Comparison of Federated Learning and Deep Learning in Mobile Applications

Yunus Emre Acar

***Abstract*—This paper presents a comparative performance analysis of Federated Learning (FL) and centralized Deep Learning (DL) approaches for Human Activity Recognition (HAR) within mobile ecosystems. The study evaluates both paradigms using a multi-layer neural network architecture—comprising dense layers with 512, 256, 128, and 64 neurons—optimized with Batch Normalization and Dropout layers to ensure training stability and prevent overfitting. Experimental evaluations are conducted on a Samsung Galaxy M34 mobile device using the UCI HAR dataset, classifying six distinct physical activities using 561 extracted features from accelerometer and gyroscope sensors. The results demonstrate that the centralized DL model achieves a peak classification accuracy of 94.64% with a rapid training duration of 32 seconds. In comparison, the FL approach provides a comparable accuracy of 93.72% while maintaining data privacy by keeping raw sensor data on-device, though it incurs a significantly higher training time of 9 minutes and 48 seconds due to distributed communication and synchronization overhead. Resource consumption metrics indicate that while centralized DL demands higher RAM (169.04 MB), the FL weight update mechanism is more memory-efficient (148.50 MB), making it suitable for resource-constrained mobile environments. This research highlights the critical trade-offs between classification performance, training efficiency, and data privacy, concluding that hybrid strategies are essential for the next generation of privacy-preserving mobile AI applications.**

***Index Terms*—Deep Learning, Federated Learning, Human Activity Recognition, Mobile Applications, Performance Evaluation, Privacy-Preserving Machine Learning**

I. Introduction

RECENT advancements in Artificial Intelligence (AI) have triggered a significant transformation across various domains, particularly in text recognition, image processing, and data analytics [7]. A prominent example of this evolution is the implementation of Human Activity Recognition (HAR) systems within smart home environments, which facilitate automation and energy optimization based on user behavior [6]. By accurately detecting physical states and mobility levels, these systems can autonomously regulate critical home infrastructure, such as lighting and heating systems, to align with the occupant's current needs.

The core logic of this sustainable approach, as illustrated in **Fig. 1**, relies on a continuous feedback loop between mobile sensors and home controllers. When the HAR model identifies a "Rest Period" characterized by low mobility—such as sitting or lying down—the smart home controller automatically transitions the environment into a low-energy mode. This proactive regulation minimizes unnecessary consumption while ensuring that high-comfort settings are only active when the user is in an engaged state. Consequently, such systems contribute significantly to global sustainability goals by reducing the carbon footprint of residential buildings through intelligent automation.

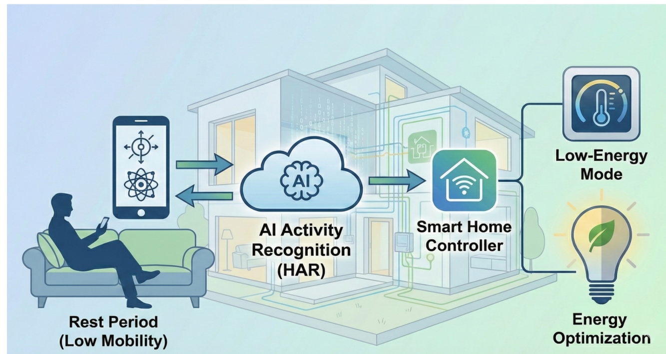


Fig. 1: The Example of Usability

Beyond the benefits of efficiency and comfort, the inherent sensitivity of domestic behavioral data presents a significant challenge for traditional cloud-based AI models.As deep learning becomes integrated into numerous human-centered systems, the inherent security vulnerabilities of cloud-centric architectures, such as potential data exposure and latency, have necessitated a shift toward decentralized learning paradigms [17]. Since raw sensor data often contains intimate details of a user's private life, transmitting this information to a central server creates substantial privacy risks. This necessitates the adoption of privacy-preserving paradigms such as Federated Learning (FL), which allows for decentralized model training directly on the user's device. In this study, we evaluate the performance of FL against traditional Deep Learning (DL) approaches to establish a robust framework for secure and energy-efficient smart home ecosystems.

Federated Learning has emerged as a decentralized learning approach that prioritizes user privacy by eliminating the requirement for central data collection. However, this paradigm faces fundamental challenges, including data heterogeneity, communication costs, and security vulnerabilities [1]. Previous research has addressed these issues through algorithms like FedAvg and differential privacy techniques [1], while others have highlighted the potential of FL in the Internet of Things (IoT) and mobile environments for bandwidth optimization [2]. Practical implementations, such as mental health monitoring frameworks and cross-platform protocols like FedKit, demonstrate the growing viability of FL in the

Manuscript received January 20, 2026; revised January 20 2026.
M. Erel-Özçevik was with the Department of Software Engineering, Manisa Celal Bayar University, 45400 Türkiye. E-mails: (muge.ozcevik@cbu.edu.tr).
Manuscript received January 20, 2026; revised January 20 2026.

mobile ecosystem [9], [12].

Conversely, Deep Learning (DL) continues to dominate mobile applications due to its ability to hierarchically learn abstract and complex patterns from large datasets. While DL models have shown remarkable success in tasks such as sentiment analysis and healthcare predictions, their deployment on mobile devices is often hindered by high resource consumption in terms of memory and processing power. To mitigate these constraints, researchers have proposed edge computing, Neural Processing Units (NPUs), and adaptive compression techniques [14], [15]. When comparing the two paradigms, FL offers superior privacy and reduced bandwidth usage, whereas DL provides higher accuracy in large-scale homogeneous datasets [1], [2], [4].

Despite the existence of studies evaluating FL and DL independently, there is a notable gap in the literature regarding a direct, comprehensive comparison of these methods on mobile hardware concerning processing time, hardware utilization, accuracy, and data privacy [13]. This study aims to fill this void by providing a detailed performance analysis of FL and DL-based systems optimized for mobile environments [1], [2]. The primary contributions of this work include: 1) A comparative evaluation of processing efficiency and resource consumption; 2) A privacy and security perspective on decentralized vs. centralized processing; and 3) A roadmap for developers to optimize next-generation AI solutions for mobile platforms [3], [5].

The remainder of this paper is organized as follows. Section II reviews the fundamental principles and related work in FL and DL. Section III details the proposed system methodology, including model architectures and API design. Section IV presents the experimental results and performance analysis. Section V discusses the research findings and potential risks. Finally, Section VI provides concluding remarks and suggestions for future work.

II. Related Work

The rapid evolution of on-device intelligence has led to extensive research in both Federated Learning (FL) and Deep Learning (DL). This section explores the state-of-the-art developments in these fields, specifically focusing on their application and performance within mobile ecosystems.

A. Federated Learning in Mobile Systems

Federated Learning was introduced to address the privacy limitations of traditional centralized machine learning by allowing models to be trained across multiple decentralized devices holding local data samples [5]. Li et al. [1] identified the core challenges of FL as data heterogeneity, communication costs, and privacy security, emphasizing the effectiveness of the FedAvg algorithm in diverse environments. Abdulrahman et al. [2] further correlated the increasing use of FL in IoT and mobile devices with its ability to optimize energy and bandwidth through decentralized training.

Recent practical frameworks have focused on implementation feasibility. For instance, FedKit [12] enables cross-platform FL protocols between Android and iOS, reducing integration barriers. In specialized applications, Suruliraj and Orji [9] proposed frameworks for mental health monitoring that ensure user data remains local, while Geng et al. [6] demonstrated privacy-preserving mobile applications for smart campuses. Despite these advancements, the impact of client selection [10] and non-IID data distributions [11] remains a critical area of study for model stability.The studies by Liu et al. [5] emphasize that meta-learning based federated approaches effectively mitigate the challenges of statistical heterogeneity by learning global initializations that adapt to diverse client environments with minimal gradient updates.

B. Deep Learning on Resource-Constrained Hardware

Deep Learning (DL) functions as a specialized subfield of machine learning, utilizing artificial neural networks with multiple hidden layers to extract high-level features and patterns from large volumes of data [16] . On the other hand, Deep Learning has become the benchmark for complex tasks such as sentiment analysis [7] and healthcare prediction [8] due to its hierarchical feature learning capabilities. However, its deployment on mobile devices is significantly limited by high memory and processing power requirements [13]. To mitigate these constraints, researchers have suggested the use of Neural Processing Units (NPUs) and edge computing accelerators [14]. Furthermore, adaptive compression techniques [15] and hybrid approaches—where the base model remains on a server while personalized parameters are tuned on-device—have been proposed to enhance efficiency without sacrificing accuracy.

C. Comparative Perspectives

While many studies evaluate FL and DL in isolation, direct performance comparisons are sparse. Darwish and Roy demonstrated that Federated Learning models (FL-CNN) can achieve a high accuracy of 95.27% in IoT malware detection, effectively competing with centralized deep learning and traditional machine learning while preserving data privacy [3]. Furthermore, Huang et al. [4] noted that while DL excels in large-scale homogeneous datasets, FL provides a distinct advantage in producing personalized models in heterogeneous data environments. This study builds upon these findings by providing a holistic comparison of processing time, hardware utilization, and accuracy specifically for Human Activity Recognition (HAR) on modern mobile hardware.

The proliferation of mobile devices has established Human Activity Recognition (HAR) as a core technology for diverse applications, ranging from healthcare monitoring to exercise assessment [19]. However, the shift toward Federated Learning (FL) in HAR systems is driven by the increasing demand for personalized models and robust data privacy, allowing collaborative training without centralizing sensitive raw data [20]. Despite these advantages, the practical implementation of FL in mobile edge networks faces significant challenges regarding communication overhead and energy consumption, necessitating cost-effective optimization strategies to ensure system convergence under resource constraints [18].

III. System Methodology and Implementation

In this section, we detail the experimental setup, including the dataset characteristics, neural network architectures for both Deep Learning (DL) and Federated Learning (FL), and the integrated system architecture consisting of a Flask-based API and an Android mobile application .

A. Dataset and Preprocessing

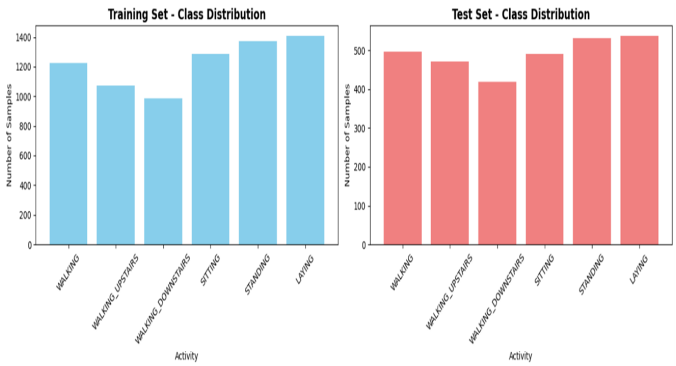
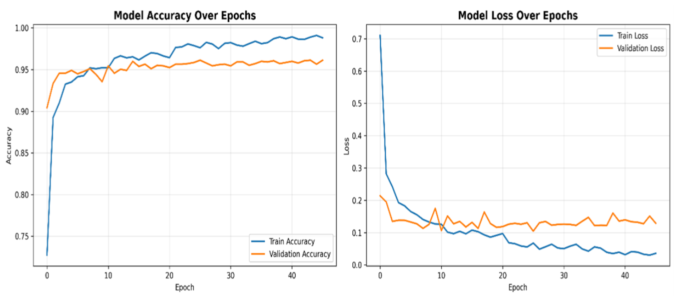


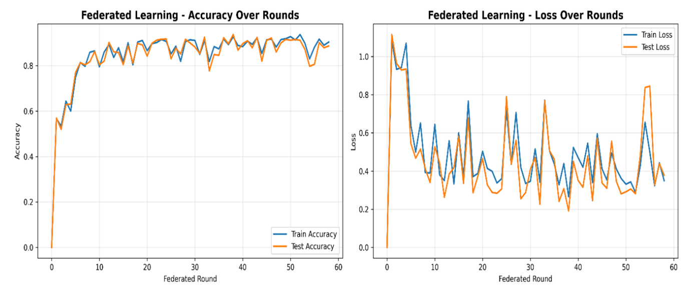
Fig. 2: Dataset Distribution

The proposed models were trained and evaluated using the UCI Human Activity Recognition (HAR) dataset. This dataset comprises sensor signals (accelerometer and gyroscope) captured from mobile devices, categorized into six distinct physical activities: walking, walking upstairs, walking downstairs, sitting, standing, and laying. To ensure a robust training process, the raw data underwent extensive preprocessing:

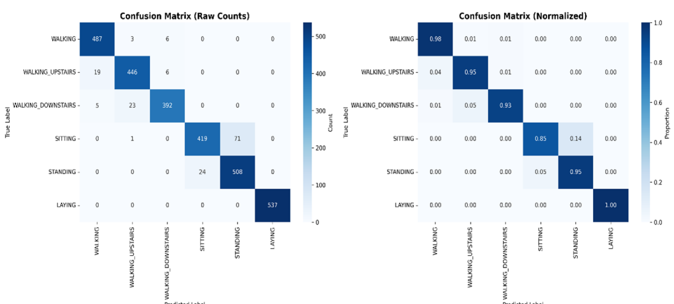
- **Feature Standardization:** All 561 features were standardized using a *StandardScaler* to improve convergence speed and stabilize training. This transformation ensures each feature has a mean of zero and unit variance, preventing data leakage between sets.
- **Label Encoding:** Categorical target labels were transformed into a One-Hot Encoded format to facilitate the use of categorical cross-entropy loss.



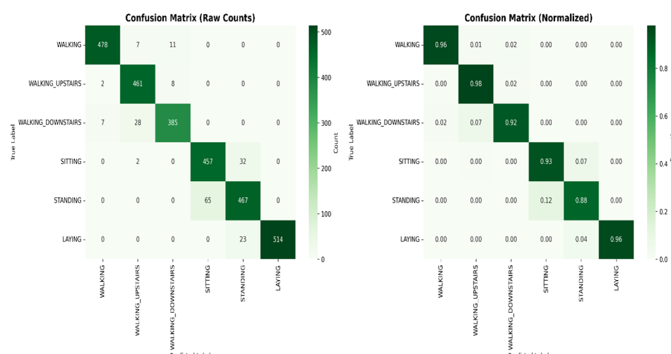
(a) Deep Learning Training Graphics



(b) Federated Learning Training Graphics



(c) Deep Learning Confusion Matrix



(d) Federated Learning Confusion Matrix

Fig. 3: Comparison of Federated Learning and Deep Learning Results

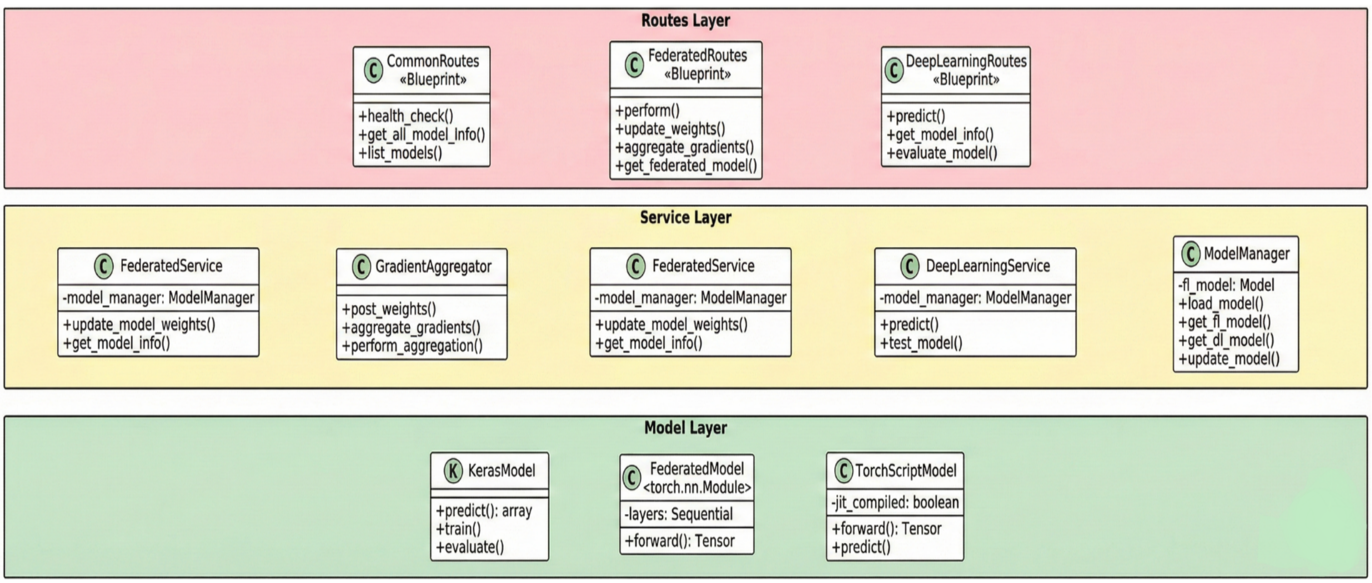


Fig. 4: API Diagram

B. Deep Learning Model Architecture and Training Analysis

For the centralized approach, a high-dimensional Artificial Neural Network (ANN) was implemented to classify human activities. The architecture comprises four dense hidden layers with 512, 256, 128, and 64 neurons, respectively.

1) **Regularization and Optimization:** To mitigate internal covariate shift, Batch Normalization was applied after each layer. Overfitting was strictly controlled through Dropout layers with decreasing rates of 0.5, 0.4, 0.3, and 0.2. The model utilized the Adam optimizer with a categorical cross-entropy loss function.
2) **Training Dynamics:**
The model was trained for 50 epochs with a batch size of 32. The integration of *EarlyStopping* ensured that the training halted once validation loss plateaued, while *ReduceLROnPlateau* dynamically adjusted the learning rate to fine-tune the weights.
As illustrated in the training curves presented in **Fig. 3a**, exhibited a consistent upward trend in accuracy, reaching a terminal test accuracy of 94.64%.As for the confusion matrix in **Fig. 3c**, the ”Laying” class achieved a perfect accuracy of 100.0%, while ”Sitting” was identified as the most challenging class at 85.3% due to its similarity to the ”Standing” position in accelerometer data.

C. Federated Learning Protocol

The FL framework utilized the same ANN architecture as the centralized model to ensure a fair performance comparison. In this decentralized scenario, each subject from the dataset acted as an independent virtual client, creating a natural Non-IID (Independent and Identically Distributed) environment . The training was conducted using the Federated Averaging (FedAvg) algorithm [1]:

1) **Training Dynamics:**
Despite the distributed nature and communication overhead, the FL model converged to a high test accuracy of 93.72% in **Fig. 3b**. The analysis of the training graphic revealed that while the accuracy is slightly lower than the DL model (a 0.92% difference), the privacy gains—achieved by never transmitting raw sensor data—justify this marginal performance trade-off.
2) **Performance Results:**
Similar to the DL approach, the FL model excelled in distinguishing dynamic activities (Walking Upstairs at 97.9%) in **Fig. 3d** but faced slight fluctuations in static activities due to the heterogeneous data distribution across different subjects.

D. System Integration: API and Mobile Application

The HAR API system is built on a Service-Oriented Architecture (SOA) and implemented as a RESTful web service using the Flask framework. This modular design facilitates the simultaneous management of Deep Learning (DL) and Federated Learning (FL) transactions across three distinct logical layers (**Fig. 4**):

- **Routes Layer (Presentation Layer):** This layer utilizes Flask Blueprints to organize the API's entry points into specialized modules. The *CommonRoutes* handle system health checks and metadata retrieval, while *FederatedRoutes* manage the coordination of distributed training rounds. The *DeepLearningRoutes* are dedicated to processing centralized inference requests. All communications are conducted via standardized JSON payloads to ensure low-latency interoperability between the mobile client and the server.
- **Service Layer (Business Logic Layer):** The core operational logic is executed within this layer. The *DeepLearningService* handles real-time sensor data prediction using centralized models, whereas the *FederatedService* manages the intake of local training updates from mobile clients. A critical component, the *GradientAggregator*, is responsible for implementing the FedAvg algorithm to synthesize decentralized weight updates into a cohesive global model.
- **Model Layer (Implementation Layer):** The *ModelManager* serves as a centralized controller for model lifecycle management. It implements an intelligent caching mechanism that loads models on demand and ensures the persistence of global weights during iterative FL rounds . This layer supports multiple backends, including *KerasModel* for high-performance centralized training and *TorchScriptModel* for seamless deployment on resource-constrained mobile hardware via PyTorch Mobile.

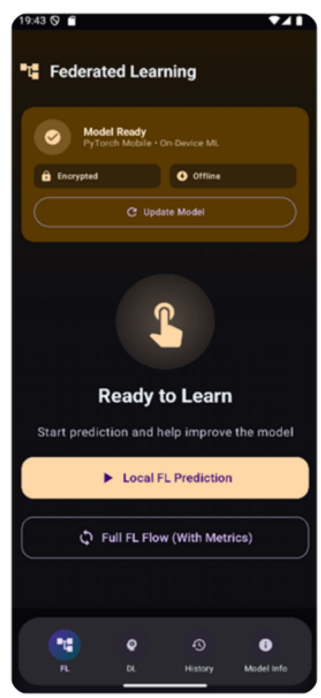


Fig. 5: Mobile Application UI Design.

The application is developed on the Android platform using the Kotlin programming language and the Jetpack Compose toolkit, adopting the Model-View-ViewModel (MVVM) pattern to decouple business logic from the presentation layer. Designed according to Material Design 3 principles, the UI consists of four primary functional modules: the Federated Learning (FL) screen for managing local and global federated training processes, the Deep Learning (DL) screen for server-side inference, the History screen for analyzing past predictions, and the Model Info screen for presenting model characteristics. This structured architecture enables the asynchronous execution of complex sensor data streams and model synchronization processes without compromising the user experience.

At the core of the system, the sensor data processing pipeline executes to transform raw accelerometer and gyroscope signals into meaningful feature vectors. Tri-axial data obtained from the mobile device hardware is sampled at a frequency of 50 Hz and segmented into sliding windows of 128 samples to capture temporal patterns. From each window, 561 distinct statistical features are extracted, covering metrics such as mean, standard deviation, signal magnitude area (SMA), energy, and correlation in both the time and frequency domains.

On an operational level, the application offers a hybrid working model. To provide local on-device inference capabilities, the models were converted to the PyTorch Mobile format, enabling them to operate independently of a network connection. In Federated Learning scenarios, local training is performed without raw user data leaving the device; only encrypted model weight updates are transmitted to the central API, demonstrating a privacy-by-design approach. To enhance the resource efficiency of the system, asynchronous data management is provided through Kotlin Coroutines and StateFlow structures.

IV. Experimental Results and Performance Analysis

In this section, we present a comprehensive performance evaluation of the proposed Federated Learning (FL) and Deep Learning (DL) systems. The experiments were conducted on a Samsung Galaxy M34 mobile device equipped with an Exynos 1280 CPU (2.4 GHz), 6 GB of RAM, and a Mali-G68 GPU. All metrics were calculated as the average of 20 independent runs to ensure statistical reliability.

A. *Resource Utilization: CPU and RAM*

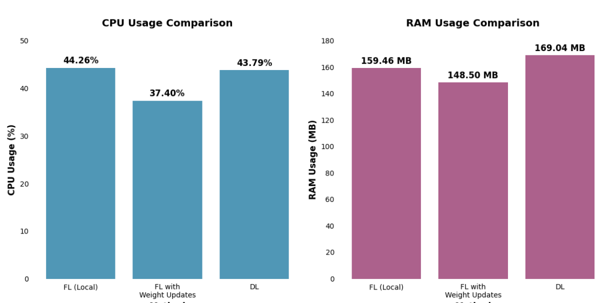


Fig. 6: CPU and RAM Comparison

The impact of each approach on mobile hardware resources is a critical factor for user experience and energy efficiency. As shown in our measurements:

- **CPU Usage:** The FL (Local) inference utilized 44.26% of the CPU, as all computations were performed on-device. In contrast, the FL approach with weight updates showed a reduced CPU load of 37.40%, as the heavy training was offloaded while maintaining prediction capabilities. The centralized DL model recorded a CPU usage of 43.79%, primarily driven by API communication and data preparation tasks.
- **RAM Consumption:** The DL method exhibited the highest memory footprint at 169.04,MB due to the overhead of managing API sessions and data buffers. The FL (Local) scenario required 159.46,MB. Notably, the FL weight update mechanism proved to be the most efficient, consuming only 148.50,MB of RAM.

B. *Inference Latency and Duration*

Latency is a key indicator of real-time usability in mobile HAR systems.

- The **FL (Local)** method achieved the fastest results with an average prediction duration of **2.780 seconds**, benefiting from the absence of network latency.
- The **Centralized DL** prediction took **3.340 seconds**, where the duration was directly influenced by network conditions and server response times.
- The **FL with Weight Updates** process required **4.470 seconds**, reflecting the additional time needed to transmit model gradients to the server after local prediction.

C. *Model Accuracy and Convergence Analysis*

The trade-off between decentralized privacy and classification performance was analyzed through two distinct training strategies.

1) **Iteration Sensitivity:**

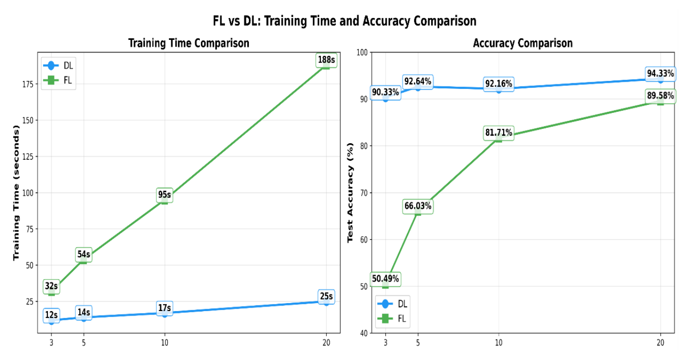


Fig. 7: Training Duration and Accuracy Comparison by Iteration

The DL model demonstrated high stability, maintaining accuracy between 90.33% and 94.33% even with limited iterations. Conversely, the FL model showed high sensitivity to the number of communication rounds, starting at 50.49% accuracy at 3 iterations and reaching 89.58% at 20 iterations.

2) **Final Performance with Early Stopping:**

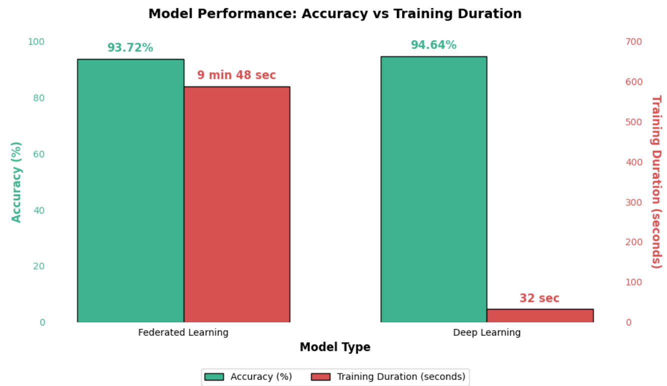


Fig. 8: Accuracy Duration by Early Stopping

When optimized using early stopping, the centralized DL model achieved a superior accuracy of 94.64% in a training duration of just 32 seconds . The FL model reached a comparable accuracy of 93.72%, though its distributed training required 9 minutes and 48 seconds to converge due to synchronization and communication overhead .

These results indicate that while DL remains the benchmark for accuracy and training speed, FL provides a robust and privacy-preserving alternative with a manageable performance penalty .

V. Discussion and Practical Implications

This section evaluates the research findings through a multi-dimensional lens, incorporating SWOT and risk analyses to provide a holistic view of the trade-offs between centralized and decentralized learning in mobile environments.

A. Strategic Analysis: SWOT Assessment

The integration of FL and DL within a single mobile framework offers unique strategic advantages. Our SWOT analysis reveals that the primary strength of the centralized DL model lies in its stable performance and high classification accuracy. Conversely, the FL architecture excels in preserving user privacy and ensuring compliance with data protection regulations by keeping sensitive sensor data on-device.

However, significant weaknesses persist. The FL approach suffers from slower convergence and higher communication overhead, which can impact device battery life and bandwidth consumption. On the other hand, the centralized DL paradigm faces inherent risks related to large-scale data breaches and centralized storage vulnerabilities. Opportunities for future growth include the proliferation of 5G and edge computing, which will likely reduce FL communication latencies and enable more complex on-device training.

B. Risk Assessment and Mitigation Strategies

The deployment of machine learning on mobile devices entails various technical and security risks:

- **Security and Privacy Risks:** While FL enhances privacy, it is susceptible to model poisoning and adversarial updates. Robust aggregation methods and anomaly detection are essential to mitigate these threats. Centralized DL remains vulnerable to inference attacks on stored data, necessitating strong encryption and strict access controls.
- **Operational Risks:** Device heterogeneity (varying CPU/RAM capacities) and high client dropout rates can degrade the quality of federated global models [10]. Implementing asynchronous FL protocols and adaptive training schedules can ensure system resilience in dynamic mobile network conditions.
- **Regulatory Compliance:** Adherence to frameworks such as GDPR and HIPAA is critical, especially for healthcare-oriented HAR systems. FL provides a significant advantage here by minimizing data movement and naturally supporting the ”right to be forgotten” principle.

C. The Privacy-Performance Trade-off

The empirical results underscore a fundamental trade-off: the 1% accuracy advantage and significantly lower training time of DL must be weighed against the superior privacy guarantees of FL. For applications where real-time personalization and data confidentiality are paramount (e.g., healthcare monitoring), the performance penalty of FL is acceptable. For general-purpose applications requiring maximum accuracy and minimal device load, centralized DL remains the preferred choice.

D. Practical Implications for Developers

Our findings suggest that a hybrid strategy—using centralized DL for initial global feature extraction and FL for local personalization—offers the most balanced solution. Developers should focus on optimizing on-device resource usage by utilizing NPU-based accelerators and implementing efficient synchronization protocols to minimize the communication-energy footprint of mobile AI applications.

VI. Conclusion

This study provided a comprehensive comparative evaluation of Federated Learning (FL) and centralized Deep Learning (DL) for Human Activity Recognition (HAR) on mobile platforms. Our experimental results, conducted on real-world mobile hardware, demonstrate that while centralized DL maintains a slight performance advantage with a classification accuracy of 94.64%, FL offers a highly competitive alternative at 93.72% while inherently ensuring user data privacy. The primary trade-off identified lies in training efficiency, where DL converges in approximately 32 seconds compared to the 9 minutes and 48 seconds required for the distributed FL architecture.

The analysis of resource utilization revealed that FL paradigms, particularly the weight update mechanism, offer a more memory-efficient profile (148.50 MB RAM) than centralized approaches, making them viable for long-term on-device deployment. However, the operational latencies associated with FL communication rounds and the technical challenges posed by device heterogeneity remain significant hurdles for widespread adoption.

We conclude that FL and DL are not mutually exclusive but rather complementary technologies. DL provides a robust foundation for global model development, while FL enables the critical personalization required for privacy-sensitive mobile applications. Consequently, a hybrid strategy—utilizing centralized pre-training followed by localized federated fine-tuning—emerges as the most effective roadmap for developing high-performance, privacy-preserving mobile AI systems.

Future research directions should focus on: The implementation of asynchronous FL protocols to mitigate synchronization delays; The development of adaptive model architectures that adjust to varying device hardware capacities; The Extensive field testing with real-world, non-IID user data to validate system resilience against environmental noise. Furthermore, strengthening FL security against emerging threats like gradient leakage and model poisoning remains paramount for the next generation of mobile intelligent systems.

REFERENCES

[1] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," in *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50-60, 2020.

[2] S. Abdulrahman et al., "A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond," in *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476-5497, 2021.

[3] R. Darwish and K. Roy, "Comparative Analysis of Federated Learning, Deep Learning, and Traditional Machine Learning Techniques for IoT Malware Detection," *2025 IEEE 4th ICAIC*, 2025.

[4] W. Huang et al., "Federated Learning for Generalization, Robustness, Fairness: A Survey and Benchmark," in *IEEE TPAMI*, vol. 46, no. 12, pp. 9387-9406, 2024.

[5] X. Liu, Y. Deng, A. Nallanathan and M. Bennis, "Federated Learning and Meta Learning: Approaches, Applications, and Directions," in *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 104-135, 2024.

[6] J. Geng et al., "FedCampus: A Real-world Privacy-preserving Mobile Application via Federated Learning & Analytics," in *Proc. of MobiHoc '24*, 2024.

[7] D. Goularas and S. Kamis, "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," *2019 Deep-ML*, 2019.

[8] C. G. Estonilo and E. D. Festijo, "Evaluation of the Deep Learning-Based m-Health Application," *2022 IEEE CCWC*, 2022.

[9] B. Suruliraj and R. Orji, "Federated Learning Framework for Mobile Sensing Apps in Mental Health," *2022 IEEE SeGAH*, 2022.

[10] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," *ICC 2019*, 2019.

[11] F. Sattler et al., "Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data," in *IEEE TNNLS*, vol. 31, no. 9, 2020.

[12] S. He et al., "Fedkit: Enabling Cross-Platform Federated Learning for Android and iOS," *IEEE INFOCOM WKSHPS*, 2024.

[13] T. Zhao et al., "A Survey of Deep Learning on Mobile Devices: Applications, Optimizations, Challenges," in *Proc. of the IEEE*, vol. 110, no. 3, 2022.

[14] T. Tan and G. Cao, "Deep Learning on Mobile Devices Through Neural Processing Units and Edge Computing," *IEEE INFOCOM 2022*, 2022.

[15] C. Hardy et al., "Distributed deep learning on edge-devices: Feasibility via adaptive compression," *2017 IEEE NCA*, 2017.

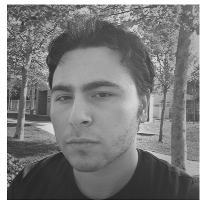
[16] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," in *Proc. 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, Bangkok, Thailand, 2017, pp. 1-6, doi: 10.1109/ICTKE.2017.8259629.

[17] W. G. Hatcher and W. Yu, "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends," in *IEEE Access*, vol. 6, pp. 24411-24432, 2018, doi: 10.1109/ACCESS.2018.2830661.

[18] B. Luo, X. Li, S. Wang, J. Huang and L. Tassiulas, "Cost-Effective Federated Learning in Mobile Edge Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3606-3621, Dec. 2021, doi: 10.1109/JSAC.2021.3118436.

[19] Y. Yin, L. Xie, Z. Jiang, F. Xiao, J. Cao and S. Lu, "A Systematic Review of Human Activity Recognition Based on Mobile Devices: Overview, Progress and Trends," in *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 890-929, Secondquarter 2024, doi: 10.1109/COMST.2024.3357591.

[20] O. Aouedi, A. Sacco, L. U. Khan, D. C. Nguyen and M. Guizani, "Federated Learning for Human Activity Recognition: Overview, Advances, and Challenges," in *IEEE Open Journal of the Communications Society*, vol. 5, pp. 7341-7367, 2024, doi: 10.1109/OJCOMS.2024.3484228.



Yunus Emre Acar is currently pursuing the B.S. degree in software engineering at Manisa Celal Bayar University, Manisa, Turkey. He is also working as a Software Engineer, focusing on the integration of artificial intelligence into mobile systems. His research interests include machine learning, federated learning, and mobile application development. He is particularly interested in the performance optimization of deep learning models on resource-constrained mobile devices.