UAV-Assisted Federated Learning with Autoencoders for IoT Image Classification

André Ribeiro Gonçalves[1]
Departamento de Informática
PUC-Rio
Rio de Janeiro, Brazil
agoncalves@inf.puc-rio.br

Bruno José Olivieri de Souza[1]
Departamento de Informática
PUC-Rio
Rio de Janeiro, Brazil
bolivieri@inf.puc-rio.br

Markus Endler[1]
Departamento de Informática
PUC-Rio
Rio de Janeiro, Brazil
endler@inf.puc-rio.br

***Abstract*—The exponential growth of the Internet of Things (IoT) has introduced unprecedented challenges in data processing, privacy preservation, and energy efficiency. Traditional centralized approaches are often unsuitable for IoT environments due to bandwidth limitations, data heterogeneity, and privacy concerns. This study proposes a novel framework combining federated learning (FL) and autoencoders to address these issues in IoT-based image classification tasks. By leveraging Unmanned Aerial Vehicles (UAVs) as intermediaries for model aggregation and distribution, the framework minimizes communication overhead while maintaining data privacy. Autoencoders are employed for unsupervised feature extraction, enabling effective data representation even in the absence of labeled data. Results demonstrate that, while autoencoders achieve lower classification accuracy compared to supervised approaches, they provide significant advantages in bandwidth efficiency, scalability, and privacy preservation. The integration of UAVs further enhances the system by optimizing communication and enabling model improvement in real-time. This framework offers a flexible and resource-efficient solution for IoT applications, particularly in scenarios where data labeling is impractical or privacy is paramount.**

***Index Terms*—Federated Learning, Autoencoders, IoT, UAVs, Image Classification, Data Privacy.**

I. Introduction

The proliferation of the Internet of Things (IoT) has generated unprecedented volumes of data from diverse connected devices [1]. Traditional centralized processing approaches face significant challenges in IoT environments, particularly regarding privacy, communication overhead, and latency, especially when data labeling is expensive or unavailable [2].

Privacy concerns arise when transmitting sensitive data from distributed devices to central servers, increasing risks of breaches and unauthorized access [3]. Additionally, IoT systems in resource-constrained environments have limited bandwidth and energy [4]. Continuous data transmission can overwhelm network resources and rapidly drain device batteries. IoT devices often use low-power, long-range radio technologies (NB-IoT, LoRa, Sigfox), which introduce challenges in transmission rates and latency. Furthermore, centralized processing introduces significant delays in time-sensitive applications like autonomous vehicles or emergency response systems [5].

Federated learning addresses these challenges by processing data locally and transmitting only model updates [6]. However, IoT deployment faces obstacles including device heterogeneity with varying computational capacities [7] and non-IID data distributions across devices [8], which can compromise global model performance, especially with sparse labeled data.

Autoencoders [9] offer a promising solution through their ability to extract compact, meaningful data representations. Using an encoder-decoder architecture, they capture critical features while filtering noise, making them valuable for heterogeneous, unstructured IoT data [10]. These compact representations preserve data structure while reducing dimensionality, enabling feature extraction without extensive labeling and providing denoising capabilities for noisy sensor data.

This paper investigates integrating autoencoders into federated learning as an alternative to supervised models. We propose a framework where IoT devices use autoencoders for image classification, with Unmanned Aerial Vehicles (UAVs) facilitating communication and model aggregation between distributed sensors [11]. Each sensor trains an autoencoder locally, and UAVs aggregate these into a global model redistributed to ensure consistent improvement.

We evaluate this approach against a supervised baseline model. While supervised learning achieves higher accuracy with labeled data, our autoencoder-based approach offers flexibility when labeled data is scarce. This paper addresses a key question: Can autoencoder-based federated learning, despite lower classification accuracy, provide meaningful results in IoT environments with limited labeled data?

II. Related Work

A. Federated Learning in IoT

Federated learning addresses privacy and efficiency challenges in IoT networks. Beitollahi and Liberti et al. [12] [13] explored preserving data privacy through localized processing with centralized aggregation, while Hsu [14] demonstrated its effectiveness with non-IID data distributions. Our work integrates autoencoders to enhance feature extraction for limited-label scenarios. Unlike prior studies, we address IoT-specific challenges through UAV-mediated communication and model aggregation. We employ quantization and compression to reduce overhead, making our approach more suitable for resource-constrained real-world IoT deployments.

B. Autoencoders for Image Classification

Autoencoders have proven to be an effective tool for feature extraction and dimensionality reduction in the context of image classification. Fagbohungbe et al. [15] proposed advancements in autoencoder architectures that improve both robustness and efficiency, making them suitable for deployment in resource-constrained environments such as IoT networks. Tschannen et al. [16] surveyed these models' ability to learn compact and meaningful representations of complex data, which are essential for unsupervised learning tasks. In this research, autoencoders are leveraged to optimize federated learning by reducing the communication overhead and enhancing the quality of the shared models.

C. Hybrid Models for Federated Learning

Recent research has explored hybrid models that combine the strengths of autoencoders and supervised learning. Zheng et al. [17] introduced a semi-federated learning framework that integrates unsupervised pre-training with supervised fine-tuning, demonstrating improved performance in scenarios with limited labeled data. Such hybrid approaches aim to leverage the feature extraction capabilities of autoencoders while benefiting from the discriminative power of supervised learning, making them potentially well-suited for IoT environments where labeled data is scarce.

D. UAVs in Data Collection

Unmanned Aerial Vehicles (UAVs) have become a critical component in IoT networks due to their ability to efficiently collect and transmit data over large areas. Recent studies [18] [19] have demonstrated how UAVs serve as vital intermediaries between distributed sensors and central servers, significantly enhancing the efficiency of data aggregation. In federated learning contexts, UAVs not only collect data but also facilitate the distribution of a global model back to the sensors, ensuring the models are continuously improved [20].

E. Path Optimization with Energy Constraints

Path optimization for UAVs, especially under energy constraints, is vital for the sustainability of IoT operations. Jim et al [21] discusses strategies to effective UAV path planning, accounting for both energy consumption and the efficiency of data collection. Their research highlights the importance of adaptive algorithms that adjust UAV flight paths based on real-time conditions, such as remaining battery life and the priority of data collection tasks. This approach not only extends the operational life of UAVs but also ensures that critical data is collected and transmitted in a timely manner, which is crucial for the reliability of UAV-assisted federated learning systems.

F. Challenges in IoT Hardware for Model Training

IoT devices typically face limitations in computational power and energy resources, which pose significant challenges for on-device model training. Zawish et al. [22] explored the use of network pruning and advanced model compression techniques to mitigate these limitations. Lan et al. [23] investigated the effectiveness of model quantization in reducing the computational load without significantly compromising accuracy. In this study, similar techniques are employed to ensure that federated learning models can be trained and deployed effectively within the constraints of IoT devices.

G. Security and Privacy in Federated Learning

Federated learning and IoT introduces unique security and privacy challenges, particularly in decentralized IoT environments where local models could be forged/intercepted and modified to disrupt the formation of the global model. Tong et al. [24] proposed integrating blockchain technology into federated learning systems to enhance the security and trustworthiness of model aggregation. Mosaiyebzadeh et al. [25] discussed the implementation of Privacy-enhancing technologies (PETs) that protect data integrity and prevent unauthorized access during the federated learning process.

III. Methodology

This section outlines the methodology applied in this research, detailing the system design, simulation setup, federated learning workflow, optimization techniques, and evaluation metrics.

A. System Design

The proposed system integrates federated learning within a UAV-assisted IoT environment. IoT sensors train local models on their data subsets, while UAVs aggregate these into a global model and redistribute it back to sensors. This design suits scenarios with scarce or unevenly distributed labeled data.

Key challenges include device heterogeneity, where sensors have varying computational capabilities and power constraints, complicating uniform performance across the network. Non-IID data distribution poses difficulties for global model generalization, as sensors collect data differing by environmental factors and locations. Communication reliability concerns arise from UAVs needing proximity to sensors, where unstable connections can cause transmission failures. The iterative aggregation process introduces latency, especially with limited bandwidth, requiring optimization between update frequency and real-time performance.

B. CIFAR-10 Dataset Considerations

The CIFAR-10 dataset provides an ideal testbed for our UAV-assisted federated learning approach. The 32×32 RGB images balance simplicity and complexity - efficiently processable on resource-constrained devices while containing sufficient detail for meaningful feature extraction. The ten balanced classes present diverse classification challenges that mirror real-world IoT scenarios.

Distributing CIFAR-10 across our four-sensor setup introduces practical challenges similar to those in IoT deployments. Each sensor receives a unique subset of examples, creating natural non-IID (non-Independently and Identically Distributed) data patterns. This characteristic is valuable for testing federated learning systems, as it replicates the heterogeneous data collection patterns common in distributed IoT sensor networks where devices capture data under different conditions.

The dataset's complexity is suitable for evaluating the core trade-offs in federated learning for IoT: model performance versus communication efficiency and privacy preservation. Without requiring computational resources impractical for IoT devices, CIFAR-10 still presents meaningful challenges for our federated learning approach.

The modest image resolution aligns with many real-world IoT scenarios where bandwidth limitations restrict image quality. This makes our experimental setup directly relevant to applications such as environmental monitoring or infrastructure inspection, where UAVs might collect model updates from distributed image sensors operating under similar constraints.

C. Simulation Setup

Our setup models environmental monitoring in remote areas, where sensors deployed across forests collect data on temperature, humidity, and acoustic signals indicating illegal activities. Due to data sensitivity and remote locations, local processing is preferred over central transmission.

A UAV periodically collects locally trained models, aggregates them into a global model, and redistributes this to sensors. The simulation used GrADyS-SIM NG [18], designed for IoT environments with UAV support. The 200×200 unit grid contained four sensors at coordinates (150,0), (0,150), (-150,0), and (0,-150), each training local models on its data subset.

A single UAV follows a predefined path with 30-unit communication range, interacting only with sensors in range. It aggregates models only when receiving updates from sensors that completed training cycles, then immediately redistributes the updated model to in-range sensors.

We deliberately set the 30m UAV-sensor distance to emulate constraints where signal attenuation and energy limitations are critical. This shorter range emphasizes connectivity challenges and route-planning complexities, providing a rigorous test scenario.

This process mirrors federated learning where distributed nodes collaborate without sharing raw data. The simulation runs for a fixed duration, allowing multiple aggregation rounds.

D. Federated Learning Workflow

The federated learning process follows these steps:

1) **Local Training:** Sensors train initial local models on their data subsets, optimizing reconstruction loss and classification accuracy.
2) **Model Update Request:** After predefined epochs, sensors await UAV communication signals.
3) **Model Transmission:** When receiving UAV signals, sensors compress and quantize model parameters before transmission.
4) **Model Aggregation:** The UAV collects and aggregates models into an updated global model.
5) **Global Model Distribution:** The updated model is sent back to sensors for next training rounds.
6) **Iterative Improvement:** This process repeats, progressively enhancing global model performance.

E. Optimization Techniques

Several techniques enhanced communication efficiency and model performance:

- **Quantization:** Model parameters were quantized before transmission using Quantization-Aware Training, reducing memory usage and communication overhead while maintaining performance.
- **Compression:** Model state dictionaries were compressed with gzip to further reduce transmission load.
- **Adaptive Learning Rate:** An adaptive scheduler optimized training, accelerating convergence while adapting to varying data distributions across sensors.

F. Evaluation Metrics

Model performance was assessed using reconstruction quality and classification accuracy metrics. Mean Squared Error measured autoencoder reconstruction accuracy, with lower values indicating better performance. Classification accuracy reflected correct categorization rates, with higher values showing more effective classification.

- **Mean Reconstruction Loss:** MSE measured how well autoencoders reconstructed original input from latent space.
- **Classification Loss:** Evaluated classification head performance during image classification.
- **Accuracy:** Overall correct classification percentage, a primary performance indicator.
- **Clustering Accuracy:** Assessed encoder feature extraction using K-means clustering.
- **Adjusted Rand Index (ARI):** Evaluated similarity between predicted clusters and true labels.
- **Confusion Matrix:** Visualized classification performance, showing correct/incorrect predictions by class.
- **t-SNE Visualization:** Projected high-dimensional data to visualize feature space separation, showing cluster distinction from both autoencoder and supervised models.

The confusion matrix illustrates classification errors between categories, while t-SNE plots represent feature space separability, indicating whether learned features form distinct clusters.

IV. System Architecture and Simulation Design

This section outlines the organization of the federated learning system, focusing on the Autoencoder model, Sensor Protocol, UAV Protocol, and the handling of communication and optimization techniques in an IoT environment.

A. Autoencoder Model and Network Layers

The autoencoder model, implemented using PyTorch, is central to the feature extraction process in the federated learning system, with the added goal of image classification.

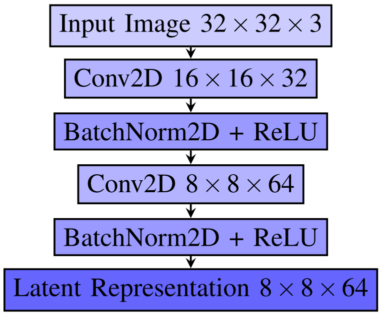

Fig. 1: Encoder Network Layers

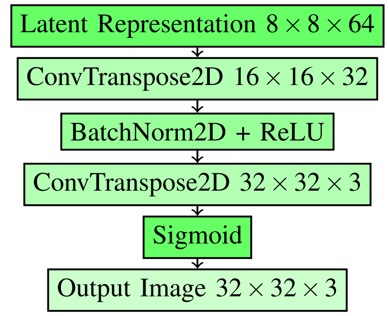

Fig. 2: Decoder Network Layers

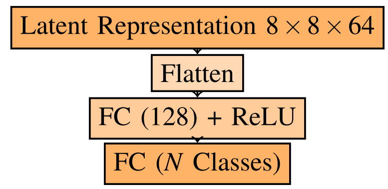

Fig. 3: Classification Head Network Layers

The network consists of three main components: the encoder, the decoder, and the classification head, each playing a crucial role in processing, reconstructing, and classifying image data.

1) Encoder, Decoder, and Classification Head: The encoder compresses the input image data into a lower-dimensional, feature-rich representation. This compression is vital for reducing the data transmission load between sensors and UAVs. The encoder, as illustrated in Figure 1, consists of two convolutional layers, each followed by batch normalization and ReLU activation functions. The first convolutional layer reduces the input image size from $32 \times 32 \times 3$ to $16 \times 16 \times 32$, and the second layer further compresses it to $8 \times 8 \times 64$. These layers are responsible for extracting hierarchical features from the image while retaining essential information.

Once the data has been compressed into a latent representation, or bottleneck, of size $8 \times 8 \times 64$, it is passed to two separate branches: the decoder and the classification head.

The decoder, shown in Figure 2, reconstructs the original image from the compressed format. It utilizes two transposed convolutional layers, each followed by batch normalization and ReLU activation, to expand the data back to its original size. The final layer of the decoder applies a sigmoid activation function to produce the reconstructed output image of size $32 \times 32 \times 3$. The quality of the reconstruction is measured using the Mean Squared Error (MSE), which guides the training process to ensure accurate image reconstruction.

In parallel, the classification head, depicted in Figure 3, flattens the latent representation and passes it through two fully connected layers. The first fully connected layer reduces the feature dimension to 128, followed by a ReLU activation. The final fully connected layer outputs the class probabilities across the predefined number of classes (e.g., 10 for CIFAR-10), enabling the model to classify the input images.

B. Sensor Protocol

The sensor protocol governs the behavior of individual IoT sensors within the network, enabling each sensor to operate independently while contributing to the collective learning process. Each sensor trains a local version of the autoencoder on its local subset of data, allowing it to adapt the model to local data distributions, which can vary significantly across different sensors. The key tasks managed by this protocol include:

- **Local Model Training:** Each sensor performs continuous training on its local dataset, periodically updating the model weights. This allows the sensor to refine its understanding of the local data, which is crucial in environments where data distribution is non-IID (non-Identically Independently Distributed) across the network. The model training process is optimized to run efficiently on the limited computational resources available on IoT devices.
- **Model Communication:** Upon receiving a request from the UAV, each sensor transmits its trained model weights. This step is vital for the federated learning process, as the aggregation of these local models forms the basis of the global model. The sensor protocol ensures that communication is efficient, transmitting only the necessary model updates to maintain data privacy and reduce the communication overhead. The protocol also manages the compression and quantization of model data before transmission to further optimize the communication process.

C. UAV Protocol

The UAV protocol orchestrates the aggregation and distribution of models across the network, serving as a central coordinator in the federated learning process. The UAV plays a pivotal role in ensuring that the collective knowledge of all sensors is integrated into a coherent global model. Key functions of the UAV protocol include:

- **Model Aggregation:** The UAV collects the model weights from multiple sensors, aggregates them into a single global model, and redistributes the updated model back to the sensors. This aggregation process involves averaging the model parameters received from each sensor, ensuring that the global model reflects the learning from the entire network, thereby enhancing overall model performance.

- **Communication Management:** The UAV manages communication with the sensors , ensuring timely data exchange with sensors within the transmission range. The UAV's mobility allows it to optimize its flight path dynamically, which is not attempted nor covered in this work
- **Global Model Evaluation:** At the end of the simulation, the UAV evaluates the performance of the global model using a separate test dataset. This evaluation is critical for identifying any performance issues and guiding further model improvements.

D. Message Handling and Optimization

The communication between sensors and the UAV is facilitated through structured messages that ensure efficient data exchange and minimize communication overhead. The primary messages exchanged include:

- **Model Update Request:** Sent by the UAV to initiate the transmission of local model weights from sensors.
- **Model Update Response:** Sent by sensors in response to a request, containing the compressed and quantized model weights.
- **Global Model Update:** Sent by the UAV after aggregating the local models, instructing sensors to update their local models with the new global parameters.

Model weights undergo gzip compression and quantization before transmission. This optimizes data transfer in bandwidth-limited IoT networks by reducing size while maintaining precision [26]. Table II shows autoencoder model size reduces from 2.197 MB to 0.562 MB after quantization, while supervised model size decreases from 2.415 MB to 0.619 MB. This significant reduction ensures efficient communication between sensors and UAVs while preserving model integrity and minimizing resource usage, enabling scalable federated learning deployment in IoT environments.

V. Experimental Results, Discussion, and Conclusion

A. Experimental Results

This section presents a detailed comparison of the autoencoder-based federated learning approach and the baseline supervised learning model. The experiment ran for 80 trials, each lasting 15,000 seconds, across both scenarios. Key metrics such as Loss, Accuracy, Mean Squared Error (MSE), Clustering Accuracy, Adjusted Rand Index (ARI), and Confusion Matrix are used to evaluate the performance. After each run, the model state was saved for subsequent runs to restart training.

1) Experimental Setup and Configurations: The simulations were run using the sensor protocol, described below, for both the autoencoder and the supervised learning models:

TABLE I: Model Configurations for Autoencoder and Supervised Learning

Parameter	**Autoencoder**	**Supervised Model**
Training Approach	Unsupervised (Autoencoder)	Supervised (Direct Classification)
Number of Training Cycles	80	80
Duration (seconds)	15,000	15,000
Learning Rate	0.001	0.001
Batch Size	32	32

The CIFAR-10 dataset was divided equally among the four sensors, ensuring each sensor had access to a unique subset of the data. The UAV followed a predefined square-shaped path, visiting each sensor in a cyclical manner to collect model updates.

The communication protocol between the UAV and sensors incorporated optimizations to reduce transmission overhead. These optimizations included:

- **Model Quantization:** Local models were quantized before transmission to reduce their size.
- **Compression:** Model state dictionaries were compressed using gzip to further minimize data transfer.

2) Scenario 1: Autoencoder-Based Federated Learning: In this scenario, the autoencoder played a dual role of feature extraction and classification. One of the primary metrics observed was the mean reconstruction loss, where the model achieved a value of 0.2618. This indicates the autoencoder's ability to effectively reconstruct input images from the learned latent representations, showcasing its strength in capturing key features of the data.

When evaluating the classification performance, the classification head of the autoencoder yielded a loss of 0.7266. This value suggests that while the autoencoder is able to extract meaningful features, there is still significant room for improvement in its classification capabilities, particularly when compared to more traditional supervised models.

The overall accuracy of the autoencoder reached 74.97%, which, although lower than the accuracy achieved by the supervised model, reflects its ability to handle unlabeled data. This accuracy, while respectable, highlights the trade-offs associated with unsupervised learning, where the focus is more on feature extraction rather than optimizing classification performance.

In terms of clustering, the autoencoder struggled. It achieved a clustering accuracy of 19.75%, indicating that the features extracted were not distinct enough to form well-separated clusters. This difficulty in clustering suggests that while the autoencoder is useful for feature extraction, the learned features are not sufficiently differentiated to improve clustering performance over a supervised approach.

The Adjusted Rand Index (ARI) for the autoencoder stood at 0.0363, further reflecting the model's moderate clustering performance. This score indicates weak alignment between the

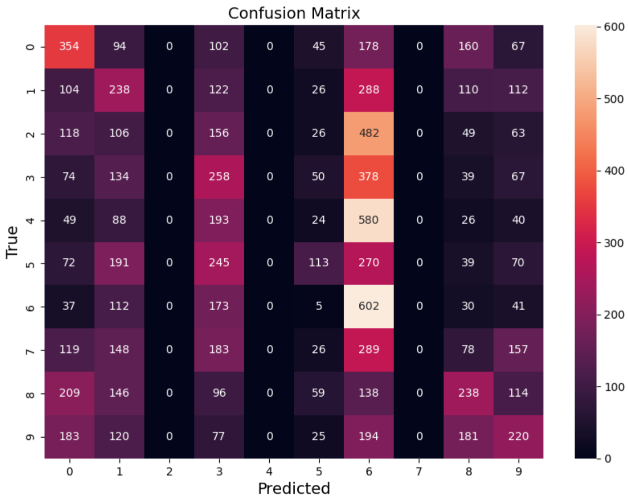

Fig. 4: Confusion Matrix for Autoencoder-Based Federated Learning

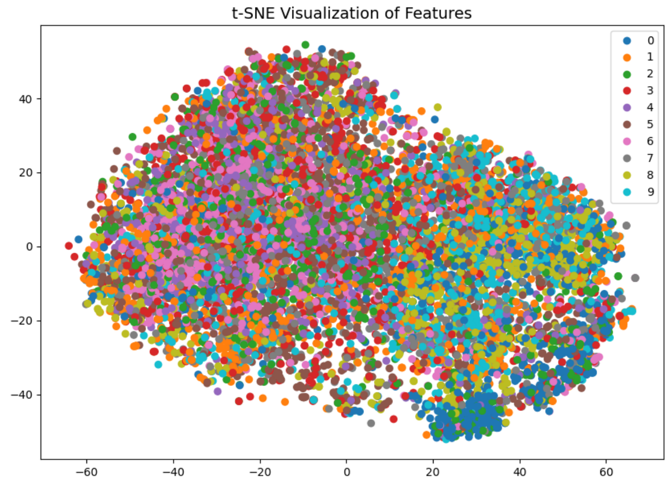

Fig. 5: t-SNE Visualization of Features for Autoencoder-Based Federated Learning

predicted clusters and the true labels, underscoring the challenges the model faced in effectively separating the classes.

The confusion matrix (Figure 4) revealed that while the autoencoder was able to classify some classes with a degree of accuracy, there were notable misclassifications. This suggests that while the model can extract basic features, it struggles to distinguish between certain classes, particularly those with overlapping features, leading to higher misclassification rates.

Finally, the t-SNE visualization (Figure 5) shows that the learned features from the autoencoder were not well-separated into distinct clusters. This lack of clear separation contributes to the model's poorer performance in clustering accuracy and highlights the challenges it faced in distinguishing between different classes.

3) Scenario 2: Supervised Learning Model: In this scenario, the supervised learning model was optimized specifically for direct classification tasks. One of the key metrics observed was the loss, where the supervised model achieved a lower loss of 0.5285. This lower loss demonstrates better convergence and training outcomes compared to the autoencoder, indicating that the supervised approach is more efficient at minimizing classification errors when labeled data is available.

The model's accuracy reached 82.4%, which is higher than the autoencoder's accuracy. This superior performance shows that the supervised model, benefiting from the availability of labeled data, was able to create clearer decision boundaries between classes, leading to more precise classification results.

The clustering accuracy of the supervised model was 27.42%, reflecting its ability to form more distinct and meaningful clusters than the autoencoder. The higher clustering accuracy suggests that the supervised model learned features that were better differentiated, enabling more accurate grouping of data points based on their class labels.

When examining the Adjusted Rand Index (ARI), the supervised model recorded a score of 0.1010. This score, though still relatively low, indicates better alignment between the predicted clusters and the true class labels compared to the autoencoder. The supervised model's ARI score highlights its improved capacity for distinguishing between different classes.

The confusion matrix (Figure 6) shows that the supervised model produced fewer misclassifications compared to the autoencoder. This reduction in errors further demonstrates the strength of the supervised model in accurately distinguishing between classes, benefiting from the direct supervision of labeled data during training.

Lastly, the t-SNE visualization (Figure 7) reveals better-separated clusters in the supervised model. This clear separation of data points into distinct clusters aligns with the higher clustering accuracy and ARI. The visualization underscores the supervised model's ability to learn discriminative features that better differentiate between the various classes, making it more suitable for tasks where classification accuracy is a priority.

B. Model Size Reduction through Quantization

Quantization effectively reduces model sizes, which is essential for deployment in resource-constrained environments. Table II shows the sizes of both models before and after quantization. Notably, the autoencoder model has a smaller size compared to the supervised model.

TABLE II: Model sizes before and after quantization

Model Type	**Non-Quantized Size (MB)**	**Quantized Size (MB)**
Autoencoder	2.197	0.562
Supervised	2.415	0.619

For the autoencoder, quantization reduced the model size from 2.197 MB to 0.562 MB, achieving a reduction of approximately 74.4%. The supervised model saw a similar

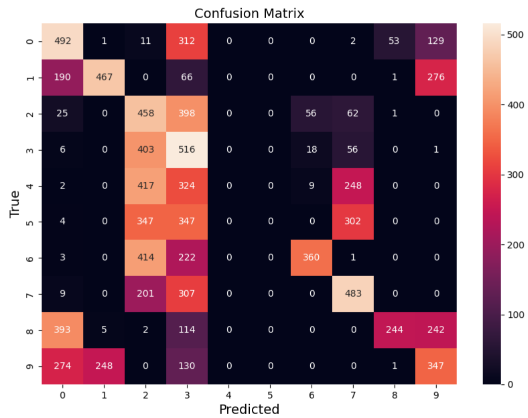

Fig. 6: Confusion Matrix for Supervised Learning Model

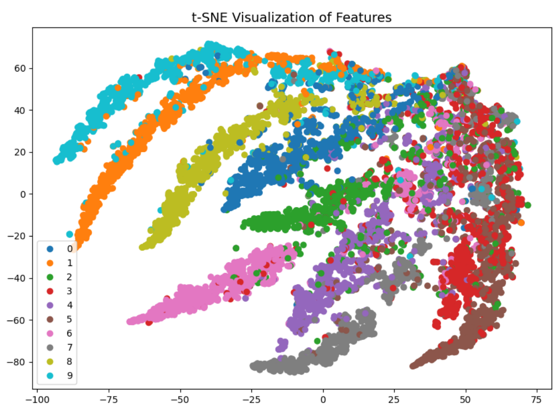

Fig. 7: t-SNE Visualization of Features for Supervised Learning Model

reduction from 2.415 MB to 0.619 MB. The smaller model size of the autoencoder highlights its efficiency, making it more suitable for devices with limited storage capacity.

Reducing model sizes not only saves storage space but also decreases communication overhead in federated learning systems. The significant size reductions demonstrate the effectiveness of quantization in optimizing models for practical, real-world applications.

C. Comparison with Standard Benchmarks

The performance of our autoencoder-based federated learning approach on the CIFAR-10 dataset (74.97% accuracy) differs from results reported in standard centralized benchmarks such as those by Thakkar et al. [27] and Recht et al. [28]. This discrepancy can be attributed to several key factors inherent to our federated learning framework and its specific design choices.

First, our implementation prioritizes data privacy and communication efficiency over maximum classification accuracy. While traditional centralized models can access the entire dataset at once with consistent hardware resources, our model operates under the constraints of distributed training across resource-limited IoT devices, with only periodic model aggregation via UAV. This fundamental architectural difference introduces several challenges not present in benchmark studies:

- **Non-IID Data Distribution:** Each sensor in our setup trains on a distinct local subset of data, leading to potential distribution shifts and model biases that can impact overall performance. The periodic nature of model aggregation may not fully mitigate these biases compared to centralized training approaches.
- **Communication Constraints:** To optimize bandwidth usage between sensors and UAVs, our models undergo quantization and compression before transmission. While these techniques significantly reduce model size (by approximately 74%), they introduce slight information loss that can affect model performance.
- **Architectural Trade-offs:** Our autoencoder architecture is deliberately designed to balance feature extraction capability against computational efficiency and model size. More complex architectures might achieve higher accuracy but would be impractical for deployment in resource-constrained IoT environments.
- **Dual Optimization Objectives:** Unlike standard classification models that optimize solely for prediction accuracy, our autoencoder simultaneously optimizes for both reconstruction quality and classification performance. This dual objective inevitably results in performance trade-offs compared to models focused exclusively on classification.

These differences highlight that direct comparisons with benchmark studies should be interpreted within the context of our system's design constraints and objectives. While our approach achieves lower classification performance than centralized benchmarks, this trade-off is deliberate and enables critical benefits in privacy preservation, communication efficiency, and adaptability to unlabeled data in resource-constrained IoT environments.

VI. Conclusion

The integration of autoencoders and supervised learning models within a federated learning framework offers a promising solution for IoT-based image classification challenges. This study highlights the strengths and limitations of autoencoders, which excel in data reconstruction and feature extraction but underperform in direct classification tasks compared to supervised models. This performance gap stems from the autoencoders' focus on minimizing reconstruction error rather than optimizing classification accuracy. Nevertheless, their ability to operate in data-scarce environments and to reduce communication overhead makes them highly valuable for IoT applications where data privacy and resource constraints are critical concerns.

Additionally, the use of UAVs to aggregate locally trained models and redistribute them across the network enhances the federated learning framework by ensuring continuous model improvement while preserving data privacy. The study emphasizes the trade-offs between autoencoders and supervised models, particularly regarding data efficiency and classification performance. While supervised models excel in scenarios with abundant labeled data, autoencoders present a more flexible and privacy-preserving alternative in contexts where data labeling is costly or impractical. Furthermore, the research highlights the importance of efficient communication protocols in bandwidth-constrained IoT environments and underscores the challenges posed by non-IID data distributions, which remain a significant obstacle to achieving robust and generalizable global models.

Future directions aim to refine and extend the application of federated learning models in IoT environments, ensuring they effectively leverage the computational capabilities of distributed devices while preserving data privacy and minimizing communication overhead. By addressing the challenges identified in this study, future research can contribute to the development of more robust, efficient, and scalable federated learning systems that meet the demands of real-world IoT applications.

ACKNOWLEDGMENT

This study was financed in part by AFOSR grant FA9550-23-1-0136.

REFERENCES

[1] M. Lahby, R. Saadane, and S. D. Correia, "Integration of iot with cloud computing for next-generation wireless technology," *Annals of Telecommunications*, vol. 78, no. 5, pp. 653–654, 2023.

[2] M. Ayache, I. El Asri, J. N. Al-Karaki, M. Bellouch, A. Gawanmeh, and K. Tazzi, "Enhanced dass-care 2.0: a blockchain-based and decentralized fl framework," *Annals of Telecommunications*, 2023.

[3] T. H. Rafi, F. A. Noor, T. Hussain, and D.-K. Chae, "Fairness and privacy-preserving in federated learning: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2306.08402

[4] D. Hemanand, C. Ambika Bhuvaneswari, S. Savitha, M. Jothi, and R. Senthil Rama, "Optimizing energy-efficient communication protocols for iot devices in smart cities using narrowband iot and lte-m technology," *Journal of Electrical Systems*, vol. 20, no. 5s, 2024.

[5] K. Yasumoto, H. Yamaguchi, and H. Shigeno, "Survey of real-time processing technologies of iot data streams," *Journal of Information Processing*, vol. 24, no. 2, pp. 195–202, 2016.

[6] J. Wang, J. Zhang, and Z. Ding, "A survey on federated learning for resource-constrained iot devices: Concepts, applications, and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3222–3241, 2021.

[7] S. Sun, Z. Zhang, Q. Pan, M. Liu, Y. Wang, T. He, Y. Chen, and Z. Wu, "Staleness-controlled asynchronous federated learning: Accuracy and efficiency tradeoff," *IEEE Transactions on Mobile Computing*, no. 01, pp. 1–14, jun 5555.

[8] T. Liu, J. Ding, T. Wang, M. Pan, and M. Chen, "Towards fast and accurate federated learning with non-iid data for cloud-based iot applications," *Journal of Circuits, Systems and Computers*, vol. 31, no. 13, p. 2250235, 2022. [Online]. Available: https://doi.org/10.1142/S0218126622502358

[9] P. Li, Y. Pei, and J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," *Applied Soft Computing*, vol. 138, p. 110176, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494623001941

[10] S. Chandar, P. Chandran, R. Bhat, and A. Chakravarthi, "Communication optimization in large scale federated learning using autoencoder compressed weight updates," 2021. [Online]. Available: https://arxiv.org/abs/2108.05670

[11] W. Y. B. Lim, S. Garg, Z. Xiong, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Uav-assisted communication efficient federated learning in the era of the artificial intelligence of things," *IEEE Network*, vol. 35, no. 5, pp. 188–195, 2021.

[12] M. Beitollahi and N. Lu, "Federated learning over wireless networks: Challenges and solutions," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 749–14 763, 2023.

[13] F. Liberti, D. Berardi, and B. Martini, "Federated learning in dynamic and heterogeneous environments: Advantages, performances, and privacy problems," *Applied Sciences*, vol. 14, no. 18, p. 8490, 2024.

[14] H.-Y. Hsu, K. H. Keoy, J.-R. Chen, H.-C. Chao, and C.-F. Lai, "Personalized federated learning algorithm with adaptive clustering for non-iid iot data incorporating multi-task learning and neural network model characteristics," *Sensors*, vol. 23, no. 22, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/22/9016

[15] O. Fagbohungbe, S. R. Reza, X. Dong, and L. Qian, "Efficient privacy preserving edge intelligent computing framework for image classification in iot," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 4, pp. 941–956, 2022.

[16] M. Tschannen, O. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning," 2018. [Online]. Available: https://arxiv.org/abs/1812.05069

[17] J. Zheng, W. Ni, H. Tian, D. Gündüz, T. Q. S. Quek, and Z. Han, "Semi-federated learning: Convergence analysis and optimization of a hybrid learning framework," *IEEE Transactions on Wireless Communications*, vol. 22, no. 12, pp. 9438–9456, 2023.

[18] T. Lamenza, M. Paulon, B. Perricone, B. Olivieri, and M. Endler, "Gradys-sim-a omnet++/inet simulation framework for internet of flying things." [Online]. Available: http://www.lac.inf.puc-rio.br/index.php/gradys/

[19] M. E. Bruno Jose Olivieri de Souza, "Ventures, Insights, and Ponderings from Real-World Experiments with FANETs and UAVs," in *29th IEEE Symposium on Computers and Communications (IEEE ISCC 2024)*, 2024, p. to appear.

[20] M. Fu, Y. Shi, and Y. Zhou, "Federated learning via unmanned aerial vehicle," *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 2884–2900, 2024.

[21] H. Jin, X. Jin, Y. Zhou, P. Guo, J. Ren, J. Yao, and S. Zhang, "A survey of energy efficient methods for uav communication," *Vehicular Communications*, vol. 41, p. 100594, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209623000244

[22] M. Zawish, S. Davy, and L. Abraham, "Complexity-driven model compression for resource-constrained deep learning on edge," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 8, pp. 3886–3901, 2024.

[23] G. Lan, X.-Y. Liu, Y. Zhang, and X. Wang, "Communication-efficient federated learning for resource-constrained edge devices," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 1, pp. 210–224, 2023.

[24] Z. Tong, J. Wang, X. Hou, J. Chen, Z. Jiao, and J. Liu, "Blockchain-based trustworthy and efficient hierarchical federated learning for uav-enabled iot networks," *IEEE Internet of Things Journal*, pp. 1–1, 2024.

[25] F. Mosaiyebzadeh, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, M. Han, L. Zhao, G. Sannino, C. M. Ranieri, J. Ueyama, and D. M. Batista, "Privacy-enhancing technologies in federated learning for the internet of healthcare things: A survey," *Electronics*, vol. 12, no. 12, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/12/2703

[26] M. S. M, P. V. Madnur, S. H. Dabade, A. Khanapure, S. Rodrigues, S. Hegde, and U. Kulkarni, "Enhancing deep neural networks through pruning followed by quantization pipeline: A comprehensive review," in *2023 2nd International Conference on Futuristic Technologies (INCOFT)*, 2023, pp. 1–8.

[27] V. Thakkar, S. Tewary, and C. Chakraborty, "Batch normalization in convolutional neural networks — a comparative study with cifar-10 data," in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, 2018, pp. 1–5.

[28] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do CIFAR-10 classifiers generalize to CIFAR-10?" *arXiv preprint arXiv:1806.00451*, 2018. [Online]. Available: https://arxiv.org/abs/1806.00451