positions, fostering both global and local search capabilities.

The selection of fitness is a substantial factor affecting the efficiency of the PFOA method. The hyperparameter selection methodology comprises the solution encoding procedure for assessing the efficiency of candidate solutions. The PFOA approach considers accuracy a primary condition for developing the fitness function.

$$Fitness = \max(P) \quad (14)$$

$$P = \frac{TP}{TP + FP} \quad (15)$$

where TP and FP are the true- and false-positive values.

4. Performance Validation

The performance analysis of the BCEVCA-ODL method is discussed briefly in terms of diverse measures [37]. The suggested technique is simulated using the Python 3.6.5 tool on a PC with an i5-8600k, 250 GB SSD, GeForce 1050Ti 4 GB, 16 GB RAM, and 1 TB HDD. The parameter settings are provided: learning rate: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and batch size: 5.

The brief precision rate (PR) results of the BCEVCA-ODL method are shown in Table 1 and Figure 5. The outputs show that the PSO-DAWRF and DAWRF models obtain the lowest PR values compared to other models. Although the CSADL-DEVM and BDEV-CAML models attain slightly boosted PR values, the BCEVCA-ODL technique shows maximum PR values under all classes. On CPUHog Class, the BCEVCA-ODL technique obtained an increased PR value of 99.96%, while the CSADL-DEVM, BDEV-CAML, PSO-DAWRF, and DAWRF models obtained decreased PR values of 99.76%, 99.56%, 98.65%, and 97.10%, respectively.

Table 1. Performance comparison of the (PR, %) across diverse classes. The table portrays the PR outcomes for various methodologies evaluated on the classes CPUHog, MemoryOF, Scanning, IOHog, and DOS, demonstrating the performance of each method in terms of fault detection precision across these categories.

Class	PR (%)				
	BCEVCA-ODL	CSADL-DEVM	BDEV-CAML	PSO-DAWRF	DAWRF
CPUHog	99.96	99.76	99.56	98.65	97.10
MemoryOF	99.74	99.53	99.23	98.11	96.99
Scanning	99.92	99.72	99.61	98.40	97.21
IOHog	99.93	99.79	99.64	98.77	97.58
DOS	99.69	99.53	99.46	97.94	96.42

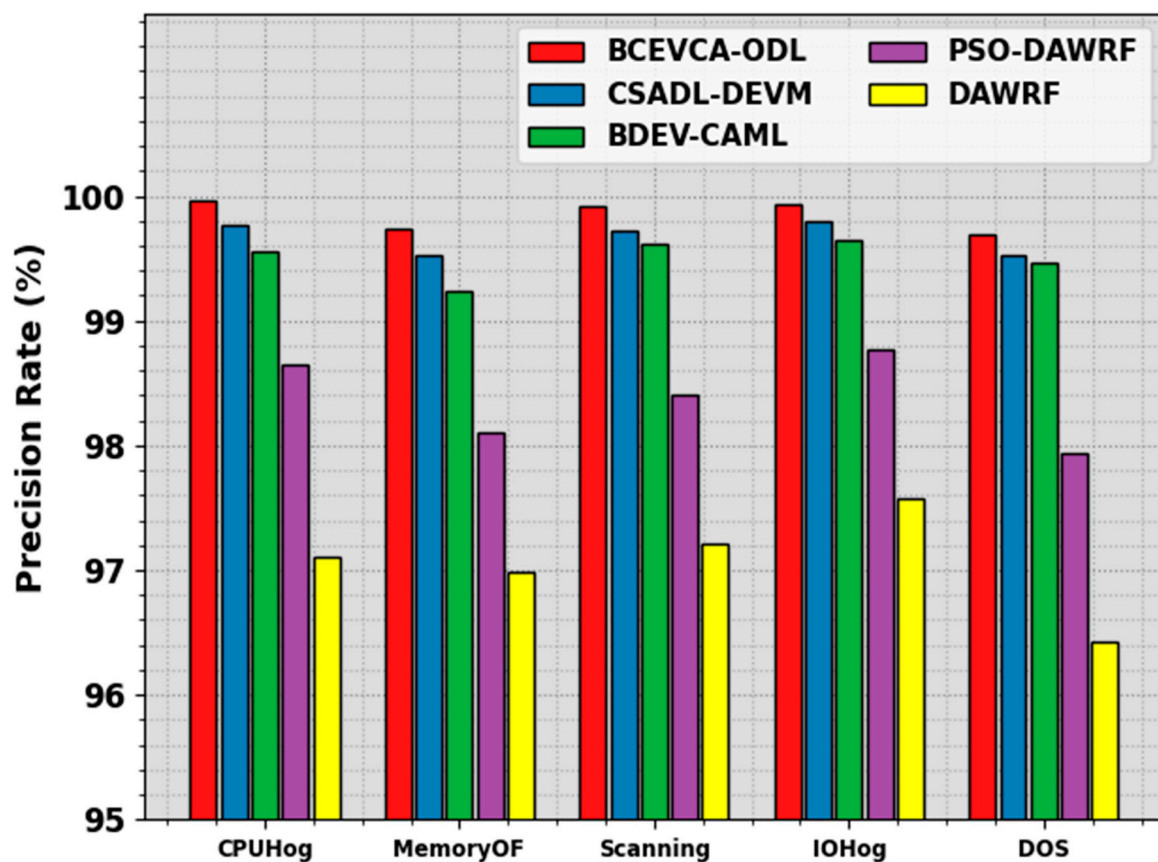


Figure 5. (PR, %) outcomes of the BCEVCA-ODL technique across diverse classes. The table compares the PR results for the BCEVCA-ODL technique against other methods in detecting faults across various classes. The values presented represent the performance in terms of precision, illustrating how the BCEVCA-ODL technique performs relative to alternative approaches for each class.

A brief recall rate (RR) outcome of the BCEVCA-ODL technique is shown in Table 2 and Figure 6. The results illustrate that the PSO-DAWRF and DAWRF approaches obtain the lowest RR values over other models. Although the CSADL-DEVM and BDEV-CAML models attain slightly boosted RR values, the BCEVCA-ODL method shows maximum values of RR under all classes. On CPUHog Class, the BCEVCA-ODL technique obtained an increased PR value of 99.96%, while the CSADL-DEVM, BDEV-CAML, PSO-DAWRF, and DAWRF techniques attained minimized RR values of 99.67%, 99.49%, 98.50%, and 97.59%, respectively.

Table 2. (RR, %) outcomes of the BCEVCA-ODL approach compared to other methods across various classes. The figure presents the RR for the BCEVCA-ODL approach and different methods across five classes: CPUHog, MemoryOF, Scanning, IOHog, and DOS. This comparison highlights the efficiency of the BCEVCA-ODL model in achieving high RRs in fault detection across these diverse categories.

Class	RR (%)				
	BCEVCA-ODL	CSADL-DEVM	BDEV-CAML	PSO-DAWRF	DAWRF
CPUHog	99.96	99.67	99.49	98.50	97.59
MemoryOF	99.61	99.43	99.25	97.92	96.27
Scanning	99.89	99.68	99.57	98.45	97.44
IOHog	99.96	99.78	99.64	98.09	96.90
DOS	99.81	99.61	99.40	97.81	96.71

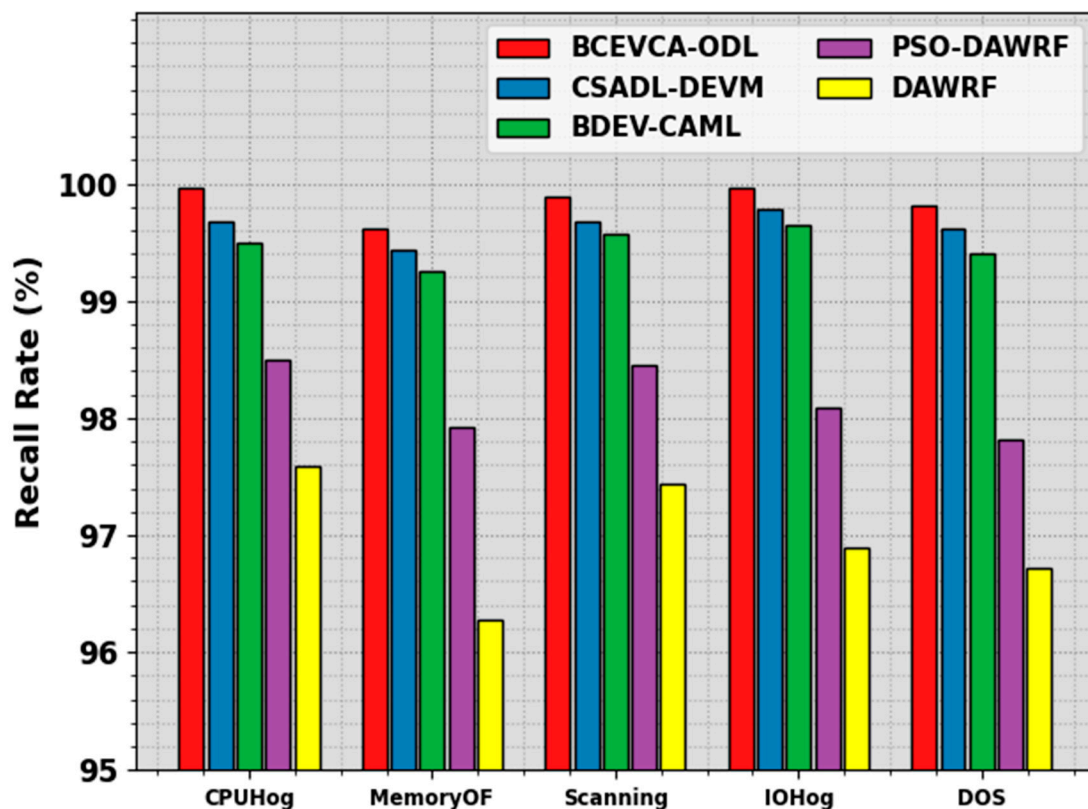


Figure 6. (RR, %) outcomes of the BCEVCA-ODL technique across various classes. The figure compares the RR results of the BCEVCA-ODL technique with other methods, across five classes: CPUHog, MemoryOF, Scanning, IOHog, and DOS. The data illustrate the recall performance of each method, illustrating how the BCEVCA-ODL method performs relative to alternative approaches in detecting faults across these diverse categories.

The brief accuracy rate (AR) results of the BCEVCA-ODL approach are shown in Table 3 and Figure 7. The outputs indicated that the PSO-DAWRF and DAWRF approaches achieve the lowest AR values over other models. Although the CSADL-DEVM and BDEV-CAML models attain slightly boosted AR values, the BCEVCA-ODL technique shows maximum values of AR under all classes. On CPUHog Class, the BCEVCA-ODL method obtained an increased AR value of 99.91%, while the CSADL-DEVM, BDEV-CAML, PSO-DAWRF, and DAWRF approaches obtained decreased AR values of 99.72%, 99.63%, 98.45%, and 96.79%, respectively.

Table 3. (AR, %) outcomes of the BCEVCA-ODL methodology compared to other methods across various classes. The figure presents the AR results of the BCEVCA-ODL methodology along with other methods computed on five classes: CPUHog, MemoryOF, Scanning, IOHog, and DOS. This comparison highlights each method's accuracy performance, demonstrating the BCEVCA-ODL model's efficiency in achieving high accuracy across these diverse categories.

Class	AR (%)				
	BCEVCA-ODL	CSADL-DEVM	BDEV-CAML	PSO-DAWRF	DAWRF
CPUHog	99.91	99.72	99.63	98.45	96.79
MemoryOF	99.76	99.55	99.43	98.04	97.04
Scanning	99.96	99.84	99.68	98.09	96.49
IOHog	99.92	99.72	99.55	98.23	96.56
DOS	99.82	99.63	99.50	98.07	96.95

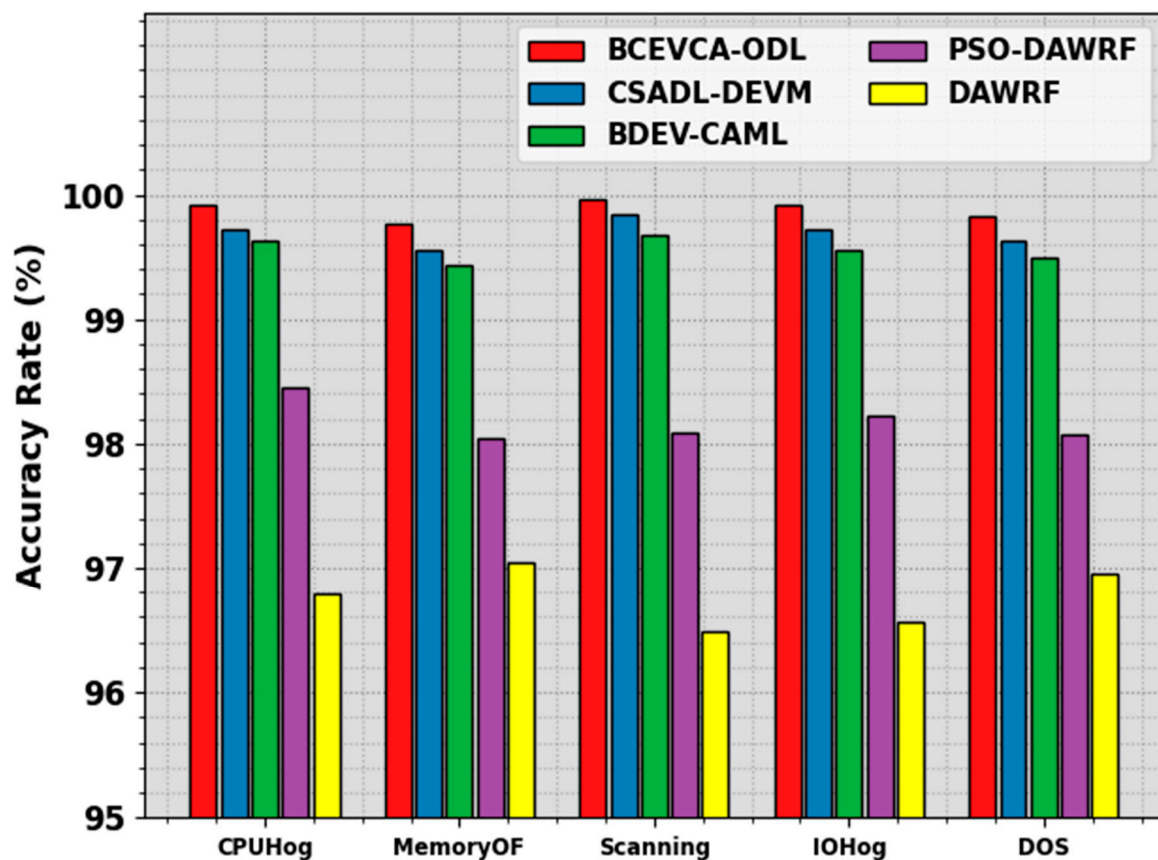


Figure 7. (AR, %) outcomes of the BCEVCA-ODL technique across diverse classes. The figure compares the AR results of the BCEVCA-ODL technique with other methods computed on five distinct classes: CPUHog, MemoryOF, Scanning, IOHog, and DOS. The data illustrate each method's accuracy, accentuating the superior performance of the BCEVCA-ODL technique in achieving high accuracy for fault detection across these classes.

Brief F-score rate (FR) results of the BCEVCA-ODL technique are shown in Table 4 and Figure 8. The outcomes indicated that the PSO-DAWRF and DAWRF techniques obtain the lowest FR values over other models. Although the CSADL-DEVM and BDEV-CAML models attain slightly boosted FR values, the BCEVCA-ODL technique shows maximum values of FR under all classes. On CPUHog Class, the BCEVCA-ODL method attained an increased FR value of 99.63%, while the CSADL-DEVM, BDEV-CAML, PSO-DAWRF, and DAWRF techniques attained minimized FR values of 99.43%, 99.30%, 97.65%, and 96.49%, respectively.

Table 4. (FR, %) outcomes of the BCEVCA-ODL technique compared to other methods across diverse classes. The figure presents the FR results of the BCEVCA-ODL technique and other methods evaluated on five classes: CPUHog, MemoryOF, Scanning, IOHog, and DOS. The data illustrate the efficiency of each method in terms of the F-score, highlighting the superior performance of the BCEVCA-ODL model in achieving higher F-scores for fault detection across all classes.

Class	FR (%)				
	BCEVCA-ODL	CSADL-DEVM	BDEV-CAML	PSO-DAWRF	DAWRF
CPUHog	99.63	99.43	99.30	97.65	96.49
MemoryOF	99.73	99.52	99.41	97.77	96.53
Scanning	99.82	99.63	99.44	97.96	96.50
IOHog	99.83	99.63	99.51	97.85	96.92
DOS	99.79	99.59	99.48	97.82	96.65

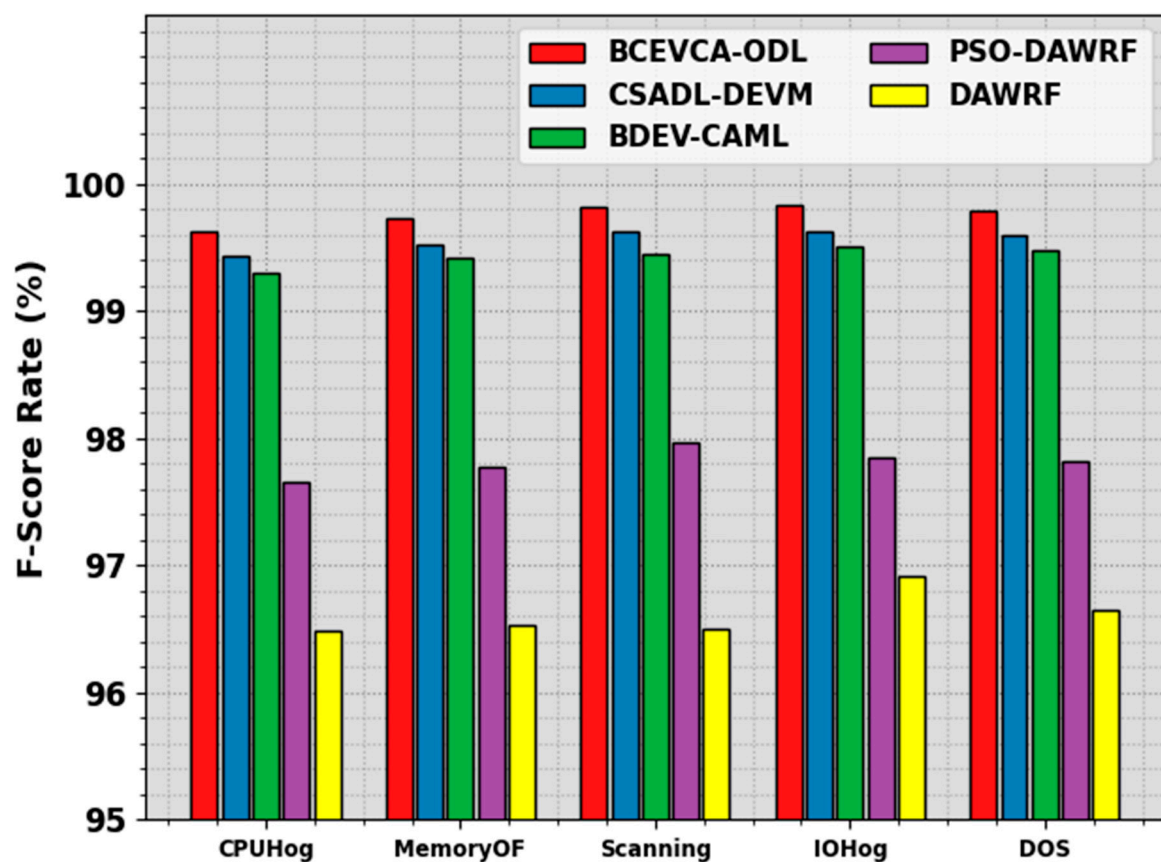


Figure 8. (FR, %) outcomes of the BCEVCA-ODL technique across diverse classes. The figure compares the FR results of the BCEVCA-ODL technique with other methods computed on five classes: CPUHog, MemoryOF, Scanning, IOHog, and DOS. The data demonstrate the performance of each method in terms of the F-score, highlighting the superior capability of the BCEVCA-ODL model to achieve higher F-scores across all the classes, indicating its efficiency in fault detection.

A brief fault detection accuracy (FDA) outcome of the BCEVCA-ODL method is shown in Table 5 and Figure 9. The outputs show that the ETXTD and NFD techniques obtain the lowest FDA values over other models. Although the PSO-DAWRF, CSADL-DEVM, and BDEV-CAML models attain slightly boosted FDA values, the BCEVCA-ODL technique shows maximum values of FDA under fault probability. At a fault probability of 0.05, the BCEVCA-ODL technique obtained an increased FDA value of 99.89%, while the CSADL-DEVM, BDEV-CAML, PSO-DAWRF, NFD, and ETXTD methods attained decreased FDA values of 99.52%, 99.16%, 97.99%, 96.79%, and 96.36%, respectively. In

addition, at a fault probability of 0.35, the BCEVCA-ODL method attained an increased FDA value of 93.41%, while the CSADL-DEVM, BDEV-CAML, PSO-DAWRF, NFD, and ETXTD models obtained decreased FDA values of 92.21%, 91.08%, 89.04%, 84.59%, and 80.52%, respectively.

Table 5. (FDA, %) outcomes of the BCEVCA-ODL technique compared to other methods under various fault probabilities. The table presents the FDA results for the BCEVCA-ODL technique and other methods across diverse fault probability levels ranging from 0.00 to 0.50. The data highlight the performance of each technique in FDA as fault probability increases, illustrating the efficiency of the BCEVCA-ODL method in maintaining high detection accuracy compared to the other approaches at various fault levels.

Fault Probability	FDA (%)					
	BCEVCA-ODL	CSADL-DEVM	BDEV-CAML	PSO-DAWRF	NFD	ETXTD
0.00	100.00	100.00	100.00	100.00	100.00	100.00
0.05	99.89	99.52	99.16	97.99	96.79	96.36
0.10	99.86	99.56	99.20	98.51	95.57	92.43
0.15	97.76	96.56	96.18	94.34	92.46	89.83
0.20	96.84	95.63	95.27	92.45	90.40	87.27
0.25	95.60	94.40	94.05	91.63	88.38	86.60
0.30	94.40	93.17	92.70	90.70	84.58	83.75
0.35	93.41	92.21	91.08	89.04	84.59	80.52
0.40	93.21	91.00	89.62	86.06	82.84	78.64
0.45	91.18	89.98	87.82	84.20	81.75	78.88
0.50	90.53	89.30	87.14	83.25	81.50	75.00

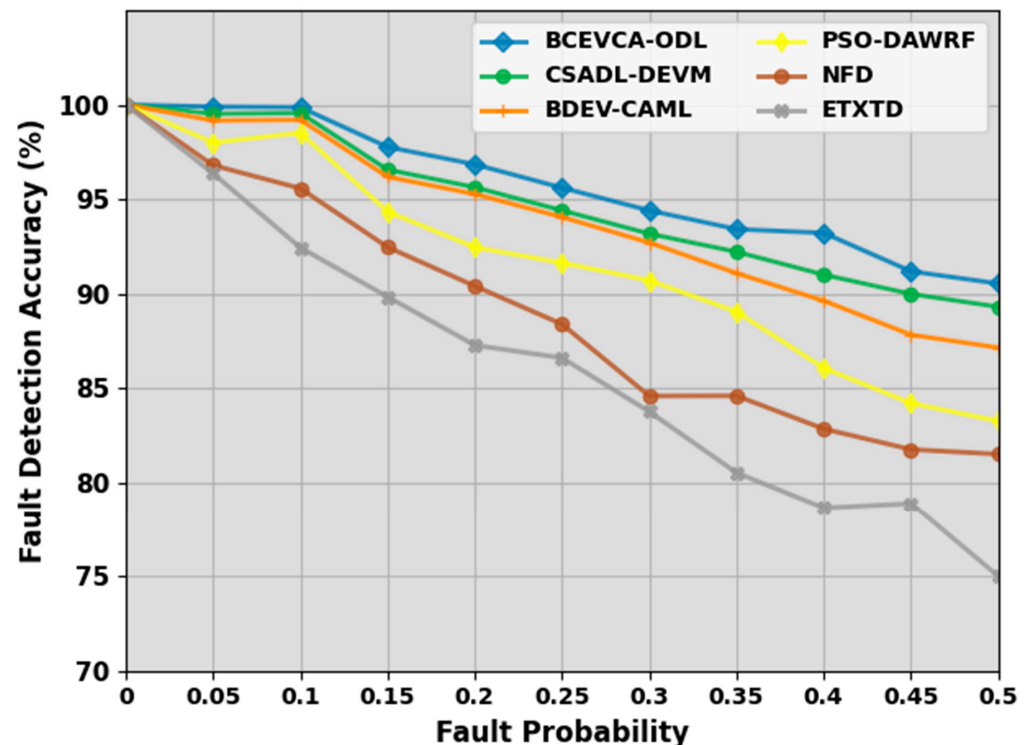


Figure 9. (FDA, %) outcomes of the BCEVCA-ODL technique across various fault probabilities. The figure illustrates the FDA performance of the BCEVCA-ODL technique in comparison to other methods under diverse fault probability levels ranging from 0.05 to 0.50. The data show how each method performs as fault probability increases, highlighting the capability of the BCEVCA-ODL model to maintain superior accuracy in fault detection across all probability levels.

The fault alarm ratio (FAR) results of the BCEVCA-ODL technique with other techniques are shown in Table 6 and Figure 10. The outputs imply that the BCEVCA-ODL method attains better outcomes over other models with minimal FAR values. At a fault probability of 0.10, the BCEVCA-ODL technique offers a reduced FAR of 0.36%, while the CSADL-DEVM, BDEV-CAML, PSO-DAWRF, NFD, and ETXTD models provide increased FAR values of 0.62%, 0.87%, 1.14%, 1.63%, and 1.99%, respectively. Moreover, at a fault probability of 0.50, the BCEVCA-ODL technique offers a reduced FAR of 5.68%, while the CSADL-DEVM, BDEV-CAML, PSO-DAWRF, NFD, and ETXTD methods provide increased FAR values of 6.33%, 9.00%, 12.56%, 15.19%, and 20.16%, respectively. These results highlight the supremacy of the BCEVCA-ODL technique.

Table 6. (FAR, %) outcomes of the BCEVCA-ODL technique compared to other methods under various fault probabilities. The table presents the FAR results for the BCEVCA-ODL technique alongside other methods, evaluated across diverse fault probability levels ranging from 0.10 to 0.50. The data illustrate the performance of each method in terms of the FAR, showing how the FAR increases with higher fault probabilities. This comparison highlights the effectiveness of the BCEVCA-ODL model in reducing the FAR compared to other approaches.

Fault Probability	FAR (%)					
	BCEVCA-ODL	CSADL-DEVM	BDEV-CAML	PSO-DAWRF	NFD	ETXTD
0.10	0.36	0.62	0.87	1.14	1.63	1.99
0.15	0.56	0.81	1.06	1.63	1.99	2.81
0.20	0.79	1.02	1.14	2.48	2.72	3.86
0.25	0.96	1.20	1.49	2.23	3.45	4.97
0.30	0.81	1.32	2.07	3.47	5.15	7.04
0.35	1.56	3.03	4.26	6.02	8.06	9.37
0.40	3.31	3.84	6.03	7.78	9.98	12.08
0.45	4.53	5.31	7.69	10.12	11.73	17.22
0.50	5.68	6.33	9.00	12.56	15.19	20.16

To evaluate the effectiveness of the BCEVCA-ODL technique, accuracy curves for the training (TRA) and validation (TES) phases are generated, as shown in Figure 11. The curves reveal that as the epoch count increases, both the TRA and TES accuracies show significant improvement. Specifically, the TRA accuracy reaches 0.995, while the TES accuracy stabilizes around 0.990. This upward trend demonstrates the capability of the model to learn and generalize effectually, recognizing patterns within both the TRA/TES datasets as the number of epochs progresses.

Figure 12 illustrates the performance of the BCEVCA-ODL technique by displaying the loss values throughout the training (TRA) process. As the epochs increase, the TRA loss steadily decreases, indicating that the method continuously refines its weights to minimize prediction errors for both the TRA and validation (TES) datasets. The loss curve depicts a significant reduction, with the TRA loss nearing 0.02 and the TES loss stabilizing around 0.04. This consistent decrease in loss values reflects the effective learning and ability of the model to fit the data. The diminishing gap between the TRA/TES losses highlights the method's successful adaptation and generalization capability across both datasets.

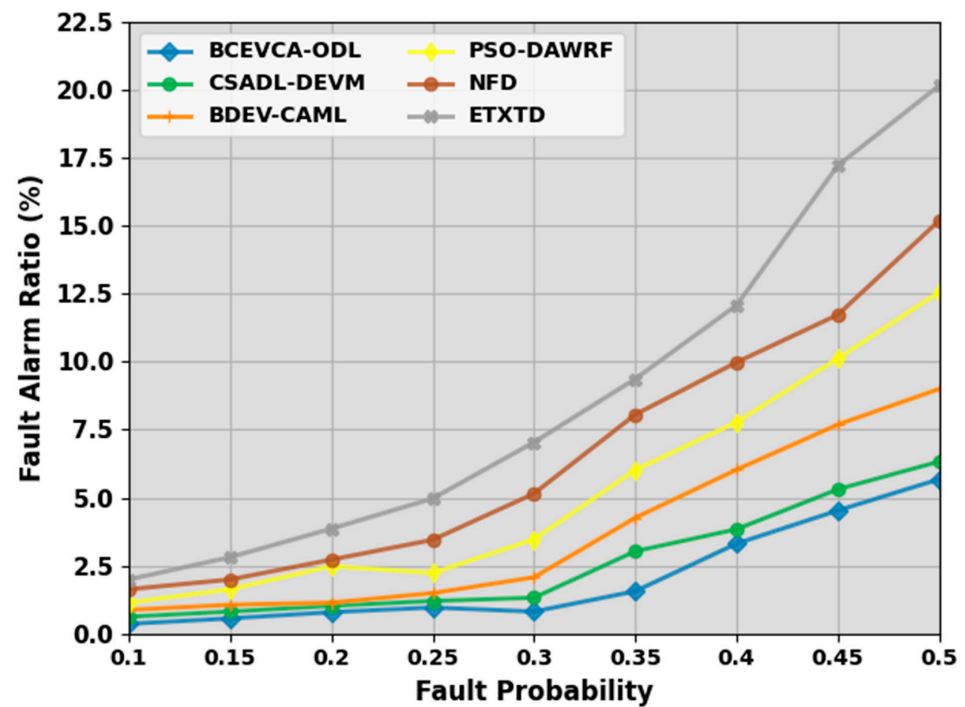


Figure 10. (FAR, %) outcomes of the BCEVCA-ODL technique across diverse fault probabilities. The figure compares the FAR results of the BCEVCA-ODL technique with other methods under fault probability levels ranging from 0.1 to 0.5. The data show how the FAR varies with increasing fault probabilities, emphasizing the performance of each technique in minimizing false alarms. The figure illustrates the effectiveness of the BCEVCA-ODL model in maintaining a lower FAR compared to other approaches across all fault probability levels.

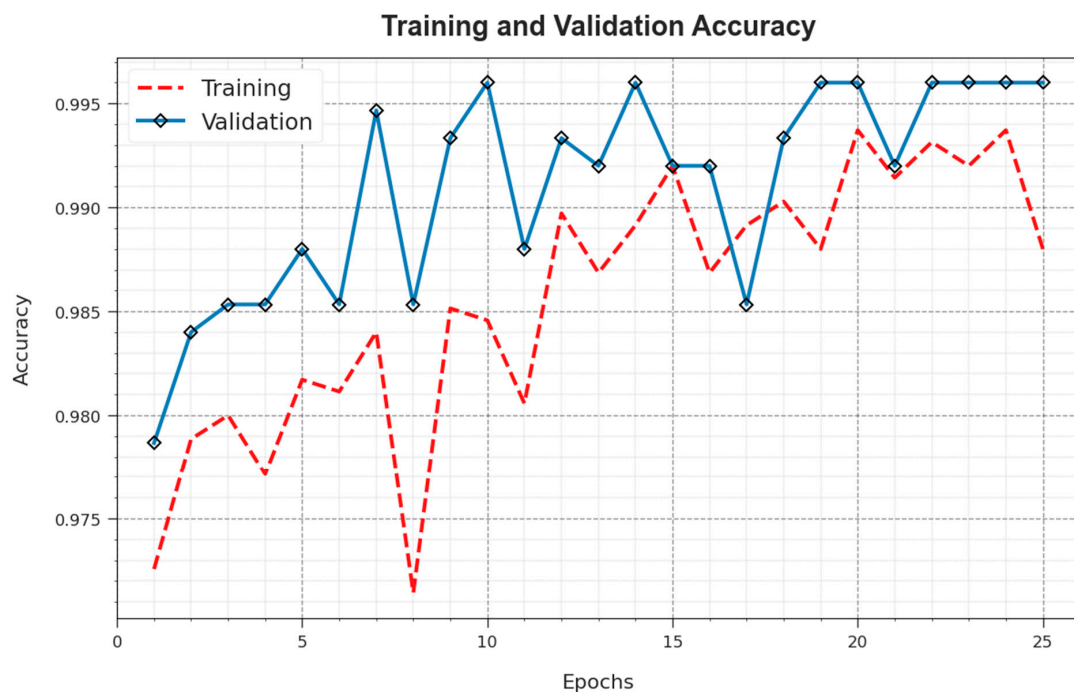


Figure 11. $Accu_y$ curve of the BCEVCA-ODL technique showing TRA and TES accuracy over epochs. The figure illustrates the change in accuracy for both TRA and TES datasets as the model progresses through epochs. The curve reflects the performance of the BCEVCA-ODL technique, with the accuracy increasing and stabilizing across the epochs. The plot provides insights into the model's capability to learn and generalize over time, highlighting how the TRA and TES accuracy increase during the learning process.



Figure 12. The loss curve of the BCEVCA-ODL technique shows TRA and TES loss over epochs. The figure illustrates the change in loss values for both TRA and TES datasets as the model progresses through epochs. It shows how the TRA and TES losses decrease over time, reflecting the model's learning process and its capability to mitigate errors during TRA. This curve provides insights into the convergence behavior of the BCEVCA-ODL model, demonstrating how well the model fits the data and generalizes over the epochs.

Table 7 and Figure 13 compare various methods based on their computation time (CT) in seconds. The BCEVCA-ODL method achieves the lowest CT at 4.49 s, indicating its efficiency in processing. The CSADL-DEVM method has a CT of 9.57 s, followed by BDEV-CAML at 8.32 s, which likely comprises some form of DL or evolutionary optimization technique. The PSO-DAWRF model has a slightly higher CT of 8.87 s. NFD takes 6.54 s, and ETXTD registers 9.50 s, suggesting a focus on specific detection or routing tasks in complex environments. The varying CTs highlight the tradeoffs between method complexity and processing speed.

Table 7. CT analysis of the BCEVCA-ODL technique compared to existing models. The figure presents the CT (in seconds) for the BCEVCA-ODL technique alongside other methods. The data highlight the computational efficiency of the BCEVCA-ODL model, demonstrating its capability to achieve faster processing times compared to the different techniques, making it a more effectual outcome for the given task.

Methods	CT (s)
BCEVCA-ODL	4.49
CSADL-DEVM	9.57
BDEV-CAML	8.32
PSO-DAWRF	8.87
NFD	6.54
ETXTD	9.50

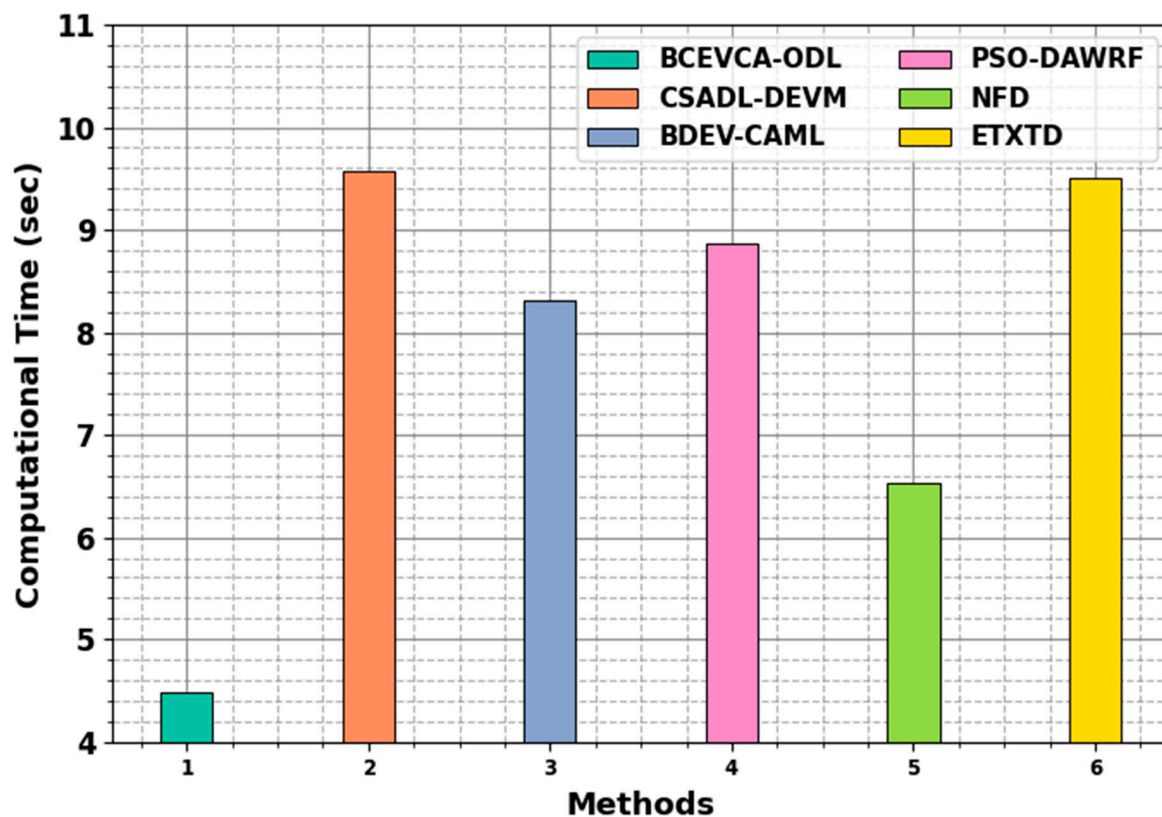


Figure 13. CT analysis of the BCEVCA-ODL technique compared to existing models. The figure presents the computational time (in seconds) for the BCEVCA-ODL technique and other methods comprising CSADL-DEVM, BDEV-CAML, PSO-DAWRF, NFD, and ETXTD. The data show the computational efficiency of each method, with the BCEVCA-ODL technique accentuating the shortest processing time compared to the others. This analysis underscores the superior speed and effectualness of the BCEVCA-ODL model in performing the task across the evaluated methods.

5. Conclusions

In this study, a new BCEVCA-ODL approach was developed for sustainable IoT fault detection. IoT devices have a substantial level of decentralized decision-making capacity in BC technology to achieve a consensus on the accomplishment of intrablock transactions. The SSAE model was employed to detect faults in the IoT network. Finally, the PFOA method was used for the optimum hyperparameter tuning of the SSAE model, which enhances the fault recognition rate. A wide range of simulations was performed to highlight the efficacy of the BCEVCA-ODL technique. The investigational outcome of the BCEVCA-ODL technique portrayed a superior FDA value, demonstrating its optimal performance in fault detection when no faults are present compared to other techniques. Furthermore, the accurate detection of the faults in the IoT network decreases system inefficiency and the carbon footprint, supporting IoT operations with global sustainability goals. This incorporation promotes the expansion of eco-friendly IoT solutions, which balances the tradeoff between technological advancement with environmental responsibility.

The limitations of the BCEVCA-ODL technique comprise its reliance on simulated data, which may not fully capture the complexities of real-world IoT environments. Additionally, the model's performance may degrade in highly dynamic or large-scale systems requiring real-time processing. This study also does not account for the impact of communication delays and network failures in BC-assisted IoT systems. Future work may focus on enhancing the scalability and robustness of the model by integrating real-time data and addressing network instability. Furthermore, integrating adaptive learning models

could improve fault detection in more unpredictable scenarios. Exploring hybrid models incorporating multiple optimization models is another potential area for future research.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The author gratefully acknowledges the funding provided by the Deanship of Graduate Studies and Scientific Research, Jazan University, Saudi Arabia, through Project Number GSSRD-25.

Conflicts of Interest: The author has expressed no conflicts of interest.

References

1. Raj, J.M.; Ranjani, S.S. A secured blockchain method for multivariate industrial IoT-oriented infrastructure based on deep residual squeeze and excitation network with single candidate optimizer. *Internet Things* **2023**, *22*, 100823.
2. Almaiah, M.A.; Ali, A.; Hajje, F.; Pasha, M.F.; Alohal, M.A. A lightweight hybrid deep learning privacy preserving model for FC-based industrial internet of medical things. *Sensors* **2022**, *22*, 2112. [\[CrossRef\]](#)
3. Fadhel, Z.; Attia, H.; Ali, Y.H. Optimized and Comprehensive Fake Review Detection based on Harris Hawks optimization integrated with Machine Learning Techniques. *J. Cybersecur. Inf. Manag.* **2025**, *15*, 11.
4. Wang, F.; Cui, J.; Zhang, Q.; He, D.; Gu, C.; Zhong, H. Blockchain-Based Lightweight Message Authentication for Edge-Assisted Cross-Domain Industrial Internet of Things. *IEEE Trans. Dependable Secur. Comput.* **2023**, *21*, 1587–1604. [\[CrossRef\]](#)
5. Abdulqadder, I.H.; Zhou, S. SliceBlock: Context-aware authentication handover and secure network slicing using DAG-blockchain in edge-assisted SDN/NFV-6G environment. *IEEE Internet Things J.* **2022**, *9*, 18079–18097. [\[CrossRef\]](#)
6. Rezazadeh, B.; Asghari, P.; Rahmani, A.M. Computer-aided methods for combating Covid-19 in prevention, detection, and service provision approaches. *Neural Comput. Appl.* **2023**, *35*, 14739–14778. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Sharma, A.; Pilli, E.S.; Mazumdar, A.P.; Jain, A. BLAST-IoT: BLockchain Assisted Scalable Trust in Internet of Things. *Comput. Electr. Eng.* **2023**, *109*, 108752. [\[CrossRef\]](#)
8. Sellami, B.; Hakiri, A.; Yahia, S.B. Deep Reinforcement Learning for energy-aware task offloading in join SDN-Blockchain 5G massive IoT edge network. *Future Gener. Comput. Syst.* **2022**, *137*, 363–379. [\[CrossRef\]](#)
9. Ajayi, O.J.; Rafferty, J.; Santos, J.; Garcia-Constantino, M.; Cui, Z. BECA: A Blockchain-Based Edge Computing Architecture for Internet of Things Systems. *IoT* **2021**, *2*, 610–632. [\[CrossRef\]](#)
10. Fan, W.; Gao, L.; Su, Y.; Wu, F.; Liu, Y.A. Joint DNN Partition and Resource Allocation for Task Offloading in Edge-Cloud-Assisted IoT Environments. *IEEE Internet Things J.* **2023**, *10*, 10146–10159. [\[CrossRef\]](#)
11. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R.; Jolfaei, A.; Islam, A.N. A blockchain-orchestrated deep learning approach for secure data transmission in IoT-enabled healthcare system. *J. Parallel Distrib. Comput.* **2023**, *172*, 69–83. [\[CrossRef\]](#)
12. Xu, S.; Liao, B.; Yang, C.; Guo, S.; Hu, B.; Zhao, J.; Jin, L. Deep reinforcement learning assisted edge-terminal collaborative offloading algorithm of blockchain computing tasks for energy Internet. *Int. J. Electr. Power Energy Syst.* **2021**, *131*, 107022. [\[CrossRef\]](#)
13. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R. BDEdge: Blockchain and deep-learning for secure edge-envisioned green CAVs. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 1330–1339. [\[CrossRef\]](#)
14. Du, X.; Chen, X.; Lu, Z.; Duan, Q.; Wang, Y.; Wu, J.; Hung, P.C. A Blockchain-Assisted Intelligent Edge Cooperation System for IoT Environments with Multi-Infrastructure Providers. *IEEE Internet Things J.* **2023**, *10*, 21227–21239. [\[CrossRef\]](#)
15. Alqaralleh, B.A.; Vaiyapuri, T.; Parvathy, V.S.; Gupta, D.; Khanna, A.; Shankar, K. Blockchain-assisted secure image transmission and diagnosis model on Internet of Medical Things Environment. *Pers. Ubiquitous Comput.* **2021**, *28*, 17–27. [\[CrossRef\]](#)
16. Zhang, S.; Wang, Z.; Zhou, Z.; Wang, Y.; Zhang, H.; Zhang, G.; Ding, H.; Mumtaz, S.; Guizani, M. Blockchain and federated deep reinforcement learning based secure cloud-edge-end collaboration in power IoT. *IEEE Wirel. Commun.* **2022**, *29*, 84–91. [\[CrossRef\]](#)
17. Gumaei, A.; Al-Rakhami, M.; Hassan, M.M.; Pace, P.; Alai, G.; Lin, K.; Fortino, G. Deep learning and blockchain with edge computing for 5G-enabled drone identification and flight mode detection. *IEEE Netw.* **2021**, *35*, 94–100. [\[CrossRef\]](#)
18. Sharadq, A.A.; Hatamleh, H.A.M.; Saloum, S.S.; Alawneh, T.A. Hybrid Chain: Blockchain Enabled Framework for Bi-Level Intrusion Detection and Graph-Based Mitigation for Security Provisioning in Edge Assisted IoT Environment. *IEEE Access* **2023**, *11*, 27433–27449. [\[CrossRef\]](#)

19. Hu, X.; Tang, T.; Tan, L.; Zhang, H. Fault Detection for Point Machines: A Review, Challenges, and Perspectives. *Actuators* **2023**, *12*, 391. [\[CrossRef\]](#)
20. Alserhani, F.M. Integrating deep learning and metaheuristics algorithms for blockchain-based reassurance data management in the detection of malicious IoT nodes. *Peer-Peer Netw. Appl.* **2024**, *17*, 3856–3882. [\[CrossRef\]](#)
21. Alyoubi, K.H.; Khadidos, A.O.; Alshareef, A.M.; Hamed, D.; Khadidos, A.O.; Ragab, M. Capuchin Search Algorithm with Deep Learning-Based Data Edge Verification for Blockchain-Assisted IoT Environment. *IEEE Access* **2023**, *12*, 351–360. [\[CrossRef\]](#)
22. Kokila, M.; Reddy, K.S. BlockDLO: Blockchain computing with deep learning orchestration for secure data communication in IoT Environment. *IEEE Access* **2024**, *12*, 134521–134540. [\[CrossRef\]](#)
23. Sekhar, G.C.; Aruna, R. A Novel Blockchain-Assisted Deep Learning Model for Secure Edge Intelligence in IoT Networks. *J. Inst. Eng. India Ser. C* **2024**, 1–14. [\[CrossRef\]](#)
24. Alamro, H.; Marzouk, R.; Alruwais, N.; Negm, N.; Aljameel, S.S.; Khalid, M.; Hamza, M.A.; Alsaid, M.I. Modelling of Blockchain Assisted Intrusion Detection on IoT Healthcare System using Ant Lion Optimizer with Hybrid Deep Learning. *IEEE Access* **2023**, *11*, 82199–82207. [\[CrossRef\]](#)
25. Kulandaivelu, K.; Rajappan, S.; Murugasamy, V. Blockchain Enabled Secure Medical Data Transmission and Diagnosis Using Golden Jackal Optimization Algorithm with Deep Learning. *Braz. Arch. Biol. Technol.* **2024**, *67*, e24240214. [\[CrossRef\]](#)
26. Perumal, E.; Arulanthu, P.; Ramachandran, R.; Singh, R. Enhanced Metaheuristics with Deep Learning Model for Blockchain Assisted Cyber Security Solution in Internet of Things Environment. In Proceedings of the 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), Vellore, India, 23–24 February 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–7.
27. Khan, Z.A.; Amjad, S.; Ahmed, F.; Almasoud, A.M.; Imran, M.; Javaid, N. A blockchain-based deep-learning-driven architecture for quality routing in wireless sensor networks. *IEEE Access* **2023**, *11*, 31036–31051. [\[CrossRef\]](#)
28. Swamynathan, C.; Shanmugam, R.; Mohan Kumar, K.P.; Subbiyan, B. Traffic Prevention and Security Enhancement in VANET Using Deep Learning with Trusted Routing Aided Blockchain Technology. *Trans. Emerg. Telecommun. Technol.* **2024**, *35*, e70004. [\[CrossRef\]](#)
29. Katib, I.; Ragab, M. Blockchain-assisted hybrid harris hawks optimization based deep DDoS attack detection in the IoT environment. *Mathematics* **2023**, *11*, 1887. [\[CrossRef\]](#)
30. Zuo, Y. Exploring the Synergy: AI Enhancing Blockchain, Blockchain Empowering AI, and their Convergence across IoT Applications and Beyond. *IEEE Internet Things J.* **2024**, early access. [\[CrossRef\]](#)
31. Mohammed, M.A.; Lakhan, A.; Zebari, D.A.; Abd Ghani, M.K.; Marhoon, H.A.; Abdulkareem, K.H.; Nedoma, J.; Martinek, R. Securing healthcare data in industrial cyber-physical systems using combining deep learning and blockchain technology. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107612. [\[CrossRef\]](#)
32. Kumar, P.; Kumar, R.; Aljuhani, A.; Javeed, D.; Jolfaei, A.; Islam, A.N. Digital twin-driven SDN for smart grid: A deep learning integrated blockchain for cybersecurity. *Sol. Energy* **2023**, *263*, 111921. [\[CrossRef\]](#)
33. Ranjan, A.K.; Kumar, P. Ensuring the privacy and security of IoT-medical data: A hybrid deep learning-based encryption and blockchain-enabled transmission. *Multimed. Tools Appl.* **2024**, *83*, 79067–79092. [\[CrossRef\]](#)
34. Yadav, A.K.; Singh, K. Comparative analysis of consensus algorithms of blockchain technology. In *Ambient Communications and Computer Systems: RACCCS 2019*; Springer: Singapore, 2020; pp. 205–218.
35. Yan, B.; Han, G. Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. *IEEE Access* **2018**, *6*, 41238–41248. [\[CrossRef\]](#)
36. Cao, S.; Qian, Q.; Cao, Y.; Li, W.; Huang, W.; Liang, J. A Novel Meta-heuristic Algorithm for Numerical and Engineering Optimization Problems: Piranha Foraging Optimization Algorithm (PFOA). *IEEE Access* **2023**, *11*, 92505–92522. [\[CrossRef\]](#)
37. Vaiyapuri, T.; Shankar, K.; Rajendran, S.; Kumar, S.; Acharya, S.; Kim, H. Blockchain Assisted Data Edge Verification with Consensus Algorithm for Machine Learning Assisted IoT. *IEEE Access* **2023**, *11*, 55370–55379. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.