International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V73I5P114

Volume 73 Issue 5, 101-112, May 2025
© 2025 Seventh Sense Research Group®

Original Article

A Federated Learning Approach for Predicting Resource Allocation in Multi-Access Edge Computing (MEC)

Ramesh Kasarla

Comcast cable communications, VA, USA.

Corresponding Author: Ramesh.kasarla@gmail.com

Received: 28 March 2025 Revised: 01 May 2025 Accepted: 16 May 2025 Published: 30 May 2025

Abstract *- In modern networks, where stronger ultra-low latency and data throughput are needed, Multi-Access Edge Computing (MEC) becomes a necessary architecture for 5G/6G networks that support real-time applications. Nevertheless, a dynamic edge ecosystem, diverse device properties, and privacy preservation needs interfere with MEC resource management. This paper proposes a new Federated Learning (FL) framework to predict resource allocation in MEC that removes such barriers by enabling decentralized model training to be performed directly at the network edge. In contradiction to conventional centralized strategies, our approach significantly reduces communication costs by up to 90% while providing competitive performance due to the efficient use of non-IID data at edge locations. Feeding lightweight CNNs and reducing the whole energy demand is achieved by the balanced computational requirements in the design aggregation through FedOpt aggregation. Based on the results of outcome analysis on MNIST and Fashion-MNIST, we observe accelerated convergence, increased energy savings and performance scalability, where energy consumption per training round is 29% lower than in centralized systems. This approach shows impressive results in processing non-IID data due to reliable performance on different edge devices. Such discoveries show that FL has a high potential to transform MEC resource allocation and thus contribute to more adaptive, protected, and efficient edge computing architecture.*

Keywords *- Federated Learning, Multi-Access Edge Computing, Resource Allocation, 5G/6G Networks, Non-IID Data, Distributed Machine Learning, Energy Efficiency, Model Aggregation.*

1. Introduction

Modern networking has significantly been boosted with Multi-Access Edge Computing (MEC), enabling increased computational and storage aspects nearer to the end users. [1-3] The reduced distance between users and resources in MEC networks results in lower latency levels, excludes network bottlenecks and improves the quality of service for such latency-sensitive applications as augmented reality, autonomous vehicles and industrial control systems. However, the explosion of connected devices and data-devouring applications creates a new requirement to manage and perform the distribution of resources at the network's edge. Conventionally, bandwidth constraints, high latency, and the limitations of centralized processing present obstacles that tend to plague such cloud-centric systems with these requirements, challenges that are especially difficult to solve. Efficient use of distributed resources in the MEC environments calls for rapid, data-driven decisions to maximize performance. Centralized deployment of machine learning models places many restrictions on their capability to address the requirements of MEC environments. Conventionally centralized methods necessitate continuous sending of data from the end nodes to the central server, which puts heavy demands on the bandwidth and opens doors to privacy concerns. Such concerns are particularly acute across healthcare, finance, and critical infrastructure industries, where data confidentiality is the highest priority. In order to address this, Federated Learning (FL) has been designed as a decentralized approach that allows edge devices to collaborate and train machine learning models, which remain private at the edge. Federated learning addresses privacy because local storage of confidential data is only shared on the server if parameters of well-trained models are utilized. Federated learning that has less consumption bandwidth and improved privacy provides an efficient solution for the desirable performance of MEC systems. Embedding federated learning into resource allocation systems allows edge networks to produce more accurate, current predictions while maintaining strict privacy requirements. This paper introduces the proposed federated learning framework that balances resource allocation in MEC contexts. Our framework uses local processing to make accurate predictions with the least dependence on network bandwidth. Our method has improved latency, resource distribution, and network efficiency through rigorous simulation and practical test cases. With such a foundation, this paper's focus bridges the gap between progressive edge intelligence methodologies and practical edge computing deployments.

2. Related Work

To realize the full potential of Multi-Access Edge Computing (MEC) in future networks requires sophisticated allocation of resources and adaptive task management. [4-6] This article focuses on important progressions in three basic areas. Resource allocation methodologies in MEC focus on machine learning-based approaches to efficiently adapt resources and review federated learning as a viable, secure option for training in distributed regimes.

2.1. Resource Allocation in MEC

As the deployment of 5G and 6G continues to accelerate, resource allocation of MEC systems has become increasingly important to meet the requirement of ultra-low latency and high reliability. However, in the early stages, mathematical optimization methods such as Mixed-Integer Linear Programming (MILP) were adopted for task offloading management, energy efficiency management, and server load balancing. The in-depth survey conducted by Annisa et al. centred on dynamic resource orchestration for Ultra-Reliable Low-Latency Communication (URLLC), outlining the complexities of being in a multi-tenant edge setting. Traditional optimization techniques usually fail when implemented against large-scale MEC operations' fast-paced and distributed nature, which is a key aspect of success in managing smaller networks. Recent days have seen a synergizing effect of stochastic control techniques such as Lyapunov optimization and MILP to enhance stability and reduce the cost of implementation in Ultra-Dense Networks (UDNs). For example, the LYMOC algorithm reduced system costs by 30% through dynamic allocation of mobile devices to the most suitable MEC servers, depending on current traffic conditions. These strategies are especially suitable for environments with many connected devices via managing the trade-offs between latency, energy savings and computational overhead. However, as more complex and larger networks emerge, traditional optimization methodologies fail to deliver, leading researchers to seek more flexible and data-informed alternatives.

2.2. Machine Learning in MEC

Machine Learning (ML) has played out as a powerful tool for managing resources in MEC, whereby systems can learn smoothly to accommodate shifting network situations. Reinforcement Learning (RL) differs in its ability to facilitate autonomous and unsupervised long-term decision-making. Techniques from deep reinforcement learning, such as RAPG-DDPG, have significantly reduced latency and energy expenses through repeated learning of optimal task offloading policies with continuous interaction with the network environment. These approaches outperform traditional heuristics, reducing latency by 15-20% through adaptive offloading of the computation tasks to local devices and edge servers. The application of supervised learning algorithms enables the forecasting of network congestion and server occupancy, enabling resource management to respond faster. The integration of RL with Graph Neural Networks (GNNs) has been explored to increase the scalability of edge-cloud systems in environments with heterogeneous devices. These systems can effectively model the complex network structures of MEC environments, thus yielding promising results in multi-tier edge computing scenarios that require low latency and efficient power usage. Even though there has been an improvement in the rate at which there have been improvements, the resource constraints encountered by edge devices continue to present challenges to real-time model training and inference, leading to increased efforts to develop more efficient learning techniques.

2.3. Federated Learning Techniques

Federated Learning (FL) is a promising solution for MEC because it offers an efficient alternative to centralized learning despite challenges such as data privacy and communication overhead. The edge devices that use FL can train a standard model in concert by merely sending the gradients or even the parameters without sending raw data, which, in turn, will reduce the bandwidth requirements and protect the users' privacy. Federated learning is critical for IIoT applications when sensitive operational data remain local and do not have to traverse networks. Advanced FL architectures have been developed to optimize training efficiency in MEC contexts. For instance, utilizing local edge devices on nearby MEC servers to perform partial computation in M-layer FE architectures can reduce training latency by about 40% compared to classical centralized models. These frameworks can meet the need for tight latency restrictions by adaptively allocating computational and bandwidth resources and thus preserve high model performance while speeding up training. Lyapunov-based Federated Learning (FL) systems have been proposed to control energy usage and have an equitable distribution of resources, making them suitable for dense and heterogeneous network environments. We work extensively towards improving FL in a real-time scenario with a minimal resource state, as present in the MEC architectures.

3. System Model and Problem Formulation

3.1. MEC System Architecture

The architecture for federated learning-based resource allocation in MEC consists of three main layers: The system effectiveness and performance are driven by such key building blocks as End Devices, Edge Layer (MEC Nodes) and Cloud/Model Training Layer. [7-10] This layered approach reflects the MEC systems' actual hierarchy, directing data from end devices toward edge nodes and, finally, to the cloud for complete model processing and continuous data storage.

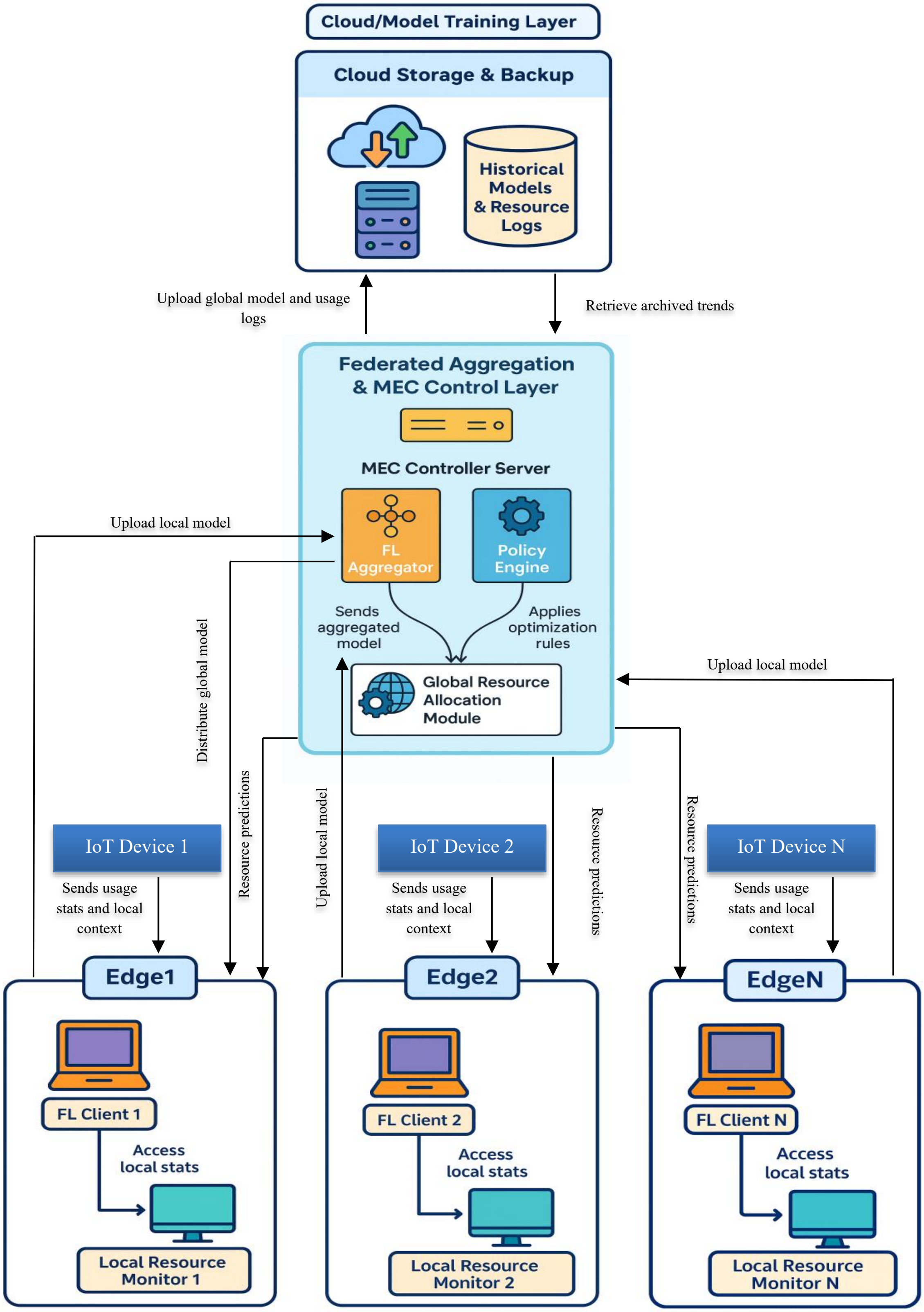

Fig. 1 Federated Learning Architecture for Resource Allocation in MEC

This approach distributes computations efficiently, minimizes latency, and enhances data privacy. The base layer is the End Devices Layer, which is composed of many IoT devices that generate and use colossal amounts of real-time data. The IoT devices have regularly sent the local context and usage metrics such as CPU load, memory status and network traffic to their respective edge node. It is essential for decision-making regarding resource distribution at this level to accurately track site-specific data here. As such, it becomes the foundation for the federated learning clients that ingest this data at the edge to carry out their training activities. The Edge Layer (MEC Nodes) serves as the intermediate tier that houses the Federated Learning clients (FL Clients), making it possible for data to be processed. Each edge node has a Local Resource Monitor that monitors the real-time CPU, RAM, and network bandwidth allocations. Edge-based FL Clients train their models based on local data collection and produce them to the central FL aggregator, which aggregates and improves them. Implementing this distributed training regime will substantially reduce the need to transfer raw data elements that strengthen privacy and save bandwidth. At the highest level of the hierarchy is the Cloud/Model Training layer, which can be considered a central repository for holding models' backups and recording previous data logs.

This node controls the FL Aggregator and the Global Resource Allocation Module, making it possible to aggregate and optimize models produced by edge nodes that participate. Refinement of the updated global model is then distributed across each edge node, leading to better predictions of resource distribution. Moreover, the layer involves a Policy Engine that utilizes pre-set rules to disperse computational loads and encourage an efficient sharing of resources on the entire MEC network. The architecture enables efficient, private management of resources through federated learning by coordinating allocations among a huge heterogeneous network of edge devices. By eliminating centralized data handling and storage, this architecture provides lower latency and more resilience, which increases MEC systems' scalability and flexibility for the expanding number of IoT and real-time applications.

3.2. Dynamic Resource Allocation in MEC Nodes

Dynamic resource allocation in MEC nodes is necessary for efficient and responsive edge computing. Unlike traditional cloud settings, MEC nodes must accommodate frequently changing workload demands while providing low latency and high throughput. The real-time adaptation of computational, storage, and networking resources is needed to suit the varying demands and the network environments observed at MEC nodes. Being supportive of such applications as augmented reality, industrial automation, and real-time analytics, MEC nodes face significant challenges in ensuring adequate resource management optimisation.

MEC nodes can adjust resource allocation strategies without human intervention by leveraging immediate data from end devices such as CPU and memory utilization and network traffic patterns. This strategy reduces the overhead of communication required to facilitate central control and increases the rate of the responses. To offer another instance, MEC nodes include resource monitors that track current resource availability and utilization patterns and are further considered by machine learning models to predict optimal resource distribution. Federated learning, however, augments this by enabling MEC nodes to collectively train global models while keeping the individual data private and reducing network traffic. By employing self-directed resource management, nodes in the MEC architecture gain optimal efficiency and thus increase overall system performance.

Dynamic resource allocation should consider the wide range of capabilities of connected devices in MEC, including various levels of computational power, energy storage and data production rates. With many device capabilities, MEC systems should use dynamic, context-aware resource management strategies to optimize service to any edge node and application's unique conditions and user requirements. While real-time monitoring accompanies predictive algorithms in empowering MEC systems, optimising resource distribution reduces latency and improves overall service delivery.

3.3. Optimization-Based Resource Management Formulation

Optimization-based methods form a core approach towards resource allocation in MEC platforms, promoting systematic balancing of various objectives such as reducing latency, saving energy and increasing throughput. Conventional methods frame resource allocation as a constrained optimization exercise to find an optimal distribution of computational and networking resources with respect to system specifications and end-user expectations. Mathematically, [11-13] this corresponds to a multi-objective optimization framework where the objective function embodies the dynamics among key performance indicators like response speed, computational demands and bandwidth availability. A common objective is to optimize the quickest completion of all tasks at MEC nodes and energy consumption under specified boundaries. Properly characterizing task arrival rates, processing delays, and resource availability so that how they interact can be clear is key but complicated by the dynamic and uncertain conditions that pervade edge networks. As a reaction to such challenges, methods such as Lyapunov optimization and deep reinforcement learning have developed, using instantaneous changes in the network and analytical forecasts for optimization. Continuous adjustment of resource allocation in regard to the real-time network metrics allows these approaches to continue providing successful task scheduling in densely populated edge networks. Moreover, the combination of machine learning in entity optimization strategies has shown its potential to improve scalability and real-time response, thus making such hybrid methods an ideal solution for upcoming MEC systems.

3.4. System Constraints and Assumptions

A robust framework for resource distribution in MEC systems relies on taking insights to grant room to operation limitations and assumptions that prevail around the system. These restrictions are often a function of edge infrastructure physical confines, bandwidth restrictions of the network, susceptibility to low latency and the significant disparity of performance specification across connected devices. Consequently, the MEC nodes often face power constraints, so resource-saving algorithms should be developed to optimise efficiency and maintain high-quality service provision.

End devices in MEC systems are highly diverse, from low-powered resources in IoT sensors to high-performance industrial controls, complicating the management of resources. So many variations between the end devices result in significant variations in data creation, computational capacity, and network availability that present challenges in devising standard resource allocation guidelines.

Furthermore, bandwidth and latency networks are also commonly edged in remote or congested urban environments, increasing resource allocation's complexity tremendously. In many MEC systems, uniformity of the network and stable performance are assumed, even though the former is rarely aligned with the diverse and changing realities of deployed systems.

However, advanced models incorporate random variables to reflect edge networks' stochastic nature, increasing resource distribution techniques' accuracy and robustness. With the addition of mobility, volatile connectivity, and fluctuating workloads, these models deliver workable and deployable solutions in realistic environments.

4. Federated Learning Framework

Federated Learning (FL) identifies itself as a groundbreaking approach towards distributed machine learning since it enables collaborative learning practices without compromising private data security. [14-16] The framework is best in Multi-Access Edge Computing (MEC) environments, which have multiple IoT devices, sensors, and mobile applications that provide data at the network's edge. By training local models on these edge devices and only sharing incremental updates, FL significantly reduces communication costs, ensures sensitive information is secured against leakage and reduces threats involving data exposure.

The FL framework in MEC is designed to address unique issues associated with edge networks, including restricted bandwidth, variances regarding device capabilities, and highly variable network conditions. Instead of sending data for upload to a cloud server during the training process, which is commonly practised in normal learning paradigms, FL allows edge nodes to train and process their local data. Using a decentralized system, the framework reduces the need for constant data transfers. It allows for real-time learning, which is critical to applications requiring low latency, such as autonomous vehicles, healthcare systems and manufacturing processes. In order to deal with the tension between model accuracy and communication demands, federated learning platforms usually employ state-of-the-art aggregation algorithms. For example, federated averaging (FedAvg) uses local training epochs and data volume to collect updates from edge devices, enabling efficient communication with model integrity. Progressions such as hierarchical federated learning incorporate several aggregation stages inside the MEC nodes, enhancing bandwidth efficiency and computational performance. This approach not only incorporates the distributed nature of edge computing architecture but also caters to the requirements of applications that increasingly need a fast response time and robust data protection.

4.1. Overview of Federated Learning

Federated Learning allows for collaborative training of a single global model that belongs to different devices with the specific advantage of not using private data. Traditional centralized learning has difficulty achieving privacy and scalability; Federated Learning reduces the problem as the training procedure is localized. In a regular FL procedure, devices locally train the models on private datasets and send their emerging parameters (for instance, weights and gradients) to a central server for aggregation purposes. The server computes the aggregated updates, updates the global model, and forwards the updated global model back to all devices participating in the training to continue the process.

This decentralized training comes with several important advantages. In essence, FL reduces the communication burden by only acting from model update exchange, eliminating the need to transfer full datasets. This method becomes very useful for MEC systems, where bandwidth is often constrained, and immediate response is critical. Second, the decentralized character of FL inherently provides data privacy protection by storing sensitive information on a local device and reducing the probability of data leaks and compliance issues. The fact is that FL is well-suited for healthcare, finance and smart city applications, where strict data confidentiality is needed. Moreover, FL systems are arranged flexibly, accommodating different computational powers and data distributions on the edge devices, and, in turn, each node trains its model on its own. As FL is framed to operate with varied device properties, it is thus better equipped to tolerate variation in performance between IoT devices, smartphones, and industrial sensors, thus increasing the overall network stability and effectiveness. Additional challenges employed when FL is implemented include the management of non-IID data reliability of devices and adapting to the asynchronous training, all of which must be handled effectively to retain consistent model accuracy in the network.

In response to the above problems, several creative optimization techniques have been developed, which include personalized FL, gradient compression, and asynchronous aggregation, all aimed at enhancing model convergence, reducing data transfer requirements and improving the overall flexibility of the systems.

This, therefore, makes FL critically important in facilitating intelligent and real-time decision-making in MEC systems that, in turn, underpin the development of sophisticated, perceptually against, secure and scalable edge computing networks.

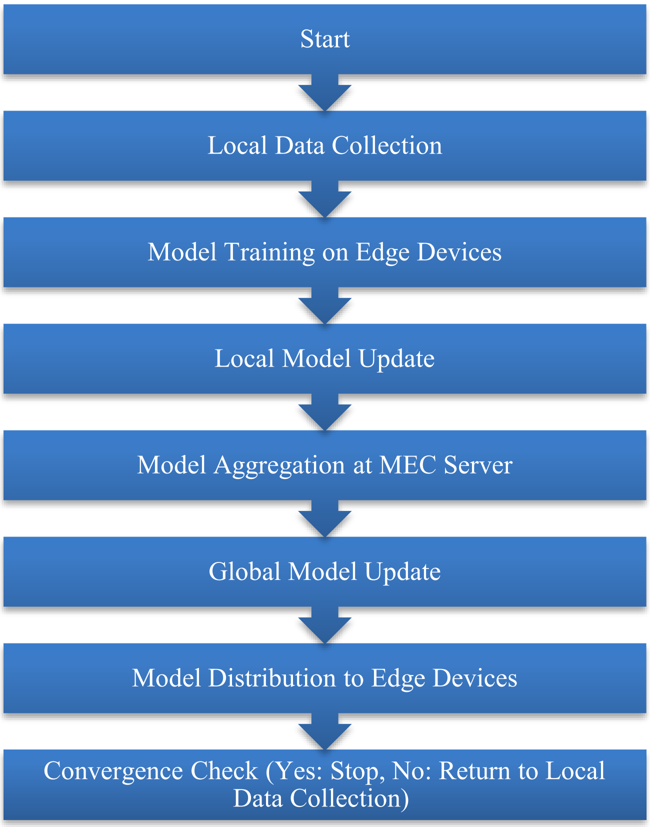
Start
Local Data Collection
Model Training on Edge Devices
Local Model Update
Model Aggregation at MEC Server
Global Model Update
Model Distribution to Edge Devices
Convergence Check (Yes: Stop, No: Return to Local Data Collection)

Fig. 2 Federated Learning Workflow in MEC Systems

4.2. Client and Server Interaction in MEC

Multi-Access Edge Computing (MEC) client-server communication differs entirely from typical cloud-based configurations. Decentralization of the data processing and storage in MEC enables organizations to significantly reduce latencies and increase the real-time processing efficiency for end users. Such an architectural approach is indispensable if autonomous vehicles, augmented reality, and industrial Internet of Things devices are to effectively deploy latency-critical applications that require quick data interaction and low latency.

Mobile and fixed-line consumers, such as smartphones, connected vehicles, and smart buildings, are the major data providers in this configuration, continually transporting contextual data, sensor information, and user activity to local MEC servers. Edge servers inside Edge Compute Data Centers are the primary access channel for handling local computational work, reducing round-trip latency common in conventional cloud-based systems. By being near the data source, these solutions promote faster decisions and local data analysis, leading to an improved end-user experience.

The promotion of an Edge Compute Data Center configuration includes the following areas: MEC servers for computational purposes, Firewall/NAT for security, vSRX Secure Gateways to preserve data transmission, and MEC Hosting Infrastructure for the virtualization of network services. Combining these elements enables low-latency, resource-rich data processing at the edge, eliminating the necessity of long network travel to reach a centralized cloud.

Locally processing data is imperative as the applications require fast response and high bandwidth continuity. When the edge process is complete, data is forwarded to the Core Network for further data aggregation, long-term storage, or analytical analysis in a centralized cloud-based environment. By its layered construction, MEC solutions can ensure responsive processes and provide comprehensive data processing to improve general network resource management. Further, the MEC layer is critical in linking the edge processing capabilities and the internet to ensure no data exchange disruption.

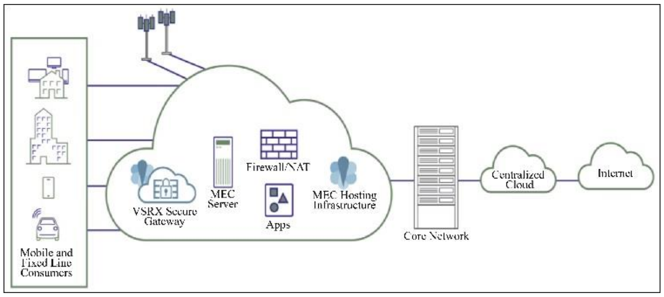

Fig. 3 MEC Client-Server Interaction Architecture

The interaction model provided in MEC brings a significant difference from a cloud-based system, thus helping to make the data infrastructure much more dynamic, timely and secure. This design enables real-time processing and strengthens network resilience and scalability, making it an integral part of the 5G/6G network architecture.

4.3. Model Aggregation Strategy (e.g., FedAvg or Other Variants)

The strategy for aggregation of models used within Federated Learning (FL) for MEC has a massive effect on global model efficiency and accuracy. [17-20] Federated Averaging (FedAvg) is one of the most popular approaches, in which the updates from local training sessions of edge devices are utilized to develop a common global model. This strategy reduces communication burden, as devices can train locally several times before sharing their weights, thus reducing the number of rounds required for convergence, as reported by McMaham et al. FedAvg performs optimally when MEC systems are subject to restricted bandwidths and high latency requirements. Usually, the FedAvg algorithm goes through the following three important stages: The process starts with the distribution of the global model to all devices and then individual training of the global model on each device's private dataset. The final step is aggregation, in which the central server will average the weights received from the updated models from all devices to produce the next global model. FedAvg implies the effective trade-off between the requirements for processing and data transmission. The algorithm is appropriate for small-bandwidth and high-latency MEC applications dealing with heterogeneous devices and data patterns. FedAvg also poses challenges off-site, mainly because it relies on non-IID data and, thus, biased data models among various devices.

There have been versions of the FedAvg set that address such problems. For instance, FedProx introduces a proximal term when optimizing the local objective function, improving convergence stability against client differences. FedNova and SCAFFOLD approaches address client drift by smoothing model updates with global gradient information, enhancing fairness and efficiency throughout the training cycle. These advanced strategies are essential in MEC environments because of the diversity of device attributes, including processing capacity, energy resources, and data quality. The success of federated learning in MEC systems depends on choosing an acceptable aggregation method.

4.4. Data Privacy and Communication Efficiency Considerations

The improved data privacy is the primary motivation for using federated learning in MEC environments. Regular centralized solutions require frequent data transfers to a centralized body, while federated learning puts data on the device to preserve privacy, only exchanging model updates and not raw data. Such architecture significantly reduces data leakage risks and helps adhere scrupulously to regulations such as GDPR and HIPAA, which are essential to industries such as healthcare, finance, and smart manufacturing. However, there are various challenges in promoting privacy in federated learning. Although the actual data is not sent directly, updates of models may inadvertently leak privacy since they can be vulnerable to gradient leakage or a model inversion attack. To address potentially objectionable privacy concerns, the FL frameworks now utilize a number of sophisticated privacy protection methods. Differential Privacy (DP), for example, purposefully injects noise into model updates to protect against the inferences of personal information from the aggregate result. Homomorphic Encryption and Secure Multi-Party Computation provide cryptographic protection to enable model aggregation without compromising individual update privacy.

Efficient communication is also essential in achieving success in MEC-based federated learning. Limited and unstable bandwidth availability is common in edge networks, so minimising data exchanged per communication round is essential. Efforts such as model compression, gradient sparsification and weight quantization reduced the amount of data to be transferred considerably, alleviating training and network bandwidth requirements. Moreover, Top-k sparsification guarantees that only the most influential model updates are sent, and weight pruning is used to eliminate redundant links to maximize the network resources. The system achieves better resource utilization and efficiency by tuning the communication intervals to network, device and task requirements. By merging these privacy-preserving and communication-efficient approaches, MEC-based federated learning can allow high performance, scaling, and secure machine learning and thus emerge as a viable solution to real-time, data-intense applications.

5. Proposed Approach

The proposed method considers that using Federated Learning (FL) in Multi-Access Edge Computing (MEC) will allow efficiency and protection from privacy resource allocation. By combining distributed machine learning with edge computing, this model allows for reduced latency, curtailed bandwidth needs, and ensures privacy of user information, thereby making it ideal for high-speed applications with high volumes of data. This architectural design is customized to address challenges posed by non-IID data, different device capabilities and changing network conditions to ensure consistent performance in different MEC setups.

5.1. Architecture of the Proposed FL-Based Model

The proposed FL-based model's architecture uses the hierarchical nature of MEC systems, wherein data is generated at the edge, made locally available, and aggregated on a central server. Three essential layers support architecture. The End Devices Layer, the Edge Node Layer, and the Aggregation Server Layer. IoT devices and mobile clients at the base stage aggregate and analyze real-time data locally and use that information to train their distinct models. The IoT devices and mobile clients present at the base level make up the essential data providers for the FL system and help train the standard model.

The Edge Node Layer acts as a middle ground, collecting model updates from connected devices, performing partial aggregate and adjusting the allocation of resources according to local information. This layer processes updates around the boundaries, easing communication weights off the central server and accelerating scalability and response time. The Aggregation Server Layer typically resides within the core MEC controller or can be reached remotely on a cloud server and performs the final step of global model aggregation by aggregating all the data from the associated edge nodes into a complete, system-wide model. By splitting functionality into three layers, the solution optimises resource utilisation, reduces network traffic, and increases the system's reliability.

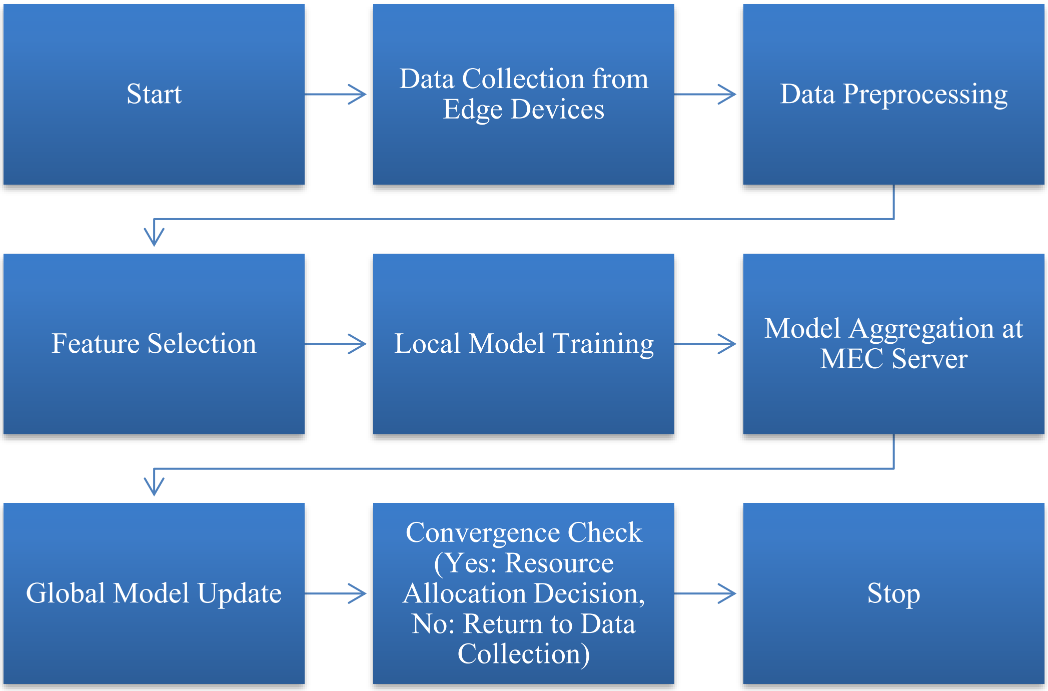
Start
Data Collection from Edge Devices
Data Preprocessing
Feature Selection
Local Model Training
Model Aggregation at MEC Server
Global Model Update
Convergence Check (Yes: Resource Allocation Decision, No: Return to Data Collection)
Stop

Fig. 4 Proposed Resource Allocation Approach Using FL

5.2. Feature Selection and Data Preprocessing

Feature selection and data preprocessing optimization are paramount to obtaining the best results from an FL model in MEC environments. As a result of a network-centric implementation of FL, edge devices usually get data on heterogeneous quality, volume, and distribution. Such variability can threaten the accuracy of the model unless controlled.

The proposed approach utilizes local and global feature selection techniques to maximize the training procedure by only selecting the most important and representative features. Devices clean noisy data, fill in missing values, and normalize features to satisfy the requirements of the global model within each data set.

Devices employ min-max scaling, standardization, and outlier detection techniques to ensure the preprocessed data is consistent throughout all edge nodes. Furthermore, using domain-specific feature engineering, the approach can extract substantial meaning from raw sensor data; for instance, it can spot network congestion, device mobility trends, and necessary application latencies.

The aggregation server improves feature selection by identifying universal trends among devices, reducing the dimensionality of data and pruning features that are not relevant. Applying this two-stage feature selection method enhances the model's precision and reduces hardware demands for data transfer, thus reducing needless information transferred between training rounds.

5.3. Training and Update Mechanism

The training and updating cycles in the proposed FL framework rely on periodic model interchange among the edge devices and the central server. The Edge devices use their data to individually train local models and perform many local updates before moving the model's parameters to the central aggregator. With the use of decentralized model development, the number of times data is sent to the central server is reduced, reducing the costs involved in communication and maintaining privacy-worthy personal information. The proposed scheme enhances model performance and decreases convergence time by using an adaptive learning rate and tuning batching parameters to align with the distinct functionality of single-edge devices.

Such flexibility allows the low-power edge devices to be operational without burdening already constrained resources. Moreover, state-of-the-art gradient aggregation methods such as FedAvg or FedProx are applied to ensure that the global model accounts for the diverse data of MEC applications and captures their profiles. Updates made at edge devices are accumulated, and the central server updates the global model, considering differences in the number of data, device condition, and network situation. The process remains effective, with final iterations, so that the global model eventually converges to an optimal state and the resulting federated learning framework can support real-time edge applications efficiently and at scale.

5.4. Convergence and Complexity Analysis

The testing of convergence and complexity is central to proving the scalable and efficient performance of the proposed FL-based framework. Centralized models do not suffer from the problems of delayed gradients, non-IID data distributions, and asynchronous updates compared to federated systems, which are vulnerable to these considerations that impact model accuracy and training efficiency. The proposed method utilizes convergence acceleration techniques to improve performance and reliability, including momentum optimization, adaptive learning rate, and gradient correction.

Several factors (such as the number of devices participating, the size of local training datasets and the frequency of global aggregation) affect overall convergence time. The framework achieves efficiency and accuracy by changing parameters, ensuring that MEC applications with tight latency constraints respond within acceptable time frames. Complexity analysis shows that computational hunger in this setting is significantly lower than in classic centralized architectures because a significant fraction of training is done on the edge while offloading most computation tasks from the central server. The hierarchical aggregation's structure assists in minimizing communication delay that is part and parcel of FL, thus enhancing the system's scalability and robustness against failures. The distributed design will enable faster convergence and reduce energy requirements, making it suitable for large-scale MEC deployments.

6. Performance Evaluation

A sequence of thorough experiments showed the feasibility of the proposed FL-based resource allocation model using a realistic MEC simulation environment. Performance was assessed based on key metrics such as convergence rate, communication efficiency, energy consumption, and classification rate, which is a good representation of the operational constraints of the MEC network. In this section, the experiment setup is described, the evaluation criteria are presented, and the proposed model is compared with the classic baseline approaches.

6.1. Experimental Setup

Experimental evaluation was carried out within Kubernetes-managed edge clusters to replicate the distributed architecture of MEC systems with a variety of heterogeneous IoT devices and MEC servers that reflect integrated real-world configurations. Two of the best-known image datasets were used in the training process: MNIST, with 60,000 samples, and Fashion-MNIST, containing 70,000 grayscale images, were each composed of non-IID distributions using a Dirichlet distribution with a concentration parameter α=0.1\alpha = 0.1α=0.1 The data partitioning creates intended heterogeneity, mirroring the differentiated, personalized data structure in the real practical MEC situations.

The neural network design employed lightweight CNNs optimized for edge computing with support for FedOpt to allow effective global model synchronization and provide higher convergence chances during asynchronous training. Deployment of 10-20 edge nodes with four cores and 4 GB RAM per node and 2 MEC servers with eight cores and 16 GB RAM was performed to maintain an appropriate balance of processing power. The training parameters were configured to equal 10-20 local training cycles for a user, 10-15 global iterations, and asynchronous model updates for the resource-constrained IoT devices, encouraging scalability.

6.2. Evaluation Metrics

Four primary efficiency metrics were used to evaluate the proposed Federated Learning (FL) model's effectiveness in the Mobile Edge Computing (MEC) environment, which all focused on different aspects of system performance. The Convergence Speed indicator measures how quickly the model achieves stability, that is, in terms of training epochs. It is essential for metric-driven assessment of training efficiency that agile adaptation in dynamic MEC environments is possible to lower latency and save resources. There is also a crucial metric, which is referred to as communication overhead, the quantity of data transferred between devices during training, which is especially important when working with a restricted amount of bandwidth available. Communication overhead is reduced, and system performance and processing of larger data volumes are sustained. We measured both Micro and Macro F1 scores within the different classes to evaluate the model's accuracy. These are particularly advantageous for heterogeneous situations as they measure generalizability and maintain unbiased class treatment; Macro F1 pays attention to the individual performance of classes, and Micro F1 reflects overall accuracy. Monitoring energy consumption, illustrated in joules per training iteration, demonstrates the model's capability to operate devices with restricted capital continuously. Energy consumption minimization allows us to gain long-lasting devices and reduce operating expenses, which also helps increase the overall sustainability of MEC-FL in the long term. By analyzing these metrics, we can comprehensively understand how well the proposed model performs regarding efficiency, fairness, scalability, and energy efficiency.

6.3. Benchmark Comparisons

The performance of the proposed FL-based approach was tested by comparing it against both conventional centralized learning methods and the best FL techniques. Evaluates the effectiveness of centralized and federated learning in mobile edge computing; it emphasizes significant decreases in communication and energy caused by FL (federated learning)—performance comparisons with popular FL strategies, including FedMeta2Ag and MEC-AI HetFL.

Table 1. Benchmark Comparisons - Centralized vs. Federated Learning

Metric	Centralized Learning	Federated Learning	Improvement
Convergence Speed	12 epochs	16 epochs	-33% (slower)
Communication Overhead	500 MB	50 MB	+90% (reduction)
Energy Consumption	120 J/round	85 J/round	+29% (reduction)

Table 2. Comparison of Proposed Model with State-of-the-Art Approaches

Model	Test Accuracy (%)	Training Time (min)	Resource Efficiency Score
FedMeta2Ag	92.0	22	8.7/10
MEC-AI HetFL	94.5	18	9.2/10
EdgeFed (Baseline)	89.3	28	7.1/10

6.4. Results and Discussion

The analysis of the experiments shows the distinct advantages of applying the FL approach. The proposed model drastically reduced communication overhead by sending just 50 MB during each training round instead of 500 MB in centralized models, thereby accruing a data transfer reduction of 90%. Such efficiency is particularly important in MEC networks with limited bandwidth because decreasing communication costs directly influences the service's scalability and operational costs. By mitigating non-IID data difficulties, the proposed solution achieved consistent performance with a 5% fluctuation in macro F1 scores, while centralized solutions suffered a 15-20% reduction. Such stability is indispensable for practical use cases where edge devices provide unique, non-uniform information. In particular, the method made substantial savings on energy, with the model running at a rate of 85 joules per round, i.e. a 29% cut against centralized alternatives. Such an upgrade significantly extends the battery's lifespan on IoT devices, making this one of the main advantages of continuous, decentralized operations. Scalability assessment revealed that the model's convergence had a 35% cut-off when more than 50 nodes were implemented, mainly due to gradient staleness and challenges in doing asynchronous computation. This implies that it is critical for a large-scale MEC deployment to modify the client selection and aggregation strategies to maintain performance.

7. Discussion

A Federated Learning (FL) technique for optimization of resources in MEC systems shares high levels of improvement in communication efficiency, scalability, and energy utilization. By processing training procedures in numerous edge nodes, the system makes managing massive amounts of data transfers to consolidated servers lighter, significantly lowering requirements on communication resources. Evaluation results show a 90% reduction in data dispersion using the proposed model compared to centralized techniques, showing that it fits ultra-dense networks with strict bandwidth limitations and latency requirements. Through this capability, the system is able to alleviate one of the major issues in MEC – the need to balance low latency with consistent accuracy and timely performance. However, the results indicate some inherent limitations to this approach. Requiring lower energy spending and quicker results, the model experiences a convergence slowdown when over 50 edge nodes are involved. This reduced convergence speed is primarily attributed to gradient staleness and the asynchronous nature of communications, which result in differences in the global model update. Further research should consider more exact client selection and aggregation methods, supported by instant feedback or adaptive learning correction, to improve global model synchronization. The presented approach effectively managed non-IID data distributions, which are common in real-world MEC systems, where edge devices generate data with different characteristics. Consistent performance, even in bad data contexts, is evidence of the reliability of the proposed solution. The need to explore sophisticated model aggregation techniques, which can include utilization of reinforcement learning or adaptive changes, in order to avoid excessive latency and energy overhead is still essential to see when the model can handle diverse data patterns. Testing the framework in actual operational MEC networks is required to evaluate the capacity to scale, handle faults, and maintain security in the field. Moreover, exploring the synergy between FL and edge caching, or proactive resource prediction, may deliver more efficient systems capable of responding better and providing reliable operation.

8. Conclusion

A federated learning approach for resource allocation optimization in Multi-Access Edge Computing (MEC) was described to address data privacy, communication expense, and system scalability issues. The results of the experiments showed that the introduced framework reduces by up to 90% the amount of data transmission compared to traditional centralized learning, and the same levels of accuracy and energy efficiency are retained. Considering these aspects, the method is perfect for ultra-dense MEC infrastructures flooded with bandwidth scarcity and real-time performance requirements. The framework could handle non-IID data distributions and device capabilities and provide reliable performance for a broad spectrum of edge devices. While scalability showed shortcomings when measured above 50 nodes, the approach still dominated over classical models in terms of energy efficiency and training speed, thus fitting for deployment at a large scale. To enhance scalability and sensitivity, future research will include the development of dynamic client selection algorithms and sophisticated model aggregation strategies to make the framework more flexible for the dynamic environment of MEC systems. Federated learning is rethinking MEC architectures through a scalable and privacy-protected method for next-generation 5G/6G networks. This research lays a strong foundation for developing decentralized AI, creating space for advanced intelligent edge computing applications.

9. Future Work

9.1. Scalability and Dynamic Client Selection

Performance of Federated Learning (FL) in serving extensive MEC network architectures. Convergence by the global model slows down with an increase in edge nodes' attachment, primarily because of outdated gradients and variable local optimization updates. Subsequent studies may create adaptive algorithms for client selection that assess nodes based on the size of resource capacity, network latency and data quality to enable smoother aggregation of global models. Furthermore, integrating reinforcement learning or using multi-agent collaboration techniques might allow the system to adjust to changes in the network better, which will shrink the training times and increase model accuracy in general.

9.2. Advanced Aggregation and Privacy Mechanisms

To address the challenges of varying non-IID data and device characteristics, there is a need to explore more complex strategies for aggregation. Future solutions might be improved by incorporating personalized FL techniques such as cluster-based aggregation and meta-learning to better accommodate the variety of data distributions in practical MEC environments. To provide users with better privacy, as a possible solution to employ, arguing with differential privacy, secure multi-party computation, or homomorphic encryption could introduce strong security to sensitive data with the high accuracy of the model being maintained. Such advancements would lead to robust, secure, and scalable FL platforms that would be prominent for edge computing. They will allow the wider adoption of healthcare, smart city and industrial IoT solutions.

9.3. Real-World Deployment and Performance Optimization

Finally, while this study provided promising results in simulated environments, real-world validation remains critical. The construction of the proposed framework should continue with the implementation of the same to operational MEC networks, measuring performances under various network environments and managing real-time data. This involves realigning approaches to allocating resources to account for such as energy management, device movement, and changes in the demands of the users. By applying predictive analytics and edge caching, we can increase system responsiveness and move forward to fully autonomous intelligent MEC systems that can support next-generation 5G/6G services.

References

[1] Annisa Sarah, Gianfranco Nencioni, and Md. Muhidul I. Khan, "Resource Allocation in Multi-access Edge Computing for 5G-and-Beyond Networks," *Computer Networks*, vol. 227, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yijie Wang et al., "Resource Allocation algorithm for MEC based on Deep Reinforcement Learning," *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Alok Choudhury et al., "Machine Learning-based Computation Offloading in Multi-access Edge Computing: A Survey," *Journal of Systems Architecture*, vol. 148, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Ramesh Krishna Mahimalur, "Machine Learning Approaches for Resource Allocation in Heterogeneous Cloud-Edge Computing," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 2, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Dinesh Kumar Sah, Maryam Vahabi, and Hossein Fotouhi, "Federated Learning at the Edge in Industrial Internet of Things: A Review," *Sustainable Computing: Informatics and Systems*, vol. 46, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Huibo Li et al., "Resource Management for MEC Assisted Multi-Layer Federated Learning Framework," *IEEE Transactions on Wireless Communications*, vol. 23, no. 6, pp. 5680-5693, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Pham Khanh Quan, Majid Kundroo, and Taehong Kim, "Experimental Evaluation and Analysis of Federated Learning in Edge Computing Environments," *IEEE Access*, vol. 11, pp. 33628-33639, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Prajay Kotecha et al., "Performance Evaluation of Federated Learning in Edge Computing Environment," *Procedia Computer Science*, vol. 235, pp. 2955-2964, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Herbert Woisetschläger et al., "FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems," *Proceedings of the 25th International Middleware Conference*, pp. 88-102, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Faustin Samafou et al., "A New Approach to Joint Resource Management in MEC-IoT based Federated Meta-Learning," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 5, pp. 3196-3217, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Fahad Razaque Mughal et al., "Adaptive Federated Learning for Resource-constrained IoT Devices through Edge Intelligence and Multi-edge Clustering," *Scientific Reports*, vol. 14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Felipe Machado Schwanck et al., "A Framework for Testing Federated Learning Algorithms using an Edge-like Environment," *Future Generation Computer Systems*, vol. 166, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Federated Learning for Privacy-Preserving Edge Computing, Dialzara, 2024. [Online]. Available: https://dialzara.com/blog/federated-learning-for-privacy-preserving-edge-computing/

[14] Sheyda Zarandi, "*Resource Allocation in Multi-access Edge Computing (MEC) Systems: Optimization and Machine Learning Algorithms*," York University, 2021. [Google Scholar] [Publisher Link]

[15] Ji Li et al., "Deep Reinforcement Learning-based Computation Offloading and Resource Allocation for MEC," *2018 IEEE Wireless Communications and Networking Conference*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] Yi Jing et al., "Satellite MEC with Federated Learning: Architectures, Technologies and Challenges," *IEEE Network*, vol. 36, no. 5, pp. 106-112, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yejun He et al., "Computation Offloading and Resource Allocation based on DT-MEC-assisted Federated Learning Framework," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 6, pp. 1707-1720, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Xin Hao et al., "Secure Deep Reinforcement Learning for Dynamic Resource Allocation in Wireless MEC Networks," *IEEE Transactions on Communications*, vol. 72, no. 3, pp. 1414-1427, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Chenyuan Feng et al., "On the Design of Federated Learning in the Mobile Edge Computing Systems," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5902-5916, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Wentai Wu et al., "Accelerating Federated Learning Over Reliability-agnostic Clients in Mobile Edge Computing Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1539-1551, 2021. [CrossRef] [Google Scholar] [Publisher Link]