الجـمهورية الجـزائرية الديمقراطـية الشـعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العـالي و البحـث العـلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جـامعة عمار ثـليجي بالأغـواط
UNIVERSITY OF AMMAR TELIDJI LAGHOUAT

كـلية العلـوم
FACULTY OF SCIENCES
قـسـم الإعـلام الآلـي
DEPARTMENT OF COMPUTER SCIENCE

MASTER THESIS

Field : Mathematics and Computer Science

Option : Computer Science

Specialization : Networks, Distributed Systems, and Applications

Submitted by : Oudenani Mohammed Khaled Ali and Ahmed Chaouch Ilies Akram

Theme

Client Selection for Federated Edge Learning in UAV Networks

Supervisor :

Dr. Youssra Cheriguene University of Laghouat

بسم الله الرحمن الرحيم

Dedications

we would like to dedicate this work to our dear parents who taught and raised us. Thanks to them we are here today, we wish for nothing but to make them proud. Dedications also go to our dear families, brothers and sisters, who have always been there for us To...

Oudenani Mohammed Khaled Ali
Ahmed Chaouch Ilies Akram

Acknowledgments

First and foremost, no volume of words is enough to express praise to Allah Almighty. All praise is due to Allah, the Lord of the Worlds.

We wish to express our deepest gratitude to our supervisor, Dr. Cheriguene Youssra who has convincingly encouraged and guided us through this work, we wouldn't have been able to achieve it without her persistent help.

My sincere thanks go to the committee members for accepting to review our work.

Oudenani Mohammed Khaled Ali and Ahmed Chaouch Ilies Akram, June 2025

Abstract

One promising way to enable distributed intelligence at the edge while protecting data privacy is to integrate Federated Learning (FL) with Unmanned Aerial Vehicles (UAV) networks. Using FL enables each UAV to cooperatively train a global model without sharing raw data, especially in UAV swarms used for surveillance, monitoring, or emergency response missions. But choosing the best clients (UAVs) for every training cycle is made extremely difficult by the dynamic and diverse character of UAV environments. These difficulties are brought on by things like fluctuating connectivity, shifting patterns of movement, and energy limitations.

In this work, we investigate the problem of client selection for Federated Edge Learning in UAV networks. We first present a taxonomy of existing selection strategies, considering criteria such as model performance and UAV mobility. Then, we propose an adaptive client selection framework that integrates both mobility-awareness and distance with speed to enhance learning efficiency and model accuracy. Extensive simulations demonstrate that our method significantly improves convergence speed and reduces communication overhead, while maintaining high model performance in dynamic UAV scenarios.

Keywords: FL, UAV Networks, Client Selection, Mobility-Aware Selection, Edge Intelligence.

ملخص

يمثل دمج التعلم الفيدرالي مع شبكات الطائرات بدون طيار (درون) حلاً واعدًا لتمكين الذكاء الموزع على الحافة مع الحفاظ على خصوصية البيانات. بشكل خاص، في أسراب الطائرات المستخدمة في المراقبة أو الاستجابة للطوارئ، يسمح التعلم الفيدرالي لكل طائرة بالمشاركة في تدريب نموذج عالمي دون مشاركة البيانات الخام. لكن الطبيعة الديناميكية وغير المتجانسة لهذه الشبكات تفرض تحديات كبيرة، خصوصًا في اختيار الطائرات المناسبة لكل جولة تدريب، نظرًا لتقلب الاتصال والحركة ومحدودية الطاقة.

في هذا العمل، ندرس مشكلة اختيار العملاء في التعلم الفيدرالي الطرفي ضمن شبكات الطائرات بدون طيار. نبدأ بتصنيف الاستراتيجيات الحالية بناءً على أداء النموذج وتنوع البيانات وحركة الطائرات، ثم نقترح إطارًا تكيفيًا لاختيار العملاء يأخذ في الاعتبار كلاً من الحركة مع المسافة و السرعة لتحسين الكفاءة والدقة. توضح المحاكاة المكثفة فعالية طريقتنا في تسريع التقارب وتقليل الحمل الاتصالي مع الحفاظ على أداء عالٍ للنموذج في البيئات الديناميكية.

الكلمات المفتاحية: التعلم الفيدرالي، شبكات الطائرات بدون طيار، اختيار العملاء، الوعي بالحركة، الذكاء الطرفي.

Contents

List of Figures

List of Algorithms

List of Acronyms

3D Three-dimensional. 2, 24

5G 5th Generation of mobile communications. 2

AI Artificial Intelligence. 2

BAFL-SVM Blockchain-Assisted FL-Driven SVM. 16

CFL Clustered Federated Learning. 19

FEEL Federated Edge Learning. 16

FL Federated Learning. iii, 2, 15

FTL Federated Transfer Learning. 11

GHz Gigahertz. 29

GNN Graph Neural Network. 16

HALE High Altitude Long Endurance. 6

HFL Hierarchical Federated Learning. 11, 19

IoT Internet of Things. 18

LoRa Long Range. 18

LoS Line-of-sight. 2

LPWAN Low-Power Wide-Area Networks. 18

MALE Medium Altitude Long Endurance. 6

MB Megabyte. 29

MEC Multi-access Edge Computing. 2

MHz Megahertz. 29

ML Machine Learning. 2

NB-IoT Narrowband Internet of Things. 18

QoS Quality of Service. 10

SGD Stochastic Gradient Descent. 29

SVM Support Vector Machine. 16

UAV Unmanned Aerial Vehicles. iii, 2

V2V Vehicle-to-Vehicle. 19

VFL Vertical Federated Learning. 11

Chapter 1

General Introduction

1.1 Context and motivation

Multi-access Edge Computing (MEC) enabled by Unmanned Aerial Vehicles (UAV) has been viewed as a promising solution for addressing the stringent demands of computation-intensive and latency-critical applications in the context of the 5th Generation of mobile communications (5G) and beyond, especially in remote and disaster-stricken areas [3]. UAVs possess several inherent advantages, including effortless Three-dimensional (3D) mobility, Line-of-sight (LoS) transmissions, rapid deployment, and enhanced ving capabilities in terms of longer flight durations and more on-board computational payloads. These are features making UAV-based MEC a successful means of offering on-demand computing capacities to thinly spread mobile ground User Equipments (UEs) in different circumstances, ranging from military operations to civilian use applications such as emergency response and disaster relief activities [4]. Apart from these, UAVs have emerged as ideal data collection agents for applications such such as remote sensing and large-scale environmental monitoring. Considering the sensitive description of the collected data, an increasing demand for Machine Learning (ML) models trained in a privacy-preserving way exists. Federated Learning (FL) is an appealing distributed Artificial Intelligence (AI) scheme that allows collaborative model training among diverse UAVs without needing to send raw data to a central point, e.g., a MEC server. However, current UAV-based FL frameworks usually apply naive client selection methods random sampling being most frequent and may lead to inferior learning performance. The superior mobility of UAV networks, frequent topology changes, constrained energy resources, and low flight durations add additional challenges. All these factors may lead to temporal-varying connectivity, increased packet loss, and ultimately devalue the overall training process. In light of these challenges, the objective of this thesis is to propose smart UAV participant choice mechanisms for edge-enabled FL systems. The mechanisms need to consider UAV-specific requirements, for instance, mobility and battery requirements, to maximize learning performance, communication efficiency, and operational life of UAV networks.

1.2 Contributions

This thesis will address the challenges of UAV-assisted FL, with a specific emphasis on the effects of UAV mobility. The main contributions will include:

- We analyze the impact of UAV mobility on FL performance, identifying key challenges such as unstable connections, increased dropout rates, and fluctuating client participation.

- We propose a mobility-aware client selection strategy that accounts for dynamic UAV behavior such as speed, displacement, and positional changes to reduce dropout incidents.

- We demonstrate that the proposed method effectively reduces the mean and variability of dropout ratios, ensuring more consistent UAV participation across training rounds.

- We show that by stabilizing client selection and reducing communication disruptions, the approach enhances the convergence rate and improves the global model's accuracy.

1.3 Organization of the Thesis

This thesis is organized as follows:

- **Chapter 2:** Introduces UAVs and discusses their classifications. It also provides an overview of Federated Learning, highlighting its functioning, applications, and use cases within UAV edge networks.

- **Chapter 3:** Reviews the state-of-the-art by illustrating various client selection methods used in literature. This chapter includes descriptions and comparisons of all discussed methods.

- **Chapter 4:** Presents our proposed client selection solution in UAV edge networks, with a focus on maximizing the global test accuracy under high mobility settings. The client selection algorithm operates by applying an heuristic lightweight method that takes into account distances between UAVs and the edge server in order to select optimal clients.

- **Chapter 5:** Concludes the thesis with a summary of findings and suggestions for future work.

Chapter 2

UAV-Assisted MEC Background

Introduction

UAVs, or drones, evolves rapidly have from technologies military-origins into tools versatile applied numerous across sectors civilian, commercial, and industrial. With sensors advanced integration increasing, algorithms AI, and systems communication, UAVs are now critical becoming tasks for like monitoring environmental, agriculture precision, inspection infrastructure, and response disaster [5].

Research recent highlights a deployment surge UAV in environments urban within, new challenging raising around management airspace, efficiency energy, and compliance regulatory [6]. Concurrently, efforts powering to UAVs with energy clean and renewable sources momentum gain ensuring operations sustainable aerial [7]. As demand the for autonomous and aware-energy UAVs grows continue, field the is witnessing a shift smartward toward, safer, greener, and aerial systems.

This chapter delves into the foundational aspects of UAV-assisted MEC, providing a comprehensive overview of their components, classifications and tasks.

2.1 Unmanned Aerial Vehicles Classification:

UAVs can be classified based on various parameters, such as size, Weight, altitude, and application .to summarize, Figure 2.1 represents classification of UAVs based on different categories. The key classifications include:

2.1.1 Classification by Size [1] :

- **Micro/Very Small UAVs:** Length or wingspan less than 50 cm.
- **Mini/Small UAVs:** Length or wingspan between 50 cm and 2 meters.
- **Medium UAVs:** Length or wingspan between 5 and 10 meters.
- **Large UAVs:** Length or wingspan greater than 10 meters.

2.1.2 Classification by Weight [1]:

- **Micro UAVs (MAV):** Weighing between 250 grams and 2 kilograms.
- **Miniature or Small UAVs (SUAV):** Weighing between 2 kilograms and 25 kilograms.
- **Medium UAVs:** Weighing between 25 kilograms and 150 kilograms.
- **Large UAVs:** Weighing more than 150 kilograms.

2.1.3 Classification by Range and Endurance [1] :

- **Close Range UAVs:** Range between 5 km and 50 km; endurance between 1 to 6 hours.
- **Short Range UAVs:** Range between 50 km and 150 km; endurance between 8 to 12 hours.
- **Medium Range UAVs:** Range between 150 km and 650 km; endurance between 12 to 36 hours.
- **Long Range UAVs:** Range greater than 650 km; endurance over 36 hours.

2.1.4 Classification by Altitude [1] :

- **Close Range UAVs:** Operating altitude up to 1.5 km.
- **NATO Type UAVs:** Operating altitude up to 3 km.
- **Tactical UAVs:** Operating altitude up to 5.5 km.
- **Medium Altitude Long Endurance (MALE) UAVs:** Operating altitude up to 10 km.
- **High Altitude Long Endurance (HALE) UAVs:** Operating altitude above 10 km.

2.1.5 Classification by Application [2] :

- **Military UAVs**: These drones are used for combat, reconnaissance, surveillance, and target acquisition. Examples include the Predator and Reaper drones.
- **Civilian UAVs**: These drones are used for a variety of non-military applications, including agriculture, disaster management, and delivery services.
- **Commercial UAVs**: These drones are used for commercial purposes such as aerial photography, real estate, and infrastructure inspection.
- **Recreational UAVs**: These are small drones used by hobbyists for personal enjoyment and photography.

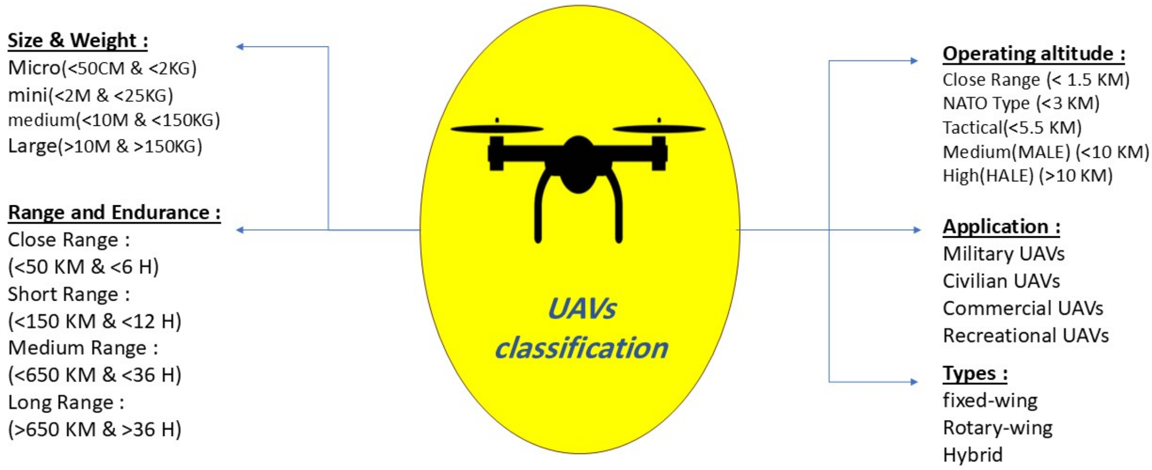

Figure 2.1: Classification of UAVs.

2.2 UAV Task Types

UAVs are used for a wide range of tasks across various industries. Below are some common UAV task types:

- **Surveillance and Monitoring:** UAVs are used for real-time monitoring and surveillance in areas such as border security, wildlife tracking, and infrastructure inspection [8].

- **Precision Agriculture:**

  UAVs are used for crop monitoring, soil analysis, and precision spraying in agriculture [9] (Fig. 2.2).

Figure 2.2: Drone in agriculture.

- **Disaster Response and Management:** UAVs are used for damage assessment, search and rescue, and delivering supplies during natural disasters [10].
- **Environmental Monitoring:** UAVs are used to monitor environmental conditions such as air quality, water quality, and deforestation [11].
- **Delivery and Logistics:** UAVs are used for delivering goods, especially in remote or congested areas [12] (Fig. 2.3).

Figure 2.3: Delivery drone.

- **Scientific Research:** UAVs are used in scientific research to collect data in fields such as meteorology, geology, and biology [13].
- **Infrastructure Inspection:** UAVs are used to inspect infrastructure such as bridges, pipelines, and power lines [14].
- **Telecommunications:** UAVs are used to provide temporary or emergency telecommunications services, such as internet connectivity in remote areas or during disaster recovery operations [15].

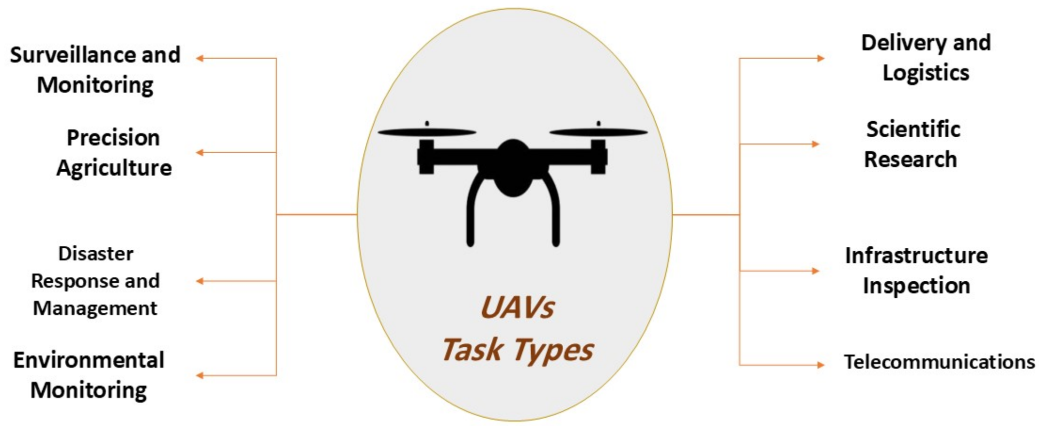

Figure 2.4: UAVs task types.

2.3 Unmanned Aerial Vehicles in MEC :

The conjunction of MEC with UAVs has surfaced a paradigm promisingly that enhances capabilities computational and diminishes latency for diverse applications. MEC servers equipped, UAVs with can present resources computing on-demand in absent infrastructure areas, like during recovery disaster or at locations remote. This conjunction facilitates processing data efficient to closer the source of the data, responses improving times and decreasing need the for transmission data extensive towards data centers centralized [16].

2.3.1 Roles of UAVs in MEC :

2.3.1.1 Aerial MEC Servers

UAVs may perform like MEC platforms mobilized, delivering resources computational to users on the ground and devices IoT. This arrangement conveys benefits particularly in situations scenarios where MEC infrastructure terrestrial is missed or compromised [16].

2.3.1.2 Computation Offloading

Devices resource-restrained can intensive tasks offload computational to MEC servers-UAV based, saving energy and performance boosting. This becomes critical for applications demanding processing real-time of data, like surveillance and monitoring environmental [16].

2.3.2 Advantages of UAV-Enabled MEC :

2.3.2.1 Flexibility and Rapid Deployment

UAVs can be quickly deployed to establish MEC services in dynamic environments, providing immediate computational support during emergencies or large-scale events.

2.3.2.2 Enhanced Coverage

By operating at varying altitudes, UAVs can extend network coverage to underserved or inaccessible areas, ensuring broader access to computational resources.

2.3.2.3 Improved Quality of Service (QoS)

Proximity to end-users allows UAV-based MEC servers to reduce latency and increase data processing speeds, enhancing the overall QoS for applications like video streaming and online gaming.

2.3.3 Challenges and Considerations :

2.3.3.1 Energy Constraints

UAVs have limited battery life, which can restrict their operational time and the duration of MEC services. Efficient energy management strategies are essential to prolong missions.

2.3.3.2 Resource Allocation

Dynamic allocation of computational resources and optimal positioning of UAVs are complex tasks that require advanced algorithms to balance load and maintain service quality.

2.3.3.3 Security and Privacy

Ensuring secure data transmission and processing in UAV-enabled MEC systems is critical, especially when handling sensitive information. Robust encryption and authentication mechanisms are necessary to protect against potential threats.

2.4 Fundamentals of Federated Learning

FL is a decentralized ML approach where multiple devices (clients) collaboratively train a shared model without sharing their raw data. Instead, only model updates (e.g., gradients) are sent to a central server, preserving data privacy. Key components of FL include:

- **Clients**: Devices (e.g., smartphones, IoT devices) that perform local training on their data.
- **Server**: Aggregates updates from clients to improve the global model.
- **Communication Rounds**: Iterative process of local training, update sharing, and global aggregation.

Types of Federated Learning

FL has evolved into different types, each designed to address specific data distribution scenarios and application needs. In the following items, each type is described along with its key characteristics and use cases [17]:

1. Horizontal Federated Learning (HFL)

Applicable when datasets across clients share the same feature space but differ in samples. For example, hospitals using the same data format but with different patients can use HFL to train a joint model.

2. Vertical Federated Learning (VFL)

Used when datasets share the same users but differ in features. For example, a bank and an e-commerce platform may have overlapping users with distinct types of data.

3. Federated Transfer Learning (FTL)

Suitable when datasets differ in both users and features. Transfer learning techniques are used to enable knowledge sharing between parties with little data overlap.

4. Hybrid Federated Learning

A combination of horizontal and vertical FL to deal with scenarios where both sample and feature spaces partially overlap.

2.5 Use Cases of FL in UAV-Assisted MEC

FL is increasingly being utilized in UAV-assisted (MEC) to enhance performance, security, and efficiency in a wide range of scenarios. Below are some prominent use cases of FL in this domain [18]:

- **Intelligent Task Offloading:**UAVs as flying edge servers can use FL to collaboratively learn efficient offloading strategies, reducing latency and conserving bandwidth.

- **Real-time Object Detection and Surveillance:** FL enables UAVs to train shared object detection models using localized video data, which is crucial for privacy in surveillance and disaster response missions.
- **Edge Caching Optimization:** FL helps UAVs to understand user content preferences and make proactive caching decisions, reducing data traffic and improving user experience.
- **Collaborative Intrusion Detection:** FL allows multiple UAV nodes to build a robust and decentralized intrusion detection system without exposing sensitive logs or patterns.

2.6 Conclusion

This chapter provides a foundational understanding of UAV-assisted MEC and its integration with FL. It highlights the potential advantages, challenges, and opportunities that this technology presents. As research and development in this field continue, UAV-assisted MEC is expected to evolve further, unlocking new possibilities for various industries. In the next chapter, we will present a classification of client selection methods in FL and their key differences.

Chapter 3

Client Selection in Federated Learning

3.1 Introduction

This chapter introduces a novel classification framework for the existing body of research in FL, specifically focusing on client selection techniques. By categorizing prior works based on their underlying approaches and objectives, this classification sheds light on prevailing trends, uncovers critical research gaps, and provides a structured foundation for advancing future developments in secure, efficient, and intelligent federated systems.

3.2 Client Selection Methods in FL

Client selection plays a crucial role in enhancing the efficiency, robustness, and fairness of FL systems. A wide range of strategies has been proposed in the literature to address various challenges such as data heterogeneity, communication overhead, limited resources, security risks, and client mobility especially in dynamic environments like UAV networks or vehicular systems.

There are many client selection methods adapted in the literature, which can be broadly classified into four main categories: (i) Data-based selection, (ii) Resource-based selection, (iii) Security-based selection, and (iv) Mobility-based selection as shown in Fig. 3.1.

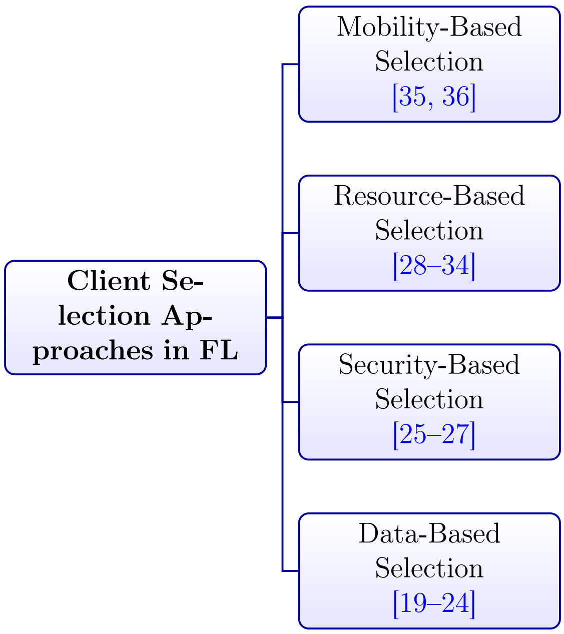

Figure 3.1: Classification of client selection methods in FL.

3.2.1 Data-Based Client Selection

In FL, client selection is a critical step that significantly impacts the efficiency and performance of the model training process [19]. Data-based client selection methods focus on choosing clients based on the characteristics and quality of their local datasets. These methods aim to prioritize clients with diverse, representative, or high-quality data to improve model convergence and generalization [23].

In scenarios involving UAVs, client selection becomes particularly challenging due to the dynamic nature of UAV networks, limited energy resources, and the heterogeneity of data collected from sensors or cameras onboard [21]. For instance, UAVs with larger datasets or those whose data distributions align closely with the global model's objectives may be selected more frequently. Techniques such as data sharding can be used; for instance, in [22], clients (including UAVs) are chosen based on the uniqueness or complementarity of their data, which can help reduce redundancy and enhance learning efficiency. Other approaches, such as the work in [24], include selecting UAVs based on data freshness ensuring that the most recent data is utilized or leveraging data importance metrics to identify UAVs whose data contributes most to model updates. These methods often require UAVs to share metadata about their datasets, such as data size, distribution, or quality metrics, which can be used to make informed selection decisions.

Advantages

- **Improved Model Performance:** By selecting clients (including UAVs) with diverse, high-quality, or representative data, the global model can achieve better generalization and faster convergence [23].

- **Efficient Resource Utilization:** Data-based selection ensures that only the most relevant UAVs or clients participate in training, reducing redundant computations and communication overhead [22].

Drawbacks

- **Communication Overhead:** Collecting and analyzing metadata from UAVs to make selection decisions can introduce additional communication overhead, which may strain the limited bandwidth and energy resources of UAV networks [21].

- **Dynamic Network Challenges:** UAVs operate in highly dynamic environments, with frequent changes in connectivity, location, and data availability. This makes it difficult to consistently apply data-based selection criteria, as the relevance of a UAV's data may change rapidly [21].

3.2.2 Security-based Client Selection

Although FL offers numerous advantages such as distributed training, it still creates critical security defects such as data poisoning, adversarial attacks, and malicious client behavior. Several studies have proposed new attack strategies, defense mechanisms, and privacy-preserving techniques for improving FL security. One study in [27] looks into label-flipping attacks in Graph Neural Network (GNN)-based FL, in which malicious clients change class labels to mislead the overall model. Unlike traditional adversarial attacks that manipulate graph structures, label-flipping attacks are stealthy and cost-effective, making them difficult to detect. The researchers present the Graph Federated Label-Flipping Attack (Graph-), which is designed to disrupt node classification tasks while minimizing the risk of detection. As well, they propose a malicious gradient computation strategy that optimizes label modifications to maximize attack success. Experimental results show that Graph-FLFA has a high attack effectiveness across multiple GNN-based FL models, even against existing defense mechanisms. Another study in [27] analyzes security issues in UAV-enabled FEEL, where untrustworthy and malicious UAV clients can degrade model performance. Due to their high level of mobility and different computing power, UAVs frequently fail to send updates, leading to straggler effects and biased model aggregation. Also, malicious UAVs can deliberately poison model updates, risking global model accuracy. To mitigate these threats, the study suggests a client selection strategy that prioritizes UAVs with high reliability scores while weeding out potentially malicious participants early in the training process. Experimental results show that this approach improves model convergence, mitigates the impact of unreliable UAVs, and increases overall FL robustness. On the other hand, the work in [26] focuses on privacy and security in FL for smart agriculture, with models trained to detect bugs that affect crops.

Traditional FL approaches run the risk of data leakage, poisoning attacks, and inconsistencies in decentralized datasets. To address these concerns, the authors introduce BAFL-SVM (Blockchain-Assisted FL-Driven SVM). This framework uses blockchain to secure model updates, homomorphic encryption to preserve privacy during computation, and secret-sharing techniques to prevent unauthorized access. Experimental results on real-world agricultural datasets show that BAFL-SVM achieves higher recognition accuracy while adhering to strict data security standards, making it a promising strategy for FL applications in settings with limited resources. Together, these investigations highlight the necessity of improved client screening procedures, robust adversarial defenses, and safe data-sharing frameworks in FL. Researchers continue to improve FL's resistance to security threats by fusing attack analysis, defense tactics, and privacy-enhancing methods, opening the door for more reliable and trustworthy decentralized learning systems.

Advantages

- **Enhanced Model Integrity :** Selecting clients with strong security measures reduces the risk of poisoning attacks, ensuring a more trustworthy and accurate model.

- **Improved Data Privacy :** Clients with robust encryption, secure storage, and privacy-preserving mechanisms help prevent data leakage and unauthorized access.
- **Reduced Risk of Malicious Clients :** By filtering out potentially compromised or adversarial clients, the system minimizes Byzantine failures and backdoor attacks.

Drawbacks

- **Limited Client Pool :** Restricting selection to only highly secure clients may reduce participation, limiting data diversity and potentially introducing bias in learning outcome.
- **Potential Exclusion of Valuable Data Sources :** Some clients with useful data but weaker security may be excluded, reducing model generalization.

3.2.3 Resource-Based Client Selection

Clients vary significantly in processing power, with high-end smartphones and edge servers offering robust computational resources, while IoT devices and UAVs often have limited capacity. Selecting clients without considering computational constraints can lead to system inefficiencies, as weaker devices may struggle to complete training tasks on time or may drop out due to overheating or excessive resource consumption. Strategies such as adaptive workload allocation and hierarchical learning architectures can help mitigate these challenges by distributing tasks according to device capabilities [28].

Another critical factor is energy availability, as FL training is inherently power-intensive. Smartphones and IoT devices rely on battery power, while UAVs face additional energy constraints due to flight requirements. A poorly designed selection mechanism that disregards battery levels can lead to premature device shutdowns, increasing model divergence. Several studies propose battery-aware selection frameworks such as [29], where clients with sufficient power reserves are prioritized, while low-energy devices either perform lightweight computations or act as relays for data aggregation. Energy-efficient optimization techniques, such as model compression, quantization, and knowledge distillation, further alleviate the energy burden on resource-limited devices. For instance, in [30], the authors propose a method that focuses on optimizing energy usage by adjusting the computational workload based on the available battery power and processing capacity of the devices.

Communication bandwidth also plays a pivotal role in FL training efficiency. FL involves frequent model updates between the central server and participating clients, making network stability and bandwidth availability crucial factors in selection. IoTdevices often use low-power wide-area networks (LPWAN) such as LoRa or NB-IoT, which impose strict constraints on data transmission rates. Similarly, UAVs experience fluctuating network connectivity, especially in long-range operations. To address this, bandwidth-aware selection algorithms have been proposed such as in [31], where clients with stable and high-speed connections are favored while employing gradient compression and differential update mechanisms to reduce transmission overhead. Additionally, edge-assisted FL has been introduced, where UAVs and IoT devices offload computations to nearby edge servers, reducing direct communication with the central model aggregator [32].

On the other hand, we find works that rely on cluster-based techniques in order to improve communication and stability. For instance, in [33], authors propose a framework where groups of mobile clients coordinate updates collectively, reducing reliance on a single node.

A combination of these resource-aware strategies leads to hybrid selection approaches such as in [34], where different types of clients contribute based on their strengths. IoT devices with stable network connections serve as data aggregators, UAVs with high processing power participate selectively in training, and smartphones contribute dynamically based on user activity and battery levels.

Advantages

- **Efficient Resource Utilization:** Resource-based client selection ensures that only devices with adequate computational power, energy levels, and bandwidth participate in training. This reduces the chances of system overload and ensures smoother training without unnecessary strain on weak clients [37].
- **Improved Model Convergence:** By selecting clients based on their ability to handle computation and communication efficiently, this approach minimizes stragglers (devices that slow down the training process). This leads to faster model convergence and better overall performance [28].

Drawbacks

- **Exclusion of Low-Power Devices:** While selecting clients based on resources improves efficiency, it may lead to the exclusion of resource-limited devices that have valuable data. This could result in biased training models if those devices contain critical data that is not represented in training [31].

- **Additional Overhead in Resource Monitoring:** Implementing resource-aware selection requires real-time monitoring of computational power, energy levels, and network bandwidth. This adds extra processing overhead and can introduce delays in client selection [37].

3.2.4 Mobility-Based Client Selection

Mobility plays a crucial role in the efficiency and convergence of FL within mobile networks such as UAV networks, influencing both communication stability and data heterogeneity. Traditional FL architectures face challenges in mobile environments due to high-speed nodes, and non-independent and identically distributed () data collected by clients. We find a number of works in this area, for instance, the Clustered Vehicular Federated Learning (CFL) approach addresses these challenges by forming dynamic clusters of vehicles that share and aggregate model updates locally through vehicle-to-vehicle (V2V) communication before sending aggregated updates to MEC servers [35]. To address the limitations of traditional FL in vehicular networks, authors in [35] proposed a clustered FL approach that optimizes both communication and learning performance by leveraging V2V interactions. The study introduced a novel clustering-based training process, where vehicles with similar mobility patterns and data distributions form groups to collaboratively train models before sending aggregated updates to MEC servers. This method not only reduces the communication burden but also improves model accuracy in non-i.i.d and unbalanced data scenarios. The authors formulated a joint cluster-head selection and resource allocation problem, taking into account both mobility constraints and data heterogeneity, and solved it using a greedy optimization algorithm to efficiently assign vehicles to clusters. On the other hand, authors [36] examined the impact of mobility on Hierarchical FL (HFL) and provided a theoretical framework for analyzing how mobility accelerates learning convergence. Unlike previous studies that viewed mobility as a limiting factor due to frequent disconnections and unstable network links, this work showed that higher mobility can improve convergence speed by facilitating data mixing across edge servers. The authors developed a Markov mobility model to quantify the effect of vehicles moving between edge servers, demonstrating that mobility enhances data diversity and model generalization.

Advantages

- **Improved Model Generalization :** Selecting mobile clients ensures diverse data collection from various locations, reducing bias and improving model performance across different environments .

- **Efficient Resource Utilization :** By selecting clients based on their mobility, FL can leverage devices with better connectivity, computational power, or lower latency at a given time.

Drawbacks

- **Unstable Participation :** Mobility introduces uncertainty; clients may move out of range, disconnect, or experience poor network conditions, leading to incomplete training rounds.
- **Complex Client Scheduling :** Managing mobility-aware client selection requires sophisticated scheduling algorithms to predict and adapt to changing client availability.
- **Lack of application within UAVs :** Most existing works in this area either focus on 2D movement in vehicular networks or assume a predefined UAV trajectory, limiting their applicability to dynamic and unpredictable UAV environments.

3.3 Edge-Based System Mode : Strengths and Gaps

The paper [22] presents a federated learning framework that operates within an **Edge Network system model** , where edge devices collaborate to train machine learning models without sharing their raw data. This design aims to overcome key challenges in real-world federated learning, such as label noise, non-IID data distribution, and heterogeneous device capabilities. By using techniques like label calibration, data augmentation, and quality-aware aggregation, the system significantly improves training efficiency and accuracy. However, despite these strengths, the model is not perfect. It does not address critical practical issues like "node dropout", "device mobility", or "speed optimization", which are essential for robust deployment in dynamic edge environments. Therefore, while the system model is well-designed for controlled settings, it still has limitations that affect its applicability in more complex real-world scenarios.

3.4 Comparison and Summary

Table 3.1 presents a comparative summary of the studies reviewed in earlier sections. A closer examination reveals a notable research gap in the domain of mobility-aware UAV edge networks. Specifically, the majority of existing works overlook the impact of UAV velocity on the convergence behavior of FL models. Moreover, energy efficiency is a critical factor in UAV operations and is frequently neglected, indicating a need for more effective approaches that jointly consider mobility dynamics and energy constraints in FL-enabled UAV networks.

Therefore, we have attempted to design a new client selection solution for UAV-assisted MEC that takes into account the non addressed challenges related to mobility. The following chapter describes the details of our new solution.

Table 3.1: Comparative Performance of Federated Learning Approaches

Approach	System Model	Dropout Nodes	Mobility	Energy	Speed
[22]	Edge Network	✗	✗	✓	✗
[24]	Edge Network	✗	✗	✗	✗
[36]	Vehicular Network	✗	2D	✗	✓
[28]	N/A	✗	✗	✓	✗
[29]	IoT Network	✗	✗	✓	✗
[34]	Wireless Network	✗	✗	✓	✗
[35]	Vehicular Network	✓	2D	✗	✓
[25]	UAV-Edge Network	✓	3D	✗	✗
[26]	Blockchain-enabled Network	✗	✗	✗	✗
[27]	N/A	✗	✗	✗	✗
Our Solution	UAV-Edge Network	✓	3D	✓	✓

Chapter 4

Mobility-Aware Client Selection in UAV-MEC Networks

4.1 Introduction

In this chapter, we look into the critical role of UAVs selection in FL, which significantly affects the capacity and the performance of distributed ML systems. Client selection algorithms decide which devices participate in each training round, steady factors such as data distribution, device availability, computational resources, and communication constraints. An active client selection does not only upgrade training speed and model accuracy but also it lowers communication costs and increases privacy. Therefore, in this chapter, we propose a novel client selection solution, aimed at improving FL by changing client participation strategies based on network conditions and device capabilities. Additionally, we will compare it against benchmark client selection strategies.

4.2 System Model

In a geographical region, We consider that a base station is deployed and attached to an edge server with computation capabilities (shown as the red triangle in Fig. ??). Additionally, the UAVs are uniformly distributed across the operational area and are equipped with GPS, allowing them to be fully aware of their geospatial positions (as illustrated by the blue circles in Fig. ??). Moreover, each UAV has collected a local dataset. Each UAV train a local model with its own data. This system allows each drone to contribute to make improvement to the global model based on its own data. Moreover, UAVs are considered to move in 3D realm with various speeds [1]. UAVs update their models and send them to the base station for further aggregation indicating that they communicate with it during the training process.

4.3 Mobility-aware Client Selection Strategy

Previously, we thoroughly presented and compared a number of related works in the area of client selection within the context of FL. These studies have provided valuable insights into various strategies and techniques used to improve the selection process. However, based on the findings and the analysis of existing approaches, it is clear that there is a significant research gap, in particular, when it comes to incorporating the mobility of clients in dynamic environments. This gap is especially obvious in UAV edge networks, where the unique characteristics of UAVs, such as their high mobility and varying communication capabilities, are not sufficiently addressed. Therefore, in this work, we aim to propose a specific client selection method that not only focuses on the traditional aspects

[1] We consider that UAVs move in and out of the communication range of the base station where green dashes indicate the connectivity between UAVs and the base station in Fig. ??.

of client performance but also takes into account the speed and mobility of UAVs in the 3D world. By doing so, we ought to enhance the robustness and efficiency of FL systems in scenarios where UAVs are involved, ensuring better participation and model accuracy

4.3.1 Mobility-Aware Client Selection for UAV Edge Networks

To address the challenges posed by the dynamic nature of UAV networks, we propose a mobility-aware client selection algorithm that strategically selects a subset of UAVs for each communication or training round. This method prioritizes UAVs that exhibit favorable mobility characteristics namely, stability, reachability, and communication reliability ensuring their sustained participation in latency-sensitive and collaborative FL tasks. Our client selection solution is described in the following sub-sections. Moreover, Algorithm 2 and Algorithm 1 summarize our contribution.

4.3.1.1 Inputs and Output

Our proposed client selection framework operates based on three key inputs: the total number of UAVs in the network V_n, the total number of communication or training rounds R_n, and a selection fraction f_p, which specifies the proportion of UAVs to be selected in each round. Using these inputs, the algorithm generates a single output, which is a structured list that records the subset of UAVs chosen for each round based on their mobility characteristics.

4.3.1.2 Utility Calculation Process

In each communication or training round, the algorithm initiates the selection process by resetting a list designated to store the utility scores of candidate UAVs. It then iterates over the set of available UAVs, evaluating each one based on a combination of eligibility criteria and performance indicators. Specifically, a UAV must satisfy the following minimum conditions: (1) its battery level must exceed a predefined threshold $\beta_{\text{threshold}}$ (e.g. 15%) to ensure task completion without interruption, and (2) it must lie within the maximum allowable communication range from the base station to guarantee reliable connectivity. Once these prerequisites are met, the UAV's mobility behavior is assessed through a set of dynamic features, including its current distance from a reference center, instantaneous speed, change in distance since the previous round, and variation in speed. These mobility attributes are then fed into a scoring function that computes a composite mobility score. Only UAVs that receive a valid (non-null or above-threshold) score are included in the candidate pool for potential selection in the given round.

We begin by defining the Euclidean distance function, which measures spatial separation between two points:

$$\text{EuclideanDistance}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}, \tag{4.1}$$

where $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2, y_3)$ denote the 3D coordinates of two entities typically a UAV and the base station, where each component corresponds to the x, y, and z axes, respectively.

Using this function, we calculate the current distance between UAV v at time t and the base station as:

$$\text{dist}_v^{(t)} = \text{EuclideanDistance}(v^{(t)}.\text{coords}, \text{BS.coords}), \tag{4.2}$$

where $v^{(t)}$.coords denotes the coordinates of UAV v in round t, and BS.coords denotes the position of the base station.

To quantify the positional change of UAV v between the current and previous rounds, we compute the displacement:

$$\Delta d_v^{(t)} = \text{EuclideanDistance}(v^{(t)}.\text{coords}, v^{(t-1)}.\text{coords}). \tag{4.3}$$

The variation in speed between rounds t and $t-1$ is computed as:

$$\Delta v_v^{(t)} = \left| v_{\text{speed}}^{(t)} - v_{\text{speed}}^{(t-1)} \right|. \tag{4.4}$$

A UAV is only considered further if it satisfies the range constraint:

$$\text{dist}_v^{(t)} < \text{max\_dist}. \tag{4.5}$$

We then compute a set of normalized feature-specific scores to assess UAV suitability. The proximity score favors UAVs closer to the base station:

$$\text{score}_{\text{dist}}^{(t)} = 1 - \frac{\text{dist}_v^{(t)}}{\text{max\_dist}} \tag{4.6}$$

.

The speed score rewards UAVs with lower velocities:

$$\text{score}_{\text{speed}}^{(t)} = 1 - \frac{v_{\text{speed}}^{(t)}}{\text{max\_speed}}. \tag{4.7}$$

The displacement score captures spatial stability:

$$\text{score}_{\Delta d}^{(t)} = 1 - \frac{\Delta d_v^{(t)}}{\max(\Delta d)}. \tag{4.8}$$

The speed variation score measures temporal consistency:

$$\text{score}_{\Delta v}^{(t)} = 1 - \frac{\Delta v_v^{(t)}}{\max(\Delta v)}. \tag{4.9}$$

Finally, the overall utility score is a weighted combination of the individual feature scores:

$$\text{final\_score}_v^{(t)} = a \cdot \text{score}_{\text{dist}}^{(t)} + b \cdot \text{score}_{\text{speed}}^{(t)} + c \cdot \text{score}_{\Delta d}^{(t)} + d \cdot \text{score}_{\Delta v}^{(t)}. \tag{4.10}$$

Here, a, b, c, and d are scalar weights (typically between 0 and 1) indicating the relative importance of each feature.

4.3.1.3 Scoring and Ranking UAVs

UAVs that remain close to the base station, exhibit lower speeds, demonstrate minimal displacement, and maintain consistent speed over time are assigned higher scores. These characteristics are indicative of UAVs that are reliable. By assigning greater weight to such mobility attributes, the scoring function ensures that UAVs with the most desirable behavioral patterns are prioritized.

Following the scoring process, all UAVs are ranked in descending order according to their final scores. A fraction f_p of the highest-ranked UAVs representing the target selection rate for each round is then selected. These UAVs are added to the selected UAVs list, which stores the identifiers of UAVs selected in each round.

Upon completion of all communication or training rounds, the algorithm returns the selected UAVs list as its output. This list serves as a complete schedule of UAV participation across rounds, based on the calculated trade-offs between mobility, availability, and energy constraints.

Algorithm 1 UAV Selection Per Round Based on Mobility Score

Input: Number of UAVs V_n, number of rounds R_n, selection factor f_p
Output: List of selected UAVs per round: `selected_UAVs_per_round`

```
selected_UAVs_per_round ← [ ]                    // List to store selected UAVs per round
foreach round r ∈ R_n do
    U ← ∅                                         // Reset utility list
    foreach UAV v ∈ V_n do
        if v_battery > β_threshold then
            dist ← EuclideanDistance(v.coords, base_station.coords)
            Δd ← EuclideanDistance(v.coords, v.previous_coords)
            Δv ← |v.speed − v.previous_speed|
            score ← ComputeMobilityScore(
              dist, MAX_COMMUNICATION_RANGE, v.speed, MAX_SPEED,
              Δd, MAX_Δd, Δv, MAX_Δv)
            if score ≠ None then
                U ← U ∪ {(score, v)}
U_sorted ← Sort U by score descending
num ← max(0, ⌊f_p|V_n|⌋)
selected_UAVs ← first num UAVs in U_sorted
selected_UAVs_per_round ← selected_UAVs_per_round ∪ {selected_UAVs}
    return selected_UAVs_per_round
```

Algorithm 2 Mobility Score Computation

Function `ComputeMobilityScore(`*dist, max_ dist, speed, max_ speed,* Δd*, max_* Δd*,* Δv*, max_* Δv`):`

- **if** *dist* $>$ *max_ dist* **then**
  - **return** 0
- $\text{score}_{\text{dist}} \leftarrow 1 - \frac{\text{dist}}{\text{max\_dist}}$
- $\text{score}_{\text{speed}} \leftarrow 1 - \frac{\text{speed}}{\text{max\_speed}}$
- $\text{score}_{\Delta d} \leftarrow 1 - \frac{|\Delta d|}{\text{max\_}\Delta d}$
- $\text{score}_{\Delta v} \leftarrow 1 - \frac{|\Delta v|}{\text{max\_}\Delta v}$
- **return** $a \cdot \text{score}_{\text{dist}} + b \cdot \text{score}_{\text{speed}} + c \cdot \text{score}_{\Delta d} + d \cdot \text{score}_{\Delta v}$

4.4 Experimental Results and Discussion

4.4.1 Experimental Setup

We evaluate our proposed mobility-aware FL framework using the MNIST dataset, which contains 60,000 grayscale images of handwritten digits. Each image has a resolution of 28×28 pixels and belongs to one of ten classes, representing digits from 0 to 9. To reflect realistic deployment scenarios where clients often possess non-uniform data, we simulate a non- setting using a $k\%$ random sampling strategy. In our configuration, 70%

of the dataset is randomly assigned across the participating UAVs, while the remaining 30% is distributed in a biased fashion, resulting in highly imbalanced and skewed local datasets.

The FL system consists of a single base station that integrates edge computing capabilities and 20 UAVs operating within its coverage area. Each UAV is assumed to have a fixed processing frequency of 2.5 GHz and a maximum communication range of 250 meters. Each UAV executes one epoch of training per communication round using a batch size of 32 samples. Data heterogeneity is explicitly incorporated by enforcing a non-IID distribution across the UAVs, and client selection is limited to 10% of the total population per round.

For model optimization, we use SGD with a learning rate of 0.001. Each local update consists of a single iteration per round. The model size transmitted by each UAV is 1 MB, with each parameter occupying 32 bytes. Communication is modeled with an uplink bandwidth of 1 MHz and a fixed UAV transmission power of 0.1 Watts. The channel environment includes background noise, modeled with a noise power of 10^{-9} Watts, consistent with typical wireless settings.

In terms of coordination overhead, model aggregation at the BS is assumed to take 0.5 seconds per round. Additionally, UAV mobility is dynamically tracked and updated at 0.1 second intervals throughout the training process.

To evaluate the performance and robustness of the framework, simulations are conducted over 100 FL rounds. During this process, we test the system's resilience under various network dynamics, including fluctuating connectivity, diverse data distributions, and communication delays. The utility function used to score UAVs is parameterized with weights set to $a = 0.3$, $b = 0.3$, $c = 0.2$, and $d = 0.2$, assigning relative importance to proximity, speed, displacement, and speed variation, respectively.

4.4.2 Benchmarks

To contextualize the effectiveness of our solution, we compare it against two baseline methods: random selection and distance-based filtering. Each method represents a different strategy for selecting clients to participate in FL rounds. In order to highlight the difference between them, we define the benchmarks as follows:

- **Random Selection Strategy** In the random selection approach, UAVs are chosen arbitrarily without considering their location, speed, or movement stability. The advantages of this method are that it is easy to understand and implement and does not take into account any element or value, but the disadvantages are that it does not choose the best drones to participate in the rounds.

- **Distance-based Filtering Strategy** This strategy selects UAVs that are geographically closest to the edge server. By reducing communication delays and the risk of disconnections, this method improves performance compared to random selection. It takes into account the distance between the center and each UAV participating in the training.

4.4.3 Evaluation Metrics

To evaluate the performance and robustness of the proposed client selection strategy in UAV-based FL, we use two primary categories of metrics: model performance and client participation stability.

The first metric assesses the global model performance, while the second group captures the dropout behavior of UAV clients across training rounds.

- **Global Test Accuracy:**
  This metric reflects the classification accuracy of the aggregated global model after each FL round. It is evaluated on a centralized test set and indicates how well the model generalizes after incorporating updates from selected UAVs. Higher accuracy values correspond to more effective client contributions and improved learning convergence.

- **Mean Dropout Ratio:**
  The mean dropout ratio in FL represents the average percentage of UAVs that fail to complete local training and upload their model updates during each round. A lower mean dropout ratio implies higher reliability and faster convergence of the global model. It is computed as:

$$\text{Mean Dropout Ratio} = \frac{\sum_{i=1}^{N} D_i}{N},$$

where D_i is the number of dropout events in the i-th round and N is the total number of UAVs.

- **Standard Deviation of Dropout Ratio:**
  This metric quantifies the variability in UAV participation across training rounds. A low standard deviation implies stable client availability, minimizing disruption to model updates. It is calculated using:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^{N}(D_i - \mu)^2}{N}},$$

  where μ is the mean dropout ratio. We note that a low mean and standard deviation in dropout ratios indicate an efficient client selection strategy.

Analysis of The Obtained Results

In this section, we analyze the performance of the proposed client selection strategy by examining its effect on model accuracy and dropout behavior under two distinct mobility scenarios. The primary objective is to evaluate how varying UAV speeds impact the global model's convergence rate and system stability.

To this end, we define two experimental scenarios that differ in terms of UAV mobility patterns:

- **Scenario 1: Low-Speed Mobility** In contrast, this configuration assumes UAV speeds between 2 and 15 meters per second, reflecting slower, more stable flight conditions typical of persistent monitoring or coordinated sensing operations.

- **Scenario 2: High-Speed Mobility** In this setting, UAVs move at speeds ranging from 30 to 40 meters per second. This scenario simulates highly dynamic environments.

The graphs (Fig. 4.3) and (Fig. 4.1) compares three different strategies for selecting drones to participate in each round of FL. The goal is to evaluate the impact of each selection strategy on the model's accuracy over 100 training rounds. As for Figures (Fig. 4.4) and (Fig. 4.2) for the bar chart, it contains 3 parts. The first part is for the average dropout for each method, while for Part 2 it represents the standard deviation values, while Part 3 represents the total number of dropout (number of UAVs).

Scenario 1:

4.4.3.1 Random Selection Strategy:

From the curve (Fig. 4.1) we notice that when the speed decreases, the acceleration of accuracy increases and becomes better , and the final value is around 95% . As for the dropout values, we notice from the figure (4.2) that they are a total of 8 out of 200, meaning that by selecting 200 drones to participate in the FL tasks, there are 8 drones that were out of range, as for the mean dropout (AVG), the value was 0.04 . This indicates that random selection is the worst of the three methods.

4.4.3.2 Distance-based Filtering Strategy:

From the curve (4.1) We notice a slight advantage in acceleration for distance filter , and It reaches 80% around round 38 , and the final value is 95.14% , which is a little better than random selection. As for the dropout values (4.2) that they are a total of 7 out of 200, and for the mean dropout AVG, the value was 0.035, and this is a little better than random and contributes to better accuracy values. However, since it does not account for the drones' motion dynamics (e.g., velocity or trajectory changes), it may still select unstable UAVs, resulting in fluctuations in model accuracy during later rounds.

Our solution Strategy

From (4.1), We notice that the acceleration and final accuracy value have become clearly better than other methods, as the final value is 95.63% . As for the dropout, from (4.2) we notice that it has become 4 out of 200, and for the mean dropout AVG, the value was 0.02, which is much better than the other strategies, and this contributed to the rapid acceleration of the accuracy value.

Scenario 2:

4.4.3.3 Random Selection Strategy :

As the graph shows (4.3), this method leads to the slowest convergence and the lowest overall performance. Accuracy remains below 60% until about round 38 and shows significant fluctuation even in later rounds , It reaches 80% only in round 50 , and this is worse than the first scenario . As for the dropout values, we notice from the figure (4.4) that they are a total of 13 out of 200, meaning that by selecting 200 drones to participate in the FL tasks, there are 13 drones that were out of range, as for the mean dropout (AVG), the value was 0.065 , and this is a large numbers compared to our solution and the distance-filter, and this contributes to reducing the accuracy value.

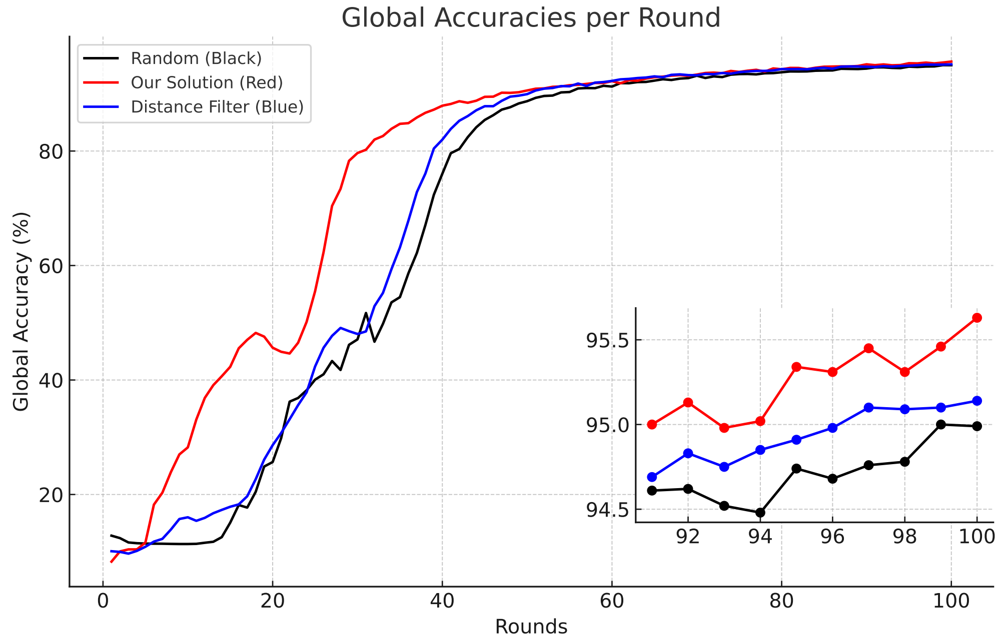

Figure 4.1: Global test accuracy comparison of participant selection methods vs. number of FL rounds in Scenario 1.

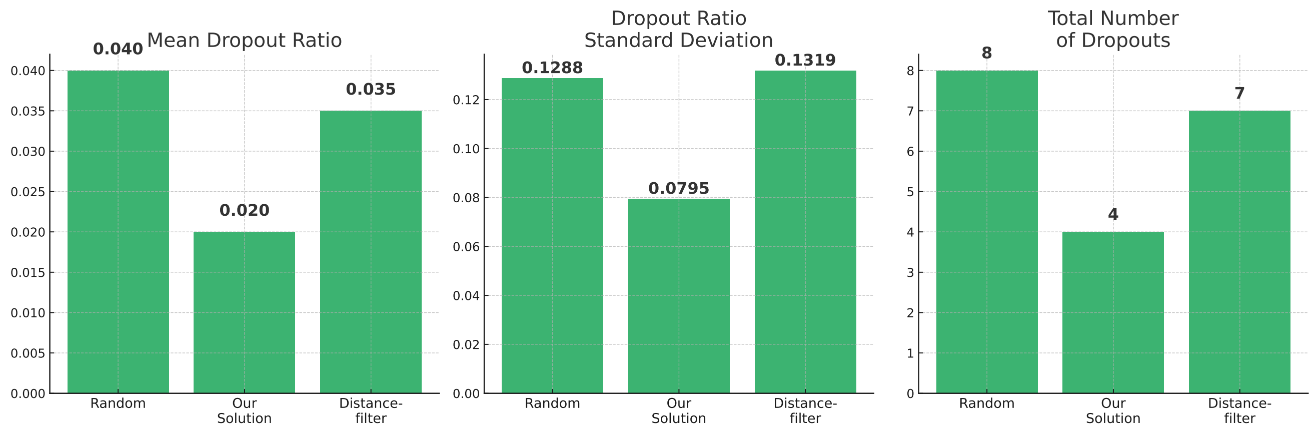

Figure 4.2: Dropout Metrics Across Selection Strategies in Scenario 1.

4.4.3.4 Distance-based Filtering Strategy :

From the curve (4.3) we notice the value of accuracy reaches 80% around round 35 , which is much better than random selection, and continues to climb steadily. As for the dropout values (4.4) that they are a total of 8 out of 200, and for the mean dropout AVG, the value was 0.04, and this is much better than random and contributes to better accuracy values. But these values remain worse than the first scenario.

4.4.3.5 Our solution Strategy :

From the curve (4.3) we notice that this strategy clearly outperforms the others. It achieves over 80% accuracy by round 27 and continues to improve rapidly. By the final round, it end in 95.50% accuracy,This indicates that our method maintains its efficiency even at higher speeds. As for the dropout values (4.4) that they are a total of 4 out of 200, and for the mean dropout (AVG), the value was 0.02, therefore, outperforming both random and distance-filter. This demonstrates the effectiveness of selecting UAVs that are not only near the server but also predictable and reliable in their mobility .

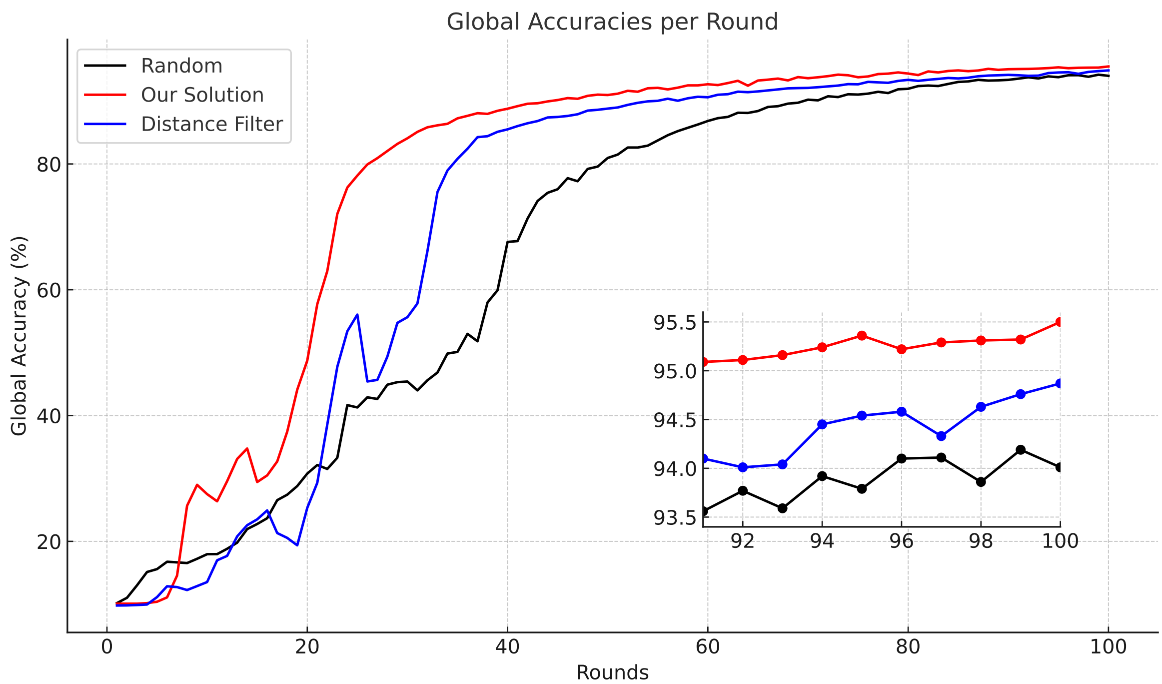

Figure 4.3: Global test accuracy comparison of participant selection methods vs. number of FL rounds in Scenario 2.

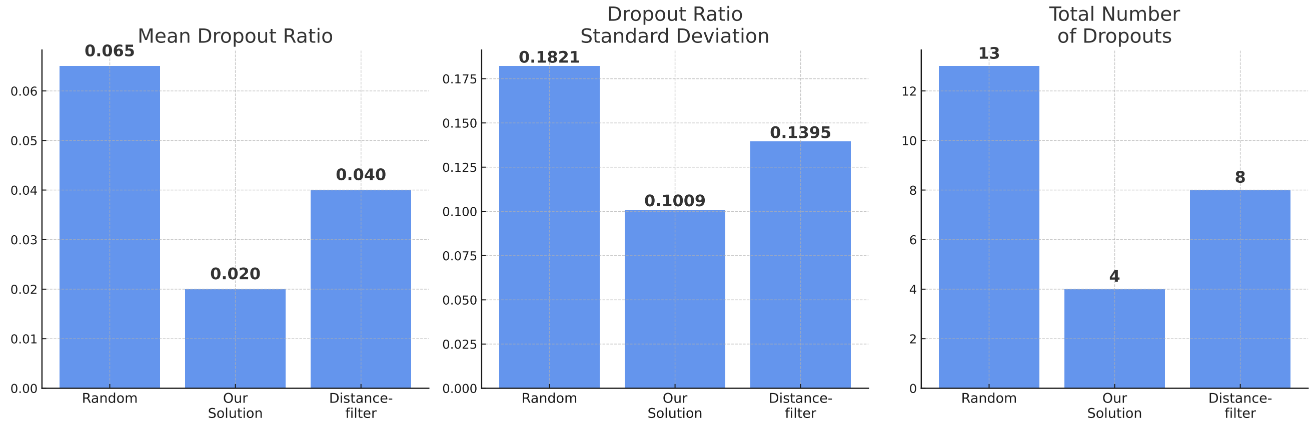

Figure 4.4: Dropout Metrics Across Selection Strategies in Scenario 2.

Scenario 1 VS Scenario 2 :

We observe that as the UAV speed increases, the performance of benchmark methods such as random selection and distance-based filtering significantly degrades, exhibiting higher dropout rates and greater variability. In contrast, our proposed solution maintains robust performance under high-mobility conditions. This resilience stems from the fact that, unlike the distance-based approach which relies solely on proximity, our method incorporates multiple mobility features including speed and its variation into the client selection process. As a result, it consistently identifies UAVs that are not only within communication range but also exhibit stable and predictable mobility patterns, thereby ensuring more reliable participation and efficient learning convergence.

Conclusion :

In this chapter, we presented a novel mobility-aware client selection solution for UAV edge networks. Through simulations, our solution strategy proved to be the most effective of the three benchmarks by incorporating both spatial and temporal movement characteristics, leading to faster convergence, higher final accuracy, less dropout, and greater training stability. This emphasizes the importance of dynamic, context-aware client selection in FL systems involving mobile and potentially unstable devices such as UAVs.

Chapter 5

General Conclusion

Summary of Work

UAVs selection strategies play a crucial role in FL. They directly impact the accuracy of the final model, the speed of training, and the efficiency of communication throughout the process.

In this work, we introduced a novel client selection method tailored for UAV-based systems, taking into account their mobility and energy limitations to enhance overall system performance.

We began by reviewing existing selection strategies: random selection, which is simple but unreliable; and distance-based filtering, which improves stability but ignores mobility dynamics. Our proposed approach improves upon these by incorporating a mobility-aware scoring function. This function evaluates UAVs based on battery level, distance to the server, speed, and stability in movement, allowing the system to prioritize drones that are both close and stable.

To evaluate performance, we conducted two simulation scenarios with varying UAV speeds: high-speed and low-speed. In both scenarios, our strategy significantly outperformed baseline methods. It achieved higher model accuracy (over 95.5%), lower dropout ratios (as low as 0.02), and faster convergence. The results confirm that maintaining stability in UAV movement contributes to higher accuracy and better system reliability.

Ultimately, our solution demonstrates that dynamic, context-aware UAV selection enhances the learning process in mobile federated systems. By reducing dropout and accelerating convergence, it validates its superiority over traditional selection strategies.

Future Perspectives

In the future, we aspire to improve our client selection strategy by improving and integrating other features into the selection method, such as adding a calculation of the size and type of data for each drone and integrating it with mobility-score, so that we obtain a better model in terms of accuracy.

Bibliography

[1] Hashim A. Hashim. Advances in uav avionics systems architecture, classification and integration: A comprehensive review and future perspectives. *Results in Engineering*, 17:100789, 2025.

[2] Richard K. Barnhart, Scott B. Hottman, Douglas M. Marshall, and Eric Shappee. *Introduction to Unmanned Aircraft Systems*. CRC Press, 2012.

[3] Y. Zhou, X. Zhang, and M. Chen. Uav-enabled edge computing in 5g: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, 25(1):34–56, 2023.

[4] Y. Qu, W. Li, and J. Huang. Disaster response with uav-assisted mec: A survey. *IEEE Transactions on Network and Service Management*, 18(4):4120–4135, 2021.

[5] H. Zhang et al. Review of technological hotspots of unmanned aerial vehicles in 2024. *Science & Technology Review*, 2025.

[6] Y. Wang et al. Unmanned aircraft systems (uass): Current state, emerging trends, and future directions. *Drones*, 9(1):59, 2025.

[7] A. Ahmed et al. A review of powering unmanned aerial vehicles by clean and renewable energy sources. *Journal of Cleaner Production*, 2025.

[8] Jeffrey Ouellette and Brian Michini. Uav surveillance and monitoring: A review of applications and technologies. *IEEE Aerospace and Electronic Systems Magazine*, 36(4):22–35, 2021.

[9] Usha Rani Mogili and B. B. V. L. Deepak. Review on application of drone systems in precision agriculture. *Procedia Computer Science*, 167:502–509, 2020.

[10] Milan Erdelj and Enrico Natalizio. Uav-assisted disaster management: Applications and open issues. *IEEE Communications Magazine*, 58(2):24–30, 2020.

[11] Luís Pádua, Ján Vanko, and Jakub Hruška. Uav-based environmental monitoring: A review of applications and challenges. *Remote Sensing*, 13(5):987, 2021.

[12] Alexander Otto, Niels Agatz, and James Campbell. Optimization approaches for civil applications of unmanned aerial vehicles (uavs). *Transportation Science*, 55(1):1–21, 2021.

[13] Karen Anderson and Kevin J. Gaston. Lightweight uavs for environmental monitoring and scientific research. *Remote Sensing in Ecology and Conservation*, 6(2):123–135, 2020.

[14] Andrew Ellenberg and Antonis Kontsos. Uav-based infrastructure inspection: A review of applications and challenges. *Journal of Infrastructure Systems*, 26(2):04020010, 2020.

[15] Mohammad Mozaffari, Walid Saad, and Mehdi Bennis. Uav-enabled wireless networks: A survey. *IEEE Communications Surveys & Tutorials*, 22(4):2624–2661, 2020.

[16] Boxiong Wang, Hui Kang, Jiahui Li, Geng Sun, Zemin Sun, Jiacheng Wang, and Dusit Niyato. Uav-assisted joint mobile edge computing and data collection via matching-enabled deep reinforcement learning. *arXiv preprint arXiv:2502.07388*, 2025.

[17] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[18] Lin Li, Zhi Zhou, Yan Liu, and Yao Zhang. Energy-aware federated learning in uav-assisted mobile edge networks. *Energies*, 16(5):2486, 2023.

[19] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

[20] Yang Liu, Yan Kang, Chaoping Xing, Tianjian Chen, and Qiang Yang. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[21] Chuankai Meng, Yong Li, Haixia Wang, and Wei Zhang. Federated learning in uav-enabled networks: Applications, challenges, and opportunities. *IEEE Communications Magazine*, 60(4):62–68, 2022.

[22] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[23] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[24] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[25] Youssra Cheriguene, Wael Jaafar, Halim Yanikomeroglu, and Chaker Abdelaziz Kerrache. Towards reliable participation in uav-enabled federated edge learning on non-iid data. *IEEE Open Journal of Vehicular Technology*, 5:125–141, 2024.

[26] Ruiyao Shen, Hongliang Zhang, Baobao Chai, Wenyue Wang, Guijuan Wang, Biwei Yan, and Jiguo Yu. BAFL-SVM: A blockchain-assisted federated learning-driven svm framework for smart agriculture. *High-Confidence Computing*, 5(1):100243, 2025.

[27] Shanqing Yu, Jie Shen, Shaocong Xu, Jinhuan Wang, Zeyu Wang, and Qi Xuan. Label-flipping attacks in gnn-based federated learning. *IEEE Transactions on Network Science and Engineering*, pages 1–12, 2025.

[28] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[29] Yang Zhao, Ming Li, Liang Lai, and H Vincent Poor. Energy-efficient federated learning for edge intelligence in 6g networks. *IEEE Internet of Things Journal*, 8(11):8852–8862, 2021.

[30] Shuai Wang, Tom Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource-constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 38(6):1246–1259, 2020.

[31] Mingzhe Chen, Walid Saad, Changchuan Yin, et al. A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 20(1):269–283, 2020.

[32] Xiaofeng Lyu, Yong Zeng, Rui Zhang, and Teng Joon Lim. Uav-aided federated learning: Joint optimization of flight trajectory and learning performance. *IEEE Transactions on Wireless Communications*, 20(7):4449–4463, 2021.

[33] Tien Dat Nguyen, N H Tran, Zhu Han, and Choong Seon Hong. Efficient client selection for federated learning in mobile edge computing. *IEEE Transactions on Mobile Computing*, 21(1):356–371, 2021.

[34] Wenjia Shi, Yuan Liu, Kun Yang, and Jianhua Xu. Communication-efficient federated learning with adaptive client selection strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1430–1443, 2022.

[35] Afaf Taïk, Zoubeir Mlika, and Soumaya Cherkaoui. Clustered vehicular federated learning: Process and optimization. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):25371–25383, 2022.

[36] Tan Chen, Jintao Yan, Yuxuan Sun, Sheng Zhou, Deniz Gündüz, and Zhisheng Niu. Mobility accelerates learning: Convergence analysis on hierarchical federated learning in vehicular networks. *IEEE Transactions on Vehicular Technology*, 74(1):1657–1673, 2025.

[37] J. Wang, X. Liu, and J. Liang. Optimizing federated learning in resource-constrained environments. *IEEE Transactions on Neural Networks*, 31(11):2003–2015, 2020.