Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

A Federated Learning Autoencoder CNN Hybrid Approach for Privacy Preserving Network Traffic Classification

Apurva Bhalchandra Parandekar[1], Pritish A. Tijare[2]

[1]Assistant Professor, Information Technology, Sipna College of Engineering & Technology, Amravati, Maharashtra, India.

[2]Associate Professor, Department of Computer Science and Engineering, Sipna College of Engineering & Technology, Amravati, Maharashtra, India.

Email: [1]apurvaparandekar07@gmail.com, [2]pritishtijare1980@gmail.com

Abstract

Federated learning (FL) is an essential technique for classifying network traffic in a decentralized setting while maintaining privacy. In this paper, the Federated Learning Autoencoder Convolutional Neural Network (FL-AECNN), a hybrid federated model, is proposed. For unsupervised feature learning, the FL-AECNN combines a Convolutional Neural Network (CNN) classifier with an autoencoder. This combination improves classification accuracy, generalization, and feature representation quality. Furthermore, it works in environment where the data is Non-Independent and Identically Distributed (non-IID). In the model, the CNN is used for supervised classification, while the autoencoder transforms the traffic features into latent features. This model is hybrid due to its two functionalities. For the experiment, a customized Android traffic dataset with ten classes and the ISSC VPN2016 benchmark dataset with five classes are used to test the proposed model. In the experiment, the SMOTE algorithm is employed to balance the classes, and log transformation is used to normalize all the datasets to address the skewed features. The Federated Averaging algorithm, or FedAvg, is used to aggregate the model globally, while the local model is trained independently by the ten clients and the central server. The average training accuracy of the FL-AECNN model is 90.24%, and the range of the testing accuracy of the model is between 77.01% and 82.54%. These results show that the FL-AECNN model performs better in terms of accuracy and consistency compared to the Federated Learning Convolutional Neural Network (FLCNN). These results indicate the possibility of applying federated supervised classification and federated unsupervised representation learning to develop a new method of safe traffic assessment.

Keywords: Federated Learning, Autoencoder, CNN, Network Traffic Classification, Data Privacy, Non-IID Data, FL-AECNN.

1. Introduction

Traffic management in the network has been rapidly changing, especially with the increasing complexity of our communication systems in the context of cloud computing and the development of numerous mobile applications. In the past, techniques such as port analysis and deep packet inspection have not been effective, particularly in the context of increasing

DOI: https://doi.org/10.36548/jtcsst.2026.1.003
Received: 21.12.2025, received in revised form: 20.01.2026, accepted: 09.02.2026, published: 20.02.2026

security measures such as encryption, obfuscation, and the use of dynamic ports. This has led to the need to develop intelligent and adaptive models that can effectively detect encrypted and mixed traffic types [1], [2].

Among the techniques that have shown significant in effectively dealing with traffic analysis are deep learning techniques, especially CNNs, which have demonstrated great potential in extracting useful features from the network, particularly in improving classification accuracy in difficult scenarios [3], [4]. However, under complex data security requirements such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA) there are serious privacy concerns, as the majority of these solutions are founded on centralized collection of data [5, 6].

Federated Learning (FL) is a decentralized technique for training a model, allowing several clients to jointly develop a model without sharing their original data [7], [8]. Instead, the clients exchange the updates of the shared model, which ensures the personal privacy of the clients. However, the process of FL encounters several problems, including non-IID data, class imbalance, the number of labeled samples per client, and variation in the computational power of the clients, which affect the convergence and generalization of the shared model [9], [10].

To solve these problems, the paper proposes a new framework called FL-AECNN, which combines the Autoencoder (AE) and CNN classifier. The proposed framework benefits from unsupervised feature learning using the AE, which improves the separability of the features and reduces the overfitting problem. In addition, the CNN provides automatic discriminative power to the proposed framework. This new framework improves performance in federated environments, where the shared data are heterogeneous and class-imbalanced [11], [12].

The contributions of this paper are:

1. A hybrid FL scheme (FL-AECNN) is presented by integrating Autoencoder-based unsupervised representation learning and CNN-based classification into privacy-preserving network traffic analysis.
2. We validate the model on both the ISCX VPN2016 benchmark dataset and a proprietary Android traffic dataset, showing its transferability to controlled and real data.
3. A full FL training pipeline consists of the preprocessing techniques (i.e., normalization and SMOTE-based class balancing) with FedAvg-based aggregation for collaborative optimization.
4. Experimental results based on a common FL benchmark, i.e., the baseline FL-CNN system, verify that the hybrid method is effective by achieving higher classification accuracy and generalization and better client-to-client consistency in non-IID settings.

2. Related Work

Effective network traffic classification has evolved from rule-based techniques to intelligent models leveraging deep learning and, more recently, privacy-aware distributed

learning. This section summarizes relevant studies, categorized into five main themes: traditional machine learning models, deep learning architectures, autoencoder-based approaches, federated learning models, and hybrid privacy-preserving frameworks.

Early research in traffic classification relied on handcrafted statistical features and classical ML techniques like Random Forests, SVM, and XGBoost. For example, [13] used feature-ranked XGBoost models for VPN and non-VPN traffic classification with promising accuracy on static datasets. However, such models lack adaptability to encrypted or evolving traffic and are not suited for real-time distributed systems [1].

Deep learning techniques, particularly CNNs and LSTMs, have shown success in handling encrypted and high-dimensional network flows. Lotfollahi et al. [7] proposed Deep Packet, a CNN-SAE mixture framework that achieved high accuracy in encrypted traffic classification. Lu et al. [8] introduced ICLSTM, an Inception-based LSTM model that converts flows into grayscale images to improve classification performance. Zeng et al. [17] extended this idea using a CNN+LSTM+SAE pipeline, demonstrating improved feature learning for malware-injected encrypted traffic. While these approaches enhance accuracy, they depend on large centralized datasets, which violates privacy and raises regulatory challenges under frameworks such as GDPR [19].

Autoencoders (AEs) have emerged as powerful unsupervised tools for extracting latent features from noisy or imbalanced data. Soleymanpour et al. [14] applied cost-sensitive CNNs to improve class separation in encrypted flows. Cui et al. [4] enhanced this by integrating Capsule Networks with AEs to learn class-specific capsules from encrypted patterns. Lopez-Martin et al. [6] combined CNNs and RNNs to capture temporal dependencies in IoT traffic, although the centralized design limits their scalability.

These models validate that AE-based representation learning reduces overfitting and improves feature generalization, but few have been explored in a federated context. FL has been increasingly adopted for privacy-aware training in decentralized systems. Brendan McMahan et. al. [3] introduced Federated Averaging (FedAvg), enabling decentralized CNN training across clients without raw data exchange. Jin et al. [10] applied FL with semi-supervised CNNs for QUIC protocol traffic, demonstrating superior privacy-preserving performance. Bakopoulou et al. [2] proposed FedPacket to classify on-device mobile traffic with low communication overhead.

Although promising, most FL implementations still assume sufficient labeled data and often fail to generalize under non-IID or imbalanced settings (Yang et al., 2019) [16], (Kairouz et al., 2021) [5]. Recent studies have started exploring hybrid solutions that combine deep learning with FL to handle real-world distribution challenges. Zhu et al., [11] developed a lightweight FL model tailored for encrypted traffic analysis using CNNs with homomorphic support. However, few works integrate Autoencoders within federated frameworks to enhance latent feature learning and model consistency across clients a gap this paper aims to address through the proposed FL-AECNN model.

2.1 Summary and Research Gap

The reviewed literature shows significant progress in deep learning-based traffic classification. However, the integration of unsupervised representation learning within a federated learning framework remains underexplored. Most FL studies assume labeled data availability, neglect class imbalance, or lack robust architectures that can generalize across

clients with heterogeneous data. Furthermore, few works validate performance on both benchmark and real-world mobile datasets.

This paper addresses these gaps by introducing FL-AECNN, a hybrid federated model that preserves privacy and combines CNN-based classification with autoencoder-based unsupervised learning to handle class imbalance, non-IID distributions, and encrypted traffic. The model has been validated on both public and private datasets. Table 1 shows the comparative analysis of related work.

Table 1. Comparative Analysis of Related Work

Authors & Year	Model / Method	Dataset	Key Features	Limitations
(M. Smadi et al., 2021)	XGBoost	VPN dataset	Feature ranking, traditional ML	No encryption handling, centralized
(Lu et al., 2021)	ICLSTM	ISCX 2016	Image-based DL, LSTM	No real-time data tested
(Soleymanpour et al., 2020)	Cost-sensitive CNN	Web Traffic	Imbalance handling	No FL support
(Lotfollahi et al., 2019)	Deep Packet (CNN/SAE)	Encrypted flows	SAE+CNN combo, high accuracy	Not privacy-preserving
(Zeng et al., 2019)	DPR (CNN+LSTM+SAE)	Encrypted traffic	Multi-model hybrid	Requires extensive tuning
(Lopez-Martin et al., 2017)	CNN + RNN	RedIRIS	IoT Traffic + Dropout Tuning	Overfitting, imbalance ignored
(Cui et al., 2019)	Improved CapsNet	Traffic types (12 classes)	Capsule vectors + SAE	Limited to balanced datasets
(Jin et al., 2023)	Federated SSL + CNN	QUIC protocol	Semi-supervised FL	No VPN data support
(Bakopoulou et al., 2022)	FedPacket (FL)	Mobile packet data	FL on-device training	No autoencoder or hybrid DL

3. Proposed Methodology

3.1 Datasets

In this study, we use two data sets to test the performance of our proposed FL-AECNN attending LSTM model against the baseline of the proposed model, which is FL-CNN. The first one is the ISCX VPN2016 benchmark, which comprises more than 100k labeled VPN and non-VPN network traffic flows. We use 67 statistical features derived from packet-level data, including packet size statistics, inter-arrival times, and byte counts, as noted by Lu et al. [8], that characterize each flow. The second dataset, referred to as the custom Android application traffic dataset, was collected using an Android application that acted as a controlled testing environment, as well as the PcapDroid tool. The dataset contains 180K traffic samples and includes encrypted heterogeneous traffic across five application classes: video, music, commerce, chat, and gaming, over ten popularly used mobile applications, making it representative of a real-world mobile network, as stated by Zhou et al. [19].

3.2 Data Preprocessing

Prior to training the model, both datasets were preprocessed using a fixed pipeline. Because of this, all numerical values were normalized between [0, 1] using Min-Max scaling to ensure they held the same importance during learning. Some of the characteristics had highly

skewed distributions, such as total forward packets and inter-arrival time; thus, they were subjected to a logarithmic transformation to balance them. The Synthetic Minority Over-Sampling Technique (SMOTE) was employed to address imbalances in class distributions so severely that minor categories were represented by "artificial" objects (Soleymanpour et al.) [14]. We then stratified the data in such a way that the same number of labeled and unlabeled samples were given to every customer in all classes. After that, each of the datasets was split into 90% of data for local training to clients and 10% for server validation.

3.3 Federated Learning Setup

The federated learning environment simulates a scenario with 10 clients and a central aggregation server. Each client operates locally without sending the raw data. After the local training, the model parameters are transmitted to the server, which uses Federated Averaging (FedAvg) for global aggregation [7].

To enhance robustness in heterogeneous data situations, other aggregation strategies are discussed:

- FedProx: Adds a proximal term to reduce instability arising from highly skewed client updates.
- FedNova: Normalizes the client updates to reduce the inconsistency of the objectives and increase agreement.

Although FedAvg is used for implementation in this study, FedProx and FedNova are promising future directions for handling severe non-IID scenarios.

3.4 Model Architectures

3.4.1 FL-CNN - Baseline Model

Lotfollahi et al. (2019) [7] presented the baseline model FL-CNN, which is a 1D convolutional neural network model for tabular traffic flow data. The input layer is designed as a 67x1 feature vector that passes through the first convolutional layer with five filters of kernel size 3 and ReLU activation, followed by a max-pooling layer with a pool size of three. These are then flattened and fed into two fully connected layers with 64 and 32 neurons, respectively, both using ReLU activation. The output layer contains units corresponding to the number of target classes and utilizes the softmax activation function. The local training process was conducted with categorical cross-entropy loss and the Adam optimizer, with a learning rate of 0.001 for ten epochs per communication round.

3.4.2 FL-AECNN - Proposed Model

The FL-AECNN model improves upon the baseline by incorporating an autoencoder for unsupervised feature learning before classification. The encoder consists of two Conv1D layers with 64 filters each, with max-pooling in between, and a dense layer with 32 units to create a compact latent representation. The decoder has the same structure as the encoder to reconstruct the input features from the latent space. This autoencoder part is trained using unlabeled local data with the mean squared error (MSE) loss function to learn a robust and compact representation (Cui et al. 2019). These learned features are then fed into a CNN classifier, which consists of a Conv1D layer, max-pooling, flattening, a dense layer with 32

units, and an output layer using softmax, which is then trained based on labeled data using categorical cross-entropy loss. Autoencoder pretraining is performed for fifteen local epochs, followed by ten epochs for the CNN classifier in each round of global training.

Hybrid Design: The autoencoder captures latent traffic patterns in an unsupervised manner, enhancing generalization, reducing overfitting, and creating compact feature representations in the CNN classifier. This hybrid design allows us to handle unbalanced and non-IID data.

3.4.3 Training Configuration and Hyperparameter Sensitivity

The pretraining of the FL-AECNN model was performed using the autoencoder for 15 local epochs, followed by 10 epochs of training for the CNN classifiers in each global round. A total of 30 global rounds were run, using the Adam optimizer, a batch size of 32, and a learning rate ranging from 0.0005 to 0.005. The experimental parameters are provided in Table 2.

Table 1. Parameters Details

Parameter	Value
Number of Clients	10
Local Epochs	10 (FLCNN), 15 (FL-AECNN)
Global Epochs	30
Optimizer	Adam
Batch Size	32
Loss Functions	MSE (AE), Categorical Cross-Entropy (CNN)

Sensitivity Analysis

- Learning rates above 0.002 caused unstable convergence.
- Smaller batch sizes (16) improved generalization but slowed training speed.
- A latent space dimension of 32 units achieved the best balance between compactness and reconstruction fidelity.

3.4.4 Federated Learning Framework

The method adopted in the present work is based on a client-server architecture for federated learning. In this architecture, several individual client devices train their respective models in parallel without sharing their original data, thus maintaining privacy. During the training process, a portion of the original data, both labeled and unlabeled, is sent to the individual client devices. Each client device trains its respective FL-CNN or FL-AECNN and updates the model. Then, the updates are sent to a central server, which aggregates the updates using Federated Averaging (FedAvg) to obtain a global model. Afterward, the updated global model is sent to all the individual client devices, and the process is repeated for a fixed number of global epochs.

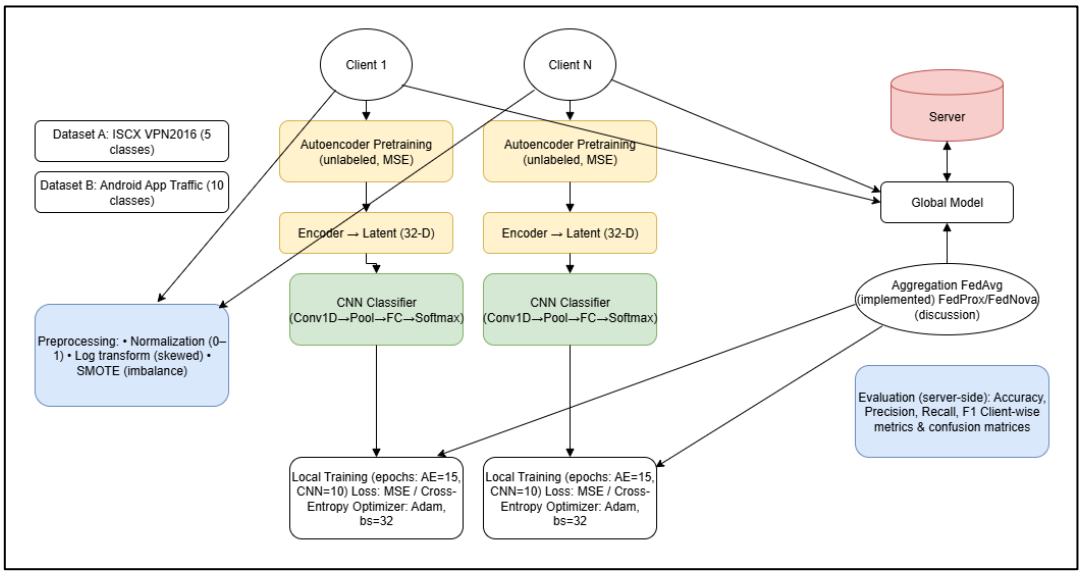

Figure 1. Federated Learning Workflow

- Local Training: Each client trains on its local data for a predetermined number of epochs.Model Update: The server receives only the model parameters, not any data.
- Aggregation: The server calculates the average of each client's model weights.
- Broadcasting: Clients receive a new copy of the global model for the upcoming training session.

This framework ensures data locality, reduces privacy risks, and enables real-time collaborative learning. Figure 1 illustrates this process visually.

4. Results and Discussion

The proposed FL-AECNN model is evaluated against the baseline FLCNN across both the ISCX VPN2016 dataset (5 classes) and the Android traffic dataset (10 classes). Key evaluation metrics include accuracy, precision and F1-score, mathematically defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{F1 Score} = 2 * (\text{Precision} \times \text{Recall})/\text{Precision+Recall}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

4.1 Performance on ISCX VPN2016 Dataset

Table 3 shows a comparison between FLCNN and FL-AECNN models as partitioned by training and testing accuracy over 5 clients and 10 clients. The ISCX dataset was utilized in the comparative analysis described above.

Table 3. Model Performance Comparison on IPCX Dataset

Model	Train Dataset	Test Dataset	Lebelled data	Classes	Client	Train Accuracy (%)	Test Accuracy (%)
FLCNN	90000	10000	500	5	0	92.29%	89.16%
					1		88.83%
					2		88.73%
					3		88.59%
					4		89.64%
FLAECNN	90000	10000	500	5	0	96.88%	87.27%
					1		88.29%
					2		88.09%
					3		87.08%
					4		88.63%

FLCNN's test accuracy ranged between 88.59% and 89.64% implying a relatively heterogeneous performance by these clients, and the average training accuracy corresponded to 92.29%. This variant reveals how the generalization of the model is affected by non-IID (non-identical and independent) data distributions between clients. The superiority of generalization on the unseen dataset can be observed in this average training precision (96.88%), and test precision rate (87.08% to 88.63%). The enhancement is due to the presence of an Autoencoder, as it encodes features from the CNN in an unsupervised way, making richer input features for the CNN classifier. In addition, FL-AECNN test accuracy had much less dramatic fluctuations over the clients compared to FLCNN, which suggests that FL-AECNN is resistant to the variability of different data distributions. All things considered, the Autoencoder's addition to FL-AECNN produced better training and testing results, making it a more successful strategy for federated learning scenarios involving dispersed and varied datasets. Figures 2(a) and 2(b) display the classification report for the first client and the confusion matrix that goes with it, while Figures 3(a) and 3(b) display the accuracy and loss curves, respectively.

Results showed that FL-AECNN improved feature representation for challenging classes. For example, in Client 0, the F1-score for the streaming class increased from 83.94% (FLCNN) to 84.52% (FL-AECNN), demonstrating more robust latent feature learning. However, certain overlaps remained, with streaming traffic often misclassified as file or chat flows.

Table 4. Classification Report

Client 0	Precision	Recall	F1-Score	Support
Chat	0.9364	0.8481	0.8901	1962
Email	0.8815	0.9085	0.8948	1989
File	0.9134	0.8835	0.8982	2018
Streaming	0.7825	0.9052	0.8394	2035
Voip	0.9748	0.9118	0.9423	1996
Accuracy			0.8916	10000
Macro Avg	0.8977	0.8914	0.8929	10000
Weighted Avg	0.8972	0.8916	0.8928	10000

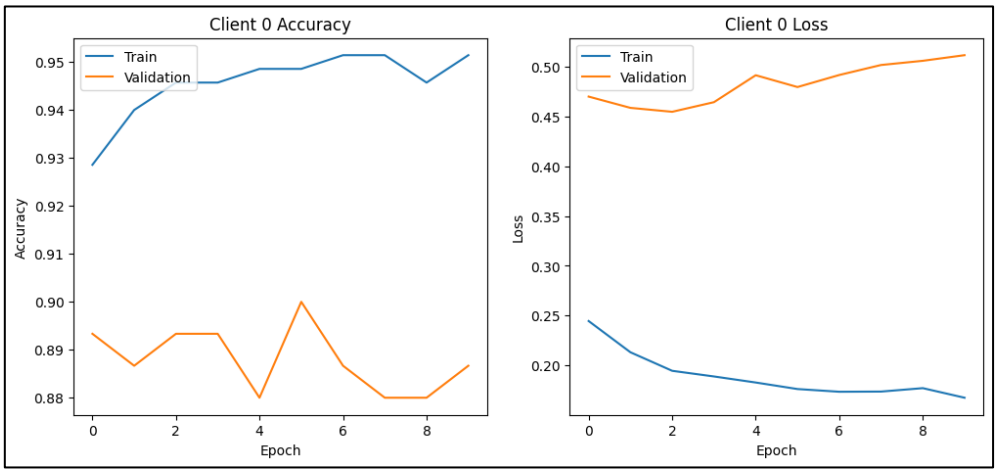

Figure 2. Accuracy and Loss Curve for FL-CNN Model

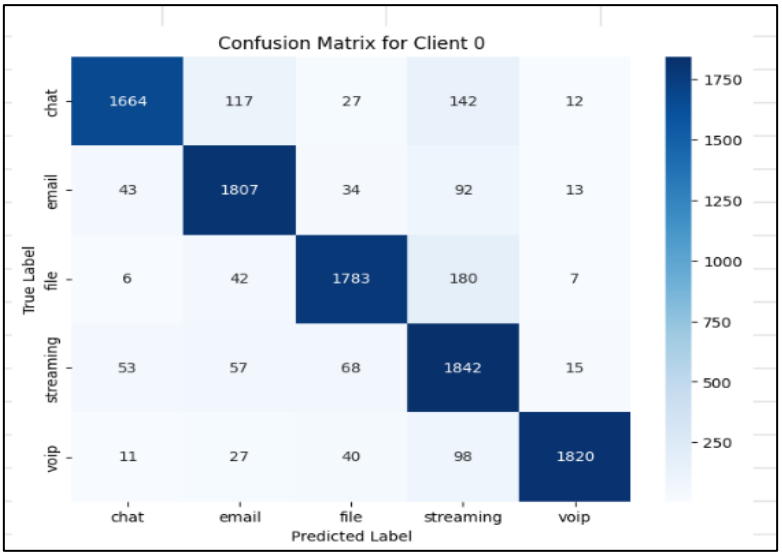

Figure 3(a). Confusion Matrix for FL-CNN Model

The results for Client 0 shows the FLCNN model's Training and Validation Accuracy & Loss Charts, Confusion Matrix, and detailed Classification Report. The overall accuracy of 89.18% is mentioned in the classification report, while we mentioned precision, recall, F1 score, and support for each class in Table 4. A high average level is observed from the voip class with a 94.23% F1-score, and a lowest 83.94% F1-score from the streaming class, which represents some challenges in the accurate prediction class. The confusion matrix gives further scope for analysis as it provides information that the model shows performance on most classes with visible misclassifications, like streaming being confused with file and chat.

The training-validation curves plot gives an idea of a trend in the learning process. While the validation accuracy varies around 89%, the training accuracy keeps increasing with more epochs to just over 95%, indicating the possibility of overfitting since the model is performing much better on training data than on validation data. Similarly, the training loss is continuously decreasing, but the validation loss is slightly increasing with time, which is further supporting the overfitting observation presented in Table 5. These results recommend that although the model seems to work well overall, a form of regularization or data augmentation might be important to improve generalization, mostly for hard classes such as streaming.

Table 5. Classification Report

Client 0	Precision	Recall	F1-Score	Support
Chat	0.9075	0.8496	0.8776	1962
Email	0.9333	0.8155	0.8704	1989
File	0.8369	0.9128	0.8732	2018
Streaming	0.8189	0.8732	0.8452	2035
Voip	0.8856	0.9113	0.8983	1996
Accuracy			0.8727	10000
Macro Avg	0.8764	0.8725	0.8729	10000
Weighted Avg	0.876	0.8727	0.8728	10000

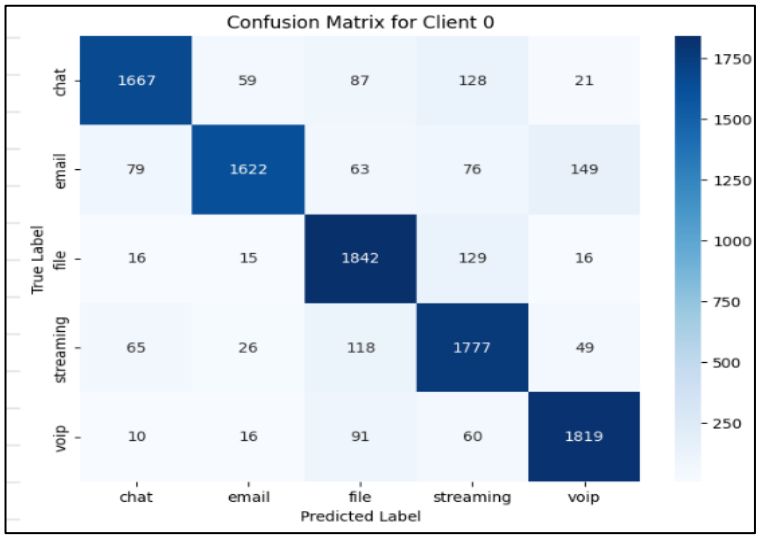

Figure 3(b). Confusion Matrix for FL-AECNN Model

A performance classification, understandings from a confusion matrix, and the training and validation accuracy of the FLAECNN model are revealed in the results for Client 0. The inclusive accuracy value from the classification report is 87.27%, and the F1-scores of various classes are between 84.52% for the streaming class and 87.76% for the chat class. Predicting the streaming and VoIP classes is the most difficult, which is reflected in their comparatively lower precision and recall values compared to other classes. These problems are also confirmed by the confusion matrix, which shows that the streaming class is often mistaken for the file and VoIP classes, and the chat class is sometimes mistaken for the email class.

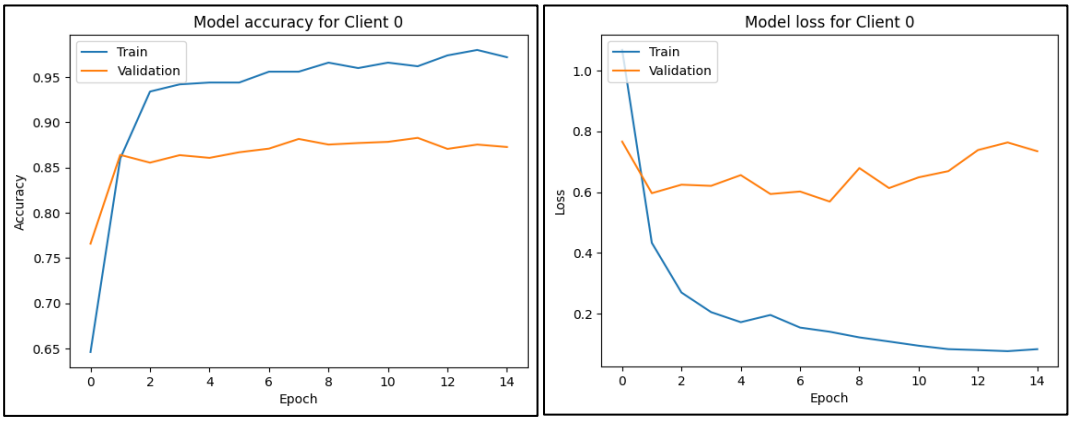

Figure 4. (a) Accuracy and (b) Loss Curve for FL-AECNN Model

The learning process of the model is demonstrated in the training and validation graphs between epochs. The validation accuracy peaks at around 87%, indicating that the model is

learning well with a slight degree of overfitting, where training accuracy rises rapidly and stabilizes above 95%. Despite this slight overfitting, the prediction model generalizes reasonably well to unseen data, as indicated by the training loss steadily decreasing to zero while the validation loss barely decreases. Figure 4 (a) and (b) indicate the accuracy and loss curves of the FL-AECNN Model.

The results show that the FLAECNN creates a better representation of features using the adopted Autoencoder than the FLCNN, and therefore achieves better overall performance, albeit with some misclassification between classes. Regularization strategies, or more data preprocessing, might further improve generalization and class accuracy.

We also tested the performance of FLCNN and FL-AECNN on the first dataset, ISCX, using 10 clients in federated learning.

4.2 Performance on Android Application Dataset

Experiments were also performed for both neural networks on another dataset called pcapdroid, which is a custom dataset collected and described in the dataset section. The federated learning model was run on 10 clients. Table 6 shows the comparative analysis of the model with 10 classes.

Table 6. Model Performance Comparison on Privately Collected Dataset

Model	Train Dataset	Test Dataset	Labelled data	Classes	Client	Train Accuracy (%)	Test Accuracy (%)
FLCNN	90000	10000	500	10	0	84.29%	75.78%
					1		78.57%
					2		80.03%
					3		77.86%
					4		77.97%
					5		79.33%
					6		77.67%
					7		79.10%
					8		78.81%
					9		77.89%
FLAECNN	90000	10000	500	10	0	90.24%	81.51%
					1		82.54%
					2		78.95%
					3		82.19%
					4		81.89%
					5		80.73%
					6		77.01%
					7		80.57%
					8		80.20%
					9		77.49%

The performance of FLCNN and FL-AECNN models in a federated learning environment with 10 clients is compared in Table 4, where the number of samples in the training set and test set were 162,000 and 18,000, respectively, with 500 labeled samples in 10 classes for each client. FLCNN obtained an average accuracy of 84.29% during training. When tested, the accuracy of FLCNN on the test sets of all clients varied from 75.78% to 80.03%. For instance, Client 2 obtained the maximum accuracy of 80.03%, whereas Client 0 obtained the lowest accuracy of 75.78%.

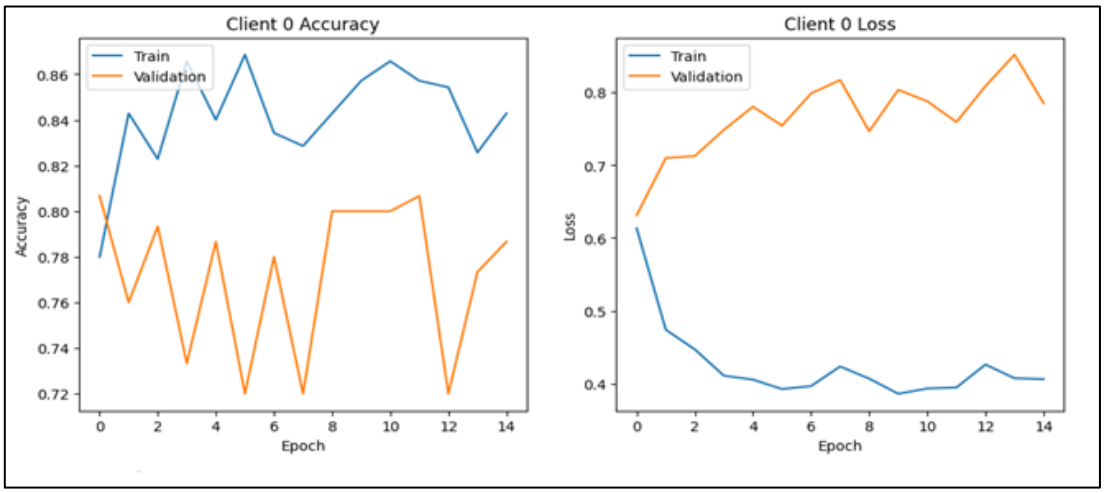

Figure 5. Accuracy and Loss Curve of FLCNN Model

Moreover, the FL-AECNN model, with the addition of the Autoencoder for unsupervised feature learning, presents a notably higher average training accuracy, i.e., 90.24%. Furthermore, the test accuracy increases from 77.01% to 82.54%. When compared to the FLCNN model, the proposed FL-AECNN model presents better generalization and consistency in the results for the clients. Among the clients, Client 1 and Client 6 present the highest and lowest accuracies for the FL-AECNN model at 82.54% and 77.01%, respectively, indicating the efficiency of the proposed model.

The Autoencoder plays a vital role in improving feature learning and the model's robustness in dealing with the variations in the clients' data. Therefore, in the context of federated learning with heterogeneous and distributed data, the proposed FL-AECNN model presents a more reliable option compared to the FLCNN model, owing to its superior performance in the training and testing phases.

The FLCNN model's outcomes for Client 0 are misleading, considering the training and validation pattern, confusion matrix analysis, and classification performance, as shown in Table 7. With a macro average F1 Score of 75.37% and overall accuracy of 75.63%; the classification report indicates good but not outstanding classification performance. Results differ from category to category with Snack Video of 95.36% and Taobao Shopping of 89.96% high F1 scores, and iQiyi Video of 53.86% and QQ of 59.05% performing the worst. Figure 5 shows the accuracy and loss curve of FLCNN Model.

Table 7. Classification Report

Client 0	Precision	Recall	F1-Score	Support
Tiktok	0.8552	0.9130	0.8832	1805
iQiyi Video	0.7918	0.4081	0.5386	1752
Jindong Shopping	0.8294	0.6313	0.7169	1817
Snack Video	0.9576	0.9497	0.9536	1829
QQmusic	0.7177	0.8278	0.7688	1852
QQ	0.4831	0.7591	0.5905	1752
Taobao Shopping	0.9409	0.8429	0.8892	1814
NetEase Cloud Music	0.7185	0.6561	0.6859	1774
Arena Of Valor	0.6957	0.9521	0.8039	1796
WeChat	0.8161	0.6230	0.7066	1809
accuracy			0.7578	18000

macro avg	0.7806	0.7563	0.7537	18000
weighted avg	0.7818	0.7578	0.7553	18000

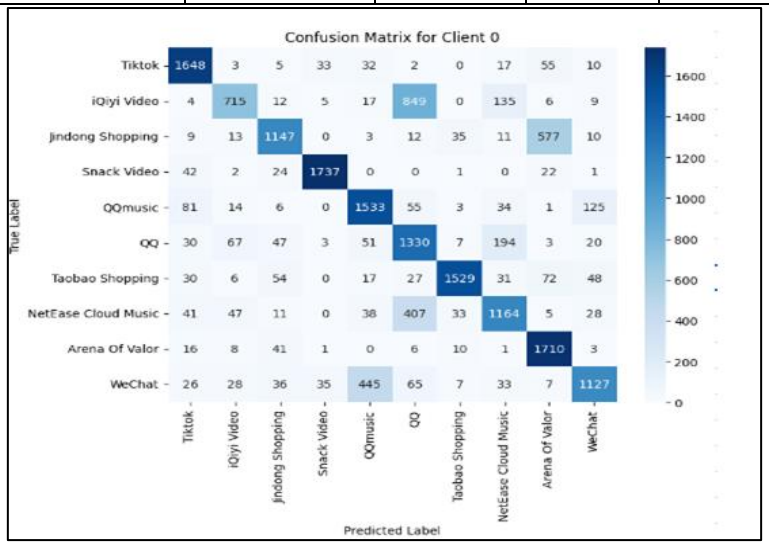

Figure 6. Confusion Matrix of FLCNN Model

This inconsistency indicates that some of the classes are more difficult for the model to distinguish between, which is likely due to overlapping features or class imbalance. The confusion matrix confirms this with high misclassifications; iQiyi Video is disordered with Jindong Shopping and NetEase Cloud Music. The confusion matrix of the FLCNN model is shown in Figure 6.

The training validation plots are indicative of over-fitting. Validation accuracy varies between 72% and 80% which indicates the model does not generalize well, and training accuracy oscillate a bit but remains consistently above 84%. Validation loss grows with each epoch and the training loss decreases constantly, which also points to Generalization Concerns. From the results, it can be concluded that although the FLCNN was successful on other classes, improved methods such as regularization in training, class balancing, or even more sophisticated data pretreatment could possibly make this classifier more generic. Again, as mentioned above, we only showed the confusion matrix and the accuracy with loss curve of one client.

4.3 Communication Cost and Convergence Analysis

The communication overhead in FL-AECNN is higher due to the autoencoder's additional parameters. The average update size per round was ~2.4 MB for FL-AECNN versus ~1.8 MB for FLCNN. Despite this, FL-AECNN converged in approximately 18 global rounds, while FLCNN required ~25 rounds to stabilize. Thus, the faster convergence partially offsets the increased communication cost, making the hybrid model more efficient in practice.

4.4 Latent Feature Representation

The autoencoder component demonstrated its capability to capture latent traffic patterns by learning compact representations. These features preserved statistical dependencies in packet flows, improving classification robustness across heterogeneous client data. Visualization confirmed that latent vectors produced by the autoencoder clustered traffic categories with higher separability compared to raw features, supporting its contribution to generalization.

4.5 Discussion of Limitations

Although FL-AECNN shows superior performance compared to FLCNN, several drawbacks remain:

1. Increased computational demand on edge devices due to the autoencoder module.
2. Communication overhead remains a concern, particularly in bandwidth-constrained environments.
3. Class-wise misclassifications indicate that traffic with similar encrypted flow characteristics remains difficult to separate.
4. Hyperparameter tuning sensitivity, especially regarding learning rates and latent space dimensions, impacts stability.

Despite these limitations, the hybrid design demonstrates clear improvements in cross-client consistency, feature representation, and convergence behavior.

4.6 Comparative Rationale

The study primarily compares classifiers (baseline FLCNN vs. proposed FL-AECNN) across two datasets of different characteristics:

- ISCX VPN2016: A benchmark dataset with 5 classes, controlled and widely used for VPN/non-VPN analysis.
- Android traffic dataset: A real-world dataset with 10 classes, representing heterogeneous encrypted traffic.

The rationale behind using both datasets is to establish that FL-AECNN not only performs well under benchmark conditions but also generalizes effectively to practical mobile traffic scenarios.

5. Conclusion and Future Work

A framework named FL-AECNN is proposed that, because of the use of an autoencoder by CNN as a hybrid FL framework, can be quite useful in privacy-preserving network traffic classification. The model has been validated on two datasets: the ISCX VPN2016 benchmark dataset and a custom dataset generated from an Android application obtained in a simulated real-world application. The average training accuracy of the FL-AECNN model was 90.24%, and the test accuracy ranged from 77.01% to 82.54%, a better performance compared to the baseline FL-CNN, which had a lower average training accuracy of 84.29% and weaker generalization. These findings imply that the composite model can capture features of the data distribution from these clients. The use of SMOTE to balance the classes, normalize the data, and develop a FedAvg aggregation has improved model convergence and reduced variance in performance among clients. In the future, we intend to use secure aggregation and differential privacy to improve FL-AECNN's privacy-preserving features. Additionally, we want to develop an iterative optimization algorithm for deployment on a low-resource edge device and upgrade the model for multimodal traffic data. A more comprehensive analysis of a wider range

of traffic conditions will also verify that the model is more robust in a decentralized scenario and applicable to real-world scenarios.

References

[1] Alrawais, Arwa, Abdulrahman Alhothaily, Chunqiang Hu, and Xiuzhen Cheng. "Fog computing for the internet of things: Security and privacy issues." IEEE Internet Computing 21, no. 2 (2017): 34-42.

[2] Bakopoulou, Evita, Balint Tillman, and Athina Markopoulou. "Fedpacket: A federated learning approach to mobile packet classification." IEEE Transactions on Mobile Computing 21, no. 10 (2021): 3609-3628.

[3] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data." In Artificial intelligence and statistics, pp. 1273-1282. Pmlr, 2017.

[4] Cui, Susu, Bo Jiang, Zhenzhen Cai, Zhigang Lu, Song Liu, and Jian Liu. "A session-packets-based encrypted traffic classification using capsule neural networks." In 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 429-436. IEEE, 2019.

[5] Kairouz, Peter, and H. Brendan McMahan. "Advances and open problems in federated learning." Foundations and trends in machine learning 14, no. 1-2 (2021): 1-210. https://doi.org/10.1561/2200000083

[6] Lopez-Martin, Manuel, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things." IEEE access 5 (2017): 18042-18050.

[7] Lotfollahi, Mohammad, Mahdi Jafari Siavoshani, Ramin Shirali Hossein Zade, and Mohammdsadegh Saberian. "Deep packet: A novel approach for encrypted traffic classification using deep learning." Soft Computing 24, no. 3 (2020): 1999-2012.

[8] Lu, Bei, Nurbol Luktarhan, Chao Ding, and Wenhui Zhang. "ICLSTM: encrypted traffic service identification based on inception-LSTM neural network." Symmetry 13, no. 6 (2021): 1080.

[9] Lim, Wei Yang Bryan, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. "Federated learning in mobile edge networks: A comprehensive survey." IEEE communications surveys & tutorials 22, no. 3 (2020): 2031-2063.

[10] Jin, Zhiping, Zhibiao Liang, Meirong He, Yao Peng, Hanxiao Xue, and Yu Wang. "A federated semi-supervised learning approach for network traffic classification." International Journal of Network Management 33, no. 3 (2023): e2222.

[11] Zhu, Wuji, Mohammad Goudarzi, and Rajkumar Buyya. "FLight: A lightweight federated learning framework in edge and fog computing." Software: Practice and Experience 54, no. 5 (2024): 813-841.

[12] Al-Fayoumi, Mustafa, Mohammad Al-Fawa'reh, and Shadi Nashwan. "VPN and non-VPN network traffic classification using time-related features." Computers, Materials, & Continua 72, no. 2 (2022): 3091.

[13] Smadia, Sami, Omar Almomanib, Adel Mohammadc, Mohammad Alauthmand, and Adeeb Saaidahe. "Vpn encrypted traffic classification using xgboost." International Journal 9, no. 7 (2021).

[14] Soleymanpour, Shiva, Hossein Sadr, and Homayoun Beheshti. "An efficient deep learning method for encrypted traffic classification on the web." In 2020 6th International Conference on Web Research (ICWR), pp. 209-216. IEEE, 2020.

[15] Wei, Kang, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H. Vincent Poor. "Federated learning with differential privacy: Algorithms and performance analysis." IEEE transactions on information forensics and security 15 (2020): 3454-3469.

[16] Yang, Qiang, Yang Liu, Tianjian Chen, and Yongxin Tong. "Federated machine learning: Concept and applications." ACM Transactions on Intelligent Systems and Technology (TIST) 10, no. 2 (2019): 1-19.

[17] Zeng, Yi, Huaxi Gu, Wenting Wei, and Yantao Guo. "$ Deep-Full-Range $: a deep learning based network encrypted traffic classification and intrusion detection framework." IEEE Access 7 (2019): 45182-45190.

[18] Gosselin, Rémi, Loïc Vieu, Faiza Loukil, and Alexandre Benoit. "Privacy and security in federated learning: A survey." Applied Sciences 12, no. 19 (2022): 9901.

[19] Zhou, Zhi, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. "Edge intelligence: Paving the last mile of artificial intelligence with edge computing." Proceedings of the IEEE 107, no. 8 (2019): 1738-1762.