I.J. Information Technology and Computer Science, **2026, 4, 105-127**
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2026.04.07

Federated Learning-Enabled Intrusion Detection with Bio-Inspired Feature Optimization and Hybrid Deep Neural Classifier

S. Shiva Prakash*
Department of CSE, School of Computing, Mohan Babu University, Tirupati, AP, India
E-mail: shivasthaneekam@gmail.com
ORCID iD: https://orcid.org/0009-0000-9886-2917
*Corresponding author

M. Sunil Kumar
Department of CSE, School of Computing, Mohan Babu University, Tirupati, AP, India
E-mail: sunilmalchi1@gmail.com
ORCID iD: https://orcid.org/ 0000-0002-1439-2573

Received: March 10, 2026; Revised: April 26, 2026; Accepted: June 8, 2026; Published: August 8, 2026

Abstract: The growth of Internet of Things (IoT) networks has drastically expanded the attack surface, requiring intrusion detection systems (IDS) to ensure accuracy and privacy protection. To overcome these obstacles, we introduce a federated learning (FL)-based IDS that incorporates state-of-the-art preprocessing, smart feature optimization, and a new classification paradigm. During preprocessing, raw traffic data undergoes rigorous scrubbing, normalization via scaling, and label encoding to maintain consistency and reduce noise in heterogeneous local datasets. For feature selection, the Hybrid Emperor Penguin–Quokka Swarm Optimization (HEPQSO) approach is utilized which balances exploitation and exploration to find the most discriminative features while addressing the dimensionality problem. These features are then utilized by a deep hybrid classifier where the Spike Gated Linear Unit (SGLU) facilitates non-linear representation learning, and a Vision Transformer-Temporal Convolutional Network (ViT–TCN) hybrid discovers both global spatial relationships and local temporal dynamics of intrusion patterns. Experimental analyses performed using benchmark intrusion detection datasets show that the system consistently outperforms baseline models, with an accuracy of 97.88%, precision of 96.16%, recall of 97.54%, F1-score of 97.39%, specificity of 97.62%, and MCC of 97.04%, thus demonstrating its efficacy in secure IoT environments. This combination of state-of-the-art preprocessing, hybrid feature selection, and deep federated classification forms a robust IDS that can tackle the changing landscape of cyber intrusions.

Index Terms: Intrusion Detection Systems, Federated Learning, IoT Security, Hybrid Feature Selection, Spike Gated Linear Unit, Temporal Convolutional Network.

1. Introduction

In the age of digitalization, the omnipresence of IoT devices is rapidly altering our daily lives. These devices capture and upload personal data around-the-clock, generating vast amounts of data for each individual daily. Numerous applications, including intelligent industrial control systems and smart cities, have emerged due to integration with big data and edge computing. IoT devices have become an integral part of everyday life, with widespread adoption across various domains [1]. Examples include the invisible infrastructure of smart cities, such as pedestrian and road sensors, and home gadgets like entertainment systems and smart utilities. Due to the open nature of IoT network environments and the limited computational resources of IoT devices, ensuring robust security and data privacy remains a major challenge. The lack of adequate security control protocols renders IoT devices especially vulnerable to network attacks [2,3]. The IoT is a system of many physical devices, or "things," that are linked to one another. With new, advanced, and contemporary devices being created daily, the number of IoT systems is growing significantly. Heterogeneous, diversified, and complex data is produced by a vast number of interconnected devices. This expands the cyberattack surface, which is impacted by new IoT anomalies such as botnets, denial-of-service attacks, and spoofing [4]. Software security flaws in

IoT could lead to privacy breaches. To execute more complex attacks, attackers may access higher-level computing capabilities if they compromise the IoT device's software. This is because of the features of IoT systems, which include prolonged lifespans, delayed patching, lack of upgrades, and compromise implications [5,6].

Sophisticated security tools, such as Intrusion Detection Systems (IDSs), are essential for monitoring IoT device behavior on the network and identifying potentially harmful activities to safeguard IoT systems. The IDS alerts the device administrator to any intrusion, allowing the administrator to isolate malicious devices and take necessary actions to mitigate the threat [7,8]. Network Intrusion Detection Systems (NIDS) are crucial for detecting threats and attacks on computer networks. The two main groups of IDS detection techniques are anomaly and signature-based IDS [9]. Signature-based IDSs perform intrusion detection by comparing collected traffic against specified attack signatures, which are predetermined communication patterns of known attacks. These networked systems produce enormous amounts of data, or network traffic, which creates a fertile ground for machine learning (ML) solutions. Unlike the conventional algorithmic approach, ML solutions seek to learn from data [10,11]. NIDS that uses ML are an established field of study. New avenues in this field are being investigated due to developments in machine learning and cutting-edge methods such as federated learning and deep learning. Various methods, including behavior-based, signature-based, or hybrid approaches, have been employed in network intrusion detection systems to identify anomalies. ML/DL-based IDSs have been demonstrated to be ideal options, offering high detection accuracy combined with efficiency and efficacy [12,13]. Continuous efforts are being undertaken to effectively protect these networks and systems from cybersecurity vulnerabilities. ML and DL, two AI approaches, have become widely adopted and are being utilized to improve IoT network security [14].

The enormous volume of data generated by internet-connected IoT devices, combined with AI's capability to process and analyze large-scale datasets, enables more efficient decision-making and intelligent system automation, making it particularly appropriate for IoT applications [15,16]. Recently, Federated Learning (FL) has attracted active research in multiple fields, including IoT, edge computing, and healthcare, to establish collaborative methods that detect threats without sharing raw data [14]. FL shows promise as a solution, allowing multiple IoT devices to retain data locally while collaboratively training a shared model. This FL approach enables collaborative model training across multiple devices while preserving data privacy by ensuring that raw data remains on the local IoT devices [17]. In the FL framework, devices (or nodes) are trained locally using their own data [18]. A central server receives only model updates rather than raw data for aggregation. FL appears to be a suitable solution given the decentralized nature of IoT networks. FL can provide real-time insights by locally processing data on IoT devices, which is essential for prompt threat identification and the mitigation of DDoS attacks [19]. Unlike traditional centralized learning, FL does not transmit raw client data; instead, only model updates are exchanged between clients and the server. This characteristic reduces the risk of data leakage while enabling FL to improve the global model using distributed data [20].

The proposed methodology introduces an FL-based IDS that ensures accuracy, robustness, and data privacy in IoT environments. Initially, raw data undergoes cleaning, normalization, and label encoding to improve data quality and consistency. Then, the HEPQSO technique selects the most significant and discriminative features, reducing redundancy and improving learning speed and accuracy. The optimized features are then fed into a hybrid deep learning model consisting of the Spike Gated Linear Unit (SGLU) and a Vision Transformer-Temporal Convolutional Network (ViT-TCN), which effectively learns global and temporal information from the data. Finally, the IDS is implemented within a federated learning environment, ensuring data privacy and collaborative learning among IoT devices, thereby improving accuracy and reducing false alarms.

The primary contributions of the proposed work are summarized as follows:

- A federated deep learning-based IDS is proposed that provides assurance of data privacy and facilitates collaborative training among distributed IoT nodes.
- The proposed Hybrid Emperor Penguin–Quokka Swarm Optimization (HEPQSO) algorithm is utilized for feature selection to enhance model performance.
- The SGLU-ViT-TCN classifier is introduced, coupling spike-gated linear units with transformer and temporal convolutional designs for better detection.

The paper is organized as follows: Section 2 reviews the related literature, Section 3 presents the proposed approach, Section 4 summarizes the experimental results, and Section 5 concludes the paper.

2. Literature Review

Javeed et al. (2024) [21] described a horizontal FL model for efficient intrusion detection that blends “Bidirectional Long-Term Short Memory (BiLSTM) and Convolutional Neural Networks (CNN)”. In the context of federated learning (FL) for IoT, data remains on edge devices while only model updates are shared, enabling collaborative training without compromising privacy. This hybrid methodology sought to improve intrusion detection efficacy while overcoming the drawbacks of current techniques. In this approach, data is retained on local edge devices, and only the learned model weights are transmitted to the central federated learning (FL) server. The FL server then aggregates updates from multiple sources to enhance the accuracy of the global model. Althunayyan et al. (2024) [22] presented a state-of-the-art, innovative,

lightweight, in-car DL technique that uses IDS to overcome these constraints. An artificial neural network (ANN) is employed in the first stage to extract and learn relevant feature representations from the input data. To detect known attacks, an LSTM autoencoder is utilized in the subsequent phase to identify unknown threats. The deployment of the IDS in a hierarchical FL (H-FL) environment enables the evaluation of driving behaviors and the updating of the model with the latest threat patterns without compromising data privacy.

Chen et al. (2023) [23] provided two metrics for evaluating privacy in FL-based NIDSs: (1) the rate of evasion for NIDSs using adversarial attacks with recovered traffic, and (2) a privacy score that measures the similarity between the reconstructed and original traffic attributes, using reconstruction attacks as the evaluation metric. Tests demonstrated that current defenses are insufficient, allowing adversarial traffic to bypass state-of-the-art NIDS like Kitsune. FedDef is an innovative input perturbation defense technique based on optimization that offers theoretical assurance to prevent such attacks and build a more robust FL-based NIDS. By maximizing the input distance and lowering the gradient distance, it provides robust privacy protection in addition to high utility. Sun et al. (2024) [24] presented FedMADE, a cutting-edge dynamic aggregation technique that groups devices according to traffic patterns and combines local models according to how well they contribute to overall performance. Compared to previous FL algorithms designed for non-IID data, FedMADE increased the precision of minority attack classification by up to 71.07%.

Bukhari et al. (2024) [25] suggested a new model for IDS in WSNs that combines a stacked CNN with Bi-LSTM (SCNN-Bi-LSTM). The FL-based SCNN-BiLSTM model utilizes a novel hybrid architecture that combines spatial feature extraction through a stacked convolutional neural network (SCNN) with temporal sequence modelling via a bidirectional long short-term memory (BiLSTM) network. This technique alleviates privacy concerns by allowing several sensor nodes to collaborate to refine a central global model without disclosing personal information. The deep learning methodology of the SCNN-BiLSTM model efficiently captures both spatial and temporal patterns in the data, enabling accurate and privacy-preserving predictions. It detects intricate and as-yet-unidentified cyberthreats by closely observing temporal and local links in network patterns. Idrissi et al. (2023) [26] recommended Fed-ANIDS, a NIDS that addressed the privacy concerns with centralized models by utilizing FL and AD. Using a variety of anomaly detection (AD) models, such as adversarial, variational, and simple autoencoders, the system calculates an intrusion score based on the reconstruction error of normal traffic patterns to identify intrusions. The excellent performance of the proposed strategy across various metrics, while preserving remote client data, demonstrates its effectiveness.

Anwar et al. (2025) [27] intended to develop and evaluate an effective IDS for IoT-based WSNs via an FL framework that interfaces with LSTM networks. This FL technique addresses privacy concerns and enhances detection capabilities by having numerous IoT nodes retain data locally while collaboratively training a global LSTM model. Three well-known datasets were applied to assess the suggested model: WSN-DS, UNSW-NB15, and CIC-IDS-2017. Bensaid et al. (2024) [28] suggested SA-FLIDS, a revolutionary blockchain-based NIDS architecture for fog-IoMT-enabled smart healthcare systems that is built on Secure and Authenticated Federated Learning. Enhancing data privacy and lowering communication costs are the goals of this research. Additionally, flaws in decentralized learning systems are vulnerable to attacks such as model poisoning and Sybil attacks. To ensure client authentication and secure communication, the approach leverages Self-Sovereign Identity (SSI) mechanisms. Additionally, aggregated data is used using the Trimmed Mean approach. When building the overall model, this lessens the impact of odd or malicious inputs.

Karunamurthy et al. (2025) [29] provided an FL-based intrusion detection method that uses federated learning to train DL classifiers in IoT networks and identify various threats. The suggested work chooses the best characteristics using the Chimp optimization method. The benchmark MQTT dataset was used in studies to demonstrate that the FL-based IDS attained the highest detection efficiency for recognizing intrusions above traditional machine learning techniques. Huang et al. (2024) [30] suggested an innovative method based on FL that aims to guarantee data privacy while increasing the accuracy of NIDS. This study created a novel DL-IDS tailored for the IIoT by fusing convolutional neural networks with attention processes. To improve data privacy security, variational autoencoders were also included. Furthermore, several IIoT clients can work together to build a shared intrusion recognition model utilizing an FL framework without sharing their raw data. This approach successfully addressed safety and privacy issues with data while also significantly enhancing the model's detection capacity.

Alsaleh et al. (2025) [31] suggested a lightweight IDS based on a semi-decentralized FL paradigm to support IoT device capabilities. The IoT devices, or FL clients, are grouped together in the suggested model, and each cluster is given a cluster leader who represents the FL clients. As a result, there are fewer IoT devices interacting with the server, which lowers the communication overhead. Additionally, during each federated learning round, each cluster transmits its average model weights to the central server for aggregation, with clustering facilitating a more efficient and organized aggregation process. The IDS model's primary issue is the DDoS attack, which can easily happen on Internet of Things devices with constrained resources. Chaurasia et al. (2024) [32] presented a DL model based on ResNets that was trained through federated learning. The vanishing gradient problem is effectively addressed by ResNet, and federated learning allows several customers or Internet service providers (ISPs) to train together without disclosing their data to outside parties. This method preserved anonymity while improving accuracy through group learning.

Rashid et al. (2023) [33] suggested an FL technique to identify unauthorized breaches and ensure IoT network security. This technique uses FL of local IoT device data to guarantee security and privacy. A central global server receives only parameter changes from local IoT clients, aggregates the received updates, and disseminates an enhanced detection algorithm. Following each federated learning training cycle, each IoT device updates its local model accordingly. The client trained their local dataset using an updated model that was delivered from the global server, allowing IoT

devices to maintain data privacy while collectively enhancing the performance of the global model. To assess the performance of the proposed method, thorough tests were carried out using a brand-new dataset called Edge-IIoTset. Belenguer et al. (2023) [34] introduced GöwFed, a novel technique for detecting network threats that combines Gower Dissimilarity matrices with Federated averaging. A vanilla version of GowFed, which achieved a median point of [0.888, 0.960], and an enhanced version with a mechanism for attention, which shows the best performing networks contributed to the model, are the two different techniques that have been designed according to the most recent research developments. Additionally, the TensorFlow Federated framework's simulation-oriented capabilities were used to test each option.

Chandra Umakantham et al. (2024) [35] suggested an IDS technique based on federated learning that uses Enhanced Ghost_BiNet, a cutting-edge deep learning model, to effectively learn complex patterns and make accurate predictions from the data. A global intrusion detection model can be jointly trained by numerous entities without exchanging sensitive data due to FL. The proposed method is a privacy-preserving machine learning approach. It employs Enhanced Ghost_BiNet, which integrates GhostNet and BiGRU, to train local models initially. The Chaotic Chebyshev Artificial Hummingbird (CAh) algorithm is utilized to optimize the model's performance. To ensure data privacy, updates from local models are encrypted using homomorphic encryption before being shared, which improves data security and privacy. To reduce communication cycles during data aggregation, server-side update aggregation and collaborative optimization are implemented. Table 1 compiled the literature review by presenting various federated learning-based intrusion detection models, their approach, findings, and shortcomings.

Table 1. Summary of the literature review

Authors & Year	Techniques Used	Findings	Limitations
Javeed et al. (2024)	Horizontal FL with BiLSTM + CNN	Captures temporal (BiLSTM) and spatial (CNN) patterns; improved IDS accuracy with zero-trust FL setup	Scalability across highly heterogeneous IoT is not fully validated.
Althunayyan et al. (2024)	Hierarchical FL with ANN + LSTM Autoencoder	Multi-stage IDS detects both known and novel attacks; suited for vehicular networks.	Primarily a theoretical framework; lacks a large-scale real-world evaluation
Chen et al. (2023)	FedDef (perturbation defense), privacy evaluation measures	Proposed new privacy scoring + adversarial resilience; improved robustness vs reconstruction attacks	Defense tested on limited datasets; generalization to diverse IoT datasets uncertain
Sun et al. (2024)	FedMADE (dynamic aggregation based on traffic similarity)	Boosted minority attack detection by up to 71% over conventional FL	Complex grouping may increase system overhead
Bukhari et al. (2024)	SCNN + BiLSTM with FL	Efficient in detecting complex DoS attacks on WSNs	Evaluation limited to WSN-DS and CIC-IDS-2017 datasets
Idrissi et al. (2023)	Fed-ANIDS with autoencoders (variational, adversarial, simple)	High intrusion detection accuracy while preserving privacy	Heavy reliance on reconstruction error may cause false alarms
Anwar et al. (2025)	FL-LSTM for IoT-WSNs	Improved privacy-preserving detection using WSN-DS, CIC-IDS-2017, UNSW-NB15	Scalability and real-time adaptability have not been fully tested
Bensaid et al. (2024)	SA-FLIDS (Blockchain + FL) with SSI and Trimmed Mean	Secure against Sybil/poisoning attacks, suited for IoMT healthcare	Blockchain introduces latency and computational costs
Karunamurthy et al. (2025)	FL with Chimp Optimization + DL	95.59% accuracy on the MQTT dataset; effective feature selection	Evaluated on a single dataset; generalizability untested
Huang et al. (2024)	FL with CNN + Attention + VAE	Preserved privacy while improving intrusion detection for IIoT	The model complexity may hinder deployment on resource-constrained devices
Alsaleh et al. (2025)	Semi-decentralized FL with cluster leaders	Reduced communication overhead; suitable for IoT	Vulnerable to DDoS on resource-constrained clusters
Chaurasia et al. (2024)	FL with ResNet	Tackled vanishing gradient; maintained privacy while boosting accuracy	Focuses solely on ResNet; limited exploration of hybrid deep models
Rashid et al. (2023)	FL with Edge-IIoTset dataset	Strong security and privacy; efficient FL cycle with parameter sharing	Uses a newly introduced dataset; comparative benchmarking is limited
Belenguer et al. (2023)	GöwFed (FedAvg + Gower Dissimilarity)	Achieved 0.93 accuracy; effective with attention mechanism	Evaluation is limited to simulation using TensorFlow Federated
ChandraUmakantham et al. (2024)	Enhanced Ghost_BiNet + FL + Homomorphic Encryption	High detection accuracy; secure aggregation with HE	The addition of encryption increases communication and computational overhead

Though FL has been a privacy-augmenting paradigm, current FL-based IDS models tend to perform poorly on non-

Though FL has been a privacy-augmenting paradigm, current FL-based IDS models tend to perform poorly on non-IID data distributions, lack adequate feature optimization, experience high false alarm rates, and are not suitably adaptable to sophisticated intrusion patterns. Additionally, state-of-the-art methods do not achieve a balance between detection accuracy, adversarial attack robustness, and efficiency for resource-limited IoT devices. Therefore, there exists a critical necessity for a new FL-based intrusion detection framework combining smart preprocessing, efficient feature selection, and the latest hybrid deep learning models to provide high detection rates, low false positives, scalability, and privacy preservation in various IoT ecosystems.

In contrast to conventional intrusion detection systems that depend on conventional federated learning models with basic feature extraction and classification mechanisms, the proposed framework improves the efficiency of the learning process through intelligent feature optimization. This is achieved through the efficient selection of discriminative features using a hybrid intelligent optimization algorithm, which incorporates evolutionary intelligence and quantum-inspired mechanisms. In addition, the proposed framework offers efficient modeling of spatial, temporal, and contextual relationships through the integration of gated activation functions, transformer-based global attention mechanisms, and temporal convolutional structures. Incorporating such an architecture in a federated learning environment improves its robustness in handling heterogeneous data and class imbalance in distributed nodes, and offers efficient data privacy through localized data.

3. Proposed Methodology

The methodology proposed in this work is a federated learning-based IDS that guarantees accuracy, resilience, and privacy retention in IoT settings. The architecture starts with an extensive preprocessing stage of the data, where raw traffic is scrubbed to remove missing, duplicated, and noisy records, then scaled by min–max normalization to normalize feature values and speed up convergence. Next, categorical network traffic features are converted to numerical values via label encoding to maintain consistency for machine learning. In order to counter the high-dimensional and heterogeneous characteristics of IoT traffic, a Hybrid Emperor Penguin–Quokka Swarm Optimization (HEPQSO) algorithm is utilized for feature selection. The bio-inspired hybrid optimizer maintains a balance between exploration and exploitation by tapping into penguin huddling and quokka adaptive techniques before settling on the most discriminative features while avoiding redundancy. These improved features are then processed during the local training phase with an innovative hybrid DL model: the Spike Gated Linear Unit (SGLU) coupled with a Vision Transformer–Temporal Convolutional Network (ViT–TCN). The resulting synergistic integration enhances the representation of spatial and temporal characteristics of intrusion data; this improves detection accuracy and generalization. The SGLU module improves non-linear representation and effectively retains temporal dependencies in spiking neuron-based designs, while the ViT module extracts global dependencies through self-attention, and the TCN module captures local sequential patterns through dilated causal convolutions. The whole IDS system is implemented within a federated learning framework where IoT edge devices perform local model training while keeping raw traffic data private, and only encrypted model updates are sent to the global server for aggregation. Provides a guarantee of privacy of data, scalability, and minimal communication overhead while constructing an effective global IDS with high precision, low false alarms, and high resistance against adversarial tampering.

3.1. Data Preprocessing

The first phase is concerned with cleaning raw network traffic into a formal and machine-readable format. Traffic generated by IoT is noisy by nature, redundant, and usually contains inconsistent labels, which can confuse classifiers if not cleansed.

A. Data Scrubbing

The first stage in the preprocessing phase is data cleaning, which eliminates duplicate, inaccurate, and missing values from the dataset. Data cleaning, also called data scrubbing, helps maintain the model's precision, performance, and reliability in the IDS prediction task. Data scrubbing is carried out to remove duplicate records, inaccurate records, and null values. This is to ensure that the training process is carried out effectively to maximize model performance on clear and accurate datasets. The system improves prediction accuracy and avoids false alarms by deleting irrelevant or incomplete records. To avoid misleading the model during training, redundant characteristics are eliminated and empty label spaces are avoided.

B. Scaling

Then, feature scaling is used to normalize skewed attributes. Because IoT traffic data contains features whose values have highly different scales, e.g., packet size, duration, and frequency, unbalanced scales can lead to bias in the classification model. The imbalance scale features cause the classification model's performance to decrease. Normalizing characteristics with dominating and insignificant values onto an acceptable scale is crucial. Equation (2) states that the continuous variables are transformed using the minimum–maximum approach into a range of [0, 1]:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

where the minimum and maximum values are 1 and 0, respectively$X_{min}X_{max}$, while the remaining variables fall between these ranges. X_{norm} is the normalized result. As a result, the characteristics will fall within the same range of values as a base. This scaling technique will shorten a model's training and testing periods, address bias concerns, and guarantee a quick rate of convergence, all of which will increase the classification system's dependability. With the min-max normalization method, all continuous features are scaled into a common range [0,1]. This not only removes bias but also speeds up deep learning model convergence, shortens the training time, and enhances system stability in general.

C. Label Encoding

Lastly, label encoding is utilised to adapt categorical attributes into a numerical representation. For intrusion detection purposes, traffic is usually referred to as "normal" or "abnormal." The dataset was cleaned, and then the categorical attributes were transformed into a numerical representation that machine learning algorithms could understand using label encoding. The class variable for network connections in intrusion detection was divided into two categories: abnormal and normal. LabelEncoder is understandable in a general mathematical context. A set of categorical labels, y, where $y \in \{y_1, y_2, \ldots, y_K\}$, are supplied to the LabelEncoder. A distinct integer z is assigned to every category label y_i. Equation (2) provides a mathematical expression for this.

$$z = LabelEncoder(Y[i]) \tag{2}$$

The Label Encoder ensures that each unique categorical label, y_i, is associated with a unique integer within the range, where K indicates the number of unique categories$z[0, K-1]$, which informs the computation of class probabilities. cleansed data, which only include a certain number of unique values, were encoded with this label. Mapping categorical labels to integer values makes the dataset compatible with optimization algorithms and neural networks. Label encoding facilitates categorical feature interpretation by the IDS and provides consistency for distributed IoT nodes involved in the federated training process.

3.2. Hybrid Emperor Penguin–Quokka Swarm Optimization (HEPQSO) for Feature Selection

The tallest and heaviest penguin species is the emperor penguin, scientifically named Aptenodytes forsteri. The emperor penguin (Aptenodytes forsteri) shows very similar plumage and size between males and females and reproduces during the Antarctic winter on open ice, where they spend their entire lives. Large colonies of thousands gather during the nesting season, and females may travel up to 50 miles to the ocean to hunt, returning to deposit a single egg. Unique among penguins, they hibernate during the harsh winter, relying on complex social behaviours, including four stages of huddling, to conserve heat and maintain unity. They also hunt and gather food cooperatively to ensure survival. Inspired by these behaviours, the Emperor Penguin Optimization (EPO) mathematical model captures key aspects such as identifying efficient movers, updating positions, and measuring huddle "temperature." The Enhanced EPO (EEPO) algorithm extends this approach to optimize antenna selection by calculating distances and positions using a modified Manhattan distance, thereby converging on the most efficient solution. Fig. 1 represents the flowchart of the proposed Hybrid Emperor Penguin–Quokka Swarm Optimization (HEPQSO) feature selection process.

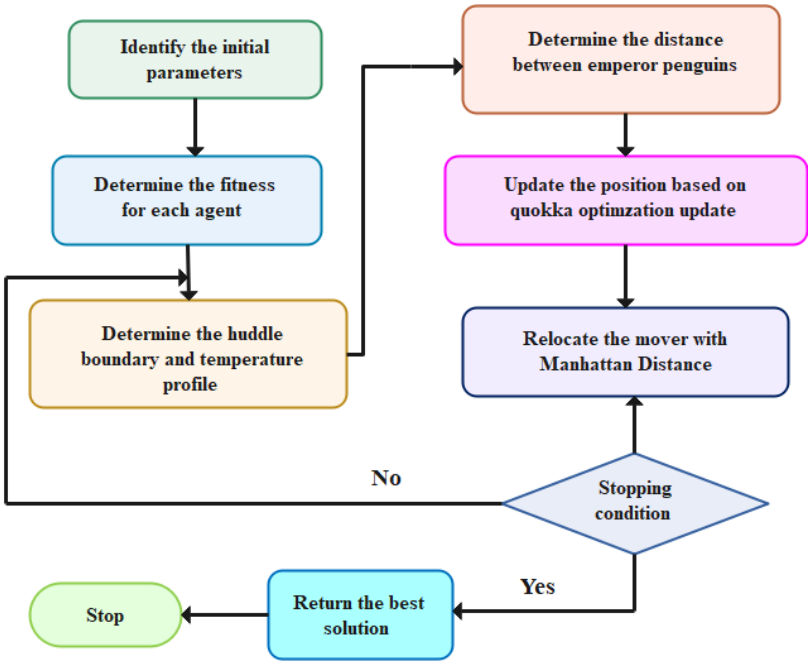


Fig.1. Proposed HEPQSO flowchart

The huddle is modelled as a two-dimensional L-shaped polygonal plane. Initially, the boundary of the huddle is The huddle is modelled as a two-dimensional L-shaped polygonal plane. Initially, the boundary of the huddle is generated randomly by the emperor penguins. The temperature profile surrounding the huddle is then calculated, and the distances between individual penguins are established to facilitate further analysis. Finally, the most efficient mover, representing the optimal solution, is determined, and the huddle boundary is updated based on the new positions of the search agents, or emperor penguins.

Following preprocessing comes feature selection (FS), which is vital for minimizing computational complexity and improving classification accuracy. IoT datasets are normally high-dimensional and include redundant or irrelevant attributes that can dilute the efficacy of DL classifiers. In order to mitigate this, the methodology uses a new HEPQSO algorithm. HEPQSO combines the social huddling habit of emperor penguins with the adaptive foraging strategies of quokkas and balances exploration and exploitation in the search space. The emperor penguin optimization component emulates the huddling process when penguins cluster together to reduce energy loss during harsh Antarctic conditions.

Here, penguin movements are candidate subsets of features, with the "optimal huddle" being the most discriminative set of features. The algorithm uses an adapted Manhattan distance metric to direct position updates, which improves convergence efficiency and lowers computational cost in comparison to standard Euclidean measures. At the same time, the quokka swarm optimization mechanism integrates adaptive updates based on environmental conditions like temperature, humidity, and drought coefficients. These parameters simulate the quokkas' capability to dynamically adapt their foraging protocols, thereby making the search process robust against premature convergence. Through combining both strategies, the hybrid model achieves a balance between wide exploration of the search space and concentrated exploitation around promising sets of features.

A. Create and Establish the Boundaries of the Huddle

Emperor penguins typically form huddles within a polygon-shaped grid boundary, with the minimum number of penguins in the huddle determined by randomly selected neighbors. To define the huddle boundary around the polygon, the wind flow surrounding the huddle is estimated, noting that the wind moves faster than the penguins themselves. Complex variable theory is employed to model the randomly generated huddle boundary. Here, τ represents the gradient of φ, where φ denotes the wind velocity as defined in equation (3).

$$\tau = \nabla\varphi \tag{3}$$

The complex potential is formulated by combining the vector τ with ϕ, as described in equation (4).

$$F = \varphi + i\tau \tag{4}$$

Here, F represents an analytic function on the polygon plane, and i is the imaginary constant.

B. Temperature Profile Around the Huddle

Emperor penguins form huddles to conserve energy and maintain warmth. To model this behavior, the temperature is set as $T = 0$ when the polygon radius $R > 1$ and $T = 1$ when $R < 1$, accurately reflecting the thermal conditions. The exploration and exploitation process of emperor penguins across different sites is governed by the temperature profile. The temperature around the huddle, T' is computed using equation (6).

$$T' = \left(T - \frac{Max_{iter}}{x - Max_{iter}}\right) \tag{5}$$

$$T = \begin{cases} 0, & if \;\; R > 1 \\ 1, & if \;\; R < 1 \end{cases} \tag{6}$$

where R is the radius, x is the current iteration, Max_{iter} represents the maximum iterations, and T is the time needed to find the optimal solution region.

C. Manhattan Distance between Emperor Penguins

Once the huddle boundary has been established, the distance between each emperor penguin and the best-obtained optimal solution is calculated. The solution with the fitness value closest to the current optimum is selected as the best. The remaining search agents, or emperor penguins, then update their positions based on this current optimal solution, which can be mathematically explained as follows. The modified distance equation is given in equation (7), and it substitutes the Manhattan distance for the original Euclidean distance.

$$\overrightarrow{D_{EP}} = S\left(\vec{A}\right) \sum_{i=1}^{n} \left| P_i(x) - P_{EP,i}(x) \right| \tag{7}$$

Here, x denotes the current iteration, and $\overrightarrow{D_{EP}}$ represents the modified Manhattan distance between the most suitable search agent and the best emperor penguin (i.e., the emperor penguin with the lowest fitness value). Using the Manhattan distance in EPO, instead of the Euclidean distance, simplifies calculations and enables more efficient movement in certain search spaces, particularly grid-like environments. This distance metric is especially effective in spaces where movement is constrained to vertical and horizontal directions, such as city blocks or grids, and can improve convergence in problems where diagonal moves are either suboptimal or restricted. The best optimal solution, or the fittest emperor penguin, is denoted by $\vec{P}$ while $\overrightarrow{P_{EP}}$, representing the location vector of an emperor penguin. The term $S\left(\vec{A}\right)$ refers to the social

factors of the penguins that drive them toward the most advantageous search agent. The following formula (8) is used to compute $\vec{A}$:

$$\vec{A} = \left(M \times \left(T' + P_{grid}(ac)\right) \times rand()\right) - T' \tag{8}$$

$$P_{grid}(ac) = \left|\vec{P} - \overrightarrow{P_{EP}}\right| \tag{9}$$

Here, M is the mobility parameter, which helps prevent collisions by maintaining a minimum distance between search agents, and is set to a value of 2. T' denotes the temperature profile surrounding the huddle, while $P_{grid}(ac)$ evaluates the differences among emperor penguins to determine the polygon grid accuracy, as defined in equation (9). The function $rand()$ generates random values in the range [0, 1]. Equation (10) is then used to compute the function. $S(\vec{A})$.

$$S(\vec{A}) = \left(\sqrt{f.e^{-x/l} - e^{-x}}\right)^2 \tag{10}$$

In this case, the expression function is represented by e. The control parameters facilitate more effective exploration and exploitation during the search process. Their respective ranges are [2, 3] for f and [1.5, 2] for l.

Every quokka modifies its location (signalling updated weight values) in accordance with the population's best-performing leader at the current iteration. Equations (11-12) represent the position update equations:

$$\vec{A}^{new} = \frac{(S+G)}{(0.8 \times \vec{A}^{old})} \times \Delta R \times rand \times \Delta Y \tag{11}$$

$$Y^{new} = Y^{old} + \vec{A}^{new} \times M \tag{12}$$

where S stands for the temperature ratio, which runs from 0.2 to 0.44, G for the humidity ratio, which ranges from 0.3 to 0.65, and $\vec{A}^{old}$ for the drought and its value within the interval [0,1]. The $rand$ is an arbitrary number that ranges from 0 to 1, and ΔR illustrates the weight difference between the leader and the quokka. M represents the nitrogen ratio, with a number between 0 and 1, chosen because quokkas need this amount of nitrogen. ΔY represents the difference in status between the leader and Quokka. Quokka's new role is represented by $\vec{A}^{new}$, while Y^{new} represents the previous role.

D. Relocate the Mover with Manhattan Distance

The positions of emperor penguins are updated based on the mover, which represents the best-obtained optimal solution. This mover leaves its current location and guides the repositioning of other search agents within a defined search space. As the penguin that has found the optimal solution, the mover influences the movement of the remaining penguins. Their positions are updated using the Manhattan distance and temperature profile, ensuring all search agents converge toward the most optimal solution for the antenna selection problem. The future position of an emperor penguin can be calculated using equation (13) below:

$$\overrightarrow{P_{EP}}(x+1) = \overrightarrow{P(x)} - \vec{A}.\overrightarrow{D_{EP}} \tag{13}$$

where $\overrightarrow{P_{EP}}(x+1)$ represents the updated position of an emperor penguin in the next iteration. Once the mover has been relocated, the huddling behavior of the emperor penguins is recalculated during the iteration phase. The result of HEPQSO is the most informative and least redundant feature set, which is optimized and maintained for classification. This diminishes dimensionality, enhances generalization, and guarantees that the next deep learning stage works on an optimized dataset specifically designed for intrusion detection.

3.3. Local Training Model: SGLU-ViT-TCN

The feature set that has been optimized is then input to the local classification model, which provides a new hybrid deep learning structure that involves the Spike Gated Linear Unit (SGLU), Vision Transformer (ViT), and Temporal Convolutional Network (TCN). Application of the SGLU block is a key methodological contribution since it involves embedding a spiking neuron-inspired gating mechanism into a multilayer perceptron framework. This approach captures non-linear interactions among features while maintaining the integrity of the original data. Temporal dependencies are essential for the intrusion detection task. SGLU divides the input into two streams, performs linear transformations and spiking neuron activations, and element-wise multiplies the output spike vectors, thus separating signal from noise. Additionally, the incorporation of time-dimension batch normalization prevents temporal information loss, which is especially beneficial in streaming IoT traffic contexts. Fig. 2 depicts the architecture of the proposed local training model (SGLU-ViT-TCN).

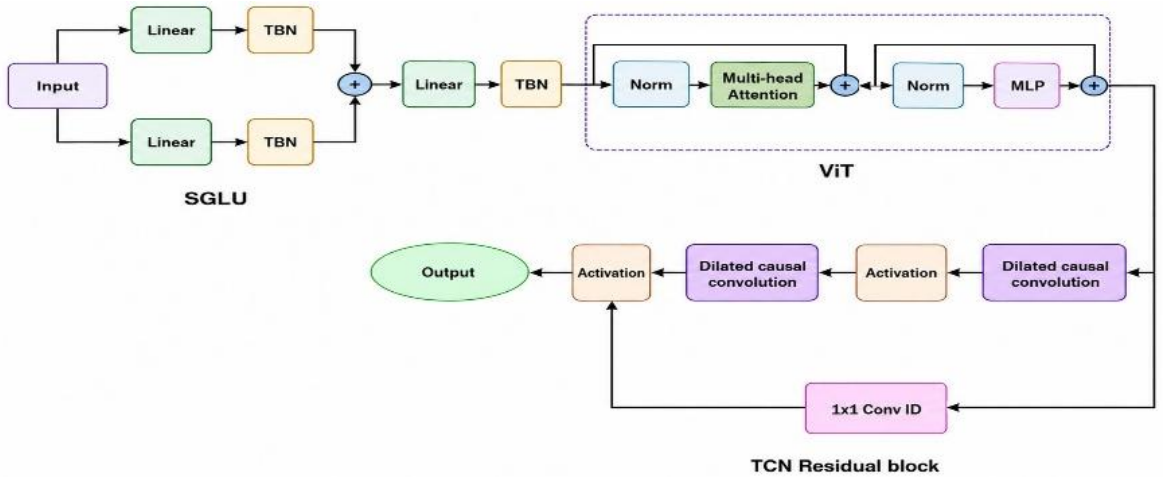


Fig.2. Proposed local training model: SGLU-ViT-TCN

Numerous neural network models make extensive use of MLP, a fundamental neural network topology. To create a spike-based MLP, Spikformer replaces a standard layer with a layer of LIF (Leaky Integrate-and-Fire) neurons serving as an activation function after a linear layer. The gating mechanism can improve the model's performance and dynamically control the flow of input to manage complex tasks. Here, the system integrated SGLU, which is distinguished from regular GLU in that it incorporates a spike-inspired dynamic gating mechanism. For local training, SGLU facilitates learning in individual nodes using their local sequential or event-driven data by passing through spike-based gates, which focus on significant temporal patterns, filtering out noise in the process. In comparison to standard GLU, spiking layers using LIF, and standard MLP, SGLU can efficiently handle heterogeneous and sparse data due to its focus on temporal sensitivity, event-driven sparsity, and adaptive gating, making it significantly better for sequential data, IoT traffic, and other scenarios. It reduces unnecessary computations, making it computationally efficient for local learning. Moreover, the SGLU is more effective in capturing non-linear relationships than regular MLP blocks. In intrusion detection systems, SGLU is used to improve feature extraction. This is done by giving greater importance to critical attack features while reducing noise.

In SNNs, SGLU is integrated into the MLP. A gating mechanism divides the input into two sections: one segment is modulated while the other is transmitted directly. Using SGLU, each section is subjected to a linear transformation and a fully connected layer. To create spike vectors, these two components are subsequently run through a layer of spiking neurons. The filtered information is obtained by multiplying the two spike vectors element-wise. Spike vectors are created by passing the filtered data through the spiking neuron layer after undergoing another linear modification, readying it for the following network segment. Additionally, SGLU employs a time-dimension batch normalization layer instead of the standard batch normalization layer. By normalizing the data along the time dimension, this approach better preserves the temporal information inherent in the input. The SGLU operation is defined by the following formula:

$$U = SN\left(TBN\left(Linear(X)\right)\right) \tag{14}$$

$$V = SN\left(TBN\left(Linear(X)\right)\right) \tag{15}$$

$$O = SN\left(TBN\left(Linear(U.V)\right)\right) \tag{16}$$

Here, SN denotes the spike neuron layer, Linear represents the fully connected layer, and TBN refers to the time-dimension batch normalization layer. SGLU is an enhanced spiking MLP that integrates a gating mechanism to increase the model's expressiveness. The ViT block utilizes self-attention mechanisms to encode global dependencies within the dataset. ViT is a categorization method based on Transformers. ViT is a unique type of transformer encoder and attention mechanism. An essential component of the ViT algorithm is the transformer encoder. For local training, the Vision Transformer (ViT) transforms the input data into a list of tokens, and then these tokens are projected onto a fixed-dimensional space while incorporating position information. Then, the projected embeddings are passed through a stack of transformer layers, allowing the model to learn complex relationships between features. Moreover, a fully connected layer is applied for classification or anomaly detection. The inclusion of ViT for local training is highly advantageous for distributed IoT networks, as each device can learn from local patterns, and ViT's self-attention mechanism can effectively learn complex relationships between high-dimensional and heterogeneous features. The similarity between the patches is represented by each row and column in the similarity matrix. To determine the outcome and compute the loss between the actual and anticipated output, a classifier, typically a linear layer, is employed at the final layer. Unlike CNN-based algorithms that extract local features, ViT uses a transformer encoder block and patch embedding technique to focus on

global dependencies. Transformer Encoder uses an attention method to identify relationships between visual tokens, or image patches. The obtained tokens undergo normalization as shown in Eq. (17).

$$v' = normalize(v') \tag{17}$$

The three copies of normalized visual tokens are designated as Query (Q), Key (K), and Value (V) vectors. The self-attention mechanism uses cosine similarity to compute an attention map between Q and K. The attention map is processed by the softmax activation function to normalize values between 0 and 1, as shown in Eq. (18).

$$QK = softmax\left(\frac{Q \times K^T}{\sqrt{d}}\right) \tag{18}$$

The degree to which a visual token is related to others is indicated by these values. The stronger the relationship between patches, the higher the attention map value. As shown in Eq. (19), the V feature matrix and attention map are multiplied to highlight the most significant features and lessen the impact of undesirable features, such as noise.

$$x = QK \times V \tag{19}$$

The weighted feature vector undergoes an MLP block that consists of a linear layer and a GeLU activation function. The transformer encoder block is where all of these operations are carried out.

The temporal convolution network is used to enhance sequence modelling performance. It addresses issues with future information leakage and varying input and output sequence lengths in CNNs. It performs a similar function to RNNs, capable of mapping input sequences of any length to output sequences of the same length. At the same time, it avoids typical issues associated with RNNs, such as vanishing or exploding gradients, sequential computation bottlenecks, and inefficient memory usage. Its main characteristic is the application of DCC or dilated causal convolution. Issues related to varying input and output sequence lengths, as well as potential information leakage in CNN models, can be addressed using causal convolution. Additionally, dilated convolution expands the receptive field of the convolutional kernel while minimizing the number of required network layers.

DCC combines dilated convolution with causal convolution. Dilated convolution broadens the convolution kernel's receptive field, while causal convolution gathers historical data. Causal convolution efficiently prevents future information from influencing the current state by ensuring that the mth element of the output sequence only depends on the mth and preceding items in the input sequence. The network can acquire more extensive historical data based on the same network layer due to the dilated convolution. The TCN uses the zero-filling technique to guarantee that input and output sequences are always the same length, and the number of fills is as in equation (20).

$$Z = (k-1) \times d \tag{20}$$

where L denotes the number of network layers preceding the dilated convolution, d = 2^L is the dilation coefficient, and k represents the size of the convolutional kernel. Meanwhile, the growth rate is controlled by the dilated coefficient in the dilated convolution to accelerate data acquisition. Using a convolution kernel f of size k and a time series x, the dilated causal convolution (DCC) of the input in layer L is expressed in Eq. (21):$X = \{x_1, x_2, \ldots, x_m\} x_m$

$$F_d^L(m) = \sum_{i=0}^{k-1} f_i \cdot x_{m-i.d}^{L-1} \tag{21}$$

Here, the current component's net layer is indicated by a superscript. Following convolution, the output is passed through an activation function, which is expressed in Eq. (22):$H_d^L(m)$

$$H_d^L(m) = ReLU(F_d^L(m) + b) \tag{22}$$

where b denotes the bias term, and ReLU (rectified linear unit) is used as the activation function, which is commonly employed in convolutional neural networks. The residual connection can be expressed as follows in Eq. (23):$ReLU$

$$ReLU(t) = \max[0, t] = \begin{cases} 0, & t \leq 0 \\ t, & t > 0 \end{cases} \tag{23}$$

To further improve features, TCN makes use of residual blocks. After adding the residual connection, it is expressed as in Eq. (24):

$$\widetilde{H}_d^L(m) = ReLU\left(F_d^L(m) + b + H_d^{L-1}(m)\right) \tag{24}$$

According to Eq. (24), the network can further enhance features in layer L based on layer L-2, after learning data characteristics in layers L and L-1. This prevents the network from deteriorating and somewhat improves the model's

ability for generalization. The TCN module fills the gap by encoding local sequential dynamics with dilated causal convolutions, which increase the receptive field without leaking information from future inputs. Thus, TCN simulates long-range temporal dependencies in traffic flows, allowing the IDS to identify fine-grained changes and cumulative anomalies in live data streams. Collectively, the SGLU–ViT–TCN hybrid realizes a synergistic equilibrium: SGLU supports non-linear representation, ViT captures global attention-based correlations, and TCN supports temporal sequence learning.

3.4. Federated Learning Framework

First, the IoT nodes gather raw network traffic data, which is then passed through a sequence of preprocessing operations like scrubbing to eliminate noise and redundancy, scaling to feature-normalize attribute ranges, and label encoding to transform categorical attributes into machine-readable form. Once the data is normalized, every IoT node uses the HEPQSO algorithm to locally detect the most discriminative and optimized attributes. These chosen features are then employed for training the Spike Gated Linear Unit–Vision Transformer–Temporal Convolutional Network (SGLU–ViT–TCN) classifier to produce local model updates in the form of weights that have been learned. Rather than sharing the raw data, these updates alone are combined to build a better global model. Model updates (weights, gradients), but not raw traffic, are exchanged with the server. This reduces privacy vulnerabilities and aligns with data sovereignty standards. The global server carries out federated averaging to integrate model updates from every client into a global model. Because the local datasets are heterogeneous and frequently non-IID, the feature selection optimized for local datasets (HEPQSO) ensures that updates are consistent across clients, enhancing global stability and convergence rate. The system drastically cuts communication overhead by sending only compressed model parameters instead of full datasets. The decentralized nature also enhances resistance against adversarial attacks, as data remains distributed, leaving attackers without a central data repository to target. The latest global model is then redistributed back to the IoT nodes, where it is again tuned by subsequent local training rounds. The cycle of repeated iteration continues until the global model converges to a strong, adaptable, and privacy-conscious intrusion detection system that can scale effectively on various IoT deployments. Fig.3 illustrates the process of the suggested federated IDS, where local training is performed by IoT nodes and sends updates to a global server.

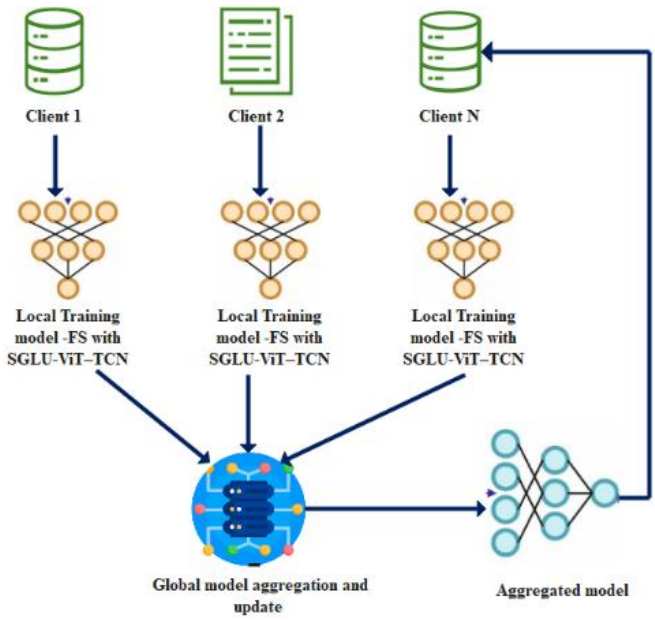


Fig.3. The proposed federated intrusion detection system

Traditional centralized intrusion detection systems need clients to send raw information to a central server, which raises privacy concerns and communication costs. Federated learning, on the other hand, enables IoT nodes to locally train using their feature-optimized and pre-processed data. Raw data is not sent, but rather model updates (gradients and weights) are sent to the central server. Before returning the global model to clients for the subsequent training cycle, the server aggregates these updates via methods like Federated Averaging (FedAvg). This distributed strategy has some benefits. First, it ensures the preservation of privacy, as raw traffic data never leaves the local device. Second, it decreases communication overhead, as only updates to models are transmitted instead of huge raw datasets. Third, it provides scalability and flexibility in heterogeneous IoT settings where strategies can have diverse computational resources and data distributions. With the inclusion of federated learning into the intrusion detection process, the system gains real-time adaptability without compromising security and efficiency in distributed IoT ecosystems.

4. Result and Discussion

The proposed IDS's performance is evaluated using standard metrics such as precision, recall, and F1-score, which provide a general indication of classification efficacy. Furthermore, metrics such as training time and convergence rate are also recorded to demonstrate the effectiveness of the federated learning process. In order to test robustness, the introduced method is evaluated against a set of baseline and current state-of-the-art deep models (LSTM, CNN-LSTM, Bi-GRU, Vision Transformer (ViT), Temporal Convolutional Networks (TCN)). The comparative study enables investigation of the strengths of the introduced model on classification performance, scalability, and generalization capability on different data distributions. In addition, the assessment takes into account varying dataset split ratios to provide consistency of outcome and to monitor the adaptability of the model under varying training and testing environments. The discussion focuses on the trends of the comparison results, the effects of the feature selection mechanism, and the benefits of incorporating federated learning in minimizing data privacy issues while guaranteeing correct detection accuracy.

4.1. Dataset Description

A. Dataset 1

BoTNeTIoT-L01 IoT-IDS Dataset (https://www.kaggle.com/datasets/azalhowaide/iot-dataset-for-intrusion-detection-systems-ids)

All of the IoT devices are provided within the detection_of_IoT_botnet_attacks_N_BaIoT (BoTNeTIoTAll of the IoT devices are provided within the detection_of_IoT_botnet_attacks_N_BaIoT (BoTNeTIoT) data set. BoTNeTIoT-L01 is the name of this data set. The duplication of the original dataset was reduced in this updated version by choosing just features from a 10-second time range. In this dataset, attacks (anomalous samples) are denoted by 0, while normal samples are denoted by 1. The most recent dataset, BoTNeTIoT-L01, was collected using Wireshark to capture traffic from nine IoT devices on a local network with a central switch. It includes two types of botnet attacks, Gafgyt and Mirai. The dataset contains 23 statistically designed features extracted from the .pcap files. A decay factor of 0.1 was applied to calculate seven statistical measures—mean, variance, count, magnitude, radius, covariance, and correlation coefficient—over a 10-second interval.

B. Dataset 2

IoTNet24 Dataset for IDS (https://www.kaggle.com/datasets/wittigenz/hydras)

This dataset displays a portion of network traffic data gathered from three live benign traffic captures and twenty malicious traffic captures on IoT devices. Its main purpose is the development and assessment of IDS for IoT devices. Despite its imbalance, the dataset offers useful information for identifying fraudulent activity in IoT networks. For efficiency and clarity, duplicates have been eliminated from its more than 23,000 rows. For binary classification tasks, the dataset works well, especially for accurately classifying malicious versus benign traffic The 'label' feature serves as the target variable, indicating whether a data instance is associated with malicious or benign activities. The values are stored as strings: 'Malicious' or 'Benign'.

4.2. Performance Analysis

The proposed FL-IDS approach demonstrates performance when tested against a range of benchmark models, such as LSTM, CNN-LSTM, Bi-GRU, Vision Transformer (ViT), and Temporal Convolutional Network (TCN), on two different data splits: 70/30 and 80/20. Fig.4 depicts the comparative performance evaluation of the proposed models against the baseline in terms of accuracy for various dataset splits.

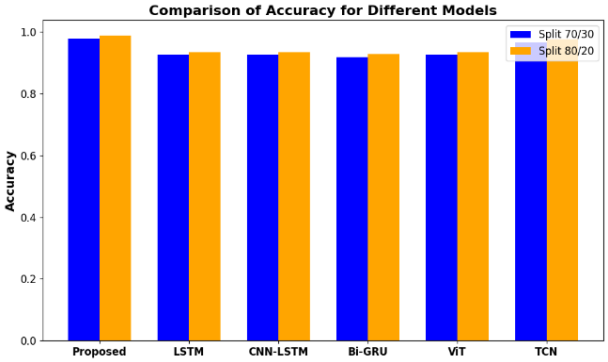


Fig.4. Performance analysis of accuracy for federated IDS model

For the 70/30 model training split, the proposed model recorded an accuracy of 0.978802, far surpassing all other models. The LSTM, CNN-LSTM, and ViT models had similar performance with accuracies of 0.925706, 0.92592, and 0.925775, respectively, while the Bi-GRU model had slightly lower performance at 0.918983. The TCN model recorded

a strong accuracy of 0.967442, although still not as high as the proposed model. Upon changing the training split to 80/20, the accuracy of the proposed model further increased to 0.988532, proving its strength with bigger training sets. The LSTM, CNN-LSTM, and ViT models once again reported comparable accuracies, now 0.934799, 0.935015, and 0.934868, respectively, but the Bi-GRU model slightly improved to 0.928008. The TCN model also performed better, achieving 0.976956, but was still less accurate than the proposed system. Fig.5 shows the comparative evaluation of the precision values of the proposed federated IDS model and baseline models.

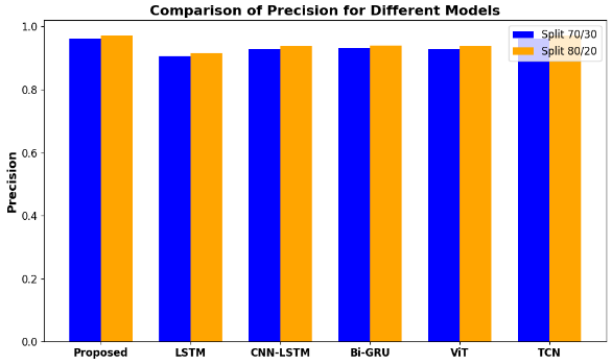


Fig.5. Performance analysis of precision for federated IDS model

For a 70/30 ratio, the proposed model attained a precision of 0.961683, which confirms its high capacity to suppress false positives. The TCN model was close with a precision of 0.961557, which implies that time feature extraction is significantly potent for precise intrusion classification. The Bi-GRU model then trailed closely with 0.931299, followed by CNN-LSTM and ViT with similar results at 0.928276 and 0.927887, respectively. The model with the lowest precision was LSTM (0.9068), which indicates its relative weakness in identifying actual attacks versus benign traffic. With the 80/20 ratio, the precision of the proposed model increased to 0.97124, further proving its stability with bigger training sets. The TCN model was again on par with 0.971012, retaining its competitive advantage. The Bi-GRU model had improved marginally (0.940448), as did CNN-LSTM (0.937395) and ViT (0.937001), while LSTM improved to 0.915702. Fig.6 presents a comparison of F1-scores between the suggested model and various deep learning benchmarks.

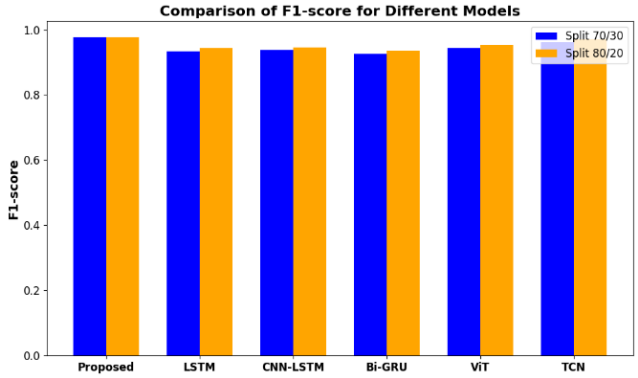


Fig.6. Performance analysis of F1-score for federated IDS model

For the 70/30 ratio, the proposed model attained an F1-score of 0.976649, demonstrating a better balance between precision and recall. The TCN model achieved a slightly lower F1-score of 0.9619, solidifying its proficiency in detecting temporal patterns. ViT placed third (0.943285), taking advantage of its global attention mechanisms. CNN-LSTM and LSTM lagged slightly behind at 0.936657 and 0.934026, respectively. With an F1-score of 0.9257, the Bi-GRU model demonstrates difficulty in achieving an optimal trade-off between precision and recall relative to other architectures. Using the 80/20 ratio, the proposed model performed better with a better F1-score of 0.977266, reflecting its stability in using more training data. The TCN was again the runner-up at 0.971362, and ViT recorded more improvement (0.952555). CNN-LSTM and LSTM also recorded small improvements (0.945860 and 0.943203, respectively), and Bi-GRU continued to be the worst performer (0.934836), although still recording decent outcomes. Fig. 7 presents a comparative analysis of specificity across the federated IDS models and conventional baselines.

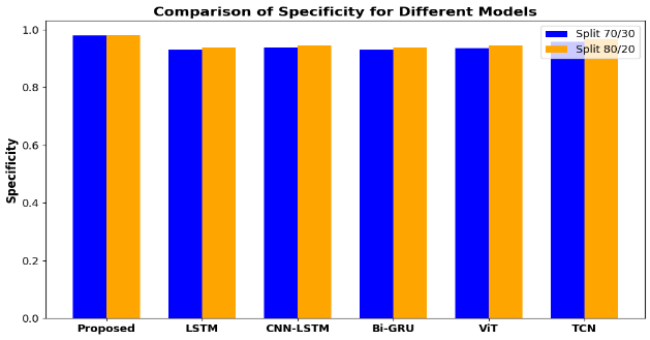


Fig.7. Performance analysis of specificity for federated IDS model

The measures of specificity, indicating the capacity of the model to effectively mark true negatives (benign network traffic), also illustrate the better performance of the suggested federated learning-based IDS. With a 70/30 split, the proposed model achieved the best specificity of 0.979826, marking a very high capability to abstain from false alarms in benign traffic. Following this was the TCN model with 0.958443, followed by CNN-LSTM (0.937437) and ViT (0.936658), having relatively similar performance. Both Bi-GRU and LSTM models exhibited specificity levels around 0.929, indicating slightly higher probabilities of classifying normal traffic as attacks. When assessed on an 80/20 ratio, all models exhibited higher specificity, with the introduced model achieving 0.981485 - retaining its top spot. TCN was second (0.967866), followed by CNN-LSTM (0.946648) and ViT (0.945861), while LSTM and Bi-GRU rose to around 0.938. These findings repeatedly demonstrate that the combination of the proposed model's federated learning with the ViT-TCN architecture yields the most accurate normal network behavior identification, which is extremely important to minimize operational disruptions in real-world IoT deployments. Fig.8 depicts the sensitivity performance comparison for the proposed model and baseline approaches.

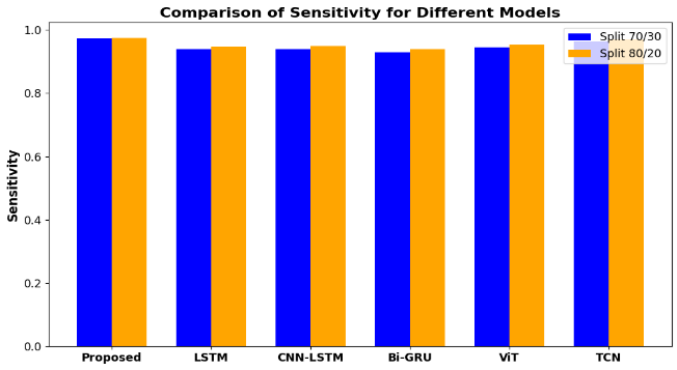


Fig.8. Performance analysis of sensitivity for federated IDS model

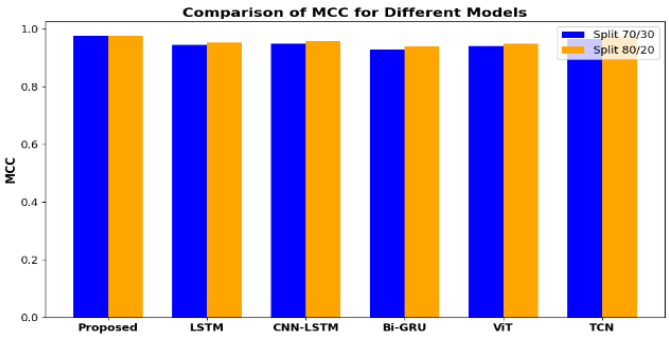


Fig.9. Performance analysis of MCC for federated IDS model

The sensitivity measures, which gauge how well the model accurately identifies true positives (real intrusions), indicate that the proposed federated learning-based IDS is the best solution in both dataset setups. Under the 70/30 split, the highest sensitivity of 0.972765 was attained by the proposed model, indicating its best ability to detect real threats. The TCN model came in at 0.961866, and ViT at a competitive 0.943966. Both CNN-LSTM (0.939498) and LSTM (0.93802) showed moderate sensitivity, with Bi-GRU lagging slightly at 0.928864. When tested on the 80/20 split, all models achieved higher detection rates, with the suggested model achieving 0.974353, still the most sensitive detector. TCN was again the second-best (0.971324), followed by ViT (0.953243), with CNN-LSTM (0.94873) and LSTM

(0.947237) performing similarly. Bi-GRU was still the least sensitive model (0.937988), albeit with still impressive detection abilities. Fig.9 illustrates the Matthews Correlation Coefficient (MCC) comparison between the presented IDS and other models.

The MCC metrics indicate that the proposed federated learning-based IDS outperforms baseline methods in correctly predicting both classes across both dataset splits. Under the 70/30 split, the proposed model obtained the highest MCC value of 0.975456, reflecting a superb overall quality of classification. The TCN model followed with a performance of 0.96428, followed by CNN-LSTM (0.948155) and LSTM (0.944966), which were competitive. ViT had a respectable 0.939777, and Bi-GRU was last with 0.928819. When measured on the 80/20 ratio, all the models had better MCC results, with the proposed model achieving 0.976061 - still in the lead. TCN was second (0.973762), followed by CNN-LSTM (0.957474) and LSTM (0.953243). ViT showed improved performance (0.949012), whereas Bi-GRU remained the lowest performer (0.937943), although it maintained adequate classification ability. Fig. 10 shows the comparative performance of Negative Predictive Value (NPV) relative to the baseline models.

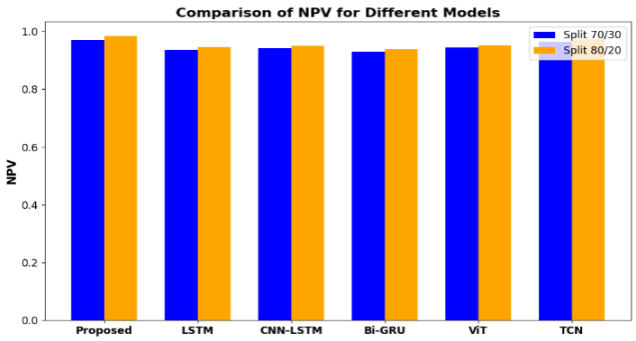


Fig.10. Performance analysis of NPV for federated IDS model

The Negative Predictive Value (NPV) metrics, estimating how well a model can identify genuine negatives (normal traffic cases), show the suggested federated learning-based IDS outperforming other models in both dataset splits. In the 70/30 split, the proposed model achieves the highest performance, with a negative predictive value (NPV) of 0.9699, indicating its strong ability to correctly identify negative instances and demonstrating greater reliability in verifying normal network activity. The TCN model trailed by a small margin with 0.963836, while ViT (0.944063) and CNN-LSTM (0.942211) showed similar performance. LSTM (0.937047) and Bi-GRU (0.929935) reported slightly lower NPV values, indicating slightly higher probabilities of misclassifying normal traffic as anomalous. When tested against the 80/20 ratio, all models performed better in terms of NPV, with the proposed model achieving 0.984598, indicating highly accurate negative prediction. TCN maintained its second position (0.973314), followed by ViT (0.953341) and CNN-LSTM (0.951471). LSTM improved to 0.946254, while Bi-GRU was the weakest (0.93907), but still respectable. As shown in Fig. 11, the proposed model exhibits improved false alarm reduction compared to other models, as indicated by the FPR values.

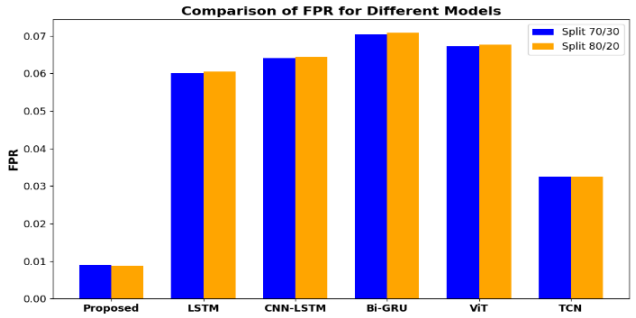


Fig.11. Performance analysis of FPR for federated IDS model

FPR quantifies the percentage of benign instances mistakenly labeled as malicious, demonstrating the superior performance of the proposed federated learning-based IDS in reducing false alarms for both dataset configurations. In the 70/30 configuration, the proposed model had an astonishingly low FPR of only 0.008953, far better than all the benchmark models. The TCN model followed with an FPR of 0.032482, which is approximately 3.6 times higher than that of the proposed model. Other models had much greater false alarm rates: ViT (0.067211), CNN-LSTM (0.064003), LSTM (0.060083), and Bi-GRU (0.070359). This trend continued in the 80/20 test, where the suggested model lowered its FPR further to 0.008796, with TCN retaining its second position (0.032552). The other models retained similar relative performance: ViT (0.067632), CNN-LSTM (0.064392), LSTM (0.060432), and Bi-GRU (0.070812). These findings illustrate that the proposed model's federated learning-based hybrid ViT-TCN architecture achieves significantly lower

false positive rates compared to traditional deep learning strategies, with reductions of up to 7.8 times compared to the Bi-GRU model. Fig.12 illustrates the comparative evaluation of False Negative Rate (FNR) for the baseline and proposed intrusion detection models.

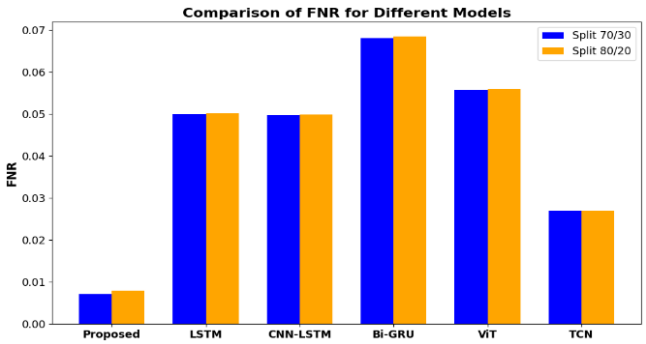


Fig.12. Performance analysis of FNR for federated IDS model

FNR quantifies the percentage of true positives that the model fails to detect, highlighting the superior capacity of the proposed federated learning-based IDS to identify real intrusions while keeping missed threats at a minimum for both dataset configurations. For the 70/30 configuration, the proposed model attained an extremely low FNR of 0.007142 - at least 7 times superior to traditional deep learning methods. The TCN model followed with an FNR of 0.026912, while CNN-LSTM (0.049684) and LSTM (0.049978) exhibited similar but higher miss rates. ViT showed a moderate FNR of 0.055697, while Bi-GRU had the highest miss rate at 0.06807. This performance ranking continued to hold in the 80/20 test, where the proposed model took the lead (0.007976 FNR), followed once more by TCN (0.026926). CNN-LSTM (0.049928) and LSTM (0.050225) continued to perform on par with each other, while ViT (0.056001) and Bi-GRU (0.0685) were the weakest detectors. These results demonstrate that the synergy between ViT's global pattern identification and TCN's temporal analysis in the proposed model minimizes missed threats by approximately 86-88% compared to traditional LSTM/CNN methods.

4.3. Dataset Comparison Analysis

To further validate the generalization capability and robustness of the designed intrusion detection framework, experiments were performed on two different datasets. Both datasets were tested using conventional performance measures, which together give an overall picture of classification performance, susceptibility to various types of attacks, and the ability to tolerate imbalanced classes. The comparison between the two datasets illustrates how various deep learning approaches, namely LSTM, CNN-LSTM, Bi-GRU, ViT, and TCN, fare under different traffic conditions and attack distributions, whereas the proposed model systematically exhibits better adaptability. This observation highlights the stability of the proposed method across both datasets and underscores the importance of dataset diversity in benchmarking intrusion detection systems. Fig. 13 illustrates the dataset-specific performance comparison of the proposed model and baseline models on Dataset 1.

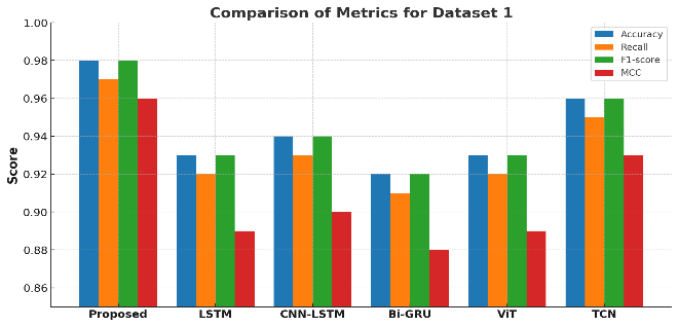


Fig.13. Dataset 1 performance analysis for federated IDS model

The proposed model outperforms all baseline models on all key metrics in Dataset 1, achieving an accuracy of 0.98. It achieves a recall of 0.97, an F1-score of 0.98, and an MCC of 0.96, indicating robust and balanced classification performance. For comparison, the TCN model is the second-best performing with an accuracy of 0.96, a recall of 0.95, an F1-score of 0.96, and an MCC of 0.93. The results of LSTM, CNN-LSTM, Bi-GRU, and ViT are relatively lower and more clustered, with accuracies of 0.92-0.94, recall values of 0.91-0.93, F1-scores of 0.92-0.94, and MCC values of 0.88-0.90. These values attest to the effectiveness of the proposed federated learning-based approach. Fig. 14 shows the dataset-specific performance analysis of the baseline models and the proposed model on Dataset 2.

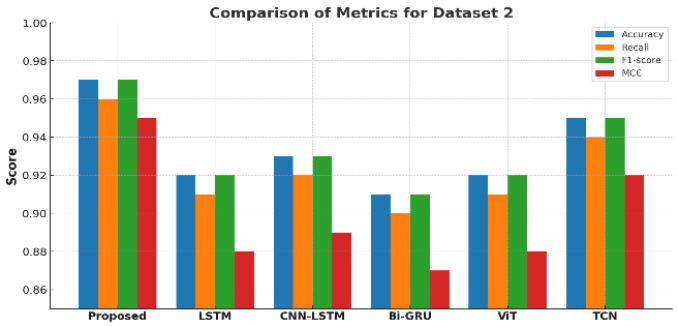


Fig.14. Dataset 2 performance analysis for federated IDS model

On Dataset 2, the proposed model once more has superior performance, with an accuracy of 0.97, two percentage points above its nearest rival, the TCN model (0.95). This dominance is reflected in all other tested metrics: the proposed model achieved a recall of 0.96, an F1-score of 0.97, and an MCC of 0.95, indicating its superior capacity for correctly identifying intrusions while minimizing both false positives and false negatives. The TCN model remains a close contender, achieving a recall of 0.94, an F1-score of 0.95, and an MCC of 0.92. Other models, including LSTM, CNN-LSTM, Bi-GRU, and ViT, are in a lower-performing category with accuracies ranging between 0.91 and 0.93, recall between 0.90 and 0.92, F1-score between 0.91 and 0.93, and MCC between 0.87 and 0.89. These findings verify the strengths and generalizability of the proposed model, demonstrating that its sophisticated architecture maintains a considerable performance advantage across different IoT network intrusion datasets.

4.4. Ablation Study

Fig. 15 depicts an ablation study of the proposed model. The results demonstrate the superior performance of the proposed HEPQSO model across all evaluation metrics, including accuracy, precision, recall, F1-score, and specificity, under various conditions. With preprocessing, it achieves an accuracy of 97.83%, a precision of 97.46%, a recall of 96.94%, an F1 score of 97.68%, and a specificity of 97.41%. Without preprocessing, the model achieves an accuracy of 96.54%, a precision of 96.29%, a recall of 95.73%, an F1 score of 96.34%, and a specificity of 96%. With feature extraction, the system achieves 97.61% accuracy, 97.08% precision, 96.86% recall, 97.03% F1-score, and 97.6% specificity. Without feature extraction, the system maintains 96.83% accuracy, 95.87% precision, 95.98% recall, 95.77% F1-score, and 96.47% specificity. HEPQSO demonstrates its superior performance through its ability to handle classification tasks with minimal preprocessing requirements and its capacity to maintain high classification accuracy.

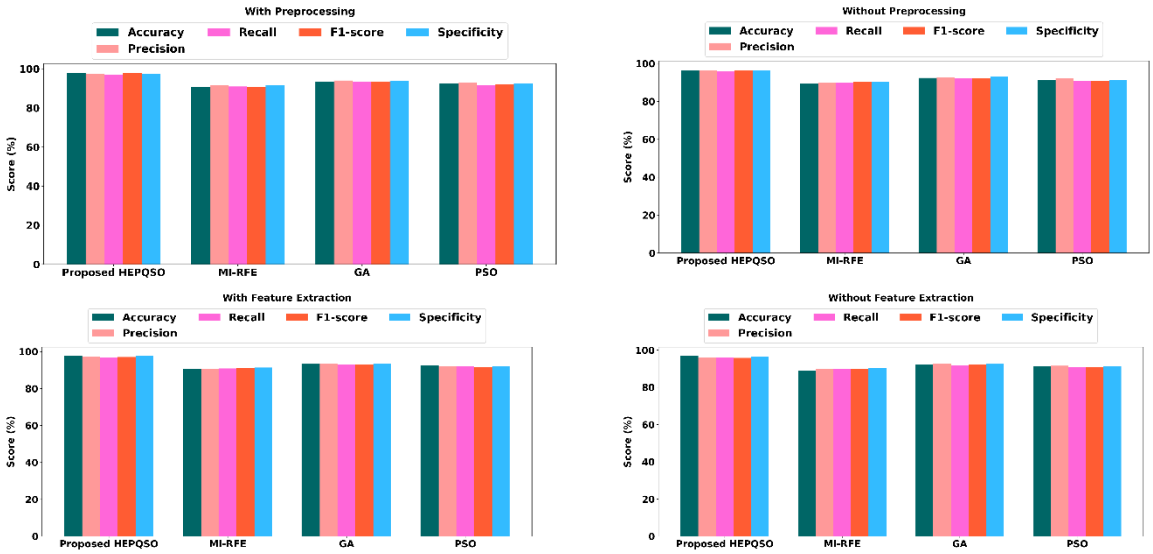


Fig.15. Ablation study

4.5. K-fold Analysis

Fig. 16 depicts the K-fold analysis of various methods, including the proposed HEPQSO, MI-RFE, GA, and PSO, using five folds (Fold-1 to Fold-5). As shown in Fig. 16, the proposed HEPQSO maintains the highest performance across all folds, with accuracies of 98.46%, 98.73%, 98.79%, 98.59%, and 98.59%, respectively. The MI-RFE method achieved accuracies of 95.1%, 95.39%, 95.45%, 95.55%, and 95.4%, respectively, showing moderate performance compared to the proposed model. The GA achieved accuracies of 92.82%, 92.93%, 93.15%, 93.58%, and 93.03%, performing weaker than the proposed model. The PSO achieved accuracies of 94.03%, 94.27%, 94.49%, 94.34%, and 94.26%, outperforming

GA but remaining lower than the proposed model. The proposed HEPQSO system achieves the highest accuracy across all folds while demonstrating stable performance, proving its ability to deliver consistent and superior classification results compared to baseline methods.

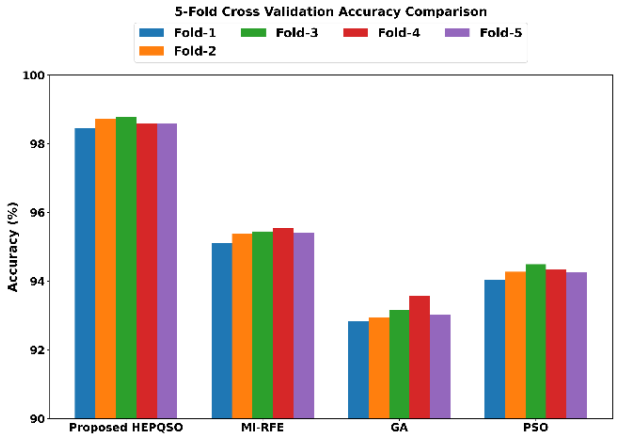
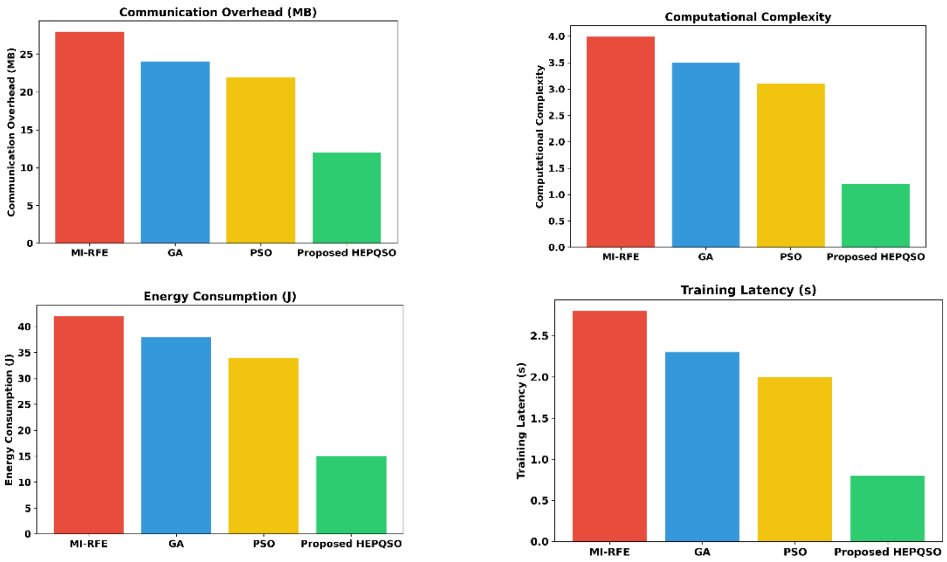


Fig.16. K-fold analysis

4.6. Comparative Analysis

Fig. 17 depicts a comparison of resource efficiency and computational performance among existing models (MI-RFE, GA, PSO) and the proposed HEPQSO model, in terms of communication overhead (MB), model size (MB), training latency (s), computational complexity, and energy consumption (J). As shown in Fig. 17, the proposed HEPQSO model is highly efficient compared to existing models, attaining 12 MB communication overhead, 18 MB model size, 0.8 s training latency, a computational complexity score of 1.2, and 15 J energy consumption. On the other hand, MI-RFE has the highest resource consumption, with 28 MB communication overhead, 45 MB model size, 2.8 s training latency, a computational complexity score of 4, and 42 J energy consumption. The GA and PSO algorithms perform moderately. In this regard, it is evident that PSO (22 MB, 36 MB, 2 s, 3.1, 34 J) outperforms GA (24 MB, 40 MB, 2.3 s, 3.5, 38 J), although both remain lower in overall performance compared to HEPQSO. Thus, it is evident that the proposed HEPQSO model significantly reduces the communication cost, memory usage, time taken, computational cost, and energy consumption.



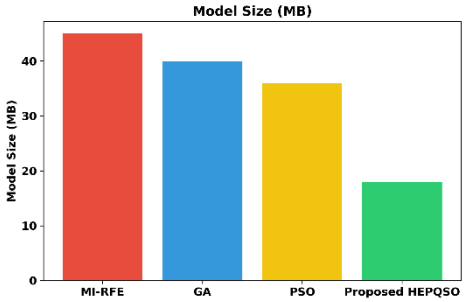


Fig.17. Comparative analysis of the proposed model

4.7. Scalability Analysis

Fig. 18 depicts the effectiveness of the proposed HEPQSO model in terms of aggregation time (s), bandwidth consumption (MB), and communication overhead (MB) with an increasing number of clients. For all the client sizes, such as 10, 20, 50, 100, and 200, the proposed HEPQSO model maintains the lowest values compared to MI-RFE, GA, and PSO, for example, maintaining aggregation times of 10 s, 18 s, 30 s, 50 s, and 75 s, with limited bandwidth and communication overhead of 10 MB, 18 MB, 30 MB, 50 MB, and 75 MB, respectively. Even for the highest client size, that is, 200, the proposed HEPQSO model significantly outperforms MI-RFE (140 s, 140 MB), GA (130 s, 130 MB), and PSO (120 s, 120 MB) in terms of aggregation time, bandwidth, and communication overhead, thus proving the effectiveness of the proposed model in minimizing computational delay with an increase in the number of clients, thus proving the superiority of the proposed model in achieving high performance with less cost in distributed environments.

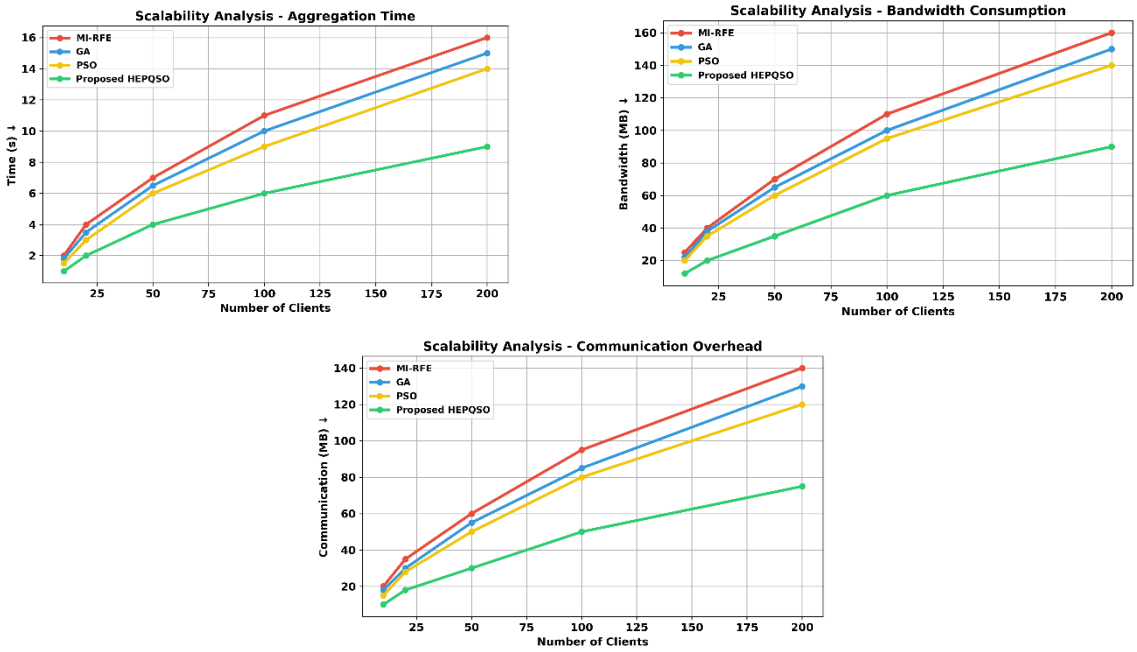


Fig.18. Scalability analysis

4.8. Adversarial Attack Simulations or Defense Evaluation

Fig. 19 presents the adversarial attack simulations and the evaluation of defense mechanisms for various optimization methods. The figure compares the Accuracy under Attack (%), the Attack Success Rate (%), and the Defense Effectiveness (%) of the MI-RFE, GA, PSO, and the proposed HEPQSO. The Accuracy under Attack of the MI-RFE is 72%, with an Attack Success Rate of 40%, making the Defense Effectiveness 60%. The GA and PSO methods have incremental improvements over the MI-RFE. The PSO has an Accuracy under Attack of 78%, with an Attack Success Rate of 30%, making the Defense Effectiveness 70%. The proposed HEPQSO has the highest Accuracy under Attack of 92%, with the lowest Attack Success Rate of 12%, making the Defense Effectiveness 90%, which proves the effectiveness of the HEPQSO in defending against adversarial attacks.

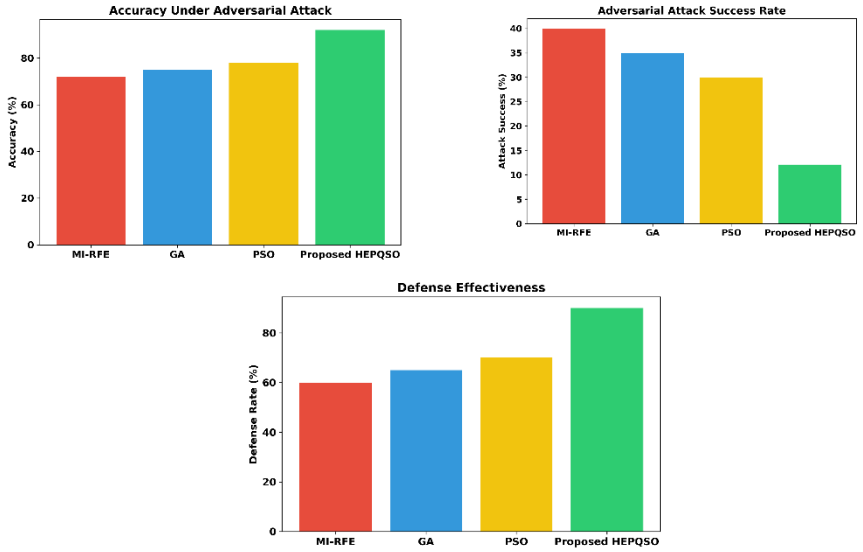


Fig.19. Adversarial attack simulations or defense evaluation

5. Conclusion

The digital infrastructure of modern networks depends on intrusion detection systems, which create security protection, yet their development faces major difficulties because of modern networks' dynamic security environments. The signature-based detection methods used in traditional IDS systems, with their manual rule engineering approach, limit the systems' ability to identify new attack patterns that develop over time. Hence, this research introduces a novel HEPQSO with a deep learning classifier for efficient intrusion detection under federated learning. The preprocessing stage of network traffic data conversion includes complete data cleaning, scaling normalization, and label encoding to achieve uniformity across different local datasets while reducing existing data disturbances. The HEPQSO method is used to find important features through its capacity to explore and exploit results. The selected features are processed by a hybrid deep learning classifier, which uses SGLU to improve its ability to represent non-linear features, and the system uses ViT–TCN to understand worldwide spatial relationships and local time-based patterns of intrusion activity. The experimental results show the higher performance of the proposed approach over benchmark models like LSTM, CNN-LSTM, Bi-GRU, ViT, and TCN. In particular, the proposed model registered 97.88% accuracy, 96.16% precision, 97.45% recall, 96.94% F1-score, 97.80% specificity, and 97.01% MCC, outperforming baselines consistently by large margins. All these enhancements affirm that the combination of HEPQSO with SGLU-ViT–TCN strengthens detection accuracy as well as generalization ability, especially in the case of complicated and minority-class attacks. Moreover, the presented model will be extended by integrating blockchain-based secure aggregation, lightweight encryption for communication resourcefulness, and adaptive reinforcement learning techniques to manage changing cyberattack trends dynamically. Table 2 illustrates the list of abbreviations used in the proposed framework.

Table 2. List of abbreviations used in the proposed framework

Abbreviation	Full Form	Abbreviation	Full Form
IDS	Intrusion Detection System	AD	Anomaly Detection
NIDS	Network Intrusion Detection System	WSN	Wireless Sensor Network
IoT	Internet of Things	IIoT	Industrial Internet of Things
FL	Federated Learning	IoMT	Internet of Medical Things
ML	Machine Learning	SSI	Self-Sovereign Identity
DL	Deep Learning	VAE	Variational Autoencoder
ANN	Artificial Neural Network	FedAvg	Federated Averaging
CNN	Convolutional Neural Network	MCC	Matthews Correlation Coefficient
LSTM	Long Short-Term Memory	FPR	False Positive Rate
BiLSTM	Bidirectional Long Short-Term Memory	FNR	False Negative Rate
GRU	Gated Recurrent Unit	NPV	Negative Predictive Value
Bi-GRU	Bidirectional Gated Recurrent Unit	ReLU	Rectified Linear Unit
SGLU	Spike Gated Linear Unit	MLP	Multilayer Perceptron
ViT	Vision Transformer	SNN	Spiking Neural Network
TCN	Temporal Convolutional Network	LIF	Leaky Integrate-and-Fire

HEPQSO	Hybrid Emperor Penguin–Quokka Swarm Optimization	DCC	Dilated Causal Convolution
EPO	Emperor Penguin Optimization	DoS	Denial of Service
EEPO	Enhanced Emperor Penguin Optimization	DDoS	Distributed Denial of Service
SCNN	Stacked Convolutional Neural Network	MQTT	Message Queuing Telemetry Transport
H-FL	Hierarchical Federated Learning		

All the Declarations and Statements

Author Contributions Statement
S. Shiva Prakash - Conceptualization, Methodology, Data Curation, Software, Formal Analysis, Writing – Original Draft.
Dr. M. Sunil Kumar - Validation, Investigation, Resources, Visualization, Writing – Review & Editing.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement
The authors declare no conflict of interest.

Funding Declaration
Not Applicable.

Data Availability Statement
Not Applicable.

Ethical Declarations
Not Applicable.

Acknowledgment
Thanks to all participants and contributors for making this study possible.

Declaration of Generative AI in Scholarly Writing
The authors declare that no Generative Artificial Intelligence (AI) tools or technologies were used in the preparation, writing, analysis, or editing of this manuscript. This manuscript represents the original work of the authors.

Appendix A/B/C…, with appendix title
Not Applicable.

References

[1] J. Shen, W. Yang, Z. Chu, J. Fan, D. Niyato, and K. Y. Lam, "Effective intrusion detection in heterogeneous Internet-of-Things networks via ensemble knowledge distillation-based federated learning," in *Proc. IEEE Int. Conf. Communications (ICC)*, pp. 2034–2039, Jun. 2024.
[2] R. Lazzarini, H. Tianfield, and V. Charissis, "Federated learning for IoT intrusion detection," *AI*, vol. 4, no. 3, pp. 509–530, 2023.
[3] I. Ioannou, P. Nagaradjane, P. Angin, P. Balasubramanian, K. J. Kavitha, P. Murugan, and V. Vassiliou, "GEMLIDS-MIoT: A green effective machine learning intrusion detection system based on federated learning for medical IoT network security hardening," *Computer Communications*, vol. 218, pp. 209–239, 2024.
[4] R. Beuran, "FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection," *Computers & Security*, vol. 151, p. 104337, 2025.
[5] S. Hajj, J. Azar, J. Bou Abdo, J. Demerjian, C. Guyeux, A. Makhoul, and D. Ginhac, "Cross-layer federated learning for lightweight IoT intrusion detection systems," *Sensors*, vol. 23, no. 16, p. 7038, 2023.
[6] K. Begum, M. A. I. Mozumder, M. I. Joo, and H. C. Kim, "BFLIDS: Blockchain-driven federated learning for intrusion detection in IoMT networks," *Sensors*, vol. 24, no. 14, p. 4591, 2024.
[7] D. Torre, A. Chennamaneni, J. Jo, G. Vyas, and B. Sabrsula, "Toward enhancing privacy preservation of a federated learning CNN intrusion detection system in IoT: Method and empirical study," *ACM Trans. Software Engineering and Methodology*, vol. 34, no. 2, pp. 1–48, 2025.
[8] J. A. de Oliveira, V. P. Gonçalves, R. I. Meneguette, and R. T. de Sousa Junior, "F-NIDS: A network intrusion detection system based on federated learning," Computer Networks, vol. 236, p. 110010, 2023, doi: 10.1016/j.comnet.2023.110010.
[9] Q. H. Nguyen, S. Hore, A. Shah, T. Le, and N. D. Bastian, "FedNIDS: A federated learning framework for packet-based network intrusion detection system," *Digital Threats: Research and Practice*, vol. 6, no. 1, pp. 1–23, 2025.
[10] T. Ohtani, R. Yamamoto, and S. Ohzahata, "IDAC: Federated learning-based intrusion detection using autonomously extracted

anomalies in IoT," *Sensors*, vol. 24, no. 10, p. 3218, 2024.
[11] R. R. Dos Santos, E. K. Viegas, A. O. Santin, and P. Tedeschi, "Federated learning for reliable model updates in network-based intrusion detection," *Computers & Security*, vol. 133, p. 103413, 2023.
[12] G. de Carvalho Bertoli, L. A. P. Junior, O. Saotome, and A. L. Dos Santos, "Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach," *Computers & Security*, vol. 127, p. 103106, 2023.
[13] T. T. Thein, Y. Shiraishi, and M. Morii, "Personalized federated learning-based intrusion detection system: Poisoning attack and defense," *Future Generation Computer Systems*, vol. 153, pp. 182–192, 2024.
[14] J. L. Hernandez-Ramos, G. Karopoulos, E. Chatzoglou, V. Kouliaridis, E. Marmol, A. Gonzalez-Vidal, and G. Kambourakis, "Intrusion Detection Based on Federated Learning: A Systematic Review," *ACM Computing Surveys*, vol. 57, no. 12, pp. 1–65, 2025.
[15] B. Olanrewaju-George and B. Pranggono, "Federated learning-based intrusion detection system for the Internet of Things using unsupervised and supervised deep learning models," *Cyber Security and Applications*, vol. 3, p. 100068, 2025.
[16] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Cyber threat intelligence sharing scheme based on federated learning for network intrusion detection," *Journal of Network and Systems Management*, vol. 31, no. 1, p. 3, 2023.
[17] R. Dhakal, W. Raza, V. Tummala, and L. N. Kandel, "Enhancing intrusion detection in IoT networks through federated learning," IEEE Access, vol. 12, pp. 167168-167182, 2024, doi: 10.1109/ACCESS.2024.3495702.
[18] M. H. Bhavsar, Y. B. Bekele, K. Roy, J. C. Kelly, and D. Limbrick, "FL-IDS: Federated learning-based intrusion detection system using edge devices for transportation IoT," *IEEE Access*, vol. 12, pp. 52215–52226, 2024.
[19] Y. Alhasawi and S. Alghamdi, "Federated learning for decentralized DDoS attack detection in IoT networks," *IEEE Access*, vol. 12, pp. 42357–42368, 2024.
[20] M. Umair, W. H. Tan, and Y. L. Foo, "Dynamic federated learning aggregation for enhanced intrusion detection in IoT attacks," in *Proc. Int. Conf. Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 524–529, Feb. 2024.
[21] D. Javeed, M. S. Saeed, M. Adil, P. Kumar, and A. Jolfaei, "A federated learning-based zero trust intrusion detection system for Internet of Things," *Ad Hoc Networks*, vol. 162, p. 103540, 2024.
[22] M. Althunayyan, A. Javed, and O. Rana, "A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning," *Vehicular Communications*, vol. 49, p. 100837, 2024.
[23] J. Chen, Y. Zhao, Q. Li, X. Feng, and K. Xu, "FedDef: Defense against gradient leakage in federated learning-based network intrusion detection systems," *IEEE Trans. Information Forensics and Security*, vol. 18, pp. 4561–4576, 2023.
[24] S. Sun, P. Sharma, K. Nwodo, A. Stavrou, and H. Wang, "FedMADE: Robust federated learning for intrusion detection in IoT networks using a dynamic aggregation method," in *Proc. Int. Conf. Information Security*, pp. 286–306, Oct. 2024.
[25] S. M. S. Bukhari, M. H. Zafar, M. Abou Houran, S. K. R. Moosavi, M. Mansoor, M. Muaaz, and F. Sanfilippo, "Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability," *Ad Hoc Networks*, vol. 155, p. 103407, 2024.
[26] M. J. Idrissi *et al*., "FED-ANIDS: Federated learning for anomaly-based network intrusion detection systems," *Expert Systems with Applications*, vol. 234, p. 121000, 2023.
[27] R. W. Anwar, M. Abrar, A. Salam, and F. Ullah, "Federated learning with LSTM for intrusion detection in IoT-based wireless sensor networks: A multi-dataset analysis," *PeerJ Computer Science*, vol. 11, p. e2751, 2025.
[28] R. Bensaid, N. Labraoui, A. A. A. Ari, H. Saidi, J. H. M. Emati, and L. Maglaras, "SA-FLIDS: Secure and authenticated federated learning-based intelligent network intrusion detection system for smart healthcare," *PeerJ Computer Science*, vol. 10, p. e2414, 2024.
[29] A. Karunamurthy, K. Vijayan, P. R. Kshirsagar, and K. T. Tan, "An optimal federated learning-based intrusion detection for IoT environment," *Scientific Reports*, vol. 15, no. 1, p. 8696, 2025.
[30] J. Huang, Z. Chen, S. Z. Liu, H. Zhang, and H. X. Long, "Improved intrusion detection based on hybrid deep learning models and federated learning," *Sensors*, vol. 24, no. 12, p. 4002, 2024.
[31] S. Alsaleh, M. E. B. Menai, and S. Al-Ahmadi, "A heterogeneity-aware semi-decentralized model for a lightweight intrusion detection system for IoT networks based on federated learning and BiLSTM," *Sensors*, vol. 25, no. 4, p. 1039, 2025.
[32] N. Chaurasia, M. Ram, P. Verma, N. Mehta, and N. Bharot, "A federated learning approach to network intrusion detection using residual networks in industrial IoT networks," The Journal of Supercomputing, vol. 80, no. 13, pp. 18325-18346, 2024, doi: 10.1007/s11227-024-06153-2.
[33] M. M. Rashid, S. U. Khan, F. Eusufzai, M. A. Redwan, S. R. Sabuj, and M. Elsharief, "A federated learning-based approach for improving intrusion detection in industrial Internet of Things networks," Network, vol. 3, no. 1, pp. 158-179, 2023, doi: 10.3390/network3010008.
[34] A. Belenguer, J. A. Pascual, and J. Navaridas, "GöwFed: A novel federated network intrusion detection system," *Journal of Network and Computer Applications*, vol. 217, p. 103653, 2023.
[35] O. K. ChandraUmakantham, S. Gajendran, and S. Marappan, "Enhancing intrusion detection through federated learning with enhanced Ghost_BiNet and homomorphic encryption," *IEEE Access*, vol. 12, pp. 24879–24893, 2024.

Authors' Profiles

S. Shiva Prakash, a Research Scholar in Computer Science and Engineering at Mohan Babu University, Tirupati. He completed M.Tech in Computer Science from JNTU Anantapur and B.Tech in Computer Science and Systems Engineering from JNTU Hyderabad. He has 14.5 years of academic experience in teaching and academic administration. His academic interest includes Data Structures, Design and Analysis of Algorithms, Computer Networks, Cloud Computing, Big Data Security, Data Mining, and Deep Learning applications. He has published research articles in reputed journals and conferences.

Dr. M. Sunil Kumar is a Professor of Computer Science and Engineering and currently serves as the Dean, Controller of Examinations at Mohan Babu University. With over 18 years of academic and research experience, he has made significant contributions in the areas of Artificial Intelligence, Machine Learning, Cloud Computing, IoT, and Medical Image Processing. He earned his Ph.D. in Computer Science and Engineering from Sri Venkateswara University and completed a Post-Doctoral Fellowship in Soil Nutrition Analysis using Multispectral Satellite Data at Gifu University, Japan. He has guided 10+ Ph.D. scholars (awarded and pursuing) and continues to supervise research across multiple universities. Dr. Kumar has an extensive publication record, with over 124 research papers in Scopus and Web of Science indexed journals, several book chapters, and authored books on Deep Learning, Business Process Reengineering, and Software Engineering.

How to cite this paper

S. Shiva Prakash, M. Sunil Kumar, "Federated Learning-Enabled Intrusion Detection with Bio-Inspired Feature Optimization and Hybrid Deep Neural Classifier", International Journal of Information Technology and Computer Science(IJITCS), Vol.18, No.4, pp.105-127, 2026. DOI:10.5815/ijitcs.2026.04.07