International Journal of Wireless and Microwave Technologies, **2026, 3, 254-276**
Published Online on June 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.03.17

Energy-Efficient and Privacy-Preserving Intrusion Detection in Edge-Based Networks Using Federated Self-Supervised Learning

Paras Kacha*
Department of Computer Science, Pimpri-Chinchwad College of Engineering, Pune, Maharashtra, India
E-mail: paras20pk@gmail.com
ORCID iD: https://orcid.org/0009-0008-2554-2320
*CorrespondingAuthor

Swati Shinde
Department of Computer Science, Pimpri-Chinchwad College of Engineering, Pune, Maharashtra, India
E-mail: swati.shinde@pccoepune.org
ORCID iD: https://orcid.org/0000-0002-8271-3025

Bal Virdee
Department of Communications Technology Research Centre, London Metropolitan University School of Computing and Digital, 166-220 Holloway Road, London N7 8DB, UK
E-mail: b.virdee@londonmet.ac.uk
ORCID iD: https://orcid.org/0000-0001-7203-0039

Ashish Khanna
Department of Computer Science, Maharaja Agrasen Institute of Technology, Delhi-110085, India
E-mail: ashishk746@yahoo.com

Received: 02 March, 2026; Revised: 01 April, 2026; Accepted: 23 April, 2026; Published: 08 June, 2026

Abstract: With the extensive adoption of edge computing and IoT infrastructure, the vulnerability landscape has expanded significantly along with stringent constraints concerning computation, energy efficiency, and data privacy. Traditional centralized IDS solutions tend to be less than ideal for such conditions, as they are highly dependent on centralized data labeling, large-scale computation, and constant traffic sharing. This paper presents FedSSL-IDS, a novel privacy-preserving IDS framework leveraging Federated Learning (FL) and Self-Supervised Learning (SSL), specifically designed for the needs of edge-based network architectures. The solution applies autoencoder-based self-supervised learning to extract informative latent feature representations of unlabeled network traffic, after which federated learning is performed on the lightweight classifier with supervised learning without any raw data sharing. In order to facilitate the implementation of the system on resource-limited edge devices, the system employs advanced model optimization methods, such as magnitude-based pruning and post-training quantization. Performance evaluations of the FedSSL-IDS framework were conducted using the CICIDS2017 dataset in a simulated federated edge environment with class-imbalanced and non-IID client distributions. According to the experimental results, the full precision model reached an average detection accuracy of 96.90% across all classes, whereas the major attack classes, like DDoS and PortScan, achieved impressive class-wise accuracy rates. Moreover, the combination of pruning and FP16 quantization greatly decreases the size of the model and computational cost during inference without compromising its near-native accuracy in detecting intrusions. Nevertheless, aggressive INT8 quantization leads to a substantial reduction in the detection performance of rare classes of attacks such as SQL injection attacks, showing that a compromise must be made between efficient compression and reliable detection in edge scenarios. Even though the presented framework increases the privacy level since there is no raw traffic exchange in federated training, sophisticated privacy-preserving techniques like differential privacy and secure aggregation are not part of the current design.

Index Terms: Intrusion Detection System (IDS), Edge Computing, Federated Learning, Self-Supervised Learning, Privacy- Preserving Security, Energy-Efficient IDS, Internet of Things (IoT), Autoencoder-Based Representation Learning, Model Pruning, Quantization, Resource-Constrained Edge Networks, CICIDS2017 Dataset

1. Introduction

The recent accelerated development of edge computing and the Internet of Things (IoT) has radically altered the manner of modern network design, deployment, and utilization. Edge-based architectures dramatically decrease the latency by moving computation and decision-making nearer to and make better use of bandwidth by moving it nearer to data sources and waste less bandwidth as well as respond immediately to processes like smart healthcare, industrial automation, intelligent transportation systems, and smart cities. Nonetheless, new and challenging cybersecurity issues are also brought by this paradigm shift, which is mainly explained by the extremely distributed, heterogeneous, and resource-scarce edge environments [2].

As opposed to the classic centralized design, an edge-based design has a multitude of geographically spread devices like IoT sensors, gateways, smart cameras, and embedded controllers. These are machines that constantly send volu- minous amounts of network traffic with a strict restriction in regard to processing power, memory capacity and power. Subsequently, the traditional security systems based on data aggregation and computationally intensive analysis are ren- dered to be impractical. Also, edge devices tend to be installed in an untrusted or semi-trusted environment which makes them an easy target of cyber-attacks including DDoS, Port Scanning, botnet propagation, and application-layer attack [20].

The intrusion Detection Systems (IDSs) are important in terms of its protection of the network infrastructure through observing traffic patterns as well as detection of malicious activities. Conventional IDS systems fall into two major categories, which include signature and anomaly systems. Although signature based IDSs are useful in detecting known attacks, it is not useful in detecting a novel or zero-day attack. Machine learning and deep learning-based anomaly-based IDSs have demonstrated potential in identifying the existence of unfamiliar attacks basing their detection on the concept of learning the intricate traffic patterns [8]. Nonetheless, the majority of currently available learning-based IDSs operate well in centralized settings and are sensitive to large amounts of labeled data, huge computing power as well as the deep penetration of sensitive network traffic-capabilities that are incompatible with the operational capabilities and privacy demands of edge-based networks.

Federated Learning (FL) is a strong tool to overcome the issue of privacy since it offers opportunities to train a more powerful global model without giving raw information to multiple clients. In FL, all nodes of the edge perform local training on their own data and only model updates are shared with a central server to be aggregated. This will greatly lower the risk of privacy and overhead communication as well as be in alignment with the decentralized edge architectures which are natural [16]. Although these benefits are available, federated learning is also affected by non-IID distributions of data, low availability of labeled samples to clients, and sensitivity to communication efficiency and model size.

Simultaneously, Self-Supervised Learning (SSL) has become an attractive aspect of the applied research on cybersecu- rity because it is capable of learnings useful representations of unlabeled information, which is available in vast quantities under natural network settings. SSL uses autoencoders, which can learn normal traffic behavior to recreate normal behav- ior, ultimately enhancing generalisation and to resist changing and unfamiliar attack hints [4]. Federated learning can be used with SSL to substantially decrease the use of labeled data and stabilize training with heterogeneous and imbalanced client datasets.

Energy efficiency and deployability is another extremely important issue in edge-based intrusion detection. Deep learning models are not only highly expressive, but they are also typically expensive to compute and memory-intensive. This renders them inapplicable to immediate implementation on roadside gadgets. Pruning and quantization are model optimization techniques, hence, determining the size of models, their inference time and energy usage that do not ne-cessitate notable effects of detection performance [3]. Nevertheless, a great deal of the current line of the IDS solutions focus on detection accuracy, protection of privacy, or computational efficiency alone, instead of a combination of all three dimensions.

Inspired by these issues, this paper presents FedSSL-IDS that is an energy-efficient and privacy-aware intrusion detec- tion system that is adjusted to edge-based networks. The proposed system will combine the principles of federated learning (decentralized and privacy-aware training), self-supervised autoencoder-based representation learning (that should elim- inate reliance on labeled data), and controlled model compression (so that it could be deployed to resource-constrained edge devices). The framework is tested on the CICIDS2017 dataset in realistic scenarios, including class imbalance and non-IID data distribution amongst clients of the federation. Via intensive experimental results, this paper has shown that great accuracy levels can be attained in intrusion detection coupled with enticing levels of privacy assurance and low-cost computation, which makes the offered method suitable in the contemporary edge and IoT setting.

2. Related Work

The rising complexity of cyber attacks as well as the large scale utilization of edge and IoT devices have prompted a great deal of investigations into machine learning based intrusion detection systems. Existing centralized IDS designs are based on the traffic and analytics of the network at a central server and have a scalability factor constraint, a communication factor and privacy factor. Deep learning-based approaches like convolutional neural networks and the

recurrent neural networks have been shown to highly detect intricate attacks, as well as DDoS attacks, probing attacks, and brute-force attacks. Nevertheless, these methods usually require the presence of abundant labeled datasets and efficient centralized computing, and thus are not applicable to implementation in distributed and resource-scanty edge computing [9] [11] [12].

Table 1. Comparative Summary of Related Intrusion Detection Approaches

Author(s)	Method	Advantages	Disadvantages
Pooja Potnurwar et al. [17]	Deep learning IDS	High detection accuracy	High computation; requires labeled data
Babatunde Olanrewaju-George et al. [16]	FedAvg (FL)	Privacy preservation	Sensitive to non-IID data
Hossain et al. [8]	FL-based IDS	Distributed and privacy-aware	No energy optimization
Razvan Beuran et al. [4]	Autoencoder IDS	Detects unseen attacks	High false positives
Amit Kumar et al. [11]	Pruning and quantization	Model compression	Accuracy loss if aggressive
Deng et al. [20]	Privacy-aware FL IDS	Strong privacy guarantees	Reduced detection accuracy

Federated learning has become a perspective in solving the privacy and decentralization issues that do not require the sharing of raw data in order to achieve collaborative model training. Federated IDS Systems Federated systems create models on edge nodes using local, private traffic data, and share model changes with a central server to be aggregated. The method will maintain data confidentiality and will also allow global model enhancement in a distributed location. It has been demonstrated by previous research that federated learning can attain detection quality that is equal to that of centralized training with much lower communications and privacy risks. However, federated IDS systems experience difficulties regarding non-IID distributions of data, heterogeneity of clients, communication efficiency and susceptibility of adverse or poisoned updates, which may influence convergence and stability [6] [10] [14].

The high reliance on labeled network traffic is also another significant constraint of the traditional IDS solutions. In practical applications, labeling attack data is expensive and time-intensive, and not always possible, especially in case of rare or dynamic attacks. Autoencoders and similar methods of self-supervised learning have thus become of much interest in the context of intrusion detection. Autoencoders are trained to encode normal traffic behaviour in compact latent space and detect anomalies as reconstructions or variations in learned feature space. It has been shown in several studies that unsupervised and self-supervised are better than supervised in enhancing generalization, improved labeling and prevention of zero-day attacks within dynamic networks [15] [1] [13].

The deep and self-supervised models can be computationally intensive and can consume high memory resources, and as such, they are not always applicable in edge devices, even despite their effectiveness. To overcome this problem, scholars have considered lightweight models and model optimization methods like pruning and quantization. Magnitude-based pruning is used to eliminate redundant parameters to decrease the model size and inference latency at a slight loss in accuracy. As well, quantization increases efficiency by decreasing the numerical accuracy of inference, and reduces energy consumption on embedded devices. Nevertheless, a previous generation of work has demonstrated that aggressive compression and in particular INT8 quantization may severely hamper the detection performance of minority attack classes, warranting the need to optimize both for balanced outcomes along with matching tradeoffs [18] [17] [7].

Recent studies have explored hybrid approaches combining federated learning, self-supervised learning, and model optimization for scalable and privacy-aware intrusion detection. However, most existing methods mainly focus on individual aspects such as privacy preservation, detection accuracy, or computational efficiency. Limited work has addressed privacy awareness, energy efficiency, non-IID robustness, and lightweight edge deployment within a unified framework, which motivates the proposed FedSSL-IDS approach in this study [19] [5].

3. Dataset Description

For the assessment of the accuracy of the proposed FedSSL-IDS framework, the CICIDS2017 dataset, which is regarded as one of the most popular and widely accepted datasets in the field of intrusion detection, has been selected. This dataset was created by the Canadian Institute for Cybersecurity (CIC) with the intention of designing modern network traffic patterns that are similar to those that occur in the real world. Unlike other intrusion detection datasets, CICIDS2017 was constructed in an environment that simulated realistic attacks on and uses of the networks. The simulation of attacks and legitimate use cases on the networks makes it an excellent dataset for testing an intrusion detection system intended for operation.

The dataset has network traffic collected over several days, featuring normal activity and multiple categories of attacks. The normal activities consist of activities carried out on the internet by regular users such as web browsing, email services, file transfer, multimedia stream usage, and other normal background network activity. In addition to normal activities, there are several types of attacks in the CICIDS2017 dataset, including Distributed Denial of Service Attacks (DDoS), port scanning, brute-force attacks, XSS attacks, and SQL injection attacks.

A flow-level representation of each sample is created by using flow-level statistical attributes computed via CICFlowMeter tool. The attributes represent the traffic pattern and timing behavior as opposed to actual payload content. It makes flow representations suitable to be used in intrusion detection systems preserving the privacy of individuals' personal data. The obtained attributes are comprised of flow duration time, sum of forward/backward packets, packet size statistics, packet inter-arrival times, number of TCP flags, and other statistical parameters describing the network traffic.

The representation of each network flow in CICIDS2017 dataset is performed on the basis of flow-based statistical features, which are obtained using the CICFlowMeter utility. The features provide the behavioral aspect of the network flows, not the payload, hence their application in the design of privacy-preserving IDSs. The feature set includes information such as flow duration, number of forward and backward packets, packet length statistics, inter-arrival times, number of TCP flags, and traffic behavioral statistics.

However, the key characteristic of the CICIDS2017 dataset is the class imbalance that is reflective of the real-world scenario of a network. Benign traffic is present in the largest portion of the dataset, whereas the occurrence of any attack type, including SQL Injection and XSS, is relatively lower. In general, types of attacks such as DDoS and PortScan are abundant as they are volume attacks. Therefore, this is an unfavorable testing environment for intrusion detection systems, where the algorithms are expected to detect attacks, which are less common but very important without generating false positives. The CICIDS2017 dataset can be used to evaluate the performance of machine learning-based IDS methods in such scenarios. On the other hand, the CICIDS2017 dataset can be used for conducting federated learning experiments, where the dataset is split into several simulated clients. Each client is assigned a specific subset of the data whose traffic profile differs from one another, reflecting heterogeneous distribution of data in a network.This partitioning scheme will make sure that the federated learning system is similar to the real-world deployment cases, where edge devices can monitor local traffic trends but are unable to provide raw data because of privacy-related considerations. With the help of training and testing the model in such a decentralized setting, the suggested architecture will be tested within the framework of a realistic privacy-preserving and distributed learning setup.

Overall, the CICIDS2017 data set is quite suitable for evaluating the efficiency, resilience, and applicability of the proposed FedSSL IDS model since it provides a variety of data for testing purposes, mimics real-world traffic patterns, and also features class imbalance, which makes it quite suitable for such purposes, considering its ability to emulate the behavior of attacks in edge networks.

4. Proposed Methodology

4.1. System Overview

The proposed methodology introduces an energy-efficient and privacy-preserving intrusion detection framework specif- ically designed for edge-based network environments, where traditional centralized security solutions are often impracti- cal. The framework integrates Federated Learning (FL), Self-Supervised Learning (SSL), and model optimization tech- niques into a unified and scalable architecture that enables collaborative intrusion detection while strictly preserving data privacy. By ensuring that raw network traffic remains localized at edge devices, the proposed system addresses critical concerns related to data sensitivity, regulatory compliance, and communication overhead.

The overall architecture consists of multiple distributed edge clients and a central federated server responsible for coordinating the training process. Each edge client represents an independent data domain, such as an IoT gateway, access point, or edge node, and continuously collects network traffic generated within its local environment. This traf- fic is inherently heterogeneous, imbalanced, and non-IID, reflecting real-world network conditions where different edge locations observe distinct traffic patterns and attack behaviors. Rather than transferring sensitive traffic data to a central- ized server, each edge client performs local model training using its private dataset. The proposed framework leverages self-supervised learning at the client level to extract meaningful feature representations from unlabeled traffic, thereby reducing dependence on annotated data. Only model parameters or updates are shared with the federated server, ensuring that sensitive traffic information is never exposed beyond the local edge environment. The federated server aggregates these updates using a secure and communication-efficient aggregation mechanism to construct a global intrusion detection model.

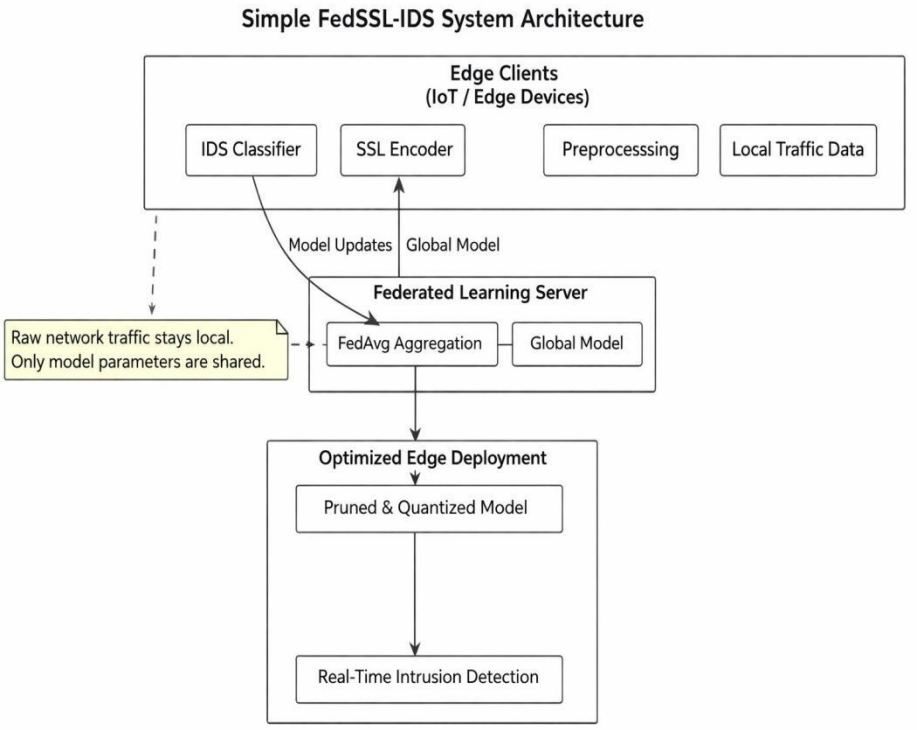

Fig. 1. System architecture of the proposed FedSSL-IDS framework for privacy-preserving and energy-efficient intrusion detection in edge-based networks.

The global model is iteratively refined over multiple federated learning rounds, during which updated parameters are redistributed to participating clients for further local training. This iterative process allows the model to continuously adapt to evolving attack patterns while maintaining robustness under non-IID data distributions. In addition, the incorporation of model optimization techniques, such as pruning and quantization, ensures that the resulting intrusion detection model remains lightweight and suitable for deployment on resource-constrained edge devices.

Overall, the proposed system architecture achieves a balanced combination of detection accuracy, privacy preservation, scalability, and deployment efficiency. By integrating FL and SSL within a carefully designed edge-centric framework, the proposed methodology provides a practical and effective solution for intrusion detection in modern edge-based network environments.

The system architecture that can be found in Figure 1 concerns the general system structure of the proposed FedSSL- IDS framework. The structure comprises of several distributed edge clients, a federated learning server and an optimized deployment layer. That is, every edge client localizes traffic over the network, conducts preprocessing and self-learnt feature learning through an encoder, and invokes intrusion classification to a lightweight classifier. Raw network traffic is contained entirely on the edge devices to maintain data privacy and model parameters are sent only to the federated server. Collaborative training is coordinated by the federated learning server which aggregates updates of the client-side model with the Federated Averaging (FedAvg) algorithm to build a global model based on intrusion detection. This inter- national model is re-distributed to edge customers on a regular basis to be refined further. Following federated training, pruning and quantization model optimization methods are used to minimize computational and memory expenses. Then, the optimized model is deployed to edge devices allowing intrusion detection in real-time and energy-efficiently.

4.2. Data Collection and Preprocessing

The data of network traffic is not only constantly gathered and measured at every edge node, but also expressed in flow-level statistical indicators of communication behavior, without inspecting the payload. These characteristics are the number of packets and bytes forward and reverse as well as flow life, interval-between-arrivals, transport-layer header flags, and aggregated calculated traffic behavior metrics. These flow representations are ideal in privacy-aware intrusion detection, not requiring the sensitive contents of packet headers to be revealed and still able to represent significant behavioral patterns of normal and attacking traffic.

Before model training, a local preprocessing pipeline is used across the edge clients to guarantee the quality and consistency of data. Raw traffic may be full of lost or infinite values, or noises, such as the loss of packets, measurement errors, or anomalies caused by an attack. To deal with this invalid and infinite values are first detected and eliminated or substituted. The median imputation method is employed to fill in missing values, and it is resistant to outliers that are usually present in the case of attack traffic and does not distort the feature distributions.

After data cleaning, feature scaling and normalization is done in order to normalise the features ranges and enhance numerical stability in training MLPs. The standard normalization methods, i.e., z-score normalization or min-max scaling, are done on an individual client in order to stabilize the gradient updates and speed up the convergence of a model. This is especially relevant in the context of federated learning, where the size of the features may vary across clients in a detrimental way to global aggregation of the model.

In case the traffic labels can be obtained, they are coded into multiclass categorical representations, which are benign traffic, and the various attack types. Self-supervised learning is also used in situations with few labeled pieces of data to utilize unlabeled traffic and use it to derive meaningful representations. Notably, the entire processing, feature trans- formation and labeling process are conducted solely on the edge client tier, meaning that raw traffic data and sensitive metadata are not ever forwarded beyond the local scope.

Local data collection and preprocessing guarantee the data privacy and communication overhead reduction and enable heterogeneous edge environments with a diverse set of traffic parameters, allowing the proposed framework. Such a decentralized preprocessing plan lays a strong basis to a powerful federated and self-managed learning, allowing powerful intrusion detection through realistic edge deployment situations.

4.3. Self-Supervised Representation Learning

In order to efficiently learn useful traffic representations and be less reliant on labeled data, the proposed FedSSL-IDS system uses edge client self-supervised representation learning. Self-supervised learning is a type of learning where the model is trained by identifying high-level and discriminative features in large amounts of unlabeled network traffic by specifying an auxiliary learning objective, which does not need manual annotation. Such a strategy is especially appropriate to intrusion detection in edge environments, where labeled attack data tend to be low in number, incomplete or expensive to have.

In this context, the self-supervised representation learning problem is formulated within a deep autoencoder framework. Here, the autoencoder is trained locally for each of the edge clients with unlabeled data of the network traffic. In an autoencoder, the encoding process reduces the input flow-level feature vectors into low-dimensional latent space vectors, which are further decoded into the original vectors by the decoding process. In our proposed autoencoder design, there are two hidden layers, which consist of 64 and 32 nodes each, followed by the latent representation of dimension 16, and finally followed by symmetric decoding layers with ReLU activations. Here, the MSE metric is utilized for the reconstruction objective.

$$z_i = f_{\theta e}(x_i) \tag{1}$$

This equation represents the encoder function, which transforms the high-dimensional input traffic feature vector x i into a compact latent representation z_i that captures essential traffic characteristics.

$$\hat{x}_i = g_{\theta_d}(z_i) \tag{2}$$

The decoder reconstructs the original input features from the latent representation, ensuring that the learned encoding preserves meaningful structural and behavioral information of network traffic.

$$L_{SSL} = \frac{1}{N} \sum_{i=1}^{N} \|x_i - \hat{x}_i\|_2^2 \tag{3}$$

Reconstruction loss is defined as the difference between the original and reconstructed input and is reduced during learn- ing to learn informative representations in the absence of labelled data. The trained latent representations have various benefits in regard to intrusion detection. Originally, they decrease the dimensionality of the input space, resulting in better computational efficiency and reduced convergence time in the subsequent supervised downstream learning. Second, they can record natural patterns of traffic less susceptible to noise and other small variations which lead to robustness when the network is dynamic and heterogeneous. Third, since the representations are learned at each edge client, they will adapt to the local traffic properties yet will be able to contribute to consistent model across the world, through federated learning. Following the self-supervised training, decoder part of the auto encoder is removed and the encoder is left as a fixed feature extractor. Frozen or fine-tuned on the subsequent supervised training, the encoder weights rely on the availability of client-specific data. This design fewer trainable parameters are trained in federated learning, and it needs fewer communication overheads, as well as can enhance training stability across non-IID clients.

Notably, self-supervised representation learning occurs on the client level all the way up, which guarantees that raw traffic data never exits the edge device. The federated learning is still private but allows joint learning because only encoder parameters or generated model updates are exchanged in the framework of the federated aggregation. In this way, by exploiting self-managed learning, the proposed FedSSL-IDS framework makes good use of not only the labeled data, but also the unlabeled data and improves the quality of features and provides a solid basis of effective and scalable intrusion detection in an edge-based network setting.

4.4. Supervised Intrusion Classification

Following self-supervised feature learning, a lightweight supervised classifier is attached to the frozen encoder at each edge client to perform multi-class intrusion detection. The classifier is trained using the available labeled samples and maps the learned latent representations to predefined traffic categories, including benign traffic, DDoS attacks, port scan- ning, and other malicious behaviors. This two-stage learning strategy allows the model to leverage rich representations learned from unlabeled data while still benefiting from supervised labels where available.

$$y_i = Softmax(Wz_i + b) \tag{4}$$

This equation computes the probability distribution over intrusion classes by applying a softmax function to the clas- sifier output based on the latent feature representation.

$$L_{\text{sup}} = -\left(\frac{1}{N}\right) \backslash sum_{i=1}^{N} \backslash sum_{c=1}^{C} y_{ic} \log (\backslash hat(y)_{ic}) \tag{5}$$

Categorical cross-entropy loss is used to train the classifier for multi-class intrusion detection by penalizing incorrect class predictions. Keeping the encoder frozen during supervised training significantly reduces the number of trainable parameters, leading to improved training stability and faster convergence at the client level. This design choice also min- imizes communication overhead during federated learning, as only the lightweight classifier parameters and selected en- coder updates need to be transmitted to the federated server. By limiting the scope of trainable parameters, the framework becomes more scalable and suitable for deployment across large numbers of edge clients. The clear separation between representation learning and classification ensures that all clients operate within a consistent and shared feature space, even when local data distributions differ significantly. This consistency is particularly important in federated learning environ- ments characterized by non-IID and heterogeneous data, as it facilitates effective model aggregation and prevents client drift. Moreover, the use of a lightweight classifier enables rapid adaptation to new or evolving attack patterns through localized fine-tuning, without requiring full retraining of the feature extractor.

Overall, this design enhances the robustness, efficiency, and adaptability of the proposed FedSSL-IDS framework, ensuring reliable intrusion detection while maintaining privacy preservation and low computational overhead in edge-based network environments.

4.5. Federated Learning Framework

The designed FedSSL-IDS framework supports the collaboration aspect of training on a global scale, facilitated by a federated learning framework in which several distributed edge clients learn a global intrusion detection model collectively but do not exchange raw network traffic. Firstly, during each federation round, the server hosting the central node will initialize and refresh the existing global model and send the model parameters safely to all clients associated with edges. This setup allows every client to initialize at the same model state, which makes stable and coordinated learning in the distributed environment possible.

When a global model is received, every client of the edge does local training using network traffic data that the client has access to, and that has been gathered locally. Local training is done during few epochs to trade off effectiveness of learning against computation efficiency and communication cost. In this step, the clients revise the model parame- ters, depending on their individual traffic distributions that can vary considerably depending on location, owing to the heterogeneous use patterns of the network and the occurrence of attacks.

$$\theta\_k\text{^}(t+1) = \theta\text{^}t - \eta \nabla L\_k(\theta\text{^}t) \tag{6}$$

The formula describes the local model update, which client is going to perform based on the global model parameters that are being refined with a help of the local loss function and the private data available to the client.

This makes it possible because this localized learning allows the framework to learn various characteristics of traffic whilst still adhering to data locality and privacy requirements.

Once a local training is done, every client only sends back its new model parameters or gradients to the central server. Notably, no uncooked traffic information, values of features and sensitive metadata are exchanged in the process, which guarantees high privacy levels. In order to minimize communication overhead further only necessary parameters, usually the ones related to the lightweight classifier and selected encoder layers are sent. The federated learning process that is designed this way is scaled and applicable to single resource-limited edge settings.

The central server combines the client updates obtained with Federated Averaging (FedAvg) algorithm which calcu- lates a weighted average of client model parameters according to the size of local datasets.

$$\theta^{(t+1)} = \sum (k=1)^{K} (N_k/N) \theta_k^{(t+1)} \tag{7}$$

The server calculates the global model through a weighted average of the locally trained models in a way that the clients with a large set of models contribute in the same direction to the updated global model.

This aggregation approach can guarantee that the overall global model would capture the knowledge acquired in all the clients involved and take into consideration the discrepancies in information volume and distribution. The collective model is then re-distributed to clients, and one round of federation is finished.

The above process is iteratively performed for several rounds of federation until certain criteria of convergence such as performance stabilization during the validation process or a specific number of iterations of communication are met. In each iteration, a random subset of edge clients is chosen to emulate realistic client participation in distributed systems. Raw data generated by the network is kept local at the edge devices, whereas lightweight updates of the models are sent back during the training process to minimize communication costs and ensure privacy.

4.6. Model Optimization for Edge Deployment

To enable efficient deployment on resource-constrained edge devices, the proposed methodology incorporates a com- bination of model optimization techniques aimed at reducing computational complexity, memory footprint, and inference latency without compromising detection reliability. These optimizations are essential for real-time intrusion detection in edge and IoT environments, where hardware resources and energy availability are limited.

During local training at each edge client, magnitude-based pruning is applied to systematically remove redundant and low-importance weights from the neural network.

$$w_i=0, if |w_i|< \\ w_i=w_i, otherwise \tag{8}$$

Weights with magnitudes below the threshold are removed, reducing model complexity while preserving parameters that contribute most to intrusion detection performance.

Pruning is performed gradually over training iterations to avoid abrupt performance degradation and to allow the model to adapt to the reduced parameter space. By eliminating unnecessary connections, pruning significantly reduces model size and computational cost while preserving the most discriminative features learned during training. This process also contributes to improved generalization by mitigating overfitting and enhancing model robustness.

After the completion of federated training, post-training quantization is applied to further compress the aggregated global model.

$$w_q = round(sw) \tag{9}$$

Quantization decreases the numerical accuracy of weights and activations, which decrease the amount of memory required and also speeds up inference. The analysis of the trade-off between computational efficiency and detection performance is provided by the evaluation of both FP16 and INT8 quantization schemes. FP16 quantization operates with performance near to full-precision model, and achieves significant memory and cost of computation saving, which is why it is appropriate to mixed-precision edge hardware. On the other hand, higher compression ratios are obtained with INT8 quantization, but with a significant loss to the accuracy of detection, especially with minority attack classes depending on the small differences in features.

All in all, the integrated process of pruning and quantization can allow the presented system of intrusion detection to attain a light and ready-to-deploy model. The optimization methods greatly minimize model size, memory, and inference time to guarantee that the developed FedSSL-IDS structure can execute effectively in real-time on edge hardware and remain able to be intrusion detectors with good results.

Phase 1 is concerned with local training and selection of a best intrusion detection model before the introduction of federated learning. The workflow will start with a selection of the CICIDS2017 dataset since it will undergo data preprocessing that will comprise the following stages: cleaning, normalization, and categorical encoding to facilitate information consistency and training stability. This is followed by feature extraction which produces a 78-dimensional numerical feature vectors used to describe flow-level and behavioral properties of network traffic.

The evaluation of several classical machine learning models, such as the Logistics Regression, the Decision Trees, the random forests, the support vectors, the nearest neighbors and the XGBoost are conducted to set the minimum perfor- mance. This is concomitantly evaluated by a systematic search of a set of neural network architectures of different depths and output configurations. Pruning and quantization are model optimization procedures that are used to determine trade off between accuracy in detection and computational efficiency. The most efficient and lightweight neural architecture is chosen on the basis of comparative assessment to be integrated into the federated learning mechanism in the framework of Phase 2.

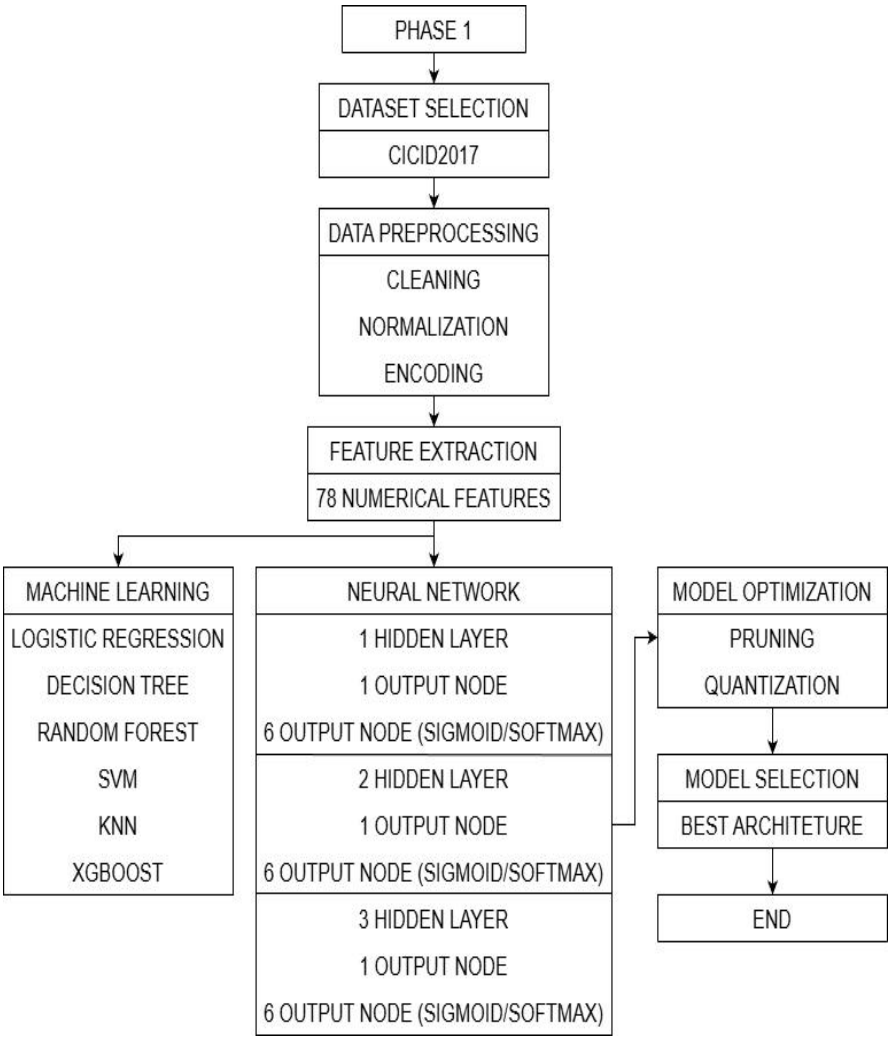

Fig. 2. Phase 1 workflow illustrating dataset preprocessing, feature extraction, classical and neural model experimentation, and model optimization for selecting the best intrusion detection architecture

4.7. Deployment and Inference

The ultimate optimized world model will be implemented on the edge devices to facilitate real-time intrusion detection in the operational networks. When deployed, local traffic within network at an edge node is constantly monitored and handled. The raw traffic is then fed through the preprocessing pipeline that was used with the training, making the features similarly represented. The raw traffic passes through the lightweight classification head and the trained self-supervised encoder and then it makes a benign or malicious flow prediction decision.

Portability and streamlinedness of the deployed model is important and contributes to practical edge deployment. The model has a small memory footprint and low computational requirements by using pruning and quantization and thus inference times are short and less energy is used. This allows network traffic to be monitored continuously and uninterrupted without overloading scarce edge resources like CPU, memory or power supply. Consequently, the intrusion detection system has the ability to run continuously even on the settings like access points, IoT gateways, and edge servers.

Moreover, the deployment plan promotes scalability and customization in a variety of edge settings. The federated learning process enables the global model to continuously be updated with the new trends in traffic patterns and attack behavior without having to collect data centrally or referephase training. The efficient distribution of updated models to the edge devices can be implemented to support the intrusion detection system, where the system is not only effective to the current threats but also does not compromise the privacy of data. Phase 2 entails the essence of federated and deployment-oriented approach of the suggested methodology. The global intrusion detection model is initiated on a federated learning server and participates in collaborative training among numerous distributed edge clients. The loading of its local dataset and preprocessing are done individually by every client to ensure the privacy of data. Self learning refers to self-supervised learning whereby an autoencoder trained on unlabeled traffic learns to represent robust features. The decoder is then discarded and the encoder is frozen after training to be used as a constant feature extractor.

An instructor classifier is then developed based on learned representations on locally available labeled data. In local training, magnitude based pruning is used sequentially to simplify models and enhance efficiency. The clients only send the updated model weights to the federated server which adds them together by the Federated Averaging algorithm to generate an updated global model. Once federated training is completed, the masks are lifted and post training quantization is used to create deployment ready models with FP16 and INT8 formats. The resulting optimized model can be used to detect intrusion in real time using edge devices that have resource constraints.

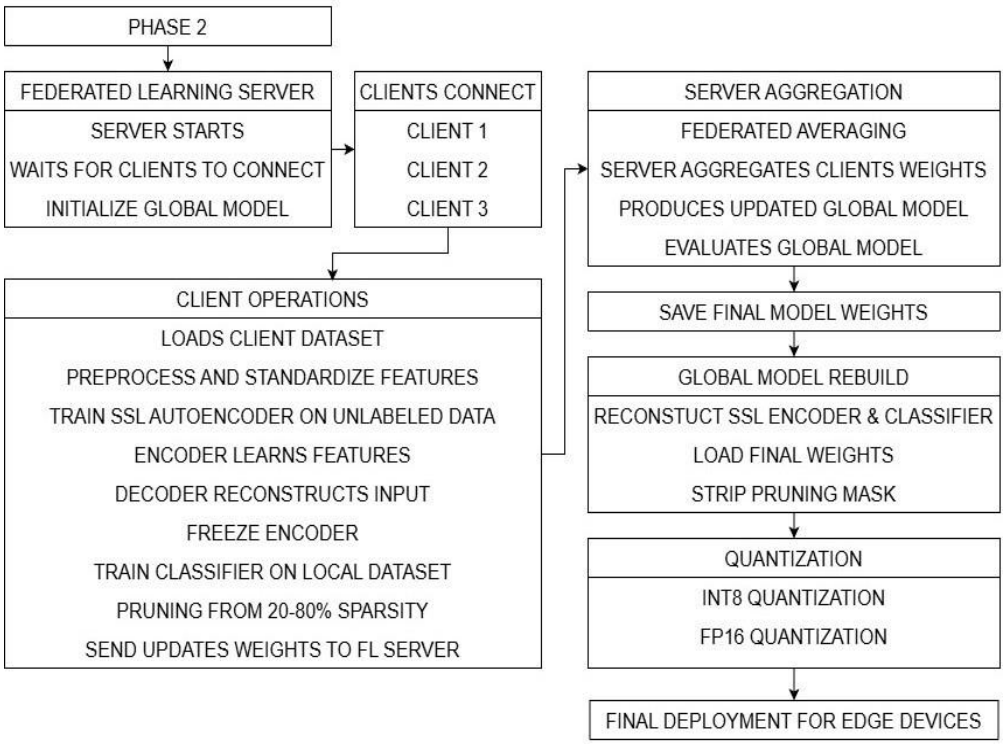

Fig. 3. Phase 2 workflow showing federated learning with self-supervised representation learning, client-side pruning, server aggregation, and post-training quantization for edge deployment.

5. Experimental Setup

All experiments were implemented via Python programming using the TensorFlow/Keras platform within a simulated federated edge scenario. Experiments were executed on a computer system equipped with an Intel Core i7 processor and 16 GB RAM, along with GPU support from NVIDIA. The federated learning approach simulated the presence of multiple edge clients in a realistic setting without exchanging network traffic data.

COD completed CICIDS2017 dataset is preprocessed and divided to many edge clients to alleviate to obtain realistic network environment conditions. Data is and the distribution is non-IID and class imbalanced, as it represents the severally varying traffic characteristics commonly seen at various edge locations. Each of the clients works independently and trains its local model on its own subset of data and without falling raw network traffic and sensitive features to the central server. This architecture will enable the test of the suggested framework within the privacy-preservation and decentralized learning environment.

A self-supervised auto encoder is then initially trained at every edge client on unlabeled network traffic to learn compact and discriminative latent representations. This kind of self-training phase allows one to make good use of huge amounts of untagged data, characteristic of more realistic network monitoring applications. After doing the representation learning, a supervised classifier is then trained on the learned latent features on some labeled samples with the constraint of the use of a lightweight supervised classifier. This is a two-phase process of learning that provides a strong extraction of features at the lowest level of use of labeled data.

Federated training is carried out in several communication rounds based on the Federated Averaging (FedAvg) algo- rithm. The global model is aired to the clients involved at the onset of each round. Each customer uses a batch size of 32, which is effective in training at the expense of the optimal balance between learning and computational efficiency. The optimization algorithm is Adam as it has adaptive learning features, and the loss function is categorical cross-entropy to perform supervised multi-class prediction. Clients after training their models locally update them to the server, which aggregates them to create an updated global model. Repeat of this until convergence is achieved.

The model optimization technique is introduced to evaluate the appropriateness of the proposed framework to be

deployed on limited resources edge devices. Magnitude-based pruning is added to the process of training gradually so that the model can be made less complex by eliminating unimportant parameters without adversely affecting detection. Once federated training is done, post training quantization is then implemented to further compress the global model. The FP16 quantization scheme and INT8 quantization schemes are compared to examine the trade-off between the computational efficiency and detection performance with reduced numerical precision.

A set of standard classification metrics taken into account for model performance, such as accuracy, precision, recall and F1-score, are used to determine the general performance in terms of detection. Further, the confusion matrices are also analyzed to investigate the behavior based on classes and also the misclassification patterns especially in class- imbalanced scenarios. The model can also be assessed in terms of the discriminative capability under various decision thresholds through ROC and Precision -Recall curves. These measures of evaluation combined will give a comprehensive analysis concerning the performance, strength, and realistic usability of the proposed FedSSL-IDS framework in edge- based network settings.

Table 2. Hyperparameter Settings Used in the Proposed FedSSL-IDS Framework

Parameter	Value
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Local Epochs	5
Federated Rounds	20
Latent Dimension	16
Activation Function	ReLU
Output Activation	Softmax
Reconstruction Loss	Mean Squared Error (MSE)
Pruning Ratio	20–40%
Quantization Types	FP16, INT8

6. Implementation Details

The proposed FedSSL-IDS framework is executed with the use of the modular software architecture that facilitates privacy-sensitive training, scalability, and efficient operations on edges-based settings. Everything is built in Python, with deep learning models built and trained in TensorFlow and Keras. The general architecture is clientserver, in which many simulated edges clients come together and interact with a central federated server to cooperatively learn an intrusion de-tector model not communicating the underlying network traffic. This intrusion detector model consists of self-supervised autoencer and a lightweight supervised classifier. Training is performed by each client directly on unlabeled network traffic on an autoencoder to learn a set of compact latent features of input features. Once the decoder has been trained on itself, the decoder is discarded and the encoder is frozen to serve as a fixed feature extractor. The design enhances training stability, minimizes training parameters, and minimizes communication overheads in the process of federated learning.

Federated learning works by the repetition of communication round with the central server. At the beginning of every round, the server gives out the latest worldwide model to the clients involved. Each customer is training on his/her own data locally and submitting new model parameters back to the server. The Federated Averaging (FedAvg) algorithm is used to implement model aggregation, such that the global model captures knowledge gained on heterogeneous and non- IID clients datasets at the same time that preserves the privacy of the identifying data. The training on the local computer is conducted by the Adam optimizer alongside a fixed learning rate and the batch size of 32. Within the context of each federated round, the clients run a specified number of local epochs to trade off a learning and communication efficiency. Categorical cross-entropy loss is the optimal classification loss that is used. The convergence is stable using this training strategy but with minimal computational overhead that fits well in edge environment.

In order to save on resources, magnitude-based pruning is used when training to gradually eliminate the least signif- icant weights of the neural network. Federated training is followed by the removal of pruning masks to provide a small dense model. The last international architecture is also optimized with the post-training quantization in which FP16 and INT8 precision format are compared to decrease the model size and inference latency. The optimizations can be used to effectively detect intrusions in edge computing devices including embedded platforms and IoT gateways. To be evaluated and reproducible, model checkpoints are stored at the end of every federated round, and performance metrics, including accuracy, precision, recall, F1-score, and confusion matrices, are calculated with an unseen test set. The implementation is made to be scalable such that it can be adapted to varying datasets, attack contexts, and deployment contexts with ease and that way illustrates the practicality of the proposed FedSSL-IDS framework.

7. Results & Analysis

This part contains an in-depth experimental analysis of the suggested Federated Self-Supervised Intrusion

Detection System (FedSSL-IDS), its efficiency, reliability, and its applicability to the purpose of deployment, in the environment of edges. The experimental analysis will evaluate how the system correctly identifies malicious network operations and yet retain the privacy of its data using federated learning and balances low computational cost using the method of optimization of models. The proposed framework integrates the self-supervised representation learning and federated training, to overcome the issues of limited labeled data, non-IID distributions of traffic, and resource constraints which are very likely to be faced at the network edge.

The test is on various aspects of system performance, such as the overall-level detection, and by the class, the preci- sion, recall, and F1-score, and error rate across the attack types. Special attention is given to the fact that the model is evaluated using realistic conditions and parameters under the conditions of imbalance of classes and uneven distributions of client data. These conditions are very realistic with respect to intrusion detection scenarios of real life settings; there is predominance of benign traffic with a small yet dangerous proportion of attack type. Besides the detection effectiveness, the current section explores the resiliency of FedSSL-IDS framework under federated learning. It is observed that the decentralized training has an effect on the fusion of the models and also generalization through evaluating between the federated rounds and comparing the result of the detection between various edge clients. The presented analysis outlines the capability of the proposed system to collaboratively learn on distributed data sources without exchanging the raw network traffic and therefore, maintain a high level of privacy and attain performance on par with centralized training methods. In addition, the effectiveness and deployability of the proposed system are investigated with the help of the ex- perimental results that identify the impact of model pruning and post-training quantization. These optimization methods are essential in minimising model size, memory footprint, and inference latency to allow real-time intrusion detection on resource constrained edge devices. The trade-offs of compression versus detection accuracy are explored thoughtfully with the special concern of the performance implications of the minority attack classes. This analysis shows in the section that the optimization strategies could be realized proper and help to achieve the high levels of energy efficiency alongside the stable detection features.

On the whole, the outcomes of this part supply powerful empirical evidences that proposed FedSSL-IDS framework achieves the equilibrium between high detection rates, privacy, and efficiency. It is shown in the analysis that the com- bination of federated learning, self-supervised representation learning and controlled model optimization is feasible to create an effective intrusion detection system that can be deployed in contemporary edge-based network settings.

7.1. Overall Detection Performance

The general effectiveness of the suggested intrusion detection model is measured in terms of conventional classifica- tion measurements, such as accuracy, precision, recall, and F1-score. The precision federated model has a high detection rate of about 96.9, which suggests that the model has a high capability to discriminate between healthy and harmful net- work traffic. This finding proves that self-supervised representation learning, using federated training, works successfully when data is non-IID.

The low false-positive rate is represented by the high accuracy of the model, the high accuracy is critical in reducing the false alarms of the model in real-world settings. Meanwhile, the high recall value is an additional confirmation that the model is capable of recognizing most attack instances, which reduces the risk of being intruded without its notice. The obtained F1-score demonstrates a fair trade-off between the precision and recall, which guarantees a sufficient and trustworthy detection performance. On the whole, these findings confirm the strength and efficacy of the proposed FedSSL-IDS system in intrusion detection in edge-based systems. The general performance of the suggested FedSSL- IDS framework was provided in Figure 4, which reports the accuracy, precision, recall, and F1-score. The accuracy of the model is high, with a figure of 96.90 which reveals that the model is highly effective to identify both benign and malicious network traffic. The percentage of precision is 95.96, which indicates that the rate of false-positive is low, indicating that benign traffic is not commonly assigned to malicious traffic.

The recall value of 96.90 is a high promoter that the model is effective in detecting most of the instances of intrusion something that is paramount when using intrusion detection systems in security sensitive settings. The F1-score demon- strates a good consistency and reliability in the detection performance as it shows a balanced trade-off of precision and recall with 96.41% F1-score. Comprehensively, all these findings support the applicability of the suggested FedSSL-IDS framework in having successful and sound intrusion search in cases of federated learning.

Figure 5 shows the performance of the proposed FedSSL-IDS model on various numerical precision parameters, i.e., original full-precision model, quantization with FP16, and the quantization with INT8. These findings demonstrate that FP16 quantization results in operation that were very nearly identical to the original model, which means that arithmetic with lower precision can be successfully employed without losing much of the detection accuracy. Conversely, INT8 quantization results in observable decrease in performance especially in terms of precision, recall and F1-score metrics. The reason behind such degradation is the weak numeric representation of the INT8 precision, which is also a major factor on the capability of the model to represent small changes in features, particularly the minority attack classes. In general, the figure implies that FP16 quantization compromises most appropriately between the aspects of computational and detection reliability with regards to the deployment on the resource-constrained edge equipment.

Table 2 & Table 3 summarizes the performance of the proposed FedSSL-IDS framework under different optimization settings. The results indicate that FP16 quantization achieves the most favorable balance between detection accuracy and deployment efficiency, while aggressive INT8 quantization leads to reduced performance, particularly for minority attack classes.

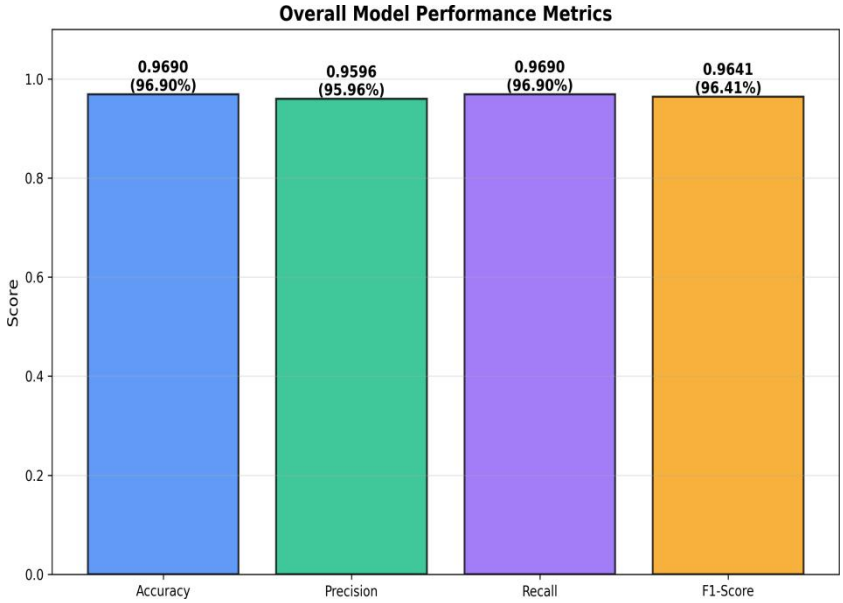

Fig. 4. Overall performance metrics of the proposed FedSSL-IDS model.

Table 3. Model Size and Performance Comparison after Quantization

Metric	Original	FP16	INT8
Size (MB)	0.09	0.04	0.02
Size Reduction	-	55.4%	75.5%
Accuracy	96.90%	96.91%	49.54%
Precision	0.9596	0.9596	0.2455
Recall	0.9690	0.9691	0.4954
F1-Score	0.9641	0.9641	0.3283

7.2. Class-Wise Performance Analysis

To examine the reliability of detection in various classes of attacks, the class-wise precision, recall, and F1-score are considered. The suggested model works very well with the leading types of attacks like the DDoS and PortScan with almost the highest recall and F1-scores. This means that the model is very efficient in terms of supporting the unique traffic patterns that are attributed to high volume attacks and scanning support attacks. High accuracy is also achieved in classifying benign traffic, which means that the model is characterized by a low detection error and that it is capable of distinguishing clearly between normal network traffic and malicious traffic.

Rather, minority attack classes (SQL Injection and XSS) are less precise: this is largely because the classes have very high situation imbalance in the dataset. In spite of this drawback, the recall of these classes still is relatively high given that this model still can detect a majority of unique and severe mauls. This action indicates that the suggested framework is sensitive towards attacks, which is a welcome statement in the intrusion detection context albeit at the expense of a relative increase in false positives. In general, the analysis by classes demonstrates the strength of FedSSL-IDS structure in its contradictory and the minority attack, as well as shows the possibility to improve it by using class-balancing or information addition methods.

Figure 6 demonstrates the distribution of classes of the test dataset, which points at the disproportion of various traffic types. Overwhelmingly the biggest part of the dataset is the benign traffic, with the next category being PortScan and DDoS attack types and the SQL Injection type having a much smaller number of samples. This distribution is realistic on conditions of a network in which normal traffic prevails and there are attack types that are not very frequent. The class imbalance is a very difficult test of intrusion detection systems since the model must also pick up uncommon attacks, and in the process, reduce on unnecessary false alarms. Through experiments based on the suggested FedSSL-IDS framework in this unbalanced environment, the experiments reveal the strength of the model and its capability of sustaining a significant detection accuracy in all dominant and minority attack classes.

Figure 7 shows that the proposed FedSSL-IDS framework has a performance in terms of precision, recall, and F1-score per classification of the traffic. The model has a high detection of Benign, PortScan and DDoS traffic with a consistent

score of high marks in all the three measures. This means that the trained representatives are effective at reflecting the distinguishing attributes of both normal traffic and high-volume attack patterns. In the case of the SQL Injection class, the performance values are very low because the extent of the class imbalance and the number of samples in the test dataset is very low. This result illustrates the problematic nature of identifying unusual types of attacks to real-life traffic to the network. Nevertheless, the general class-wise assessment course does justify the fact that the suggested FedSSL-IDS model does not suffer any severe detection issues between the most prevalent attack types, but it is aware of the high-priority and uncommon intrusion.

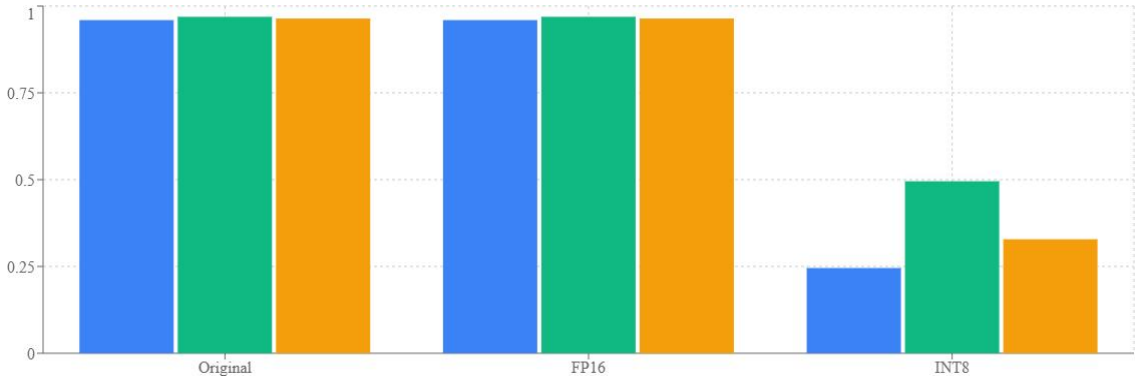

Fig. 5. Comparison of detection performance under different quantization strategies.

Table 4. Model Efficiency Observations

Model	Remarks
Full-Precision	Highest detection perfor- mance
Pruned	Reduced model size with minimal accuracy loss
FP16 Quantized	Best trade-off between effi-ciency and accuracy
INT8 Quantized	Higher compression with noticeable accuracy drop

7.3. Confusion Matrix and Error Analysis

The confusion matrix enables the comprehensive understanding of patterns of misclassification of the proposed FedSSL-IDS model as it will show the distribution of true and predicted class labels of the attack. Figure 6 displays that most benign, DDoS and PortScan samples are correctly identified demonstrating good discrimination ability both against normal traffic as well as high frequency attack behavior. The clumps of values having high concentration along the main diagonal attest to the fact that the model systematically invariably labels most traffic flows correctly, which is an indication of the stable learning and good performance of the generalization among prevalent classes.

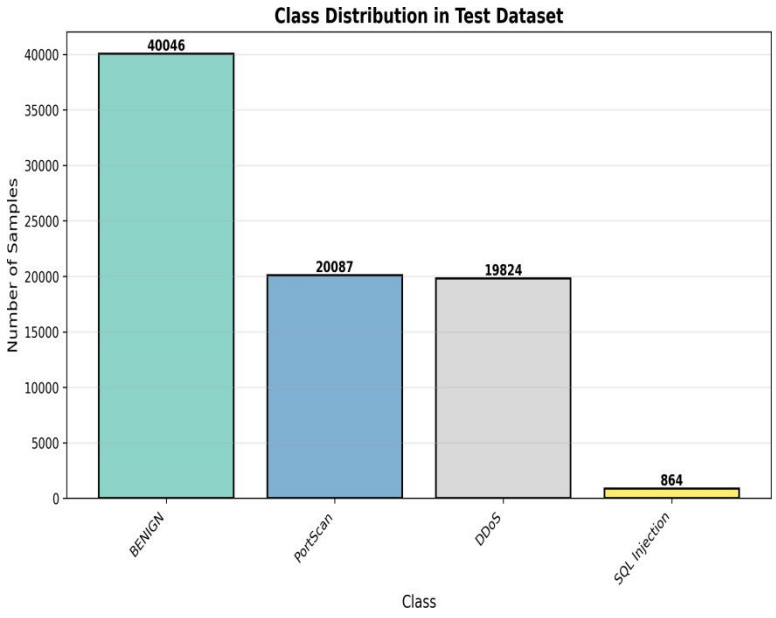

Fig. 6. Class distribution of the test dataset used for evaluation.

There is a less severe level of misclassification between benign traffic and the high DDoS flows. This is not

surprising, since some honest bursts of traffic, like flash crowds, or high-throughput data transfers, can share some volumetric attack behaviour, like high packet rate and longer flow duration. Although such overlap exists, misclassified samples are rather few, which proves that the model offered manages to identify minor features differences between malicious and non- malicious traffic in conditions of a real network. The normalized confusion matrix also helps to underscore the strength of the proposed approach by underscoring the class-wise true positive rates regardless of the frequency of the classes. Large normalized values on the diagonal of the dominant attack classes ensure that the model is still characterized by a high level of detection reliability in the case of severe imbalance between classes. It is especially notable in intrusion detection systems, where different distributions of classes can frequently cause models to be biased to dominating classes and reduce the detection capabilities. Notably, the attacks by the minority classes like SQL Injection and XSS also get significant true positive values, even though they are underrepresented in the dataset. This observation means that the self-supervised feature learning mechanism allows the model to learn informative representations to even the infrequent and rare attack patterns. Despite all the misclassification of these classes, the capability of identifying a large fraction of rare attacks is essential to the security-vulcanized applications, in which a limited number of unidentified attacks can be severe.

In general, the analysis of the confusion matrix proves that FedSSL-IDS framework proposed provides an appropriate, consistent, and explainable classification results, including various categories of traffic. The majority of misclassifications are in predictable and understandable cases of overlapping traffic behavior and not in systematic model breakdown. These results confirm the appropriateness of the proposed method to be implemented on the real-world edge, where the traffic of the network is dynamical, heterogeneous and may overlap in its behavior.

Fig 8. shows the confusion matrix of the proposed FedSSL-IDS framework in both terms that are the absolute counts of samples and normalized percentages of the percentages and this will give further information on how the model behaves in terms of classifying traffic within various traffic classes. The count-based confusion matrix is a plot that shows both the count of the correctly and incorrectly categorized samples per class but the normalized one points out the class-wise true positive rates without regard to the frequency of classes. It reveals that the benign and DDoS as well as PortScan traffic are classified with high accuracy as indicated by the high diagonal values in both matrices. There are low levels of confusion between these dominant classes and this implies that there is proper discrimination of high volume attack patterns and normal traffic. In the case of the minority SQL Injection, the SQL Injection class, the model attains significant detection rate given the small size of samples, which is sensitive to rare but significant attack kinds.

Altogether, the confusion matrix analysis proves the strength and the consistency of the suggested FedSSL-IDS frame- work under the real and unbalanced traffic load, as well as reflects the contributors to the misclassification errors that are to be expected and justified under the conditions of the close traffic behavior relationship.

7.4. ROC and Precision–Recall Curve Analysis

In determining the discriminative of the proposed FedSSL-IDS model, the trade-off between the true positive rate (TPR) and the false positive rate (FPR) at different classification thresholds are used with the help of Receiver Operating Characteristic (ROC) curves. The Area Under the Curve (AUC) values of the main traffic classes such as the Benign, DDoS, and PortScan, as shown in Figure 7 are higher than 0.99, which is an excellent class separability and a high level of confidence in the predictions of the model. High AUC values of this order indicate that the proposed framework continues to give greater levels of confidence to samples of the desired attack, as opposed to benign traffic, even under varying decision levels.

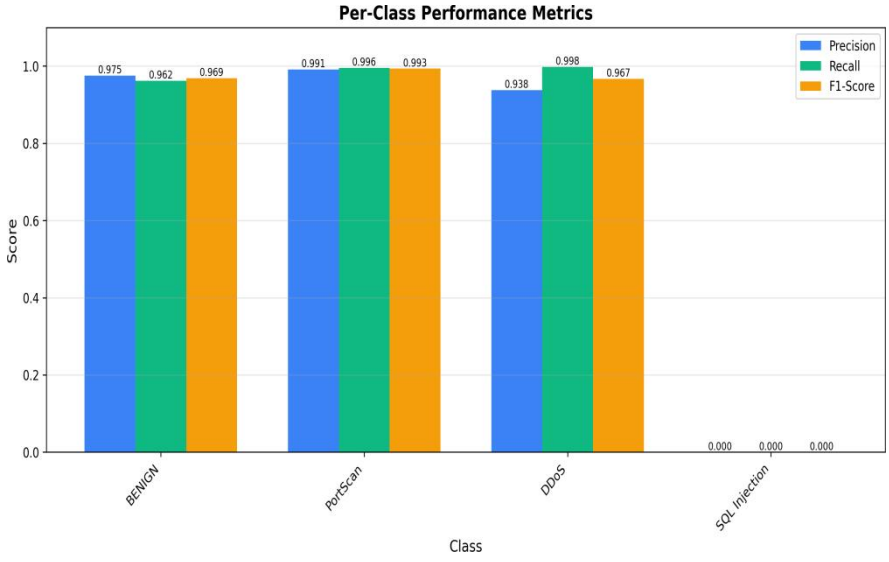

Fig. 7. Class-wise precision, recall, and F1-score of the proposed FedSSL-IDS model.

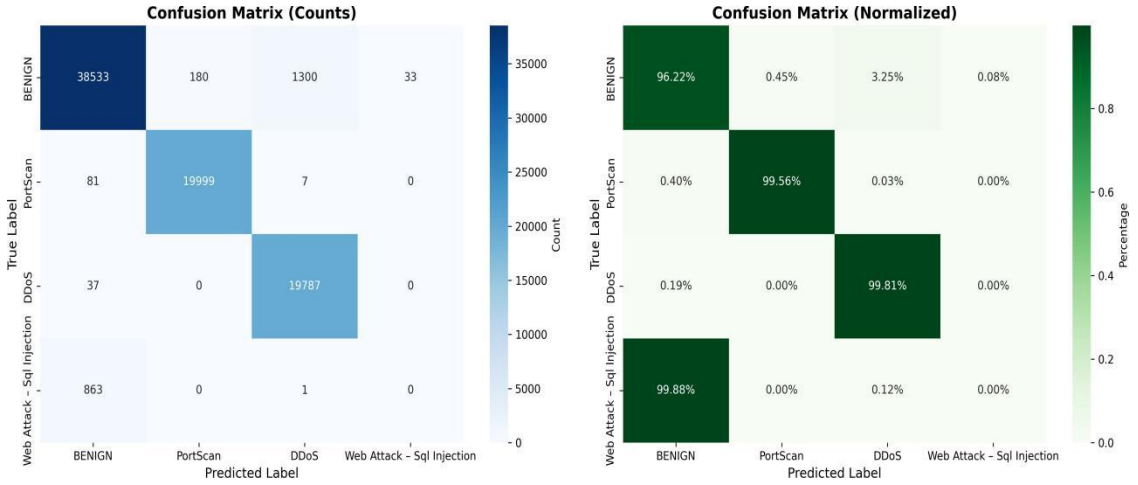

Fig. 8. Confusion matrix of the proposed FedSSL-IDS model: (a) absolute counts and (b) normalized percentages.

The ROC analysis also confirms that the self supervised feature representations learned at the edge clients are getting features that are highly discriminative traffic characteristics. Such action is especially profound within federated learning settings where heterogeneous and non-IID data distributions are very likely to negatively affect the model generalization. Even with these difficulties, the proposed FedSSL-IDS model shows high discrimination ability of all the key types of traffic, which shows the strength of self-supervised learning, combined with federated aggregation.

Besides the ROC analysis, Precisionrecall (PR) curves are also analyzed to give the analysis a more informative assessment under conditions of a class imbalance that is typical in real intrusion detection datasets. The PR curves show that the precision is constant and high over a large range of recall amounts in major types of attack like DDoS and PortScan. This behavior is especially significant in an operational network setting since it implies that the model can identify a high percentage of malicious traffic and minimize false-positive rates, which reduces the number of alerts and overhead obtained as a result of operational settings.

Nokia In case of the minority attack classes, the PR curves are more pronounced in terms of trade-off between preci-sion and recall and that is due to the small amount of samples available to be used in the training process. Nevertheless, the model also has maintainable recall levels, meaning that it can detect rare and yet a critical attack cases. Minority at- tack sensitivity is imperative to intrusion detection systems where failure to detect frequent attacks may lead to the tragic security breach.

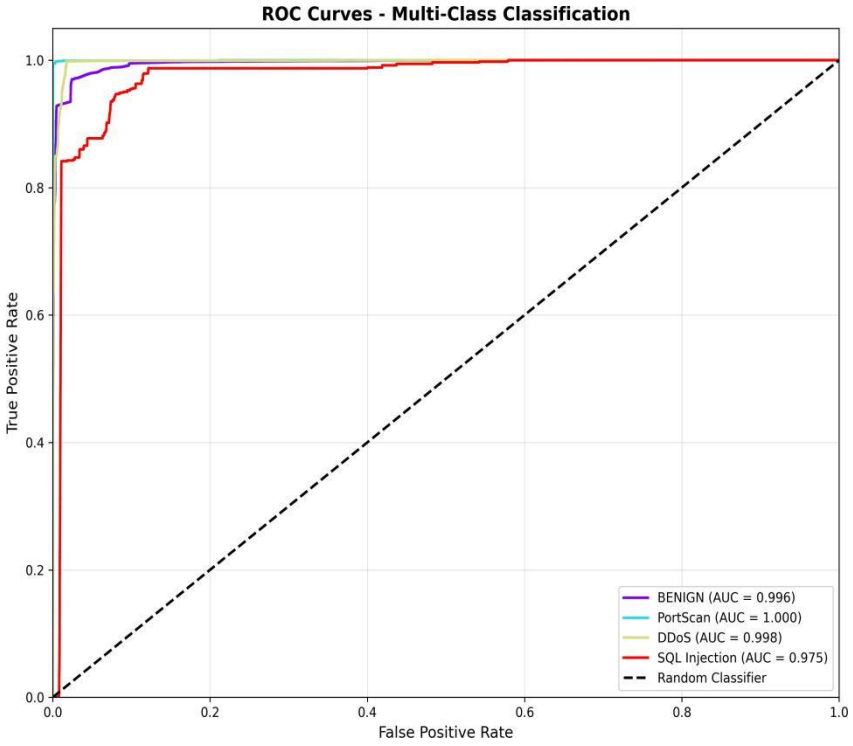

Fig. 9. Receiver Operating Characteristic (ROC) curves for multi-class intrusion detection using the proposed FedSSL-IDS model.

All in all, the joint ROC and Precision-Recall studies show that the developed FedSSL-IDS system has a high discrim- inative power, resistance to the existence of imbalance classes, as well as, consistent detection on a wide range of traffic types. These findings confirm the applicability of the suggested strategy to be implemented in real-life edge-based net- work settings that involve unstable traffic distribution and uneven data distributions. Figure 9 shows Receiver Operating Characteristic (ROC) plots of multi-class intrusion detection, which measures the discriminptions power of the proposed FedSSL-IDS framework in relation to various category of traffic. The ROC curves are given as a representation of true positive rate versus false positive rate under different classification thresholds and the slant line is that of the performance of a random classifier.

The findings indicate that the Area Under the Curve (AUC) of the key classes of the beneficial behaviour of Benign, PortScan and DDoS are near 1.0 meaning high predictability and a high confidence of prediction. The SQL Injection model is also able to identify rare patterns of attacks thereby shown by the high AUC despite serious class imbalance in the SQL Injection class. On the whole, the ROC analysis will verify the high level of discriminative performance and strength of the proposed FedSSL-IDS framework at realistic and imbalanced traffic scenarios.

7.5. Impact of Model Optimization

The effect of the model optimization methods optimizing pruning and quantization techniques are understood systematically to determine the deployability of the proposed FedSSL-IDS framework on resource-constrained edge devices. These approaches are important in minimizing the computational complexity of the computations, memory use, communication, and inference latency which are important aspects in making intrusion detection properties of edges and IoT systems using limited computational and energy resources.

Fig. 10. Accuracy comparison of the proposed FedSSL-IDS model under different precision levels.

Magnitude based pruning is used in the process of training to remove small-value weights in the neural network that leave the most significant parameters. The empirical findings prove that such high level of redundancy in the initial network structure can notably be reduced through a significant change in the model parameters at the cost of a degree of negligible deterioration in detection. This observation proves that the self-learned feature representations obtained by the model are compact and strong in nature. In addition to saving memory, pruning also reduces inference time and level of energy use making the optimized model better suited to deployment continuously on edge hardware.

After pruning, post-training quantization is used to optimize the model even more to be deployed. Of the analyzed methods, FP16 quantization allows to maintain detection accuracy similar to the full-precision model but much smaller model size and lower computational load. It is also believed due to the low interpretation and execution of the material with FP16, that half-precision arithmetic is adequate to reflect the learned weights and activations used in intrusion detection work. FP16 quantization is a useful and efficient optimization technique that can be used on edge devices capable of executing mixed-precision arithmetic.

By contrast, securing a significant loss in detection accuracy is incurred under INT8 quantization, especially with minority attack classes, which rely on fine and subtle distinctions between the feature. The violent divulsion of numerical accuracy restricts its representational ability of the model and has the effect of increasing the false classification of uncommon attack patterns. Because figure 8 shows, the difference between accuracy in FP16 and INT8 quantization points to the sensitivity of intrusion detection models to extreme compression. Moreover, Figure 9 also offers the trade-off between the model size and accuracy of the detection, with INT 8 quantization yielding the highest compression ratio, but at the expense of lower detection reliability. In general, such findings indicate that FP16 quantization offers the most desirable trade-off of efficiency and detect capabilities, thus being the optimization of choice to deploy the proposed FedSSL-IDS framework as an edge-based detector. Another insight that was made in the analysis is that the model compression strategies should be selected cautiously based on the resultant application needs such as in systems that rely on security concerns where detection reliability is of greater concern than compressing to the limit. Such results support the feasibility of the presented structure to be applied in practical settings in terms of edge devices that are energy- and resource-constrained.

involve unstable traffic distribution and uneven data distributions. Figure 10 offers the comparison of the detection accuracy of proposed FedSSL-IDS model in three numerical precision modes: full-precision (original) and FP16: quantization and INT8: quantization. The findings suggest that the accuracy of the FP16 quantization is almost the same as that of the full-precision model, which proves that the reduced-precision computation can be successfully utilized without the observable decrease in the detection performance.

Conversely, INT8 quantization is found to cause an apparent decrease in accuracy, which underscores the constraints of quantizing aggressively features in applications that use fined-grained feature representations since intrusion detection is an example. These results underscore the fact that FP16 quantization provides a more appropriate trade-off between computing efficiency and detecting reliability and is thus well suited to be used on resource-constrained edge devices. Figure 11 shows that quantifying the proposed FedSSL-IDS framework to size reduction works. This comparison is presented on the baseline full-precision model, FP16 quantized model and INT8 quantized model. The findings here show that FP16 quantization achieves a very high model size with high detect resistance and INT8 achieves maximum compression.

Even though the INT8 quantization has the least model footprint, it is concluded at the expenses of lower detection accuracy, as has been previously seen. This result supports the trade-off between model size and performance at detection, as FP16 quantization is a more balanced and realistic optimization approach to IM, when it comes to deploying models on edge devices with limited resources.

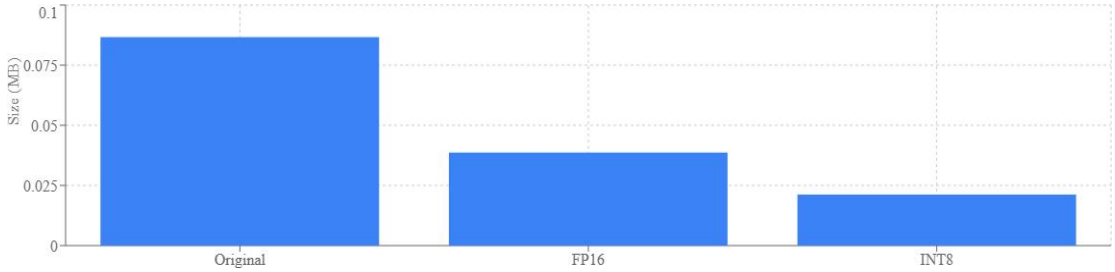

Fig. 11. Comparison of model size under different quantization schemes.

Table 5. Concise Summary of FedSSL-IDS Experimental Results

Category	Outcome
Detection Accuracy Class-wise Performance	High accuracy with strong precision, recall, and F1-score under non-IID data Excellent detection of DDoS and PortScan; reliable recall for minority attacks
Robustness	Low misclassification and strong class separability in confusion matrix and ROC analysis
Model Optimization Deployment Readiness	Pruning reduces complexity with minimal loss; FP16 provides best trade-off for edge deployment Lightweight, low-latency, and energy-efficient for edge environments

7.6. Summary of Results

The controlled testing proves that the suggested FedSSL-IDS model has a strong and balanced rate of detection accuracy, preservation of privacy, and efficiency in deployment in topology mode network settings. The federated model with full precision achieves high total accuracy, as well as strong values in precision, recall, and F1-score, which proves that it is a strong model capable of discriminating benign traffic fairly well and various forms of attacks in non-IID and highly unbalanced data sources. These findings underscore the potential of the suggested framework in ensuring stability and consistency of performance despite the nature of the problem of decentralized and heterogeneous edge data.

The class of the performance analysis also shows good detection of most powerful types of attacks including DDoS and PortScan that are one of the most common and harmful risks in current networks. Simultaneously, the framework allows significant recall of minor attack type classes proving sensitive to rare yet critical intrusions generally missed by traditional detection frameworks. All three findings are confirmed by the confusion matrix and ROC-based analyses which indicate that the class separabilities are strong, the true positive are high and the misclassification levels are low in the varied traffic classes and these results demonstrate the strength of the learned representations in the real network environment.

Besides the effectiveness of detection, the optimisation experiments of the model prove the feasibility of the suggested method. Magnitude-based pruning greatly simplifies model complexity and parameter quantities with no or only minor effects on detection performance, allowing efficient and efficient implementation in resource-constrained edge devices. FP16 quantization has the best balance between accuracy and computation efficiency of the available quantization strategies, with significant model size and inference latency reduction achieved without compromising detectability.

On the whole, these findings support a solid empirical position that the suggested FedSSL-IDS architecture is not just correct and powerful, but it is also scalable and can be implemented in practice, so it works well in edge-based and IoT application intrusion detection, where resource consumption, access to energy, and information privacy are vital factors.

The table provides a concise summary of the key experimental outcomes of the proposed FedSSL-IDS framework. It highlights the model's strong detection performance under non-IID data distributions, robustness across diverse attack categories, and the effectiveness of model optimization techniques in enabling efficient and scalable deployment on resource-constrained edge devices.

8. Discussion

The experimental outcomes prove that FedSSL-IDS framework is a useful tool that helps to overcome some critical issues related to intrusion detection in edge-based and IoT architecture, such as data privacy, limited labeled data, non-IID, and resource-constrained data distribution. The performance in terms of self-supervised learning and federated learning demonstrates that the combination of specifically the former and the latter can be considered an efficient and effective approach to creating scalable and privacy-aware security systems.

Among the most significant findings of the results is that the model can be used to sustain high performance even in the federated training circumstances. The distribution of traffic patterns is generally very different between points in practice edged environments, and it is commonly not possible to train centrally because of privacy. The suggested framework proves that even without the raw data exchange, collaborative learning can result in a strong global model, which implies that federated learning is quite suitable in intrusion detection settings where the sensitivity and decentralization of the data are the main priorities.

The analysis of the performance of the classes gives additional information on the strengths and weaknesses of the system. The high level of detection of the DDoS and PortScan attacks shows that the model works well in capturing high level traffic behavior related to scanning and volumetric types of attacks. These forms of attacks are some of the most common in actual networks and it is important that reliable detection is done so that the network is not lost. Simultaneously, this structure preserves a sensible percent of recall of minority attack classes, indicating that it is sensitive to infrequent yet high-impact intrusions. Precision to these classes is compromised by class imbalance though, such behaviour is a trade-off that is usually favoured between attack spread and false miss of intrusion, and is very common in security-critical applications.

The interpretation and the reliability of the approach is supported by the confusion matrix and ROC-based analysis. The majority of classification mistakes are made in cases of traffic classes having overlapping features, e.g. benign traffic and high-volume DDoS traffic. This finding is in agreement with actual network behavior since legitimate network traffic bursts can be similar to attack patterns. The high value of AUC and stable Precision- Recall curves imply that the model can be discriminative at different levels with respect to the threshold and can be adjusted into its operation needs such as low false-positive or more attack-sensitive.

A deployment approach, in particular, the discussion of model optimization is especially important. Lightweight models are always needed in edge devices since they usually have strict energy and memory constraints. The findings indicate that magnitude-based pruning is a good method to reduce the complexity of models without affecting the detection performance, which proves that the learned representations are succinct and robust. Moreover, FP16 quantization comes as the most feasible optimization choice, which has a desirable efficiency to accuracy ratio. The fact that the performance decays with the INT8 quantization is an important lesson that in security-sensitive applications aggressive compressions can undermine reliability, and should be implemented with caution.

All in all, it can be affirmed that the proposed FedSSL-IDS framework is not just factual, but also feasible to be deployed in real-world edge settings to produce a practical effect. Privacy-preserving learning with its powerful detection and efficient optimization of models makes the framework a viable solution to the next-generation intrusion detection systems. The findings also indicate opportunities of improvement in the future, including better countermeasures to extreme imbalance of classes, adaptive thresholding depending on network conditions, and combination with online learning measures to counterattack the changing patterns of attacks.

9. Conclusion & Future work

In this paper, the author introduced FedSSL-IDS, a fresh and holistic intrusion detection framework that aims to deal with increasing security issues of edge-based networks and Internet of Things (IoT) networks. The suggested solution is based on combining self-supervision learning with federated learning to make joint training of the model possible without violating processes of data privacy and decreasing the usage of labeled data. Representation learning on the edge directly addresses privacy concerns related to centralized intrusion detection systems by transferring only model parameters throughout federated training so that the data is stored on the decentralized edge devices that are characteristic of contemporary edge systems. Part of the analysis done on CICIDS2017 dataset shows that the proposed FedSSL- IDS framework can deliver high detection rates with realistic conditions that have non-IID data distributions and severe class imbalance. The model achieves sufficient overall accuracy and balanced F1-score values as well as recall and precision, which implies it is effective in discriminating between benign traffic and various types of network attacks. Specifically, the framework has superb detection of the most common and disruptive attacks in real network environment, which include DDoS and PortScan. Simultaneously, the system has meaningful recall on minority attack classes, which points to its sensitivity to rare and critical intrusions which can be hard to identify. The strong performance analysis in terms of confusion matrices, ROC curves, Precision Recall curves convince further the strength and soundness of the suggested approach. These studies indicate that there is high separability of classes as well as high true positive and low misclassification across large traffic classes. Notably, the majority of classification errors are committed in beneficial situations where similar traffic patterns e.g. benign traffic bursts that look like volumetric

attacks, not because the model has failed in a systematic manner. This fact makes the proposed intrusion detection system more interpretable and more credible. Besides detection accuracy, this work also puts a high emphasis on deployability and efficiency, which is also an important edge-based security solution requirement. Magnitude-based pruning application shows that it is possible to make enormous cuts in model complexity and the number of parameters with little or no effect on detection performance. In addition, comparison of post-training quantization methods indicates that FP16 quantization offers the best tradeoff between computational complexity and detection accuracy. The optimizations achieve lower memory footprint, lower inference latency and energy-efficiency, which allows real-time intrusion detection on resource-constrained edge devices over the internet of things (IoT) gateways and embedded systems. Nevertheless, even though the results are promising, there are different directions to be pursued in the future research. To begin with, future research can look into extrapolating the evaluation to other and more recent intrusion detection datasets, and real-time traffic of working networks to further test the external validity of the proposed framework. Second, more sophisticated methods like data augmentation, class rebalance, and cost sensitive learning can be deployed to enhance detection performance of absolutely rare attack classes still further. There was no deliberate use of any balancing technique like SMOTE, focal loss, cross-entropy, and data augmentation. The impact of this weakness was felt especially in cases of SQL Injections and XSS, since the minority samples of these attacks made the entire process less robust. Third, to increase the resilience of federated systems to poisoning and inference attacks in adversarial settings, it is possible to integrate strong federated learning features, such as secure aggregation, adversarial client detection and differential privacy.

Finally, it is possible to further augment the framework by incorporating adaptive and continual learning to better address dynamic network threats. While the proposed framework enhances privacy by preventing any direct exchange of data in plaintext form, more sophisticated techniques like differential privacy, secure aggregation, and protection against poisoning attacks are not incorporated into the present framework and are left for future work. In conclusion, the FedSSL-IDS framework is a promising intrusion detection scheme that addresses scalability, privacy, and energy consumption.

All the Declarations and Statements

Author Contributions Statement

Paras Kacha was involved in the conceptualization of the study, design of the methodology, development of software, data curation, formal analysis, investigation, and visualization of the results. Dr. Swati V. Shinde was involved in the overall supervision and guidance of the study, data curation, formal analysis, validation of the results of the research, project administration, and acquisition of resources and support. The authors also drafted the original manuscript.
The other co-authors were involved in the conceptualization and design of the methodology of the study, provision of resources and technical support for the experimental setup, investigation and validation of the results, and review and editing of the manuscript.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Paras Kacha	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Swati Shinde	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Bal Virdee	✓	✓			✓	✓	✓			✓	✓	✓	✓	
Ashish Khanna	✓	✓			✓	✓	✓			✓	✓	✓	✓	

C	: Conceptualization	I	: Investigation	Vi	: Visualization
M	: Methodology	R	: Resources	Su	: Supervision
So	: Software	D	: Data Curation	P	: Project Administration
Va	: Validation	O	: Writing – Original Draft	Fu	: Funding Acquisition
Fo	: Formal Analysis	E	: Writing – Review & Editing		

Conflict of Interest Statement

The authors declare that there are no conflicts of interest regarding the publication of this research work.

Funding Declaration

This research received no external funding.

Data Availability Statement

This study utilized the publicly available CICIDS2017 dataset provided by the Canadian Institute for Cybersecurity (CIC). The dataset can be accessed from the official repository at:
https://www.unb.ca/cic/datasets/ids-2017.html Accessed on: April 2026.

Ethical Declarations

This study does not involve human participants, human data, animal experiments, or any ethical-sensitive procedures. Therefore, ethical approval and informed consent were not required for this research.

Acknowledgments

The authors would like to express their sincere gratitude to the Canadian Institute for Cybersecurity (CIC) for providing the CICIDS2017 dataset, which served as a valuable benchmark for evaluating the proposed intrusion detection framework. The availability of this dataset greatly supported the experimental analysis and validation of the research outcomes.

The authors also acknowledge the support and guidance received from faculty members and mentors during the course of this research. Their constructive feedback and technical insights contributed significantly to the improvement of the proposed methodology. Additionally, appreciation is extended to the open-source research community for providing tools and libraries that facilitated the implementation and experimentation of this work.

Finally, the authors would like to thank all individuals and institutions who directly or indirectly supported this research and contributed to its successful completion.

Declaration of Generative AI in Scholarly Writing

During the preparation of this manuscript, the authors used Generative AI tools for language refinement, grammar correction, formatting assistance, and improving the clarity of technical writing. The authors carefully reviewed, validated, and edited all generated content and take full responsibility for the final content of the publication. No AI tool was used for generating experimental results, scientific interpretations, or research conclusions.

Abbreviations

The following abbreviations are used in this manuscript:

AI – Artificial Intelligence
NLP – Natural Language Processing
DL – Deep Learning
ML – Machine Learning
IDS – Intrusion Detection System
FL – Federated Learning
SSL – Self-Supervised Learning
IoT – Internet of Things
DDoS – Distributed Denial-of-Service
CNN – Convolutional Neural Network
RNN – Recurrent Neural Network
FP16 – 16-bit Floating Point Precision
INT8 – 8-bit Integer Precision
AUC – Area Under Curve
ROC – Receiver Operating Characteristic
PR – Precision-Recall
KNN – K-Nearest Neighbors
SVM – Support Vector Machine
XSS – Cross-Site Scripting
SQL – Structured Query Language
CIC – Canadian Institute for Cybersecurity
FedAvg – Federated Averaging

Appendix A\B\C…, with appendix tile

APPENDIX A

Additional Experimental Results

Additional performance evaluations, confusion matrices, and class-wise analysis results of the proposed FedSSL-IDS framework are presented in this appendix.

APPENDIX B

Hyperparameter Configuration
This appendix contains the detailed hyperparameter settings used during federated learning, self-supervised training, pruning, and quantization experiments.

APPENDIX C
Implementation Environment
This appendix provides information regarding the software libraries, hardware configuration, TensorFlow/Keras setup, and edge deployment specifications used in the experiments.

References

[1] Khalid Alsubhi. A secured intrusion detection system for mobile edge computing. Applied Sciences, 14(4), 2024.
[2] Aitor Belenguer, Jose A Pascual, and Javier Navaridas. A review of federated learning applications in intrusion detection systems. Computer Networks, page 111023, 2025.
[3] Safa Ben Atitallah, Maha Driss, Wadii Boulila, and Anis Koubaa. Strengthening network intrusion detection in iot environments with self-supervised learning and few shot learning. In International Conference on Computational Collective Intelligence, pages 83–96. Springer, 2024.
[4] Razvan Beuran et al. Fedmse: Semi-supervised federated learning approach for iot network intrusion detection. Computers & Security, page 104337, 2025.
[5] Niccolo` Borgioli, Linh Thi Xuan Phan, Federico Aromolo, Alessandro Biondi, and Giorgio Buttazzo. Real-time packet-based intrusion detection on edge devices. In Proceedings of Cyber-Physical Systems and Internet of Things Week 2023, pages 234–240. 2023.
[6] Ping Deng and Yong Huang. Edge-featured multi-hop attention graph neural network for intrusion detection system. Computers & Security, 148:104132, 2025.
[7] Trong-Minh Hoang, Tuan-Anh Pham, Van-Viet Do, Van-Nhan Nguyen, and Manh-Hung Nguyen. A lightweight dnn-based ids for detecting iot cyberattacks in edge computing. In 2022 International Conference on Advanced Technologies for Communications (ATC), pages 136–140. IEEE, 2022.
[8] Shajjad Hossain, Sidi-Mohammed Senouci, Bouziane Brik, and Abdelwahab Boualouache. A privacy-preserving self-supervised learning-based intrusion detection system for 5g-v2x networks. Ad Hoc Networks, 166:103674, 2025.
[9] Danish Javeed, Muhammad Shahid Saeed, Muhammad Adil, Prabhat Kumar, and Alireza Jolfaei. A federated learning-based zero trust intrusion detection system for internet of things. Ad Hoc Networks, 162:103540, 2024.
[10] Zhigang Jin, Junyi Zhou, Bing Li, Xiaodong Wu, and Chenxu Duan. Fl-iids: A novel federated learning-based incremental intrusion detection system. Future Generation Computer Systems, 151:57–70, 2024.
[11] Amit Kumar, Vivek Kumar, and Abhay Pratap Singh Bhadauria. Optimizing intrusion detection in edge computing network: A hybrid ml approach with recursive feature elimination. International Journal of Intelligent Engineering & Systems, 18(1), 2025.
[12] Naila Latif, Wenping Ma, and Hafiz Bilal Ahmad. Advancements in securing federated learning with ids: a com- prehensive review of neural networks and feature engineering techniques for malicious client detection. Artificial Intelligence Review, 58(3):91, 2025.
[13] Pontus Lidholm, Tijana Markovic, Miguel Leon, and Per Erik Strandberg. Network intrusion detection using ma- chine learning on resource-constrained edge devices. In 2024 International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2024.
[14] Nguyen Ngoc Minh, Pham Trung Hieu, Vu Hai, and Nguyen Huu Thanh. Dga-based intrusion detection system using federated learning method on edge devices. In 2024 International Conference on Information Networking (ICOIN), pages 509–514, 2024.
[15] Mohamad Moawad, Vong Wan Tze, Patrick Then Hang Hui, and Jason Yong Hsien Ming. Deep learning based intrusion detection for internet of things and edge devices. In 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), pages 01–06, 2023.
[16] Babatunde Olanrewaju-George and Bernardi Pranggono. Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models. Cyber Security and Applications, 3:100068, 2025.
[17] Pooja Potnurwar, Ayush Ainchwar, Rahul Neware, and Vrushali Bongirwar. Intrusion detection system for big data environment using deep learning. MDPI AG., 2024.
[18] Jahangir Shaikh, Yasir Awais Butt, and Hira Fatima Naqvi. Effective intrusion detection system using deep learning for ddos attacks. The Asian Bulletin of Big Data Management, 4(1):Science–4, 2024.
[19] Asis Kumar Tripathy, Alekha Kumar Mishra, and Rashmi Panda. Performance comparison and analysis of machine learning and deep learning models for network intrusion detection in iot-edge frameworks. In 2024 Second Inter- national Conference on Emerging Trends in Information Technology and Engineering (ICETITE), pages 1–6. IEEE, 2024.
[20] Renjie Xu, Guangwei Wu, Weiping Wang, Xing Gao, An He, and Zhengpeng Zhang. Applying self-supervised learn- ing to network intrusion detection for network flows with graph neural network. Computer Networks, 248:110495, 2024.

Authors' Profiles

Paras Kacha received his Diploma in Information Technology and his Bachelor of Engineering (B.E.) in Information Technology, and is currently pursuing an M.Tech in Computer Science. He has worked on several projects including a Facial Recog- nition Based Attendance System, an NLP-based Medical Health Checker chatbot, a Data Mining project on the Impact of Extracurricular Activities on Students' Per- formance, and an Edge-based Intrusion Detection System using Deep Learning and the NSL-KDD dataset. His research interests include Artificial Intelligence, Machine Learning, Edge Computing, Intrusion Detection Systems, and Data Analytics. He can be contacted at paras20pk@gmail.com.

Swati Shinde received her Ph.D. in Computer Science and Engineering from Swami Ramanand Teerth Marathwada University, Nanded. She completed her M.E. from Bharati Vidyapeeth, Pune, and B.E. in Computer Science and Engineering from SRTMU, Nanded. She is currently a Professor at Pimpri Chinchwad College of En- gineering, Pune, with over 20 years of teaching experience and previously served as Head of the IT Department. She has published more than 70 research papers, filed research patents, and is a recognized Ph.D. guide. Her research interests include Data Mining, Machine Learning, Soft Computing, Artificial Neural Networks, and Fuzzy Logic.

Bal Virdee is Professor and Director of the Centre for Communications Technol- ogy and Senior Professor at London Metropolitan University, UK. He completed his Ph.D. in Communications Engineering from the University of London and previously worked in R&D roles at Philips and Teledyne Defence & Space. He has published extensively in reputed international journals and conferences and has served on edi- torial boards of several journals. His research interests include wireless and satellite communications, microwave and millimeter-wave systems, IoT, biomedical sensors, robotics, assistive technologies, and wireless network security.

Ashish Khanna received his Ph.D. in Computer Science and Engineering from NIT Kurukshetra and his M.Tech from GGSIPU, Delhi. He completed postdoctoral re- search at the IoT Lab, INATEL, Brazil, and at the University of Valladolid, Spain. He has published over 250 research papers and authored multiple books and book chap- ters. His research interests include Distributed Systems, MANET, FANET, VANET, IoT, Machine Learning, and Evolutionary Computing. He actively contributes to the academic community as a keynote speaker, reviewer, guest editor, and conference or- ganizer.

How to cite this paper

Paras Kacha, Swati Shinde, Bal Virdee, Ashish Khanna, "Energy-Efficient and Privacy-Preserving Intrusion Detection in Edge-Based Networks Using Federated Self-Supervised Learning", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.3, pp. 254-276, 2026. DOI:10.5815/ijwmt.2026.03.17