electronics

Article

Multi-Constrained and Edge-Enabled Selection of UAV Participants in Federated Learning Process

Sofiane Dahmane [1,2], Mohamed Bachir Yagoubi [1], Bouziane Brik [3,*], Chaker Abdelaziz Kerrache [1], Carlos Tavares Calafate [4] and Pascal Lorenz [5]

1 Laboratoire d'Informatique et de Mathématiques, Université Amar Telidji, Laghouat 03000, Algeria; s.dahmane@lagh-univ.dz (S.D.); m.yagoubi@lagh-univ.dz (M.B.Y.); ch.kerrache@lagh-univ.dz (C.A.K.)
2 Département de Mathématiques, Ecole Normale Supérieure Abderrahmane Taleb, Laghouat 03000, Algeria
3 DRIVE EA1859, University Bourgogne Franche-Comté, 58000 Nevers, France
4 Computer Engineering Department (DISCA), Universitat Politècnica de València, 46022 Valencia, Spain; calafate@disca.upv.es
5 GRTC Laboratory, University of Haute Alsace, 68008 Colmar, France; lorenz@ieee.org
* Correspondence: bouziane.brik@u-bourgogne.fr

Abstract: Unmanned aerial vehicles (UAVs) have gained increasing attention in boosting the performance of conventional networks due to their small size, high efficiency, low cost, and autonomously nature. The amalgamation of UAVs with both distributed/collaborative Deep Learning (DL) algorithms, such as Federated Learning (FL), and Blockchain technology have ushered in a new paradigm of Secure Multi-Access Edge Computing (S-MEC). Indeed, FL enables UAV devices to leverage their sensed data to build local DL models. The latter are then sent to a central node, e.g., S-MEC node, for aggregation, in order to generate a global DL model. Therefore, FL enables UAV devices to collaborate during several FL rounds in generating a learning model, while avoiding to share their local data, and thus ensuring UAVs' privacy. However, UAV devices are usually limited in terms of resources such as battery, memory, and CPU. Some of the UAV devices may not be able to build a local learning models due to their resources capacity. Hence, there is a great need to select the adequate UAVs at each FL round, that are able to build a local DL model based on their resource capacities. In this paper, we design a novel and S-MEC-enabled framework that optimizes the selection of UAV participants at each FL training round, named FedSel. FedSel considers the available UAVs along with their resource capacities, in terms of energy, CPU, and memory, to determine which UAV device is able to participant in the FL process. Thus, we formulate the UAV selection problem as an Integer Linear Program, which considers the aforementioned constraints. We also prove that this problem is NP-hard, and suggest a Tabu Search (TS) metaheuristic-based approach to resolve it. Moreover, FedSel is built on top of blockchain technology, in order to ensure a secure selection of UAV participants, and hence building reliable FL-based models. Simulation results validate the efficiency of our FedSel scheme in balancing computational load among available UAVs and optimizing the UAV selection process.

Keywords: UAV networks; Federated Deep Learning; UAVs selection; blockchain; edge computing

check for updates

Citation: Dahmane, S.; Yagoubi, M.B.; Brik, B.; Kerrache, C.A.; Calafate, C.T.; Lorenz, P. Multi-Constrained and Edge-Enabled Selection of UAV Participants in Federated Learning Process. *Electronics* **2022**, *11*, 2119. https://doi.org/10.3390/electronics11142119

Academic Editor: Martin Reisslein

Received: 28 May 2022
Accepted: 1 July 2022
Published: 6 July 2022

1. Introduction

The unmanned aerial vehicles (UAVs) paradigm is a revolutionary innovation with a strong potential for civic and industrial applications. Internet of Drones (IoD) or Internet of Flying Things (IoFT) is an actual result of UAVs connections to other smart devices by means of a powerful multi-sensor platform, communication technologies, computation units, and IP-based Internet connectivity. Many service providers are investigating to leverage UAV devices for products transportation and shipping, home package delivery, crop monitoring and agricultural surveillance, road traffic monitoring, and lastly, as a search and rescue assistance technology [1].

Recently, tech giants such as Amazon and Google expressed interest in implementing new drone-based parcel delivery systems, igniting a trend that has spawned a slew of commercial UAV-based services and applications. The Federal Aviation Authority (FAA) and civil aviation authorities in other governments; however, they steadfastly refuse to open up the national airspace (NAS) to such services and applications, until they fully meet specific safety and security standards [1,2].

As depicted in Figure 1, the high level of heterogeneity, pervasiveness, and large scale of UAV systems, are expected to increase security risks to modern aerospace, which is becoming increasingly reliant on humans, drones, and robots in multiple combinations. Furthermore, existing security countermeasures and privacy enforcement policies cannot be directly applied to UAV technology, due to its low computational power and payload capacity. As a result, in order to attain UAVs full potential and obtain widespread adoption, appropriate confidentiality, privacy, and trust models must be designed for the heterogeneous UAV environment [2].

Figure 1. MEC-enabled Internet of Flying Things for Smart Cities.

Indeed, security could well be assessed from three main perspectives: (*i*) foremost, confidentiality and integrity of data ought to be assured for both stationary and transferred data, as well as controlling the access and authorisation to the system by other drones and users; (*ii*) users' personal information and data should be well preserved, since UAVs acquire and handle sensitive data; (*iii*) trust aspect since UAV environment is composed of several and heterogeneous devices/users, that handle data of various sorts [1–3].

In this context, on one hand, blockchain is emerging as a promising technology to provide the confidentiality and integrity of exchanged data between UAV devices, and generate trust between the involved UAVs, without the need for a trusted third entity [4,5]. On the other hand, collaborative/distributed Deep Learning (DL), such as Federated Learning (FL), emerges also to not only optimize UAV network management and thus meeting the requirement of their emerged applications, but also to ensure the UAVs' privacy by keeping the needed data (i.e., sensed data) on UAV devices [6–10]. Specifically, FL enables UAV devices to leverage their sensed data to build local DL models. The latter are then sent to a central node for aggregation, in order to generate a global DL model. Therefore, FL enables UAV devices to collaborate during several FL rounds in generating a learning model, while avoiding to share their local data, and thus ensuring UAVs' privacy.

Hence, the deployment of blockchain technology as a ledger to provide distributed FL training is a viable option to address the aforementioned security issues [11,12].

In addition, the amalgamation of UAVs with both collaborative FL and Blockchain technology have ushered in a new paradigm of Secure Multi-Access Edge Computing (S-MEC), where the objective is to bring computation very close to UAV devices and in a secure way. Indeed, a S-MEC node may act as a central node in the federated learning in order to aggregate the UAVs' local models and build a global DL model.

However, UAV devices are usually limited in terms of resources such as battery, memory, and CPU. Thus, some of the UAV devices may not be able to build a local learning models due to their resources capacity. Hence, there is a great need to select the adequate UAVs at each FL round, that are able to build a local DL model based on their resource capacities [5,13].

In this paper, we design a novel and S-MEC-enabled framework, that optimizes the selection of UAV participants at each FL training round, named FedSel. FedSel considers the available UAVs along with their resource capacities, in terms of energy, CPU, and memory, to determine which UAV device is able to participant in the FL process. Thus, we formulate the UAV selection problem as an Integer Linear Program, which considers the aforementioned constraints. We also prove that this problem is NP-hard, and suggest a Tabu Search (TS) metaheuristic-based approach to resolve it as in [14]. Generally speaking, an NP-hard problem may be addressed in three different ways: (*i*) applying an iterative method that produces an optimal solution, (*ii*) using an approximation algorithm that runs in a polynomial time, or (*iii*) using a heuristic technique without any prior guarantee for both solution quality and computing time. Local search methods are commonly used to locate nearly optimal solutions to an NP-hard problem in a reasonable computing time. In this context, tabu search (TS) is one of the most effective heuristic algorithms available. Moreover, FedSel is built on top of blockchain technology, in order to ensure a secure selection of UAV participants, and hence building reliable FL-based models. Simulation results validate the efficiency of our FedSel scheme in balancing computational load among available UAVs and optimizing the UAV selection process. Our contributions are summarized as follows:

- We study the problem of participants selection in FL. The problem is formulated as an Integer Linear Program (ILP), where the objective is to select a subset of UAVs that are able to build local DL models, while increasing the learning accuracy. Noting that we leverage Python's PuLP optimization package [15] to resolve our ILP.
- We show that the problem can be NP-hard and design a novel Tabu Search-based (TS) algorithm to determine near-optimal solutions. This is critical when the number of UAVs and their applications is very high and exact solutions are computationally costly.
- We validate the performance of the proposed scheme via simulation, showing our scheme to succeed in selecting the suitable UAV participants for FL process, while optimizing the aggregated accuracy of the generated learning models in FL.

The following is how the rest of the paper is organized: In Section 2, we present the most relevant existing works. Our proposed UAV selection model formulation and a metaheuristic solution are presented in Section 3, followed by experimental results and performance evaluation in Section 4. Finally, Section 5 concluded this peace of work. Note that used acronyms in this work are illustrated in Table 1, in alphabetical order, for the ease of reference.

Table 1. Acronyms List.

Acronym	Definition
B5G	Beyond 5 Generation
CF	Coalition Formation
CPU	Central Processing Unit
D2D	Device-to-Device
DL	Deep Learning
ETSI	European Telecommunications Standards Institute
FAA	Federal Aviation Authority
FDL	Federated Deep Learning
FL	Federated Learning
GLS	Guided Local Search
HF	Hyper-ledger Fabric
ILP	Integer Linear Program
IoD	Internet of Drones
IoFT	Internet of Flying Things
IoT	Internet of Things
MAB	multi-armed bandit
MEC	Multi Access Edge Computing
ML	Machine Learning
NAS	National Airspace
NP	Non Polynomial
NP-Hard	Non Polynomial Hard
OCF	Overlapping Coalition Formation
RL	Reinforcement Learning
SA	Simulated Annealing
S-MEC	Multi Access Edge Computing
TL	Tabu List
TS	Tabu Search
TZ	Tracking Zone
UAV	Unmanned Aerial Vehicle

2. Related Work

This work addresses the UAVs selection in FL process based on blockchain technology, due the resource-limited nature of UAV nodes. Hence, related work discussion is divided into two broad categories: Client/Participant selection whatever the network/participant nature, and resource-constrained client/participant selection in FL process.

2.1. Client/Participant Selection in FL Process

In general context, a wide range of solutions have been designed, that investigate client/participant selection in FL. The authors in [16], suggested Oort as a strategy to enhance federated training and testing performance, by guided participant selection. Intending to increase model training time-to-accuracy performance, the proposal focuses on the use of customers who have both data, that can help improve model accuracy as well as time training complexity. Based on its present loss and estimated delay, each client is assigned a utility. Each epoch, they recalculate the utility of each client accessible for training and choose the top k clients. In their work, they consider captured statistical heterogeneity, but only to a low level of the loss function, used in model training.

The authors in [17] introduced a tier-based FL system (Tifl), which divides clients into tiers depending on training results. By optimizing both accuracy and training time, the algorithm dynamically selects participating clients from the same tier for each training session. As a result, the performance challenges caused by data and resource heterogeneity are reduced. It is a resource-demanding hardware-based solution. The overall training duration for each client is unknown initially, so they assess all of the clients at the beginning.

FedSAE is a self-adaptive FL system introduced in [18], which adaptively selects clients with higher local training losses, in each training cycle to accelerate global model convergence. To increase device dependability, a prediction technique for each client's affordable workload is also suggested. This would allow for dynamic modification of the number of local training epochs for each client. The authors measure their model convergence by training loss.

The authors in [19] introduced FAVOR, which selects a subset of participating clients for each training round to offset the bias generated by non-IID data. To maximize accuracy while reducing the number of communication rounds, a deep Q-learning formulation for client selection is employed. However, the proposed algorithm requires a significant amount of offline training, and the data on each device should remain consistent during the process.

Similarly, in [20], a novel client selection method is designed, which is based on the Multi-Armed Bandit formulation to choose the subset of clients, with the least amount of class imbalance. They measure local class distributions, by comparing the similarity of FL server local gradient updates with gradients inferred from a balanced proxy data set on the server. It is necessary to understand the global data distribution, in order to generate such a proxy data set, which is difficult to do in FL contexts due to privacy-preservation issues.

2.2. Resource-Constrained Client/Participant Selection in FL Process

Implementing FL in resource-constrained networks has gained a lot of study attention in the past few years. Various studies investigated the use FL to increase learning efficiency and enhance network performance. Nonetheless, research targeting UAV selection in distributed learning is still in its infancy.

In [21], the authors intended to extend the FL process to interact with heterogeneous clients in a cellular network. To enhance the training process, they present a client selection scheme for FL at the mobile edge. They aim to solve the problem of longer update/upload times, due to insufficient computational capabilities or bad wireless channel conditions. To improve future selections and help design efficient service pricing schemes, it is necessary to evaluate the contribution of every single client in the training process. Proposing trust and reputation models to evaluate the reliability of the participating clients is also essential.

Another new resource allocation algorithm for UAV networks based on multi-agent collaborative environment learning is proposed in [22]. It aims to overcome the communication delay and enhance the network efficiency, caused by the centralized architecture. In a distributed architecture, they model each UAV as a self-contained agent, that enhances the utility of UAV networks, through dynamic selection decisions considering the UAV's deployment position, transmission power, and occupied sub-channels.

In [9], the authors introduced Federated Deep Learning concept as a potential solution for many resource-constrained UAV-enabled wireless applications. In the meantime, several issues need to be more investigated such as the optimal number of UAVs (clients), as well as the frequency of local and global model updates. UAVs are not always connected to the FDL due to energy and connectivity constraints. In this context, FDL algorithms should be robust to client loss by predicting such scenarios.

In [23], to handle dynamicity, the authors proposed, using a multi-armed bandit (MAB)-based strategy, to achieve learning convergence effectively in a short period. In permissioned blockchain context such as Hyper-ledger Fabric (HF), the authors addressed the trade-off in peers' number, caused by HF unique execute-order procedure. Meanwhile, advanced technologies such as federated learning have recently been a better solution for wireless system self-adaptation problems.

The authors in [24] introduced a combination of algorithms to maximize UAV data collection from ground sensors, while remaining within time constraints in both offline and online settings. They use an K-means method to group sensor nodes and deploy selected cluster heads, then, a UAV-based data collection is used. Tabu-search, simulated annealing (SA), and guided local search (GLS) were among the offline solutions, while reinforcement learning (RL) techniques were used online.

2.3. Comparative Study and Discussion

Table 2 gives a comparative study between the presented related works, according to several criteria such as, used technology, the heterogeneity support, and considered parameters. From this table, we can clearly observe that even these works have addressed the clients selection in FL process, however, the described works in Section 2.1 provided new selection schemes in general way, i.e., without considering the constraints of the clients that will be involved in the FL process. Therefore, such works are not suitable for UAV-based networks due to the resources limited nature of such networks. In addition, few works have been proposed to deal with UAV selection for FL process, provided in Section 2.2. Only two solutions among these works that have considered energy, CPU and memory parameters in their selection. However, these two works did not leverage any security-related technology, in order to ensure a secure selection of UAV participants. Furthermore, no solution of these two works have leveraged Edge computing, in order to enable a distributed UAV selection with a minimum of latency. To alleviate these issues, we design a novel selection optimization scheme of UAVs in FL, that uses Tabu-search and considers UAVs' energy, CPU, and memory capacities in selecting the suitable UAV clients for each FL round. In addition, our scheme leverages Blockchain and edge computing paradigms to not only ensure a secure UAV selection, but also to provide a distributed and low-latency UAV selection.

Table 2. A Comparative Study.

	Ref.	Year	Used Technologies					Selection	Heterogeneity	Considered Parameters		
			ML	FL	BC	Edge	UAV			Energy	CPU	Memory
Res-F networks	[16]	2021		✓				✓	✓			
	[17]	2020		✓				✓	✓			
	[18]	2021		✓				✓	✓		//	
	[19]	2020	✓	✓				✓	✓			
	[20]	2021		✓		✓		✓	✓			
Res-C networks	[21]	2019		✓		✓		✓	✓		✓	✓
	[22]	2022		✓			✓	✓	✓	✓		
	[9]	2020	✓	✓			✓	✓	✓	✓	✓	✓
	[23]	2022	✓		✓		✓	✓	✓			
	[24]	2021	✓				✓	✓	✓	✓	✓	✓
	Our FedSel	2022		✓	✓	✓	✓	✓	✓	✓	✓	✓

Res-F: Resource-Free, Res-C: Resource-Constrained.

3. FedSel: Our S-MEC-Enabled Selection of UAVs Participants Framework

In this section, we describe our UAVs selection framework for an effective FL process. We present the problem formulation of UAVs selection as Integer Linear Program (ILP). We then provide our tabu-search algorithm to resolve our ILP and find a near optimal solution. Before we proceed, we first give the system model of our UAVs selection problem.

3.1. System Model

Our system considers several geographical regions, that are divided into a set of Tracking Zones (TZ), as defined by ETSI UAS [25]. Each TZ is covered by only one MEC infrastructure, as depicted in Figure 2. Thus, each region is covered by several UAVs and MEC infrastructures, where each UAV is associated with only one MEC infrastructure.

We consider $\mathcal{U}_i^r = \{u_1, u_2, \ldots, u_n\}$ as a set of n UAVs flying in a TZ_i, with a region ID $r = 1, \ldots, k$ and $i = 1, \ldots, z$ the ID of a TZ in the region r. Note that n, the number of UAVs may vary from one zone to another. We also consider $\mathcal{A}pp_j = \{app_1, app_2, \ldots, app_m\}$ as a set of running DL-based applications, for which we need to build deep learning models in federated way. Such applications may be related to energy consumption of each UAV device, UAVs placement, routing performance, etc. For more details, the readers may refer to [9].

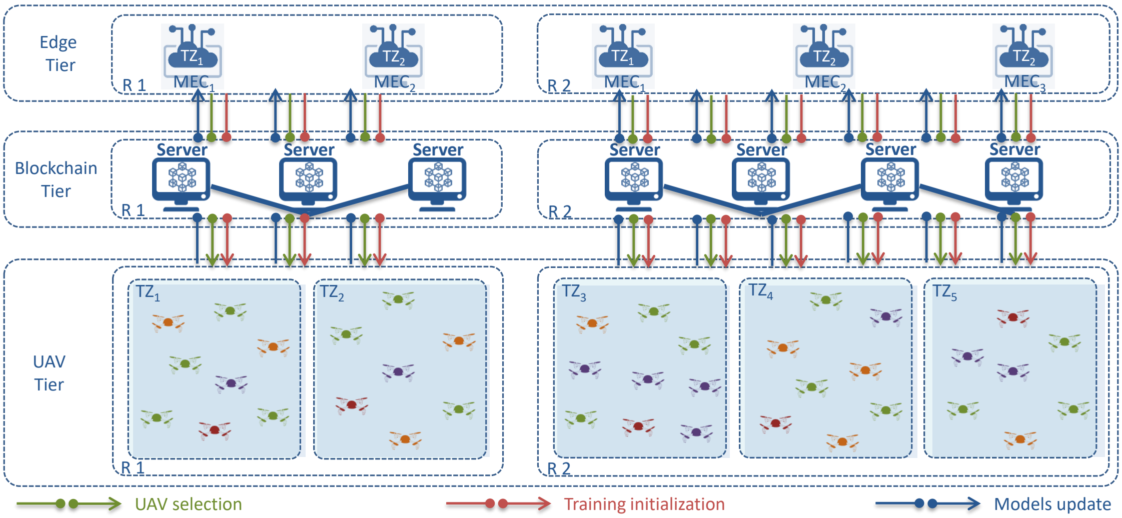


Figure 2. Envisioned Architecture.

As illustrated in Figure 3, for each considered application, its deep learning model can be built in FL way, as follows: ❶ Based on our model (detailed in Section 3.2), each MEC node selects the suitable UAVs to participate in the FL process related to each running application. ❷ Each MEC node initiates the federated learning process by building a first global model and sends it, along with its main parameters, such as learning rate, neural network architecture, and activation functions, to the involved UAVs, through blockchain layer. ❸ Each selected UAV starts to build its local learning model using its sensed data, in distributed way. ❹ finally, each UAV sends its local model (weights) to its MEC node. The latter aggregates the received local models and generates a global learning model, for each considered application.

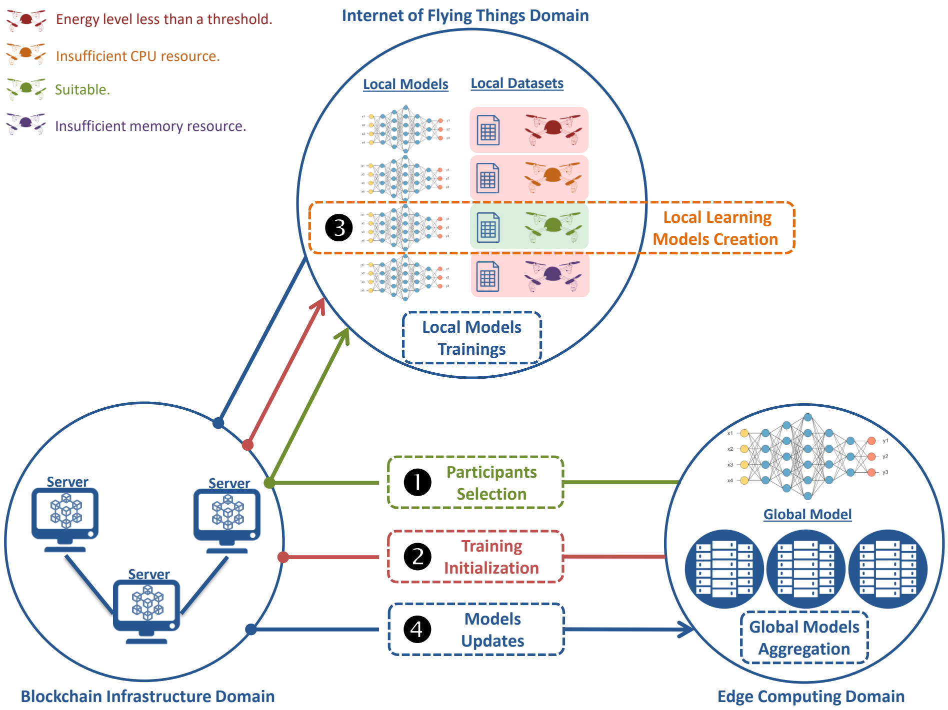


Figure 3. Overview of UAV Selection Problems in Edge-enabled Secure IoFT.

Noting that blockchain acts as an intermediate layer between the MEC nodes and the UAV devices, in order to ensure a secure, trust, and confidential FL process. In addition, this process is repeated during several iterations/rounds, that will be decided by the MEC node. Hence, our model is applied at each round to decide which UAVs will participate in the FL process.

3.2. Uavs Selection Problem Formulation

As mentioned before, we aim to design an optimization model that will help us in selecting the suitable UAVs, for each considered application $\mathcal{A}pp_i$, to participate in building the needed deep learning model in FL manner, while considering UAVs' constraints in terms of resources as well as the leaning accuracy that each UAV can provide. Therefore, we consider the following formulation (Equation (1)):

$$
\begin{aligned}
&\mathcal{M}ax \sum_{i=1}^{n} \sum_{j=1}^{m} (\mathcal{C}[i,j] * \mathcal{A}ccuracy_i) \\
&with: \ \mathcal{C}[i,j] = \begin{cases} 1 & ; UAV_i \ is \ selected \ for \ \mathcal{A}pp_j \\ 0 & ; otherwise \end{cases} \\
&(1) \ \sum_{j=1}^{m} \mathcal{C}[i,j] = 1, \ \forall i = 1, \ldots, n \\
&(2) \ \sum_{i=1}^{n} \mathcal{C}[i,j] >= 2, \ \forall j = 1, \ldots, m \\
&(3) \ \mathcal{C}[i,j] * energy_i \geq required\_energy_j, \ \forall i \in \{1,n\}, \ j \in \{1,m\} \\
&(4) \ \mathcal{C}[i,j] * cpu_i \geq required\_cpu_j, \ \forall i \in \{1,n\}, \ j \in \{1,m\} \\
&(5) \ \mathcal{C}[i,j] * memory_i \geq required\_memory_j, \ \forall i \in \{1,n\}, \ j \in \{1,m\} \\
&(6) \ \forall i \in \{1,n\}, \ j \in \{1,m\}, \ \mathcal{C}[i,j] \in \{0,1\}
\end{aligned}
\tag{1}
$$

where, $\mathcal{C}[i,j]$ is a binary decision matrix. The objective function aims to select the adequate subset of UAVs, among the UAVs covering a tracking zone, maximizing the accuracy of their local learning models $\mathcal{A}ccuracy_i$. The first constraint ensures that a UAV is participating in one and only one FL process for a specific application j, while the second constraint ensure that for each application, at least two UAVs are involved in building its (application) learning model. The third constraint guarantees that the energy required to generate the local model for an application j does not exceed the UAV_i battery level. The fourth constraint aims to ensure that the resource required to build a local model for an application j, in terms of CPU does not surpass that of UAV_i. The fifth constraint ensures that the resource required for an application j in terms of memory does not exceed that of UAV_i. The last constraint maintains the coefficient matrix $\mathcal{C}[i,j]$ as a binary matrix.

It is clear that from the above formulation, our model aims to optimize the participant (UAVs) selection in the FL process. It ensure that the selected UAVs not only are able to build their local models in terms of computing resources, but also improve the accuracy of their learning models. In fact, the UAVs participant selection problem (1) is NP-hard, where the proof can be deduced by reduction from the partition problem, which is also proved to be NP-complete.

Besides, the problem is formulated as an integer linear program, which is known to be NP-complete. In fact, for the matrix $C[UAVs, apps]$, the solutions space size is 2^n ($n = UAVs \times apps$); so, in the worst case, an exact algorithm has to perform a constant number of 2^n operations, which is $(O(k \times 2^n))$.

3.3. Tabu-Search Problem Resolution

For large instances of the problem, its complexity can make its solution computationally extremely expensive. In these instances, meta-heuristics such as the Tabu search [24,26,27] are key to finding a sub-optimal solution. Before carrying on, we first hover at how the Tabu Search method works.

1. *Overview of Tabu Search*: Tabu search (TS) is a mathematical optimization approach that uses a metaheuristic local search method to find sub-optimal solutions to large combinatorial problems, in many practical scenarios. To avoid cycles, TS prevents previously visited solutions or others using user-provided rules and short-term memory. The Tabu List (TL) is formed of these memory structures and comprises a list of recently visited locations. As a result, until a termination condition is met, a local search algorithm is applied to move from one solution to another within the neighborhood solution space. A predefined number of algorithm iterations or a threshold value is usually used as a termination condition.
To construct an initial potential solution $\mathcal{C}_{init}$, the TS algorithm begins with an initialization phase. Note that the farther this solution is from the optimal solution, the greater is the overall execution time.
2. *TS-based UAVs participant selection*: In the following, we describe how we use TS to optimize the selection of UAVs to participate in FL process. We first present the main elements of Tabu Search approach:
    - For the first step, we select the available UAVs for an $\mathcal{A}pp_j$, excluding those that do not supply the $\mathcal{A}pp_j$'s required services.
    - A potential solution $\mathcal{C}$ is a $(N \times M)$ assignment matrix ensuring that all the constraints in our formulation are met:

$$\mathcal{C} = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \mathcal{C}_{ij} & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$

    - To switch from one solution to another, we simply swap the assignments of two applications to two UAVs that are randomly chosen. As a result, a move $m(N,M)$ is a matrix with all of its values equal to zero except the values corresponding to the new and old assignment positions, which are set to one.
    - To achieve a neighborhood solution of $\mathcal{C}$ or a new solution $\mathcal{C}'$, we use the XOR function: $\mathcal{C}' = \mathcal{C} \oplus m$.
    - The attribute of each solution is the value of its objective function. The TL is then updated by including the $\mathcal{C}_{best}$ attribute, which represents the best-obtained solution.

In particular, including the attribute of the best-obtained solution to the TL will allow to avoid returning to previously visited solutions. As a result, both TL computational time and required memory are reduced.

To generate an initial potential solution that meets all of our problem constraints, we apply a "greedy-based" algorithm. To this aim, and for each application, we randomly choose two UAVs at least, verifying both energy and computational requirements. This approach is maintained until all UAV applications are assigned. As shown in Algorithm 1, once an initial solution is obtained, we use the TS-based algorithm to explore the solution space.

Algorithm 1: TS-based UAVs participant selection.

Require: An initial solution $\mathcal{C}_{init}$, number of UAVs N, number of applications M, $required\_energy(M)$, $required\_cpu(M)$, $required\_memory(M)$, and Max_{iter};

Ensure: An optimal (sub-optimal) assignment matrix C_{best};

Procedure `Move(`$\mathcal{C}$: *matrix*`)`**:**

  Generate a neighbor $\mathcal{C}'$ of the current solution $\mathcal{C}_{best}$ by applying a move $m \in Neighbors(\mathcal{C}_{best})$;

Procedure `Attribute()`**:**

  Compute the attribute of a new solution $\mathcal{C}'$ (Objective function value);

Procedure `UpdateTL()`**:**

  Add the $Attribute_{\mathcal{C}'}$ to the TL;

Program `Main`**:**

  $\mathcal{C}_{best} \leftarrow \mathcal{C}_{init}$

  $Attribute_{best} \leftarrow Attribute_{init}$

  while $iter \leq Max_{iter}$ **do**

    `Move(`$Neighbors(\mathcal{C}_{best})$`);`

    `Attribute();`

    if $Attribute_{best} > Attribute_{\mathcal{C}'}$ **then**

      $\mathcal{C}_{best} \leftarrow \mathcal{C}'$;

      $Attribute_{best} \leftarrow Attribute_{\mathcal{C}'}$;

      `UpdateTL();`

    $iter \leftarrow iter + 1$;

  return $\mathcal{C}_{best}$;

4. Experimental Results

In this section, we validate our proposed FedSel UAV selection scheme through simulation for various scenarios as well as comparative study.

4.1. Simulation Setup

The evaluation process was carried out over python implementation of our TS-based approach. Python is developed under an OSI-approved open source license, making it freely usable and distributable [28]. In our implementation, we consider one tracking zone covered by a MEC node, in addition to n UAVs flying in this tracking zone and m MEC applications requiring federated learning. Specifically, we evaluate our UAV selection algorithm in terms of objective function outcomes, while varying the number of MEC applications as well as the number of UAVs. In our simulation, we aim to validate the efficiency of our scheme in maximizing the objective function value (learning accuracy) with means of the resource capacity of selected UAVs. Table 3 provides the main simulation parameters we used. We also note that we generate randomly the different matrices respecting to the aforementioned formulation constraints. Furthermore, our proposal is compared to three solutions: (i) *energy-based UAV selection*, which considers only the energy capacity in its objective function formulation, (ii) *CPU-based UAV selection*, which replaces the accuracy by the CPU capacity in the objective function, and (iii) *Memory-based UAV selection*, which is based only on the storage capacity in its objective function.

Table 3. Simulation Parameters.

Parameter	Value
Number of UAVs	30, 45, 60, 75, 90, 105
Number of MEC apps	3, 6, 9, 12, 15, 18 25
UAV accuracy	[15 , 100]
CPU capacity	[15 , 100] GHz
CPU required	[1 , 15] GHz
Memory capacity	[15 , 100] GB
Memory required	[1 , 15] GB
Energy capacity	[15 , 100] watt
Energy required	[1 , 15] watt

4.2. Performance Evaluation of Our UAV Selection Scheme

The main performance metric considered in the evaluation process of the developed approach is the result of the objective function considered in two scenarios: (i) we varied the number of applications, while the number of UAVs deployed takes two fixed values: 30 and 105. (ii) we varied the number of UAVs, while the number of MEC applications takes two fixed values: 18 and 25 as mentioned in Table 3.

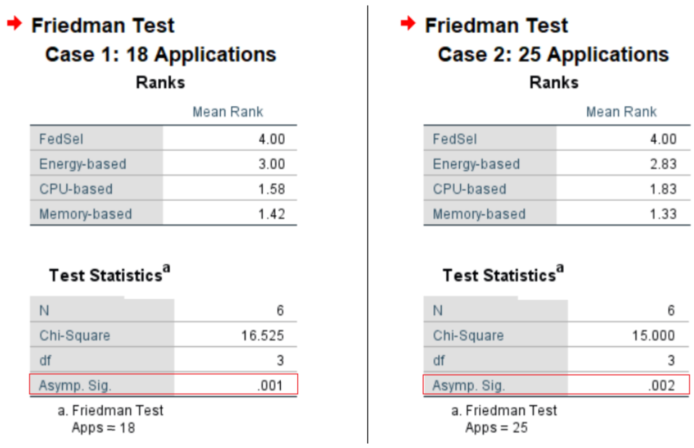
Table 4 recaps the main results relative to our FedSel approach, along with three comparative proposals. The obtained objective function scores are presented in the two different aforementioned cases. In case 1, when the number of UAVs is 90, our FedSel approach clearly outperforms the other single-criterion algorithms, while the results are relatively stable for the other UAVs number variations. In case 2, by changing the number of UAVs each time, the results are relatively convergent, with a clear advantage of FedSel.

Table 4. Objective function numerical results.

	Case 1: 18 Apps				Case 2: 25 Apps			
UAVs Number	**FedSel**	**Energy-Based**	**CPU-Based**	**Memory-Based**	**FedSel**	**Energy-Based**	**CPU-Based**	**Memory-Based**
30	1717	1668	1457	1539	1763	1727	1661	1514
45	2739	2620	2447	2351	2645	2463	2321	2375
60	3439	3350	3139	2915	3638	3529	3481	3212
75	4414	4177	3891	3891	4270	4241	4096	4013
90	5184	5162	4803	4996	5179	5158	5160	5113
105	5780	5745	5650	5405	6265	6210	5935	5962

To prove the supremacy of our FedSel algorithm, we used the Friedman rank test on the data presented in Table 4. Figure 4 depicts the outcomes of the Friedman rank test. It clearly shows that the p-values for both scenarios are strictly less than 0.05, which means that the differences between the obtained objective function scores for each number of UAVs, and for both cases, are statistically significant.

Figure 5a,b depict the results of the objective function when we increase the number of applications. As we observe, our TS-based scheme outperforms the other schemes by maximizing the objective function values, whatever the number of MEC applications in both scenarios. In other words, our scheme succeeds to select the sub-set of UAVs, for the federated learning process, that maximize the learning accuracy with respect to their (UAVs) resource capacities. Therefore, these results demonstrate the efficiency of our TS-based algorithm to find an optimal (sub-optimal) solution, that improves both learning accuracy as well as the offered computational resources and energy.



Friedman Test

Case 1: 18 Applications

Ranks

	Mean Rank
FedSel	4.00
Energy-based	3.00
CPU-based	1.58
Memory-based	1.42

Test Statistics[a]

N	6
Chi-Square	16.525
df	3
Asymp. Sig.	.001

a. Friedman Test
Apps = 18

Friedman Test

Case 2: 25 Applications

Ranks

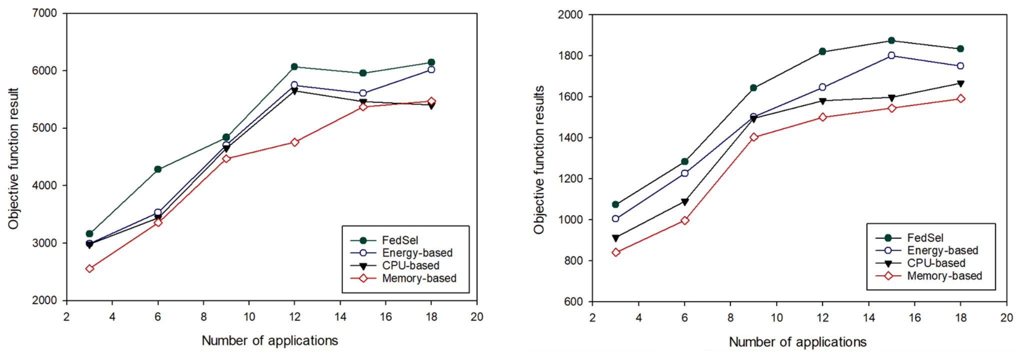
	Mean Rank
FedSel	4.00
Energy-based	2.83
CPU-based	1.83
Memory-based	1.33

Test Statistics[a]

N	6
Chi-Square	15.000
df	3
Asymp. Sig.	.002

a. Friedman Test
Apps = 25

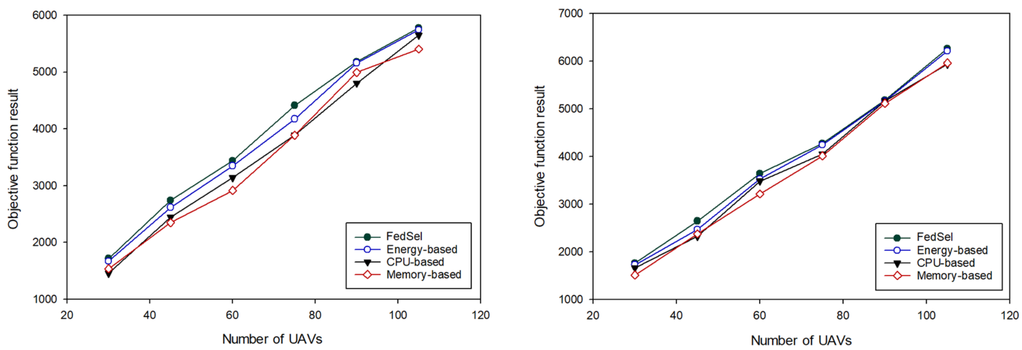
Figure 4. Friedman rank test results.



(**a**) (**b**)

Figure 5. Performance evaluation of our UAV selection scheme in terms of objective function results while varying the number of applications [3, 6, 9, 12, 15, 18]. (**a**) the number of UAVs deployed is 30, (**b**) the number of UAVs deployed is 105.

Figure 6a,b illustrate the results of our TS-based solution, when we vary the number of UAVs, while fixing the number of applications to 18 and 25 applications, respectively. We clearly see that our scheme outperforms the other schemes, by maximizing the objective function values whatever the number of UAVs. Thus, our scheme enables to optimize the UAVs selection even when we fix the number of applications and vary the number of UAVs.



(**a**) (**b**)

Figure 6. Performance evaluation of our UAV selection scheme in terms of objective function results while varying the number of UAVs [30, 45, 60, 75, 90, 105]. (**a**) the number of applications is 18, (**b**) the number of applications is 25.

In general, the evaluation of our TS-based solution confirms beyond doubt the utility of load-balancing the applications among UAVs to provide the required computing resources while considering the energy issue of UAVs. It validates the efficiency of our TS-based algorithm to select and find the suitable UAVs sub-set that can be involved in the federated learning process and built an accurate learning models for the MEC applications.

5. Conclusions and Future Directions

This work describes FedSel, as a novel S-MEC-enabled framework that optimizes the selection of UAV participants at each FL training round. It considers the learning accuracy of each UAV node, along with its computational resources and energy, to determine whether a UAV device is able to participate in the FL process or not. Thus, we formulate the UAV selection problem as an Integer Linear Program, which considers energy, CPU, and memory constraints. The problem is proved to be NP-hard, and we suggested a Tabu Search (TS) metaheuristic-based approach to resolve it. In addition, FedSel is built on top of blockchain technology, in order to ensure a secure selection of UAV participants, and hence building reliable FL-based models. The evaluation results validate the efficiency of our FedSel scheme in balancing computational load among available UAVs and optimizing the UAV selection process in terms of learning accuracy. They also show through the Friedman rank test on realistic data that the differences between the obtained objective function scores for different numbers of UAVs are statistically significant.

As a future work, we plan to evaluate the scalability of our scheme in a realistic UAVs-enabled environment.

Author Contributions: Conceptualization, S.D.; Project administration, P.L.; Supervision, C.T.C.; Validation, C.A.K.; Writing—review and editing, M.B.Y. and B.B. The authors equally contributed to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work is derived from R&D project RTI2018-096384-B-I00, funded by MCIN/AEI/10.13039/501100011033 and "ERDF A way of making Europe".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Billah, M.; Mehedi, S.; Anwar, A.; Rahman, Z.; Islam, R. A Systematic Literature Review on Blockchain Enabled Federated Learning Framework for Internet of Vehicles. *arXiv* **2022**, arXiv:2203.05192.
2. Syed, F.; Gupta, S.K.; Hamood Alsamhi, S.; Rashid, M.; Liu, X. A survey on recent optimal techniques for securing unmanned aerial vehicles applications. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4133. [CrossRef]
3. Zhang, X.; Chen, X. UAV task allocation based on clone selection algorithm. *Wirel. Commun. Mob. Comput.* **2021**, *2021, 5518927.* [CrossRef]
4. Kayalvizhi, M.; Ramamoorthy, S. Review of Security Gaps in Optimal Path Selection in Unmanned Aerial Vehicles Communication. In *Sustainable Advanced Computing*; Springer: Singapore, 2022; pp. 439–451.
5. Nguyen, D.C.; Hosseinalipour, S.; Love, D.J.; Pathirana, P.N.; Brinton, C.G. Latency Optimization for Blockchain-Empowered Federated Learning in Multi-Server Edge Computing. *arXiv* **2022**, arXiv:2203.09670.
6. Liu, X.; Deng, Y.; Mahmoodi, T. A Novel Hybrid Split and Federated Learning Architecture in Wireless UAV Networks. In *Proceedings of the IEEE ICC, Seoul, South Korea, 16–20 May 2022*; IEEE: Manhattan, NY, USA, 2022.
7. Brik, B.; Messaadia, M.; Sahnoun, M.; Bettayeb, B.; Benatia, M.A. Fog-Supported Low-Latency Monitoring of System Disruptions in Industry 4.0: A Federated Learning Approach. *ACM Trans. Cyber-Phys. Syst.* **2022**, *6*, 14. [CrossRef]
8. Yang, H.; Zhao, J.; Xiong, Z.; Lam, K.Y.; Sun, S.; Xiao, L. Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3144–3159. [CrossRef]
9. Brik, B.; Ksentini, A.; Bouaziz, M. Federated learning for UAVs-enabled wireless networks: Use cases, challenges, and open problems. *IEEE Access* **2020**, *8*, 53841–53849. [CrossRef]
10. Brik, B.; Ksentini, A. On Predicting Service-oriented Network Slices Performances in 5G: A Federated Learning Approach. In Proceedings of the 2020 IEEE 45th Conference on Local Computer Networks (LCN), Sydney, NSW, Australia, 16–19 November 2020; pp. 164–171. [CrossRef]
11. Saraswat, D.; Verma, A.; Bhattacharya, P.; Tanwar, S.; Sharma, G.; Bokoro, P.N.; Sharma, R. Blockchain-Based Federated Learning in UAVs Beyond 5G Networks: A Solution Taxonomy and Future Directions. *IEEE Access* **2022**, *10*, 33154–33182. [CrossRef]
12. Abou El Houda, Z.; Brik, B.; Ksentini, A.; Khoukhi, L.; Guizani, M. When Federated Learning Meets Game Theory: A Cooperative Framework to secure IIoT Applications on Edge Computing. *IEEE Trans. Ind. Inform.* **2022**, 1. [CrossRef]
13. Otoum, S.; Al Ridhawi, I.; Mouftah, H. A Federated Learning and Blockchain-enabled Sustainable Energy-Trade at the Edge: A Framework for Industry 4.0. *IEEE Internet Things J.* **2022**. [CrossRef]

14. Mathlouthi, I.; Gendreau, M.; Potvin, J.Y. A metaheuristic based on Tabu search for solving a technician routing and scheduling problem. *Comput. Oper. Res.* **2021**, *125*, 105079. [CrossRef]
15. Mitchell, S.; OSullivan, M.; Dunning, I. *PuLP: A Linear Programming Toolkit for Python*; The University of Auckland: Auckland, New Zealand, 2011; Volume 65.
16. Lai, F.; Zhu, X.; Madhyastha, H.V.; Chowdhury, M. Oort: Efficient federated learning via guided participant selection. In Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21), Santa Clara, CA, USA, 14–16 July 2021; pp. 19–35.
17. Chai, Z.; Ali, A.; Zawad, S.; Truex, S.; Anwar, A.; Baracaldo, N.; Zhou, Y.; Ludwig, H.; Yan, F.; Cheng, Y. Tifl: A tier-based federated learning system. In Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing, Stockholm, Sweden, 23–26 June 2020; pp. 125–136.
18. Li, L.; Duan, M.; Liu, D.; Zhang, Y.; Ren, A.; Chen, X.; Tan, Y.; Wang, C. FedSAE: A novel self-adaptive federated learning framework in heterogeneous systems. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen , China, 18–22 July, 2021; pp. 1–10.
19. Wang, H.; Kaplan, Z.; Niu, D.; Li, B. Optimizing federated learning on non-iid data with reinforcement learning. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 1698–1707.
20. Yang, M.; Wang, X.; Zhu, H.; Wang, H.; Qian, H. Federated learning with class imbalance reduction. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 23–27 August 2021; pp. 2174–2178.
21. Nishio, T.; Yonetani, R. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7. [CrossRef]
22. Dai, Z.; Zhang, Y.; Zhang, W.; Luo, X.; He, Z. A Multi-Agent Collaborative Environment Learning Method for UAV Deployment and Resource Allocation. *IEEE Trans. Signal Inf. Process. Over Netw.* **2022**, *8*, 120–130. [CrossRef]
23. Kim, S.; Ibrahim, A.S. Byzantine-Fault-Tolerant Consensus via Reinforcement Learning for Permissioned Blockchain-Empowered V2X Network. *IEEE Trans. Intell. Veh.* **2022**. [CrossRef]
24. Ghdiri, O.; Jaafar, W.; Alfattani, S.; Abderrazak, J.B.; Yanikomeroglu, H. Offline and Online UAV-Enabled Data Collection in Time-Constrained IoT Networks. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 1918–1933. [CrossRef]
25. Brik, B.; Frangoudis, P.A.; Ksentini, A. Service-oriented MEC applications placement in a federated edge cloud architecture. In Proceedings of the ICC 2020–2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
26. Asim, M.; Mashwani, W.K.; A. Abd El-Latif, A. Energy and task completion time minimization algorithm for UAVs-empowered MEC SYSTEM. *Sustain. Comput. Inform. Syst.* **2022**, *35*, 100698. [CrossRef]
27. Qi, N.; Huang, Z.; Zhou, F.; Shi, Q.; Wu, Q.; Xiao, M. A Task-driven Sequential Overlapping Coalition Formation Game for Resource Allocation in Heterogeneous UAV Networks. *IEEE Trans. Mob. Comput.* **2022**, 1. [CrossRef]
28. Welcome to Python. Available online: https://www.python.org/ (accessed on 20 June 2022).