Computer Modeling in Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.062549

ARTICLE

Privacy-Aware Federated Learning Framework for IoT Security Using Chameleon Swarm Optimization and Self-Attentive Variational Autoencoder

Saad Alahmari[1,*] **and Abdulwhab Alkharashi**[2]

[1]Department of Computer Science, Applied College, Northern Border University, Arar, 91431, Saudi Arabia
[2]Department of Computer Science, College of Computing and Informatics, Saudi Electronic University, Riyadh, 11673, Saudi Arabia
*Corresponding Author: Saad Alahmari. Email: saad.alahmari@nbu.edu.sa

Received: 20 December 2024; Accepted: 27 February 2025; Published: 11 April 2025

ABSTRACT: The Internet of Things (IoT) is emerging as an innovative phenomenon concerned with the development of numerous vital applications. With the development of IoT devices, huge amounts of information, including users' private data, are generated. IoT systems face major security and data privacy challenges owing to their integral features such as scalability, resource constraints, and heterogeneity. These challenges are intensified by the fact that IoT technology frequently gathers and conveys complex data, creating an attractive opportunity for cyberattacks. To address these challenges, artificial intelligence (AI) techniques, such as machine learning (ML) and deep learning (DL), are utilized to build an intrusion detection system (IDS) that helps to secure IoT systems. Federated learning (FL) is a decentralized technique that can help to improve information privacy and performance by training the IDS on discrete linked devices. FL delivers an effectual tool to defend user confidentiality, mainly in the field of IoT, where IoT devices often obtain privacy-sensitive personal data. This study develops a Privacy-Enhanced Federated Learning for Intrusion Detection using the Chameleon Swarm Algorithm and Artificial Intelligence (PEFLID-CSAAI) technique. The main aim of the PEFLID-CSAAI method is to recognize the existence of attack behavior in IoT networks. First, the PEFLID-CSAAI technique involves data preprocessing using Z-score normalization to transform the input data into a beneficial format. Then, the PEFLID-CSAAI method uses the Osprey Optimization Algorithm (OOA) for the feature selection (FS) model. For the classification of intrusion detection attacks, the Self-Attentive Variational Autoencoder (SA-VAE) technique can be exploited. Finally, the Chameleon Swarm Algorithm (CSA) is applied for the hyperparameter fine-tuning process that is involved in the SA-VAE model. A wide range of experiments were conducted to validate the execution of the PEFLID-CSAAI model. The simulated outcomes demonstrated that the PEFLID-CSAAI technique outperformed other recent models, highlighting its potential as a valuable tool for future applications in healthcare devices and small engineering systems.

KEYWORDS: Federated learning; internet of things; artificial intelligence; chameleon swarm algorithm; intrusion detection system; healthcare IoT devices

1 Introduction

The rapid development and growth of the Internet of Things (IoT) has led to significant improvements in data created on the edge of the network [1]. This development and growth have created novel challenges to the traditional cloud-based centralized techniques for the study of data in two main ways. Firstly, centralized techniques are not suitable for the 5G/6G period owing to the very high transmission and storage costs involved in combining data from millions or even billions of IoT devices [2]. Secondly, data gathering is increasingly regarded as a threat to user privacy. With cloud-based centralized techniques, user data can

be distributed or sold to numerous corporations, negatively affecting data security and breaching privacy rights, additionally fueling public suspicion of data-focused applications [3]. Thus, a distributed privacy-maintaining technique for inference-based applications and data-driven learning is required for efficacy and to mitigate privacy concerns.

The exposure of data in the IoT environment creates opportunities for several adverse activities, such as malware exploits, Denial of Service (DoS) attacks, phishing schemes, IoT botnet intrusions, routing operations, and challenges associated with the security of the cloud [4]. In a zero-trust security method, intrinsic IoT vulnerabilities are recognized and trust can be frequently retained. Authentication is needed for every attempt to access network sources and widespread security is introduced against the various threats initiated either internally or externally in the IoT system [5]. The defense of IoT systems against cyber-attacks has become a vital study field recently. Artificial intelligence (AI)-based techniques, mainly deep learning (DL) and machine learning (ML), are used in the IoT security context, as they can identify, alleviate, and predict abnormalities and unusual traffic behaviors [6]. However, many current IDSs cannot perform as effectively with the fast development of IoT-associated devices, and they enforce a tremendous burden on systems. Hence, current DL/ML-based IDSs are not appropriate for 6G-aided IoT, and intrusion detection can become less precise with these rapid changes [7]. To address this difficulty, Federated learning (FL) has emerged as a cooperative model, which improves edge intelligence while preserving confidentiality.

FL has gathered extensive attention as a security solution owing to its consistency and ability to preserve privacy [8]. FL allows users to study the method cooperatively by sharing local parameters without exposing private data. Instinctively, it is more secure than centralized training; however, recent experiments have revealed that it also leads to many security concerns. For instance, with distributed parameters, a malicious attacker could initiate attacks to retrieve images from a device that uses face detection [9]. Additionally, adversaries can effortlessly intercept the medical data of patients who wear gadgets. This not only invades individuals' privacy, but, more importantly, could lead to personal or economic damage. These security threats in FL are a significant feature impeding the advancement and further improvement of IAI-based applications [10]. Fig. 1 presents the structure of FL.

Existing federated learning (FL)-based intrusion detection systems (IDSs) for IoT environments face challenges in terms of feature selection, hyperparameter tuning, and anomaly detection, limiting their accuracy, privacy preservation, and scalability. Current frameworks struggle to adapt to dynamic cyber threats while ensuring efficient intrusion detection in resource-constrained IoT networks.

Research Question: Can the integration of Self-Attentive Variational Autoencoder (SA-VAE), Chameleon Swarm Algorithm (CSA), and Osprey Optimization Algorithm (OOA) in a federated learning-based IDS enhance intrusion detection accuracy, privacy preservation, and scalability in IoT networks?

This study involves the development of a Privacy-Enhanced FL for Intrusion Detection using the Chameleon Swarm Algorithm and Artificial Intelligence (PEFLID-CSAAI) technique. The PEFLID-CSAAI technique involves data preprocessing using Z-score normalization to transform the input data into a beneficial format. Additionally, the PEFLID-CSAAI method uses an Osprey Optimization Algorithm (OOA) for the FS model. For the classification of IDS attacks, the Self-Attentive Variational Autoencoder (SA-VAE) technique can be exploited. Then, the Chameleon Swarm Algorithm (CSA) is applied for the hyperparameter fine-tuning process that is involved in the SA-VAE model. An extensive variety of experiments were conducted to verify the execution of the PEFLID-CSAAI model.

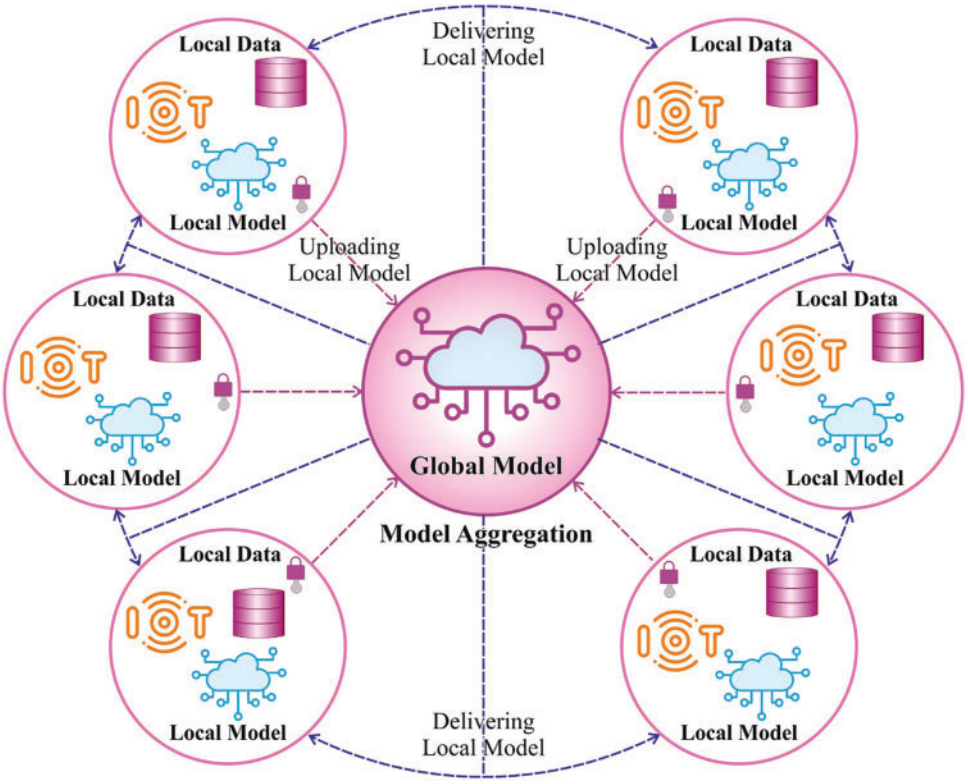

Figure 1: Federated learning architecture

2 Review of the Literature

Hamdi [11] presents a federated-based IDS. To study the implementation of the technique, the author studied the client-side assessment whereby users send local data to the server, which then combines them using an upgraded global method. Users estimate the global method locally and transfer the outcomes back to the server to be combined utilizing metric aggregating functions. Li et al. [12] introduce a new and collaborative technique for intrusion detection in the IoT that can help to solve specific security problems. The proposed technique selects the most significant features, which are superior in defining transmission among entities, by utilizing a Black Hole Optimizer (BHO). The control of all subnets can be achieved by using a controller node that utilizes an equivalent association of convolutional neural networks (CNNs) to identify the occurrence of defense risks in the traffic that travels across its subnet. Angelin et al. [13] present the execution of a DL-based network IDS (NIDS) intended to classify various kinds of attacks in industrial IoT (IIoT) contexts. To improve the efficiency of the NIDS, the technique incorporates a CNN with an autoencoder (AE). The technique enhances the recognition results by decreasing the features of the data through the use of the AE.

Ullah et al. [14] developed an IDS for improving the security of the IoT, which uses transfer learning (TL) and multimodal big data representation. Initially, Packet Capture (PCAP) files were dragged to recover essential bytes and attacks. Next, Spark-based big data optimizer methods were used to manage the massive amount of data. Then, a TL technique similar to word2vec recovered semantic-based features. Finally, the method was advanced to change the system bytes into images, and text attributes were obtained by designing attention-based ResNets. Ghasemi et al. [15] proposed a hybrid IDS based on Grey Wolf Optimizer (GWO) and Support Vector Machine (SVM), which uses the improvements of these methods. In the presented

technique, SVM was utilized to differentiate between abnormal records and normal records, and GWO was utilized to discover the FS, kernel function, and fine-tuned optimum hyperparameters for SVM to enhance the classification. Hanafi et al. [16] proposed a Long Short-Term Memory (LSTM) system and enhanced Binary Golden Jackal Optimizer (BGJO) to improve a novel IDS method for IoT systems. Initially, the GJO is enhanced by opposition-based learning (OBL). BGJO utilizes the OBL approach to enhance the execution of the GJO and prevents the method from becoming stuck in the local optimum by monitoring the primary population.

Javeed et al. [17] propose a horizontal FL method that incorporates CNN and BiLSTM for an effectual IDS. This hybrid model aims to overcome the boundaries of current techniques and improve the efficiency of IDSs in the FL context for IoT systems. In particular, CNN can be utilized for the extraction of spatial features, allowing the method to determine the local patterns distinctive of possible attacks, whereas the BiLSTM module takes the temporal dependency and studies subsequent patterns in the data. The authors in [18] proposed a novel Chaotic Cuckoo Search Optimizer (CCSO) method with optimum wavelet kernel ELM (OWKELM) called the CCSO-OWKELM method for IDSs. The CCSO-OWKELM contains a CCSO-based FS method that combines the notions of chaotic maps with CSO. Moreover, the OWKELM method can be used for classification and intrusion detection.

2.1 Motivation for the Use of Federated Learning

IDSs often rely on centralized data collection, leading to significant privacy risks, scalability limitations, and communication bottlenecks [19]. FL addresses these challenges by enabling local model training on devices, thereby reducing data exposure risks and enhancing privacy preservation [20]. Specifically, the PEFLID-CSAAI framework employs FL to reduce dependency on a centralized database, ensuring enhanced privacy-preserving capabilities. Moreover, FL enables improved scalability, as IDS models can be updated locally without transferring raw data, which is particularly beneficial in large-scale IoT environments. Additionally, FL enables IDS to adapt to dynamic network threats more efficiently through decentralized learning, ensuring faster response times against emerging cyber threats [21]. Furthermore, by minimizing bandwidth usage and computational overhead, FL significantly enhances resource efficiency, making it suitable for constrained IoT environments [22]. These advantages establish FL as a promising paradigm for next-generation IDSs, particularly in IoT and edge computing ecosystems.

2.2 Comparison with Existing Techniques

Despite the advancements in FL-based IDSs for IoT contexts, the existing techniques exhibit notable limitations, as outlined in Table 1.

Table 1: Existing techniques and limitations

Existing technique	Limitations	How PEFLID-CSAAI overcomes limitations	Source	Objective
CNN-LSTM IDS	High computational cost, poor adaptability to evolving attacks	SA-VAE enhances adaptability by learning dynamic attack patterns	[23]	Improve IDS detection using CNN and LSTM hybridization
CNN-LSTM model	Struggles with long-time dependencies and uneven feature distribution	SA-VAE enhances feature extraction and reduces computational load	[24]	Improve IoT anomaly detection efficiency and accuracy
CNN-BiLSTM IDS	Requires extensive labeled data for training	FL allows learning without centralized data sharing	[25]	Improve IDS efficiency by combining CNN and BiLSTM

(Continued)

Table 1 (continued)

Existing technique	Limitations	How PEFLID-CSAAI overcomes limitations	Source	Objective
IDS-IoT (Traditional FL)	Privacy vulnerabilities due to model poisoning risks	CSA enhances robustness by optimizing hyperparameters dynamically	[26]	Apply federated learning for IDS in IoT systems
LSTM-Attn IDS for SDNs	Prone to cyberattacks due to centralized control	PEFLID-CSAAI improves detection accuracy and adaptability	[27]	Enhance IDS for dynamic networks
GAN-Based IDS	Prone to mode collapse and unstable training	SA-VAE stabilizes training and ensures diverse attack detection	[28]	Use GANs for anomaly-based IDS with adversarial learning
Hybrid SVM-RF IDS	Requires high computational resources for large datasets	OOA reduces feature dimensions, optimizing model efficiency	[29]	Combine SVM and RF to improve classification accuracy
VAE-Based IDS	Lacks confidence estimation in anomaly detection. Struggles with reliable classification of cyberattacks	SA-VAE improves confidence metrics using latent space representations	[30]	Enhance IDS reliability and accuracy in detecting network anomalies

3 Methodology

In this study, we developed a novel PEFLID-CSAAI technique. The main intention of the PEFLID-CSAAI methodology is to identify the existence of attack behavior in IoT networks. It contains four distinct stages: Z-score normalization, feature selection, attack detection using SA-VAE, and CSA-based parameter optimization. Fig. 2 depicts the workflow of the PEFLID-CSAAI technique.

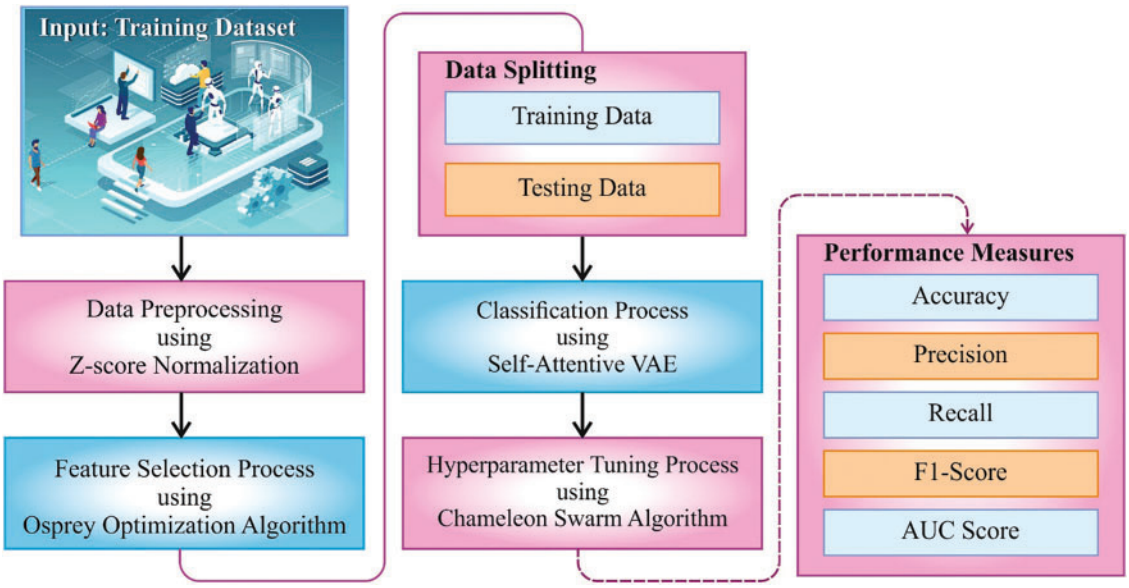

Figure 2: Workflow of PEFLID-CSAAI technique

3.1 Stage I: Z-Score Normalization

The PEFLID-CSAAI technique involves data preprocessing using Z-score normalization to convert the input data into a beneficial format. Z-score normalization, also called standardization, is a method utilized in IoT intrusion detection to improve the precision of anomaly detection methods [31]. By converting raw data into a typical format, this technique computes the Z-score for every data point, representing how many standard deviations it is from the mean. This aids in classifying outliers and uncommon patterns, which can indicate intrusions. Z-score normalization can be mostly effective while handling variable measures in IoT data, ensuring that features are subsidized similarly to the method of execution. Finally, it enhances the reliability and robustness of IDSs in dynamic IoT environments.

3.2 Stage II: OOA-Based Feature Selection

The PEFLID-CSAAI method uses the OOA for the feature selection process. The OOA simulates the osprey's hunting behaviors, involving recognizing the location of fish, carrying fish, and hunting fish [32]. In the OOA, this hunting behavior can be numerically demonstrated as the exploration stage, and the carrying fish behavior can be expressed as the exploitation stage. This structure allows the OOA to offset exploitation and exploration and obtain the optimum value of optimizer difficulties. The factors of the OOA are specified as follows.

In the OOA, along with other metaheuristic methods, the locations of each osprey are created at random in the search space by utilizing Eq. (1).

$$X_{i,j} = rand \times \left(ub_j - lb_j\right) + lb_j, j = 1, 2, \ldots, D \tag{1}$$

where $rand$ represents the randomly generated value among $(0, 1)$, and lb_j and ub_j are the lower and upper limits of the search spaces. D defines the optimizer difficulty size, and $X_{i,j}$ signifies the created location of the individual i in dimension j.

Additionally, in the process of iteration, the OOA includes the dual stages of hunting and carrying fish; these stages are described in more detail below.

Exploration stage

The initial phase of the OOA involves executing a global search and avoiding a decrease in the local optimum. In this stage, ospreys attack fish and attempt to search the entire search space. The fish locations are identified by utilizing Eq. (2).

$$PP_i = \{X_k | k \in \{1, 2, \ldots, N\} \cap F_k < F_i\} \cup \{X_{best}\} \tag{2}$$

where PP_i denotes the chosen location of fish for the osprey i, and F_i signifies the individualfitness i. X_{best} represents the best individual location.

After determining the fish locations, the osprey will attack these targets, which can be numerically expressed as in Eq. (3).

$$X_{i,j}^{P1} = X_{i,j} + r_1 \times \left(SF_{i,j} - I_{i,j} \times X_{i,j}\right), i = 1, 2, \ldots, N, j = 1, 2, \ldots, D \tag{3}$$

where r_1 is a randomly generated value in $[0, 1]$. $SF_{i,j}$ denotes the location that can be selected from the identified fishes. $I_{i,j}$ is an arbitrary integer in [1, 2]. Following this, the novel location will substitute the

preceding location if its fitness is superior to the preceding value, as described in Eq. (4).

$$X_i = \begin{cases} X_i^{P1}, & if\ F_i^{P1} < F_i \\ X_i, & otherwise \end{cases} \tag{4}$$

where F_i^{P1} denotes recently created location fitness.

Exploitation stage

During the second stage, the ospreys are required to determine an appropriate location after hunting a fish. In this phase, the OOA executes a local search and determines the optimum performance. The numerical method is presented in Eq. (5).

$$X_{i,j}^{P2} = X_{i,j} + \frac{lb_j + r_2 \times \left(ub_j - lb_j\right)}{t}, t = 1, 2, \ldots, T \tag{5}$$

where ub_j and lb_j are the upper and lower search space limits. r_2 is a random number between zero and one. T and t are the maximum and present iteration numbers. Similar to the preceding stage, a superior solution should now be implemented by the novel individual that can be identified by utilizing Eq. (6).

$$X_i = \begin{cases} X_i^{P2}, & if\ F_i^{P2} < F_i \\ X_i, & otherwise \end{cases} \tag{6}$$

where F_i^{P2} signifies the recently created location fitness. The pseudocode and flowchart of the OOA are displayed in Algorithm 1.

Algorithm 1: OOA pseudocode

Input the maximum iterations (T) and population size (N).
Initialize the locations of each and every ospreys by utilizing Eq. (1).
for $t = 1$ to T do
  for $i = 1$ to N do
    Stage 1: Finding the search space and hunting of fish
    Identify the location of fish utilizing Eq. (2).
    Compute the novel location of osprey utilizing Eq. (3).
    Ensure the limit settings and upgrade the $i - th$ osprey utilizing Eq. (4).
    Stage 2: Using the search space and carrying the fish
    Compute the novel location of osprey utilizing Eq. (5).
  Ensure the limit settings and upgrade the $i - th$ osprey utilizing Eq. (6).
  end for
end for
Output the optimal solution.

The fitness function (FF) applied in the OOA has been planned to have an equalize among the amount of chosen features in every solution (minimum) and the classifier precision (maximum) obtained by employing these selected features. Eq. (7) denotes the FF to assess solutions.

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

where $\gamma_R(D)$ signifies the error classifier rate of the presented classifications, $|R|$ denotes the chosen subset cardinality, $|C|$ means the overall value of the features in the dataset, and α and β are dual parameters equivalent to the significance of the classifier subset length and quality. $\in \beta = 1 - \alpha$ and [1, 0].

3.3 Stage III: Attack Detection Using SA-VAE

For the classification of intrusion detection attacks, the SA-VAE technique can be exploited. This section presents a foundational summary of the important structural blocks for the introduced SA-VAE: VAEs and self-attention mechanisms and presents the SA-VAE architecture [33].

VAE is an effective propagative method that incorporates the AE structure with various interpretation methods. Moreover, VAEs include a period of regularization, normally depending on Kullback–Leibler (KL) divergence, to help stimulate the learning of latent versions to pair with a previous distribution, namely, normal Gaussian distribution. This VAE comprises two important modules: a decoder and an encoder. The encoder converts input data in latent space, efficiently taking the natural distribution and framework inside the data. Training the Variational Autoencoder (VAE) involves minimizing the reconstruction loss, which measures the similarity between the decoder's output and the input data, along with a regularization term that aligns the latent space with the prior distribution. Once trained, the VAE employs an encoder-decoder framework to learn a latent variable (z) from the input data and reconstruct its dimensions. The primary objective is to minimize the reconstruction error, ensuring that the reconstructed data closely resembles the original input, ideally approaching zero reconstruction loss. This encoder performs as a feature extractor, reducing the input data dimensionalities to obtain a lower-dimension demonstration.

This encoder estimates the posterior inference $(z|x)$, while the decoder estimates the likelihood probability $p\varphi(x|z)$. Here, θ and φ represent the encoding and decoding parameters, respectively.

The loss function plays a critical role in VAE training. The maximal probability learning of parameters is stated as:

$$\log p_\varphi(x') = D_{KL}[q_\theta(z|x) \| p_\varphi(x)] + \mathcal{L}(\theta, \varphi; x) \tag{8}$$

where $D_{KL}[.]$ signifies the divergence of KL, and $\mathcal{L}$ refers to the probability of the parameters of encoding and decoding θ and φ. The loss function can be expressed as follows:

$$\mathcal{L}(\theta, \varphi) = \mathbb{E}z \sim q\theta(z|x)[\log p(x'|z)] - D_{KL}(q_\theta(z|x) \| p_\varphi(z)) \tag{9}$$

The main aim is to build an effective VAE technique that reduces reconstruction loss. These loss terms enable the decoder to learn the reconstruction of data. Furthermore, the term regularization, which depends on the divergence of KL, encourages differences between $q_\theta(z|x)$, the encoder distribution, and $p_\varphi(z)$, prior distribution. The optimization process includes upgrading the decoder and encoder parameters over gradient descent, most importantly to a well-organized hidden space and the capability to make new data models from $\sim p_\varphi(z)$.

Assuming that $p_\varphi(z) = \mathcal{N}(z; 0, I)$, the distribution of encoder $q_\theta(z|x)$ captures the multivariate Gaussian form, considered by a standard deviation, σ, and a mean, μ. The representation of latent space, z, is attained over a $g\varphi$ deterministic function using parameter φ, performed using an auxiliary noise variable ε and the input data x derived from $p(\varepsilon)$:

$$z = g_\varphi(x, \varepsilon) = \mu + \sigma \odot \varepsilon \tag{10}$$

where ⊙ signifies element-to-element multiplication. Therefore, the error of reconstruction can be stated as:

$$\mathcal{L}(\theta, \varphi, x) = \frac{1}{2}\sum_i (1 + \log((\sigma_i)^2) - (\mu_i)^2 - (\sigma_i)^2) + \frac{1}{L}\sum_{l=1}^{L} \log\left(p_\varphi\left(x|z^{(l)}\right)\right). \tag{11}$$

The VAE parameters' iterative optimizer is achieved over a gradient descent model.

This approach enhances attention by minimizing outside information requirements (source to target) and considering the interior input data correlation. An important feature of self-attention is its flexibility in terms of use in certain layers that signify a series of data such as a time sequence, which increases the interior learning of input structures by concentrating on the association between components of similar input (series). This method creates a novel depiction of the feature space over a certain weight of feature extraction with just a single data series as input in comparison with the attention mechanism. The self-attention data handling begins by calculating the weights (named the score) among data points in locations j and i for a specified input data series $\mathcal{X}$, as follows:

$$\varepsilon_{ij} = \frac{(W_a\mathcal{X}_i)^T(W_a\mathcal{X}_j)}{\sqrt{d}} \tag{12}$$

Here, W_a denotes the weighted matrix of the self-attention approach calculated during the training, and d represents a size of $(W_a\mathcal{X}_i)$. The separation by d creates quicker convergence. Weight normalization ε_{ij} is carried out to characterize the weights as a likelihood (the totality of all weight values equivalent to one), with a transformation of $softmax$:

$$\mathcal{A}_{ij} = softmax\left(\varepsilon_{ij}\right) = \frac{\exp\left(\varepsilon_{ij}\right)}{\sum_j \exp\left(\varepsilon_{ij}\right)} \tag{13}$$

The last self-attention method output is stated as:

$$O_i = \sum_{j=1}^{n} \mathcal{A}_{ij}\left(W_a\mathcal{X}_i\right) \tag{14}$$

This output efficiently improves the quality of the feature extraction and more precisely defines the interior connection of the input features. Fig. 3 depicts the structure of the SA-VAE model.

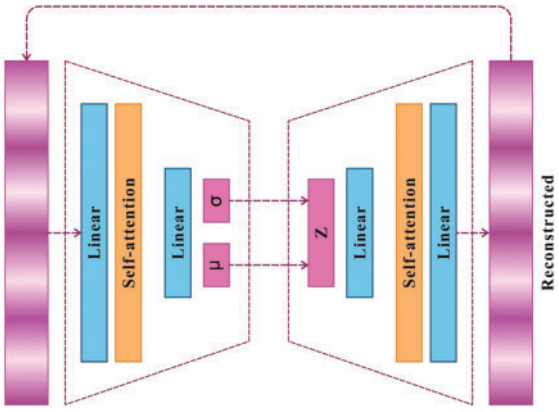

Figure 3: Structure of SA-VAE

3.4 Stage IV: CSA-Based Parameter Optimizer

In the final stage, the CSA is applied for the hyperparameter fine-tuning process in the SA-VAE model. The newly introduced meta-heuristic CSA model is based on the foraging behavior of chameleons [34]. This model efficiently searches for optimum solutions using a three-phase process that follows the hunting behavior of chameleons, including the phases of prey search, target detection, and eye rotation. The CSA model starts by initializing solutions with a random uniform distribution, as presented in Eq. (15).

$$X_j = lb_j + rand \times \left(ub_j - lb_j\right) \tag{15}$$

Here, lb_j and ub_j represent the lower and upper boundaries of the search area for the jth dimension, but $rand$ signifies randomly generated numbers that emulate a uniform distribution between zero and one.

Searching for prey

Chameleons utilize an accurately modeled location-upgrading method, defined by Eq. (16), to optimize their foraging strategy. During the search process. The locations in the search zone can be dynamically adjusted while searching for the best solutions.

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} + p_1 \times \left(P_{i,j}^{t} - G_j^{t}\right) \times r_2 + p_2 \times \left(G_j^{t} - X_{i,j}^{t}\right) \times r_1 & r_i \geq P_p \\ X_{i,j}^{r} + \mu \times \left(lb_j + r_3 \times \left(ub_j - lb_j\right)\right) \times sgn\left(rand - 0.5\right) & r_i < P_p \end{cases} \tag{16}$$

The $X_{i,j}^{t+1}$ *a* variable represents the location of the ith chameleon in the jth size for the following iteration; $X_{i,j}^{t}$ signifies the location of the ith individual and jth dimension in the present iteration; positive integers regulate exploration p_2 and p_1, including 1.50 and 0.25, respectively. $P_{i_j}^{t}$ refers to the optimum location for the ith chameleon, whereas G_j^{t} stands for the optimum location for each chameleon. r_1, r_2, and r_3 are randomly generated numbers that are distributed uniformly between zero and one.

The function $sgn(rand{-}0.5)$ offers both 1 and −1 to choose the exploration direction. μ is in inverse proportion to the iteration counts and is defined using Eq. (17).

$$\mu = \gamma e^{(\alpha t/T)^{\beta}} \tag{17}$$

The T and t variables symbolize the maximum and current iterations, while α and β are controller parameters with values of l.0, 3.0, and 3.5, respectively.

The eye rotation of chameleons

Chameleons have the unexpected ability to identify prey by rotating their eyes, providing them with a 360° observation of their searching region. In the algorithm, the eye rotation stage is a typical optical multiplexer that includes four stages to effectively identify the position of prey:

- Reference Frame Creation: The initial positions of the chameleons are initialized relative to the source, establishing a standard frame of reference for the search process.
- Calculation of Rotation Matrix: The model computes a rotation matrix to convert the coordinates of prey to make them parallel with the present spatial orientation of the chameleons.
- Transformation of Position: The locations are updated using the rotation matrix, enabling the exploration of various orientations within the search region.
- Repositioning to Original Coordinates: The converted locations are reverted to their original positions.

Eq. (18) briefly signifies the mathematical formulation for this updated position method using the eye rotation approach.

$$X_i^{t+1} = XR_i^t + XC_i^t \tag{18}$$

The X_i^{t+1}variable denotes the upgraded locations of the ith chameleon for the subsequent iteration. XR_i^t in Eq. (19) represents the rotated middle locations during the searching space.

$$XR_i^t = m \times XT_i^t \tag{19}$$

Here, m characterizes a rotation matrix. For every chameleon, the coordinates of its midpoint are represented in XT_i^t. To regulate the XT_i^t and m, Eqs. (20) and (21) are applied, respectively.

$$XT_i^t = X_i^t - XC_i^t \tag{20}$$

where the position of the ith chameleon for the current iteration is denoted by X_i^r, and the midpoint of the chameleon for the present location is signified by XC_i^t.

$$m = R\left(\theta, z_1, z_2\right) \tag{21}$$

Here, z_1 and z_2 are orthogonal vectors with nth dimensions, θ is the rotation angle, and the rotation matrices with a specific axis are denoted by R. Eqs. (22) and (23) are applied to calculate the values of R andθ, respectively. The rotation matrices regarding the x and y axes are denoted by R_x and R_y, respectively.

$$\theta = rand \times sgn(rand - 0.5) \times 180\hat{}\circ \tag{22}$$

$$R_\chi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}, R_y = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \tag{23}$$

Hunting prey

In this stage, chameleons hunt by spreading their tongues out toward the prey, achieving a distance of double their own height. This stage represents the exploitation of the search space, and the locations of chameleons are upgraded based on Eq. (24).

$$X_{i,j}^{t+1} = X_{i,j}^r + \frac{V_{i,j}^r - V_{i,j}^{t-1}}{2a} \tag{24}$$

Here, the chameleon value that is transformed for the subsequent iteration is denoted by $X_{i,j}^{t+1}$; $V_{i,j}^t$ stands for the ith chameleon velocity within the present iteration for the jth size; and the rate of acceleration of the projection of the chameleon's tongue is denoted by a. The processing time reaches a maximum of 2590 ms per second, indicating the system's peak computational load. The variable value a can be decided by using Eq. (25).

$$a = 2590 \times \left(1 - e^{-\log(t)}\right) \tag{25}$$

The values of velocity for the chameleon within the present iteration are decided using Eq. (26).

$$V_{i,j}^{t+1} = \omega V_{i,j}^r + c_1 \times \left(G_j^r - X_{i,j}^r\right) \times r_1 + c_2 \times \left(P_{i,j}^r - X_{i,j}^r\right) \times r_2 \tag{26}$$

where c_1 and c_2 are positive constants that define the influence of G_j^t and $P_{i,j}^t$ on the present chameleon; ω signifies the inertia weight, which reduces linearly for the iteration count upsurges; and parameter ρ controls the exploitation level, with its value set to a constant of one.

$$\omega = (1 - t/T)^{\left(\rho\sqrt{\frac{t}{T}}\right)} \tag{27}$$

The CSA model simulates the foraging behavior of chameleons using Eqs. (16), (18), and (24). These equations help the algorithm to select the best solutions for the optimization problem. The CSA develops an FF to achieve enhanced classifier execution. It identifies a positive integer to indicate the best executions of the candidate solutions. In this research, the reduction of the classification error rate can be examined using the FF, as specified in Eq. (28).

$$fitness\,(x_i) = ClassifierErrorRate\,(x_i)$$

$$= \frac{no.\,of\,misclassified\,samples}{Total\,no.\,of\,samples} * 100 \tag{28}$$

4 Experimental Results and Analysis

In this section, the validation of the PEFLID-CSAAI method is shown using the BoT-IoT database [35]. The BoT-IoT Binary database consists of 2056 samples under two class labels, as displayed in Table 2.

Table 2: Details of BoT-IoT Binary database

BoT-IoT Binary database	
Class	**No. of instances**
"Attack"	1579
"Normal"	477
Total samples	**2056**

Fig. 4 shows the classification of the PEFLID-CSAAI algorithm using the BoT-IoT Binary database. Fig. 4a,b displays the confusion matrices for the precise classification and recognition of two classes on a 70:30 TRAP/TESP. Fig. 4c shows the study of PR, representing superior execution over two class labels. Finally, Fig. 4d shows the rate of ROC, signifying effectual values with greater outcomes of ROC for different classes.

Note: Training Accuracy Percentage (TRAP): The proportion of correctly identified instances during the training phase, reflecting how well the model is learning from the training dataset.

Testing Accuracy Percentage (TESP): The proportion of correctly identified instances during the testing phase, indicating the model's performance and generalisation on unseen data.

In Table 3 and Fig. 5, the attack recognition of the PEFLID-CSAAI technique is validated using the BoT-IoT Binary database. The findings show that the PEFLID-CSAAI model accurately distinguishes each sample. At 70% TRAP, the PEFLID-CSAAI algorithm offers an average accuracy ($accu_y$) of 96.74%, precision ($prec_n$) of 98.44%, recall ($reca_l$) of 96.74%, F1 score ($F1_{score}$) of 97.56%, and area under the curve score (AUC_{score}) of 96.74%. Moreover, at 30% TESP, the PEFLID-CSAAI methodology exhibits an average $accu_y$ of 97.38%, $prec_n$ of 97.89%, $reca_l$ of 97.38%, $F1_{score}$ of 97.63%, and AUC_{score} of 97.38%.

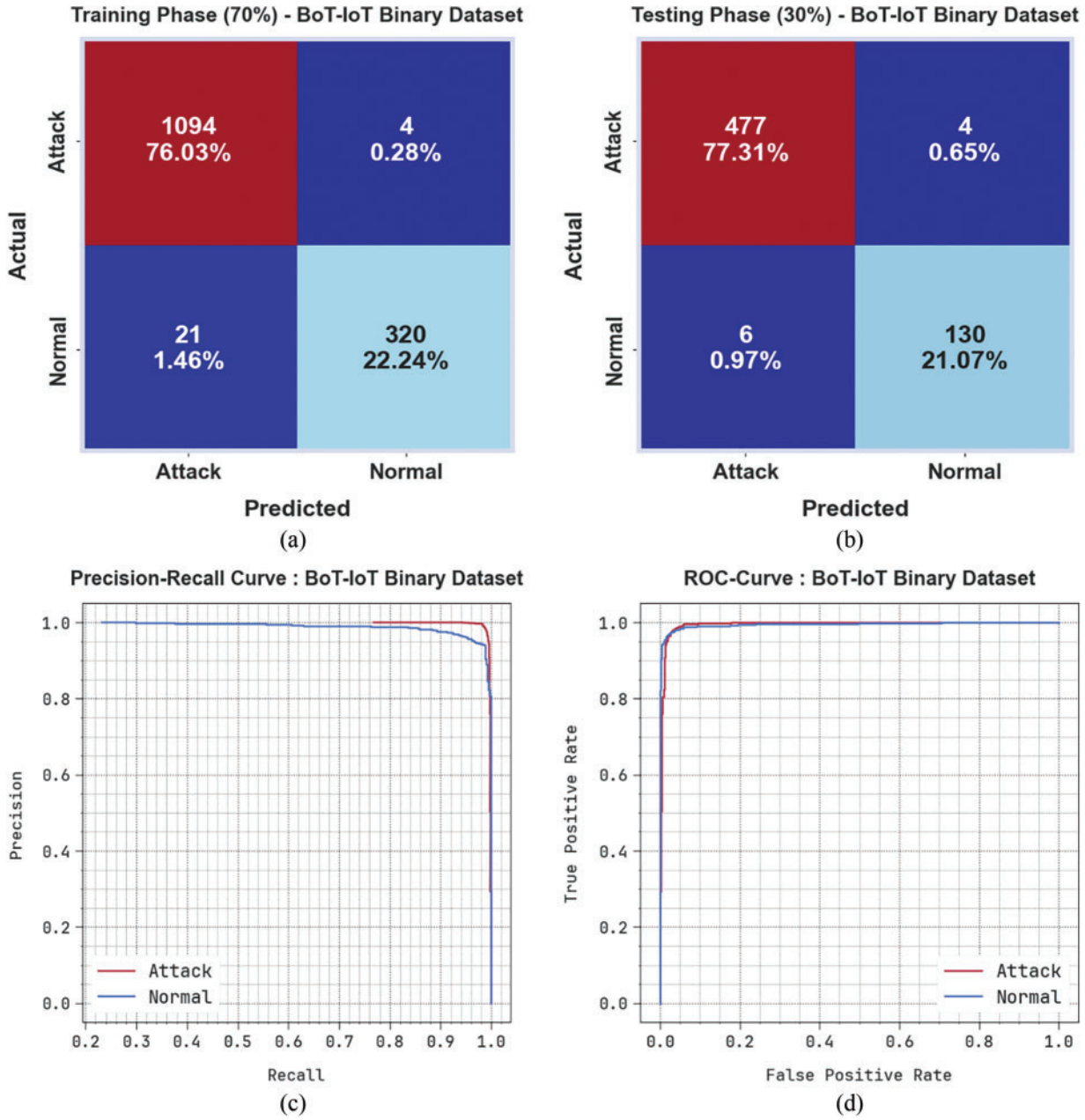

Figure 4: BoT-IoT Binary database: (a, b) confusion matrices and (c, d) PR and ROC curves

Table 3: Attack detection of PEFLID-CSAAI technique using BoT-IoT Binary database

Class	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$	AUC_{score}
			TRAP (70%)		
Attack	99.64	98.12	99.64	98.87	96.74
Normal	93.84	98.77	93.84	96.24	96.74
Average	**96.74**	**98.44**	**96.74**	**97.56**	**96.74**

(Continued)

Table 3 (continued)

Class	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$	AUC_{score}
			TESP (30%)		
Attack	99.17	98.76	99.17	98.96	97.38
Normal	95.59	97.01	95.59	96.30	97.38
Average	**97.38**	**97.89**	**97.38**	**97.63**	**97.38**

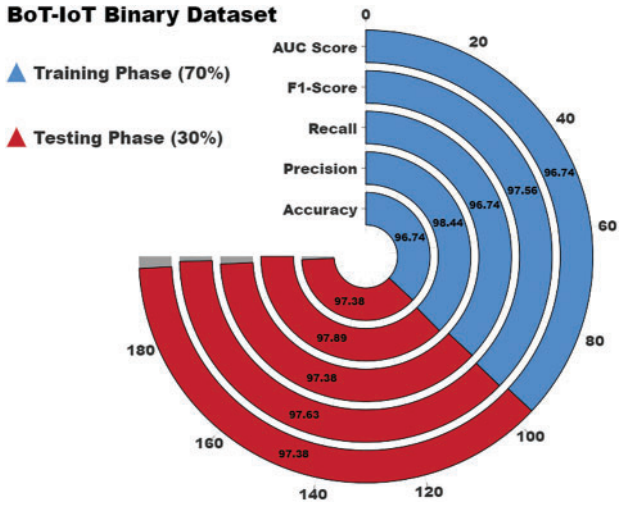

Figure 5: Average of PEFLID-CSAAI technique using BoT-IoT Binary database

In Fig. 6, the training $accu_y$ (TRAAC) and validation $accu_y$ (VLAAC) of the PEFLID-CSAAI technique using the BoT-IoT Binary database are exhibited. The rate of $accu_y$ is estimated for 0–50 epoch counts. The figure shows that the TRAAC and VLAAC values demonstrate an increasing trend, indicating that the PEFLID-CSAAI method is capable of greater execution over various iterations. Moreover, the TRAAC and VLAAC values remain closer over the epochs, highlighting the lower minimum overfitting and superior execution of the PEFLID-CSAAI methodology and ensuring the consistent prediction of hidden samples.

Fig. 7 presents the TRA loss (TRALS) and VLA loss (VLALS) graphs of the PEFLID-CSAAI model when using the BoT-IoT Binary database. The rate of loss is estimated for 0–50 epoch counts. The TRALS and VLALS values demonstrate a lower trend, indicating the ability of the PEFLID-CSAAI algorithm to correspond to a trade-off between generalization and data fitting. Further, the constant decrease in the loss rate suggests the ability of the PEFLID-CSAAI methodology to better execute and fine-tune the prediction outcomes over time.

An execution study of the PEFLID-CSAAI model was conducted using the BoT-IoT multiclass database. This database consists of 2056 instances under five class labels, as indicated in Table 4.

Fig. 8 presents the classification study of the PEFLID-CSAAI system using the BoT-IoT multiclass database. Fig. 8a,b displays the confusion matrices with the precise classification and recognition of five classes on a 70:30 TRAP/TESP dataset. Fig. 8c shows the PR study, signifying better performance across eight class labels. Finally, Fig. 8d illustrates the ROC study, exhibiting efficient values with greater ROC values for separate classes.

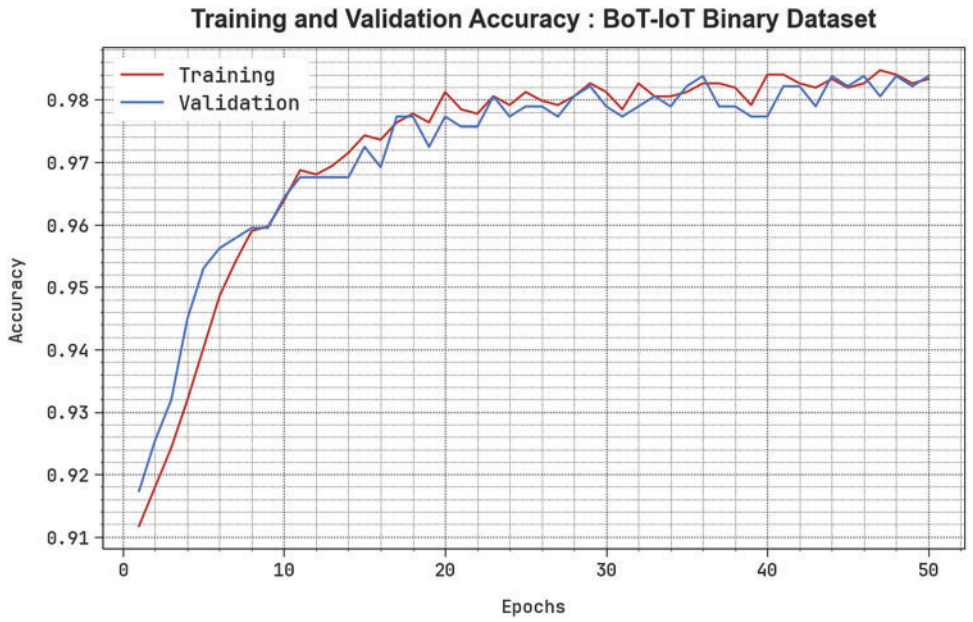

Figure 6: Accuracy curve of PEFLID-CSAAI technique when using BoT-IoT Binary database

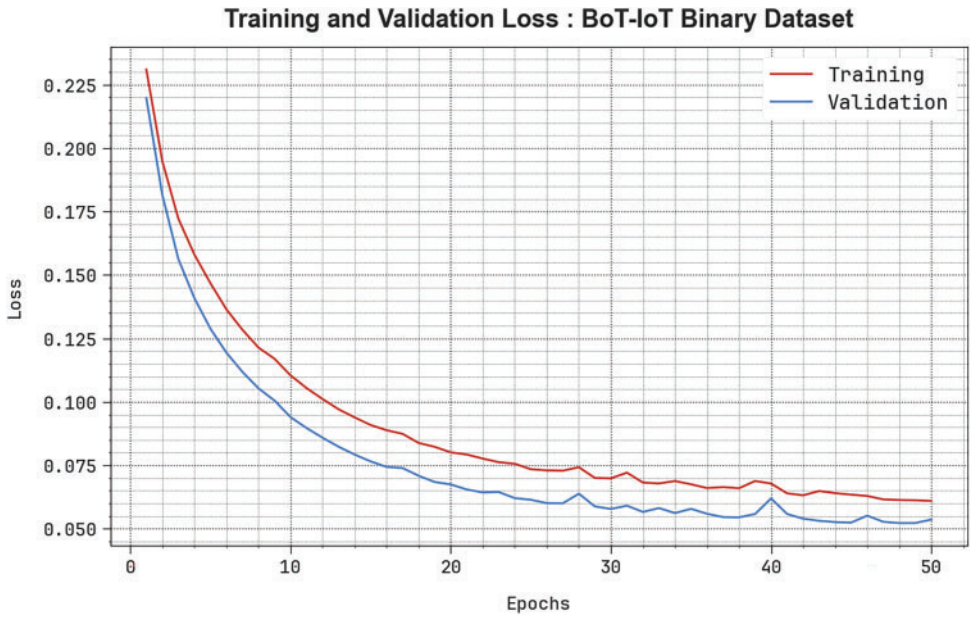

Figure 7: Loss curve of PEFLID-CSAAI technique when using BoT-IoT Binary database

Table 4: Details of BoT-IoT Multiclass database

BoT-IoT Multiclass database	
Classes	**No. of instances**
"DDoS"	500
"DoS"	500

(Continued)

Table 4 (continued)

BoT-IoT Multiclass database	
"Recon"	500
"Theft"	79
"Normal"	477
Total instances	**2056**

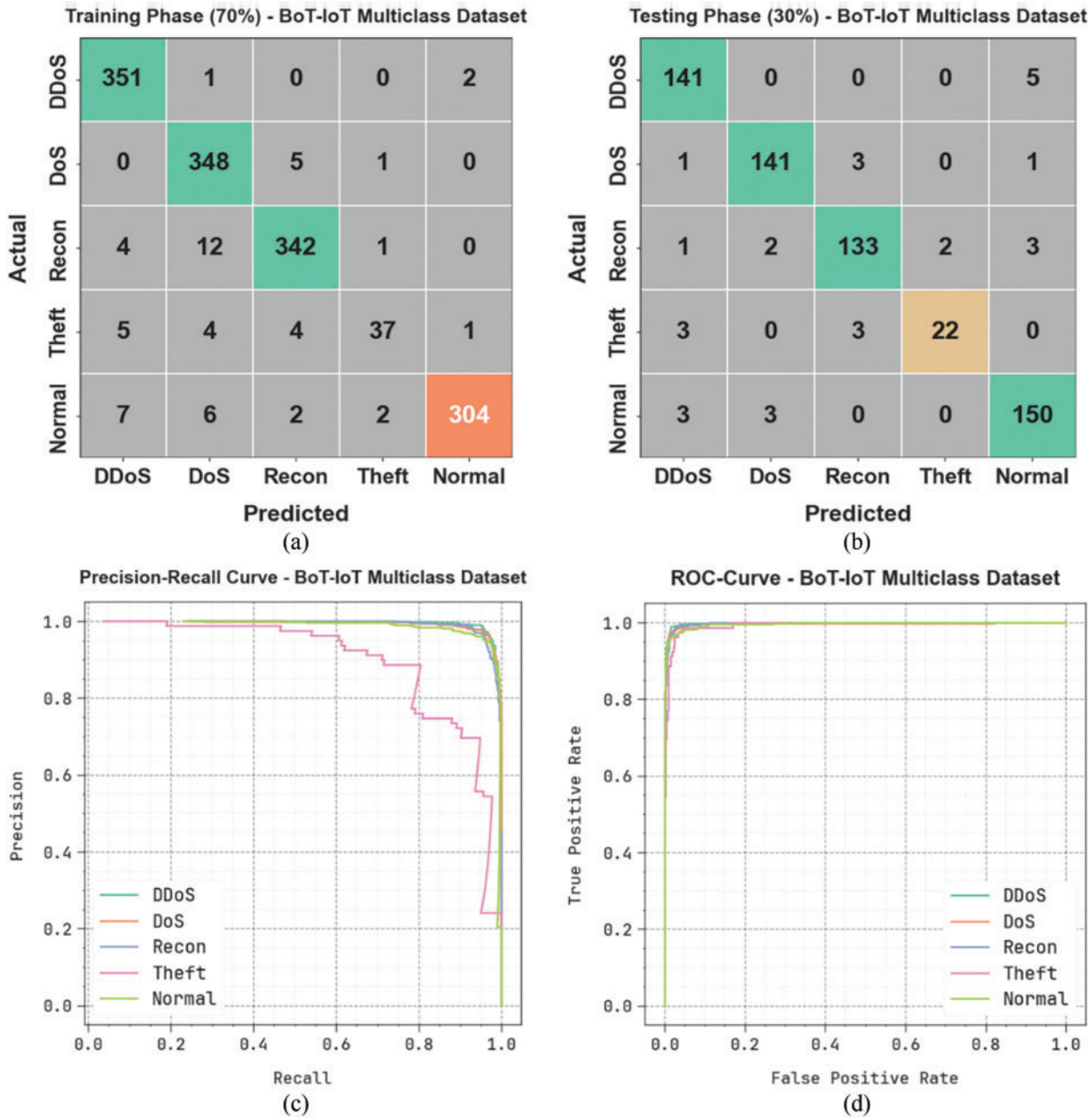

Figure 8: BoT-IoT Multiclass database: (a, b) confusion matrices and (c, d) PR and ROC curves

Table 5 and Fig. 9 present the attack recognition values for the PEFLID-CSAAI method when using the BoT-IoT Multiclass database. The findings show that the PEFLID-CSAAI algorithm appropriately distinguished eight samples. For 70% TRAP, the PEFLID-CSAAI methodology provides an average $accu_y$ of 98.42%, $prec_n$ of 95.12%, $reca_l$ of 92.00%, $F1_{score}$ of 93.34%, and AUC_{score} of 95.48%. Furthermore, for 30% TESP, the PEFLID-CSAAI approach offers an average $accu_y$ of 98.06%, $prec_n$ of 94.58%, $reca_l$ of 92.44%, $F1_{score}$ of 93.40%, and AUC_{score} of 95.59%.

Table 5: Attack detection of PEFLID-CSAAI technique when using BoT-IoT Multiclass database

Class	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$	AUC_{score}
TRAP (70%)					
DDoS	98.68	95.64	99.15	97.36	98.84
DoS	97.98	93.80	98.31	96.00	98.09
Recon	98.05	96.88	95.26	96.07	97.12
Theft	98.75	90.24	72.55	80.43	86.13
Normal	98.61	99.02	94.70	96.82	97.22
Average	**98.42**	**95.12**	**92.00**	**93.34**	**95.48**
TESP (30%)					
DDoS	97.89	94.63	96.58	95.59	97.44
DoS	98.38	96.58	96.58	96.58	97.76
Recon	97.73	95.68	94.33	95.00	96.53
Theft	98.70	91.67	78.57	84.62	89.12
Normal	97.57	94.34	96.15	95.24	97.10
Average	**98.06**	**94.58**	**92.44**	**93.40**	**95.59**

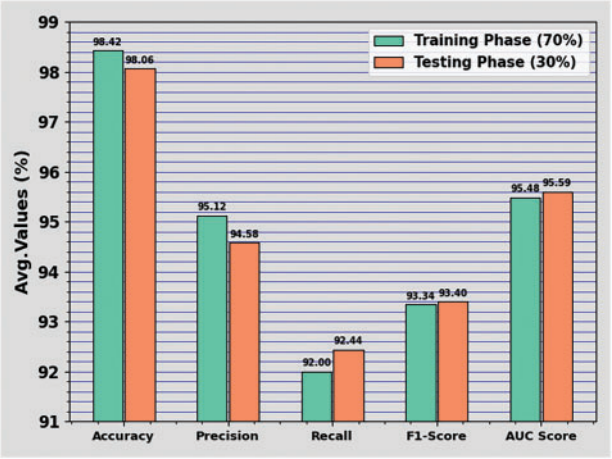

Figure 9: Average of PEFLID-CSAAI technique when using BoT-IoT Multiclass database

Fig. 10 presents the TRAAC and VLAAC values of the PEFLID-CSAAI technique when using the BoT-IoT Multiclass database. The rate of $accu_y$ is estimated for 0–50 epoch counts. The figure shows that the TRAAC and VLAAC values exhibit an increasing trend, indicating the superior execution of the PEFLID-CSAAI algorithm over various iterations. Furthermore, the TRAAC and VLAAC remain closer over the

epochs, highlighting the lower minimum overfitting and improved performance of the PEFLID-CSAAI system and its ability to make consistent predictions of hidden samples.

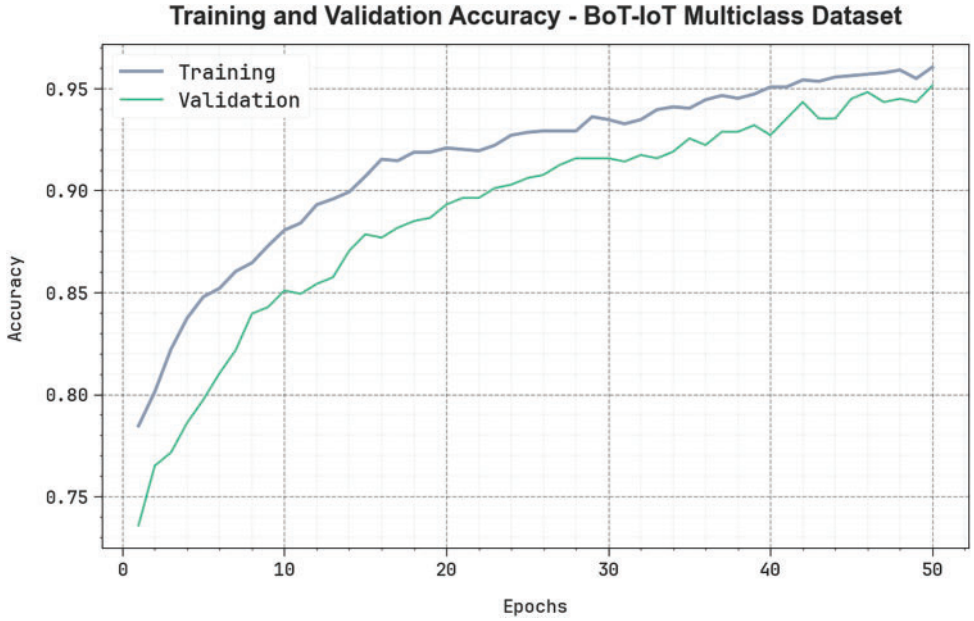

Figure 10: Accuracy curve of PEFLID-CSAAI technique when using BoT-IoT Multiclass database

Fig. 11 presents the TRALS and VLALS loss graph of the PEFLID-CSAAI technique when using the BoT-IoT Multiclass database. The loss rate is estimated for 0–50 epoch counts. The figure shows that the TRALS and VLALS values demonstrate a reducing trend, indicating the ability of the PEFLID-CSAAI system to balance a trade-off between generalization and data fitting. The constant reduction in the rate of loss also highlights the superior execution of the PEFLID-CSAAI algorithm and its ability to fine-tune the prediction values over time.

A comparison between the PEFLID-CSAAI methodology and current methods is displayed in Table 6 and Fig. 12 [35,36]. The experimental findings indicated that the PEFLID-CSAAI algorithm outperformed in terms of superior executions. The PEFLID-CSAAI system has greater accuracy at 98.42% while the Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM), Recurrent Neural Network (RNN), Convolutional Neural Network-Bidirectional Long Short-Term Memory (CNN-BiLSTM), Deep Neural Network (DNN), upport Vector Machine with Radial Basis Function (SVM-RBF), Intrusion Detection System for Internet of Things (IDS-IoT), Autoencoder-Multi-Layer Perceptron (AE-MLP), and Non-Federated Learning Privacy-Enhanced Federated Learning Intrusion Detection using Chameleon Swarm Algorithm and Artificial Intelligence (Non-FL-PEFLID-CSAAI) models have lower accuracy values at 93.98%, 89.86%, 93.12%, 94.19%, 95.16%, 97.40%, 98.09%, and 97.76%, respectively. In addition, for precision, the PEFLID-CSAAI methodology achieved 95.12% whereas the CNN-LSTM, RNN, CNN-BiLSTM, DNN, SVM-RBF, IDS-IoT, AE-MLP, and Non-FL-PEFLID-CSAAI systems had values of 92.26%, 94.05%, 94.17%, 90.21%, 90.80%, 94.91%, and 94.59%, respectively. Lastly, in terms of F1 score, the PEFLID-CSAAI algorithm achieved 93.34%, whereas the CNN-LSTM, RNN, CNN-BiLSTM, DNN, SVM-RBF, IDS-IoT, AE-MLP, and Non-FL-PEFLID-CSAAI models has lower scores of 85.59%, 87.55%, 91.25%, 86.70%, 92.00%, 94.53%, 94.13%, and 92.72%, respectively.

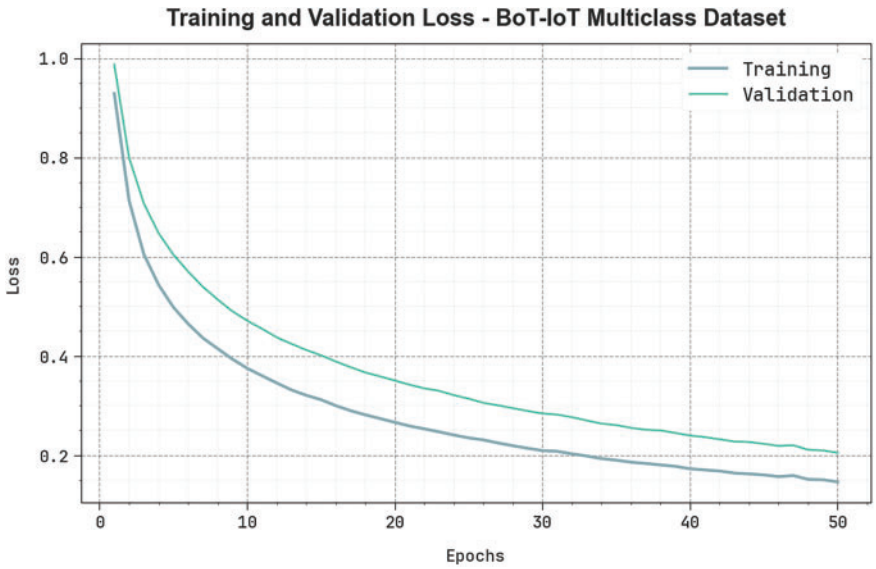

Figure 11: Loss curve of PEFLID-CSAAI technique when using BoT-IoT Multiclass database

Table 6: Comparative analysis of PEFLID-CSAAI technique against recent models

Methods	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{Score}$
CNN-LSTM	93.98	92.26	87.63	85.59
RNN Algorithm	89.86	94.05	89.52	87.55
CNN-BiLSTM	93.12	94.17	89.10	91.25
DNN algorithm	94.19	90.21	89.67	86.70
SVM-RBF classifier	95.16	90.80	90.69	92.00
IDS-IoT	97.40	94.80	90.90	94.53
AE-MLP	98.09	94.91	91.31	94.13
Non-FL-PEFLID-CSAAI	97.76	94.59	91.38	92.72
PEFLID-CSAAI	98.42	95.12	92.00	93.34

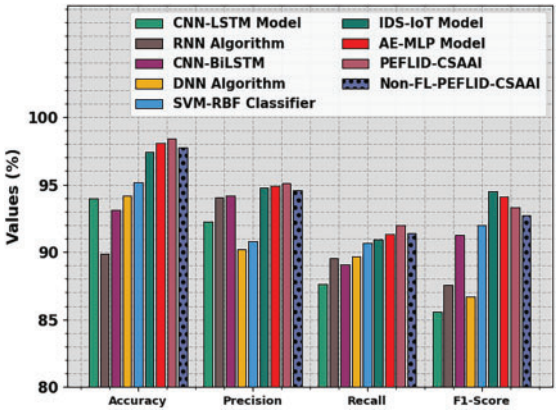

Figure 12: Comparative analysis of PEFLID-CSAAI system against recent models

4.1 Comparative Analysis of PEFLID-CSAAI against Benchmarked Schemes

These results in Table 7 indicate that PEFLID-CSAAI outperforms traditional deep learning and machine learning models, offering higher detection accuracy, improved generalization, and enhanced robustness in intrusion detection for IoT environments. The integration of federated learning (FL), Self-Attentive Variational Autoencoder (SA-VAE), and metaheuristic optimization (CSA and OOA) contributes to its superior efficiency, making it a promising solution for real-world cybersecurity applications.

Table 7: Comparative analysis of PEFLID-CSAAI against benchmarked schemes

Model	Accuracy	Precision	Recall	F1 score
CNN-LSTM	93.98%	92.26%	87.63%	85.59%
RNN	89.86%	94.05%	89.52%	87.55%
CNN-BiLSTM	93.12%	94.17%	89.10%	91.25%
SVM-RBF	95.16%	90.80%	90.69%	92.00%
PEFLID-CSAAI	**98.42%**	**95.12%**	**92.00%**	**93.34%**

5 Discussion

Significant security and privacy issues are associated with the growing use of Internet of Things (IoT)-based devices, particularly in healthcare systems where applications such as wearable medical technology, telemedicine, remote patient monitoring, and smart hospital systems are utilized [29,37]. These systems are vulnerable to cyberattacks because they gather and send extremely sensitive patient health data. For this reason, data privacy is crucial in these systems, and laws such as GDPR and HIPAA require stringent data protection procedures [38].

In utilizing the proposed PEFLID-CSAAI framework for IDSs in healthcare systems, we focused on detecting potential security breaches across interconnected medical devices and networks. For example, in healthcare settings, devices such as smart wearables, patient monitoring systems, and linked medical equipment routinely communicate sensitive data [39], leaving them vulnerable to cyberattacks including unauthorized access and data manipulation.

Using FL, each medical device or healthcare system node can train an intrusion detection model using its own network traffic and operational data without revealing sensitive patient or system information [40]. This ensures privacy while providing effective intrusion detection across the network. The OOA can be used to extract the most important features from network data that indicate intrusion attempts, whilst the SA-VAE can detect irregular patterns that indicate prospective intrusions. Finally, the CSA may adjust the hyperparameters to improve the model's performance across a range of healthcare equipment. By using this architecture, healthcare systems may proactively detect and respond to security threats, protecting the integrity and confidentiality of critical medical data while maintaining optimal network security [41].

5.1 Addressing Gaps in the Existing Literature

A major limitation in existing research is the lack of privacy-preserving techniques in intrusion detection for IoT networks. Traditional machine learning-based intrusion detection systems rely on centralized data collection, exposing sensitive information to security threats [36]. Federated learning has emerged as a promising approach to mitigate these privacy concerns by allowing distributed model training without sharing raw data, yet existing FL-based models fail to efficiently handle dynamic and heterogeneous IoT environments [42–44]. Several studies have explored privacy-preserving FL-based IDS solutions, such as

differential privacy-enhanced FL for industrial cyber–physical systems [45] and privacy-preserving FL for the Internet of Healthcare Things (IoHT) [26]. However, these approaches often lack robust anomaly detection mechanisms. To address this gap, Self-Attentive Variational Autoencoders (SA-VAEs) have been introduced to enhance anomaly detection accuracy by capturing complex attack patterns in real time, as demonstrated in multi-head attention-based VAEs for anomaly detection and dual variational autoencoder-based anomaly detection frameworks [30]. Additionally, the scalability and adaptability of FL-based IDSs can be improved using metaheuristic algorithms such as the Chameleon Swarm Algorithm (CSA) and Osprey Optimization Algorithm (OOA), which optimize feature selection and hyperparameter tuning to improve detection performance and computational efficiency [46]. The PEFLID-CSAAI framework integrates these cutting-edge techniques to overcome limitations in existing IDS models, providing a scalable, adaptive, and privacy-preserving solution for securing IoT environments [47].

5.2 Consideration for Dataset Selection and Generalizability

While the BoT-IoT dataset effectively represents a variety of IoT-based cyberattacks, additional datasets could further validate the generalization of the proposed model. Future research can explore datasets such as NSL-KDD and CICIDS2017, which provide broader network intrusion patterns and simulate real-world cybersecurity threats [48]. The TON_IoT dataset offers valuable insights by containing data from industrial IoT systems, enabling the assessment of model adaptability across critical infrastructures [49]. Additionally, the Edge-IIoT dataset serves as a suitable benchmark for evaluating the robustness of intrusion detection systems in smart industrial environments [50]. The BoT-IoT dataset was selected for this study due to its rich attack variability, covering a diverse range of cyberattacks, including Denial of Service (DoS), Distributed DoS (DDoS), reconnaissance attacks, and information theft [51]. Unlike NSL-KDD, which is designed for traditional network environments, BoT-IoT captures IoT-specific features, making it more suitable for IoT-oriented intrusion detection models. Furthermore, its diverse network traffic composition, including both benign and attack scenarios, makes it an ideal choice for FL-based IDS evaluation [52]. To support claims of generalizability, we propose incorporating a cross-dataset validation section in future research, assessing the performance of the proposed model across multiple datasets to ensure its effectiveness in detecting cyber threats across various IoT environments and enhancing its applicability in real-world scenarios [21].

5.3 Contribution of the Paper

This research introduces Privacy-Enhanced Federated Learning for Intrusion Detection (PEFLID-CSAAI), an innovative framework designed to enhance intrusion detection in decentralized environments while preserving data privacy. The proposed approach integrates federated learning to enable collaborative intrusion detection while safeguarding sensitive data, ensuring compliance with security regulations such as GDPR and HIPAA [36,42]. To improve anomaly detection accuracy, the Self-Attentive Variational Autoencoder (SA-VAE) is employed, effectively capturing complex attack patterns with greater precision [53]. Additionally, the framework utilizes the Chameleon Swarm Algorithm (CSA) and Osprey Optimization Algorithm (OOA) for optimized feature selection and hyperparameter tuning, enhancing both efficiency and adaptability across diverse IoT environments [46,54]. The experimental results demonstrate that PEFLID-CSAAI outperforms existing IDSs in terms of accuracy, resilience, and scalability, making it a viable solution for real-world cybersecurity applications [55].

6 Conclusion

This paper presents a novel Privacy-Enhanced Federated Learning Intrusion Detection System (PEFLID-CSAAI) that addresses the security and privacy challenges faced by IoT devices in the modern

digital landscape. By leveraging FL, the framework enables distributed training of intrusion detection models without sharing sensitive data, enhancing privacy preservation in highly distributed IoT environments. The integration of the Osprey Optimization Algorithm (OOA) for feature selection, the Self-Attentive Variational Autoencoder (SA-VAE) for anomaly detection, and the Chameleon Swarm Algorithm (CSA) for hyperparameter tuning further improves the system's efficiency and accuracy in detecting cyber threats. Through extensive experiments using the BoT-IoT dataset, the PEFLID-CSAAI model demonstrates superior performance over recent methods in terms of accuracy, precision, recall, and F1 score for both binary and multiclass attack detection. The proposed approach provides an effective and scalable solution to IoT security, ensuring that privacy and computational constraints are respected while maintaining high detection accuracy. Future work could explore the application of this model in real-world IoT systems, further enhancing its adaptability and robustness against evolving cyber threats.

However, this study is limited to the BoT-IoT dataset, which may affect its generalizability to other IoT environments. Additionally, the PEFLID-CSAAI method has not been evaluated in real-time IoT deployments. Future work should explore cross-dataset validation and real-time implementation to enhance its robustness and adaptability.

Acknowledgement: The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia, for funding this research work.

Funding Statement: This research was funded by the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia, under grant number NBU-FFR-2025-451-6.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Saad Alahmari and Abdulwhab Alkharashi; methodology, Saad Alahmari; software, Saad Alahmari; validation, Saad Alahmari and Abdulwhab Alkharashi; formal analysis, Saad Alahmari; investigation, Saad Alahmari; resources, Saad Alahmari; data curation, Saad Alahmari; writing—original draft preparation, Saad Alahmari; writing—review and editing, Saad Alahmari and Abdulwhab Alkharashi; visualization, Saad Alahmari; supervision, Saad Alahmari; project administration, Saad Alahmari; funding acquisition, Saad Alahmari. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article [35].

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Abdellatif AA, Mhaisen N, Mohamed A, Erbad A, Guizani M, Dawy Z, et al. Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data. Future Generat Comput Syst. 2022;128(1):406–19. doi:10.1016/j.future.2021.10.016.
2. Eskandari M, Janjua ZH, Vecchio M, Antonelli F. Passban IDS: an intelligent anomaly-based intrusion detection system for IoT edge devices. IEEE Int Thin J. 2020;7(8):6882–97. doi:10.1109/JIOT.2020.2970501.
3. Huong TT, Bac TP, Long DM, Luong TD, Dan NM, Thang BD, et al. Detecting cyberattacks using anomaly detection in industrial control systems: a federated learning approach. Comput Ind. 2021;132(7):103509. doi:10.1016/j.compind.2021.103509.
4. Li J, Lyu L, Liu X, Zhang X, Lyu X. FLEAM: a federated learning empowered architecture to mitigate DDoS in industrial IoT. IEEE Transact Indus Inform. 2021;18(6):4059–68. doi:10.1109/TII.2021.3088938.

5. Alajlan NN, Ibrahim DM. TinyML: enabling of inference deep learning models on ultra-low-power IoT edge devices for AI applications. Micromachines. 2022;13(6):851. doi:10.3390/mi13060851.
6. Yadav K, Gupta BB, Hsu C-H, Chui KT. Unsupervised federated learning based IoT intrusion detection. In: 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE); 2021; Kyoto, Japan: IEEE. p. 298–301.
7. Zhang T, He C, Ma T, Gao L, Ma M, Avestimehr S. Federated learning for internet of things. In: Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems; 2021.
8. Attota DC, Mothukuri V, Parizi RM, Pouriyeh S. An ensemble multi-view federated learning intrusion detection for IoT. IEEE Access. 2021;9:117734–45. doi:10.1109/ACCESS.2021.3107337.
9. Shahid O, Mothukuri V, Pouriyeh S, Parizi RM, Shahriar H. Detecting network attacks using federated learning for IoT devices. In: 2021 IEEE 29th International Conference on Network Protocols (ICNP); 2021; Dallas, TX, USA: IEEE. p. 1–6.
10. Man D, Zeng F, Yang W, Yu M, Lv J, Wang Y. Intelligent intrusion detection based on federated learning for edge-assisted internet of things. Secur Commun Netw. 2021;2021(1):9361348. doi:10.1155/2021/9361348.
11. Hamdi N. Federated learning-based intrusion detection system for Internet of Things. Inte J Inform Secur. 2023;22(6):1937–48. doi:10.1007/s10207-023-00727-6.
12. Li P, Wang H, Tian G, Fan Z. A cooperative intrusion detection system for the internet of things using convolutional neural networks and black hole optimization. Sensors. 2024;24(15):4766. doi:10.3390/s24154766.
13. Angelin JAB, Priyadharsini C. Deep learning based network based intrusion detection system in industrial internet of things. In: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT); 2024; Bengaluru, India: IEEE. p. 426–32.
14. Ullah F, Turab A, Ullah S, Cacciagrano D, Zhao Y. Enhanced network intrusion detection system for internet of things security using multimodal big data representation with transfer learning and game theory. Sensors. 2024;24(13):4152. doi:10.3390/s24134152.
15. Ghasemi H, Babaie S. A new intrusion detection system based on SVM-GWO algorithms for Internet of Things. Wirel Netw. 2024;30(4):1–13. doi:10.1007/s11276-023-03637-6.
16. Hanafi AV, Ghaffari A, Rezaei H, Valipour A, Arasteh B. Intrusion detection in Internet of things using improved binary golden jackal optimization algorithm and LSTM. Cluster Computing. 2024;27(3):2673–90. doi:10.1007/s10586-023-04102-x.
17. Javeed D, Saeed MS, Adil M, Kumar P, Jolfaei A. A federated learning-based zero trust intrusion detection system for Internet of Things. *Ad Hoc* Netw. 2024;162(6):103540. doi:10.1016/j.adhoc.2024.103540.
18. Gopi R, Sheeba R, Anguraj K, Chelladurai T, Alshahrani HM, Nemri N, et al. Intelligent intrusion detection system for industrial internet of things environment. Comput Syst Sci Eng. 2023;44(2):1567–82. doi:10.32604/csse.2023.025216.
19. Giraldo J, Sarkar E, Cardenas AA, Maniatakos M, Kantarcioglu M. Security and privacy in cyber-physical systems: a survey of surveys. IEEE Des Test. 2017;34(4):7–17. doi:10.1109/MDAT.2017.2709310.
20. Khan LU, Saad W, Han Z, Hossain E, Hong CS. Federated learning for internet of things: recent advances, taxonomy, and open challenges. IEEE Commun Surv Tutor. 2021;23(3):1759–99. doi:10.1109/COMST.2021.3090430.
21. Kim H, Park J, Bennis M, Kim S-L. Blockchained on-device federated learning. IEEE Commun Lett. 2019;24(6):1279–83. doi:10.1109/LCOMM.2019.2921755.
22. Roy S, Li J, Bai Y. Federated learning-based intrusion detection system for IoT environments with locally adapted model. In: 2023 IEEE 10th International Conference on Cyber Security and Cloud Computing (CSCloud)/2023 IEEE 9th International Conference on Edge Computing and Scalable Cloud (EdgeCom); 2023; Xiangtan, China: IEEE. p. 203–9.
23. Khan MA, Khan MA, Jan SU, Ahmad J, Jamal SS, Shah AA, et al. A deep learning-based intrusion detection system for MQTT enabled IoT. Sensors. 2021;21(21):7016. doi:10.3390/s21217016.
24. Li Q, Wu X, Cao Z, Ling J. Anomaly detection of IoT traffic based on LSTM and attention mechanism. In: Proceedings of the 2023 15th International Conference on Machine Learning and Computing; 2023; New York, NY, USA. p. 457–63.

25. Dai W, Li X, Ji W, He S. Network intrusion detection method based on CNN, BiLSTM, and attention mechanism. IEEE Access. 2024;12:53099–111. doi:10.1109/ACCESS.2024.3384528.
26. Bhavsar M, Bekele Y, Roy K, Kelly J, Limbrick D. FL-IDS: federated learning-based intrusion detection system using edge devices for transportation IoT. IEEE Access. 2024;12(12):52215–26. doi:10.1109/ACCESS.2024.3386631.
27. Mustafa O, Ali K, Naqash T. C-RADAR: a centralized deep learning system for intrusion detection in software defined networks. In: 2023 International Conference on Communication, Computing and Digital Systems (C-CODE); 2023; Islamabad, Pakistan: IEEE. p. 1–6.
28. Zhao X, Fok KW, Thing VL. Enhancing network intrusion detection performance using generative adversarial networks. arXiv:240407464. 2024.
29. Mathur P, Choudhary A, Kunndra C, Pareek K, Choudhary G. SVM-RF: a hybrid machine learning model for detection of malicious network traffic and files. In: International Conference on Cryptology & Network Security with Machine Learning; 2022; Singapore: Springer.
30. Pitsiorlas I, Arvanitakis G, Kountouris M. Trustworthy intrusion detection: confidence estimation using latent space. In: 2024 22nd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt); 2024; Seoul, Republic of Korea: IEEE. p. 92–8.
31. Larriva-Novo X, Villagrá VA, Vega-Barbas M, Rivera D, Sanz Rodrigo M. An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets. Sensors. 2021;21(2):656. doi:10.3390/s21020656.
32. Zhou L, Liu X, Tian R, Wang W, Jin G. A modified osprey optimization algorithm for solving global optimization and engineering optimization design problems. Symmetry. 2024;16(9):1173. doi:10.3390/sym16091173.
33. Zavrak S, Iskefiyeli M. Anomaly-based intrusion detection from network flow features using variational autoencoder. IEEE Access. 2020;8:108346–58. doi:10.1109/ACCESS.2020.3001350.
34. Mosbah AS, Mostafa RR, Barakat SI. Automatic detection and classification of grape leaf diseases based on deep learning and enhanced chameleon swarm algorithm. Mansoura J Comput Inform Sci. 2024;18(1):1–21. doi:10.21608/mjcis.2024.291256.1004.
35. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Fut Generat Comput Syst. 2019;100(7):779–96. doi:10.1016/j.future.2019.05.041.
36. Olanrewaju-George B, Pranggono B. Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models. Cyber Secur Applicat. 2025;3(46):100068. doi:10.1016/j.csa.2024.100068.
37. Newaz AI, Sikder AK, Rahman MA, Uluagac AS. A survey on security and privacy issues in modern healthcare systems: attacks and defenses. ACM Transact Comput Healthcare. 2021;2(3):1–44. doi:10.1145/3453176.
38. Shah W. Preserving privacy and security: a comparative study of health data regulations-GDPR vs. HIPAA. Int J Res Appl Sci Eng Technol. 2023;11(8):2189–99. doi:10.22214/ijraset.2023.55551.
39. Antunes RS, André da Costa C, Küderle A, Yari IA, Eskofier B. Federated learning for healthcare: systematic review and architecture proposal. ACM Transact Intell Syst Technol. 2022;13(4):1–23. doi:10.1145/3501813.
40. Gu X, Sabrina F, Fan Z, Sohail S. A review of privacy enhancement methods for federated learning in healthcare systems. Int J Environ Res Public Health. 2023;20(15):6539. doi:10.3390/ijerph20156539.
41. Coelho KK, Nogueira M, Vieira AB, Silva EF, Nacif JAM. A survey on federated learning for security and privacy in healthcare applications. Comput Commun. 2023;207(1):113–27. doi:10.1016/j.comcom.2023.05.012.
42. Dauterman E, Corrigan-Gibbs H, Mazières D, Boneh D, Rizzo D. True2F: backdoor-resistant authentication tokens. In: 2019 IEEE Symposium on Security and Privacy (SP); 2019; San Francisco, CA, USA: IEEE. p. 398–416.
43. Gharib M, Mohammadi B, Dastgerdi SH, Sabokrou M. Autoids: auto-encoder based method for intrusion detection system. arXiv:191103306. 2019.
44. Liberti F, Berardi D, Martini B. Federated learning in dynamic and heterogeneous environments: advantages, performances, and privacy problems. Appl Sci. 2024;14(18):8490. doi:10.3390/app14188490.
45. Li B, Wu Y, Song J, Lu R, Li T, Zhao L. DeepFed: federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Transact Indust Inform. 2020;17(8):5615–24. doi:10.1109/TII.2020.3023430.

46. Braik MS. Chameleon Swarm Algorithm: a bio-inspired optimizer for solving engineering design problems. Expert Syst Appl. 2021;174(1):114685. doi:10.1016/j.eswa.2021.114685.
47. Harahsheh K, Alzaqebah M, Chen C-H. An enhanced real-time intrusion detection framework using federated transfer learning in large-scale IoT networks. Int J Adv Comput Sci Applicat. 2024;15(12). doi:10.14569/issn.2156-5570.
48. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. Comput Secur. 2019;86(1):147–67. doi:10.1016/j.cose.2019.06.005.
49. Moustafa N. A new distributed architecture for evaluating AI-based security systems at the edge: network TON_IoT datasets. Sustain Cities Soc. 2021;72:102994. doi:10.1016/j.scs.2021.102994.
50. Ferrag MA, Friha O, Hamouda D, Maglaras L, Janicke H. Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access. 2022;10:40281–306. doi:10.1109/ACCESS.2022.3165809.
51. Anwar RW, Zainal A, Outay F, Yasar A, Iqbal S. BTEM: belief based trust evaluation mechanism for wireless sensor networks. Future Generat Comput Syst. 2019;96(1):605–16. doi:10.1016/j.future.2019.02.004.
52. Nguyen DC, Ding M, Pathirana PN, Seneviratne A, Li J, Poor HV. Federated learning for internet of things: a comprehensive survey. IEEE Commun Surv Tutor. 2021;23(3):1622–58. doi:10.1109/COMST.2021.3075439.
53. Kingma DP. Auto-encoding variational bayes. arXiv:1312.6114. 2013.
54. Dehghani M, Trojovský P. Osprey optimization algorithm: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems. Front Mech Eng. 2023;8:1126450. doi:10.3389/fmech.2022.1126450.
55. Rahman S, Pal S, Mittal S, Chawla T, Karmakar C. SYN-GAN: a robust intrusion detection system using GAN-based synthetic data for IoT security. Int Things. 2024;26(7):101212. doi:10.1016/j.iot.2024.101212.