frontiers in Communications and Networks

ORIGINAL RESEARCH
published: 12 July 2021
doi: 10.3389/frcmn.2021.709946

On Addressing Heterogeneity in Federated Learning for Autonomous Vehicles Connected to a Drone Orchestrator

Igor Donevski, Jimmy Jessen Nielsen and Petar Popovski*

Department of Electronic Systems, Aalborg University, Aalborg, Denmark

In this paper we envision a federated learning (FL) scenario in service of amending the performance of autonomous road vehicles, through a drone traffic monitor (DTM), that also acts as an orchestrator. Expecting non-IID data distribution, we focus on the issue of accelerating the learning of a particular class of critical object (CO), that may harm the nominal operation of an autonomous vehicle. This can be done through proper allocation of the wireless resources for addressing learner and data heterogeneity. Thus, we propose a reactive method for the allocation of wireless resources, that happens dynamically each FL round, and is based on each learner's contribution to the general model. In addition to this, we explore the use of static methods that remain constant across all rounds. Since we expect partial work from each learner, we use the FedProx FL algorithm, in the task of computer vision. For testing, we construct a non-IID data distribution of the MNIST and FMNIST datasets among four types of learners, in scenarios that represent the quickly changing environment. The results show that proactive measures are effective and versatile at improving system accuracy, and quickly learning the CO class when underrepresented in the network. Furthermore, the experiments show a tradeoff between FedProx intensity and resource allocation efforts. Nonetheless, a well adjusted FedProx local optimizer allows for an even better overall accuracy, particularly when using deeper neural network (NN) implementations.

Keywords: federated learning, contribution, incentive, staleness, convergence, UAV, heterogeneous network, fedprox

OPEN ACCESS

Edited by:
Mingzhe Chen,
Princeton University, United States

Reviewed by:
Jiawen Kang,
Nanyang Technological University, Singapore
Zhaohui Yang,
King's College London, United Kingdom

****Correspondence:***
Igor Donevski
igordonevski@es.aau.dk

Specialty section:
This article was submitted to Data Science for Communications, a section of the journal Frontiers in Communications and Networks

Received: *14 May 2021*
Accepted: *22 June 2021*
Published: *12 July 2021*

Citation:
Donevski I, Nielsen JJ and Popovski P (2021) On Addressing Heterogeneity in Federated Learning for Autonomous Vehicles Connected to a Drone Orchestrator. Front. Comms. Net. 2:709946. doi: 10.3389/frcmn.2021.709946

1 INTRODUCTION

The adoption of ubiquitous Level-5 fully independent system autonomy in road vehicles (as per the SAE ranking system (SAE, 2016)) is barred from progress due to the omnipresence of chaotic traffic in legacy traffic situations. Moreover, a 38% share of prospective users are skeptical of the performance of the autonomous driving systems (Nielsen and Haustein, 2018). As such, lowering the number of negative outcome outliers in autonomous vehicle operation, particularly ones that lead to fatal incidents, can be addressed with an overabundance of statistically relevant data (Yaqoob et al., 2019). Thus, given the privacy requirements and the abundance of the data that is produced by road vehicles and/or unmanned erial vehicles (UAVs) in the role of traffic monitors, the machine learning (ML) problem can be addressed by treating the participatory vehicles as learners in a federated learning (FL) network.

In more detail, FL is an ML technique that distributes the learning across many learners. In this way, many separate models are aggregated in order to acquire one general model at server side (Konečnỳ et al., 2016). In FL, each learner does not have to send heaps of data to a common server for processing, but maintains the data privately. As such, the concept of FL is an extension of distributed ML with four important distinctions: 1) the training data distributions across devices can be non-IID; 2) not all devices have similar computational hardware; 3) FL scales for networks of just few devices to vast networks of millions; 4) FL can be engineered in a way in which privacy is conserved. Given the vast complexity of implementing FL in autonomous vehicular traffic, particularly related to the quickly changing environment, in this paper we focus on solving the issues of non-IID data learnt across several devices with unequal processing power. A list of relevant symbols, and their descriptions are contained in **Table 1**, and a review on relevant FL literature follows below.

1.1 State of the Art

FL is an emergent field that has gained immense popularity in the last five years. From the relevant literature we highlight several works. (Li et al., 2020) covers the state of the art regarding computational models (Yang et al., 2019), contains a clear understanding of the FL potential and its most prominent applications (Lim et al., 2020). and (Aledhari et al., 2020) provide comprehensive coverage on the communications challenges for the novel edge computation (Niknam et al., 2020), analyzes scenarios of FL where learners use wireless connectivity. Challenges and future directions of FL systems in the context of the future 6G systems is given in (Yang et al., 2021), while (Savazzi et al., 2021) elaborates upon the applications of FL on connected automated vehicles and collaborative robotics (Khan et al., 2020). covers resource allocation and incentive mechanisms in FL implementations. Most of the works on FL concerning UAVs treat the devices as learners (Wang et al., 2020; Zeng et al., 2020; Zhang and Hanzo, 2020). This requires mounting heavy computational equipment on-board, and therefore it is an energy inefficient way of exploiting drones. In contrast, in our prior work (Donevski et al., 2021a) we have investigated techniques for reducing staleness when a UAV acts as an orchestrator by optimizing its flying trajectory.

There is also an interest in wireless resource allocation optimization for FL networks, as covered in the topics that follow. The work of (Chen et al., 2020a) proposes a detailed communications framework for resource allocation given complex wireless conditions and an FL implementation on IID data. This work has a strong contribution to the topic of convergence analysis of wireless implementations of FL with very detailed channel model. The work of (Amiri and Gündüz, 2020) does a detailed convex analysis for distributed stochastic gradient decent (SGD) and optimizes the power allocation for minimizing FL convergence times. The work of (Tran et al., 2019) formulates FL over wireless network as an optimization problem and conducts numerical analysis given the subdivided optimization criteria. However, the aforementioned works perform their analysis on SGD which has been shown to suffer in the presence of non-IID data and unequal work times (Li et al., 2018). The novel local subproblem that includes a proximal optimizer in (Li et al., 2018) achieves 22% improvements in the presence of unequal work at each node.

The learning of both single task and multi task objectives in the presence of unequal learner contributions is a difficult challenge and has received a lot of attention, e.g. in the works of (Smith et al., 2017; Li et al., 2019; Mohri et al., 2019). This also leads to the question of analyzing contributions among many learners with vastly different hardware that is considered in works covering FL incentive mechanisms, by (Kang et al., 2019a; Chen et al., 2020b; Khan et al., 2020; Nishio et al., 2020; Pandey et al., 2020). The incentive based FL implementations rely on estimating each learner's contribution and rewarding them for doing the work. Hence calculating appropriate rewards becomes a difficult challenge that also comes at the price of computation and communications as shown by (Kang et al., 2019b). Such mechanisms are useful when orchestrating an FL where learners would collect strongly non-IID data and learn with vastly different processing capabilities.

TABLE 1 | Relevant symbols of variables, constants and functions.

Symbol	Definition
$h_{MAX}()$	Utility function that maximizes the number of computed epochs
$h_{AAS}()$	Utility function that minimizes the average anchored staleness
$h_{ACT}()$	Utility function that maximizes based on the estimated contributions from each learner
$F()$	Local machine learning optimization function
$f()$	Global (network-wide) optimization function
$E()$	Model evaluation function
$\omega_{g\setminus\{k\},i}$	Custom model aggregator that excludes the k learner's model
i	An integer indicating the FL cycle/round
k	Learner index number
K	Total number of leanrners in the MA
T_i	Vector representation of the epochs computed across all learners for round i
G_i	Vector representation of the contributions computed, for all learners, for round i
S_i	Vector representation of bandwidth allocated for each learner for round i
$G_{k,i}$	Estimated contribution for learner k, at round i
$\omega_{g,i}$	The global ML model weights for round i
$\omega_{k,i}$	The ML model weights produced at learner k for round i
$\tau_{k,i}$	Epochs computed at learner k for round i
B	The size of the batch computed at each epoch
μ	Proximal term intensity in the FedProx FL implementation
f_k	Processing capability of learner k in terms of epochs per millisecond
W	Total bandwidth allocated for the system
D	Total data transmitted in both directions to a single learner within a single round
R_{avg}	Channel data rate in symbols per hertz
$S_{k,i}$	Bandwidth allocation coefficient for learner k and round i
α	Computation phase duration coefficient (in milliseconds)
β	Communication phase duration coefficient (in milliseconds)
S_{min}	The lower bound of the bandwidth allocation coefficient
S_{max}	The extreme bound of the bandwidth allocation coefficient

1.2 Drone Traffic Monitors as Federated Learning Orchestrators

Unmanned erial vehicles (UAVs) or drones could provide an essential aid to the vehicular communication networks by carrying

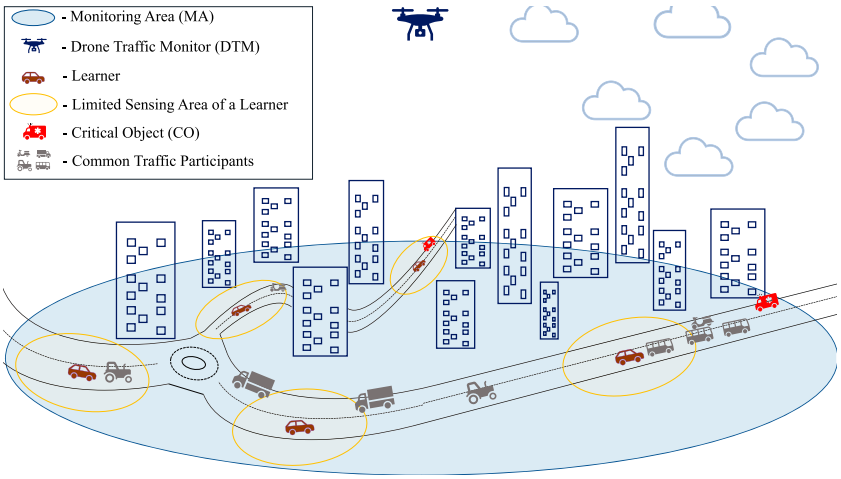

FIGURE 1 | Illustration of the DTM covered monitoring area, with five scattered learners.

wireless base stations (BSs). In combination with the 5G standardization and the emerging 6G connectivity, drone-aided vehicular networks (DAVNs) (Shi et al., 2018) are capable of providing ultra reliable and low latency communications (URLLC) (Popovski et al., 2018; She et al., 2019) when issuing prioritized and timely alarms. In accord, most benefits of DAVNs come as consequence of the UAV's capability to establish line of sight (LOS) with very high probability (Mozaffari et al., 2019). The good LOS perspective also benefits visual surveillance, hence enabling UAVs to offer just-in-time warnings for critical objects (COs) that can endanger the nominal work of autonomous vehicles. Though DAVNs expect many roles from the drone, we draw inspiration from UAVs in the role of drone traffic monitors (DTMs) that continuously improve and learn to perform timely and reliable detections of COs. To avoid requiring a plethora of drone-perspective camera footage of the traffic, we propose DTMs that take the role of a federated learning (FL) orchestrator, and autonomous vehicles participate as learners.

This FL architecture with a drone orchestrator, illustrated in **Figure 1**, exploits the processing and sensing enabled vehicles contained in the monitoring area (MA) to participate both as learners and supervisors. The vehicle-learners receive the drone provided footage, and do the heavy computational work of ML training for the task of computer vision. This is possible since the vehicle-learners have robust sensing capabilities, and when they have the CO in view, can contribute to the learning process due to their secondary perspective (Chavdarova et al., 2018) on the object, and their deeper knowledge of traffic classes. However, even when assuming perfect supervision by the learners, FL is not an easy feat since some knowledge can be obfuscated among omnipresent information and/or contained at computationally inferior straggler learners. In accord, we use a combination of state of the art FL implementation with a novel resource aware solution for balancing work times and learner contributions, which are described in the overview that follows.

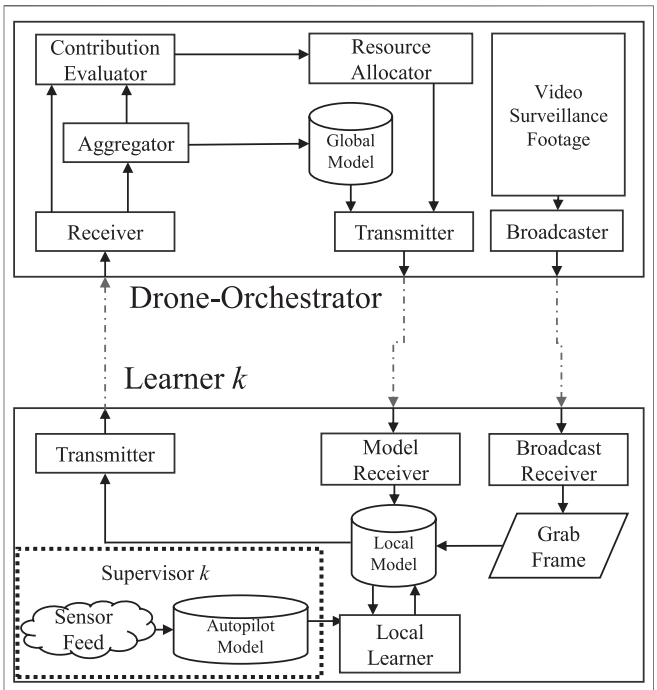

FIGURE 2 | System model illustration.

1.3 Main Contributions

In this paper, we provide a novel perspective on continuous DTM improvements through an FL implementation onto vehicle-learners. Moreover, we aim to provide a robust and adaptable resource allocation method for improved FL performance in the presence of chaotic, quickly changing, and most importantly imbalanced and non-IID data. Since both computational and data bias cannot be analytically extracted before sampling the ML model received from each learner, we assume heuristic measures such as maximizing the epochs computed, or equalizing the epochs computed across the learners. Moreover, the core

contribution of this work is a dynamic resource allocation method based on each learner's past contributions. To provide full compatibility with heterogeneous learners and non-IID data, we employ these methods in combination with the FedProx algorithm. Finally, we developed an experimental analysis in which the performance is evaluated through its capability to learn an underrepresented class of the dataset, while also balancing overall system accuracy.

The paper organization goes as follows. **Section 2** introduces the learning setup and the communications resource allocation setup. **Section 3** defines the optimization problem and lists several static and reactive heuristic measures for improving the learning performance, and introduces the learner contribution calculations. This is followed by **Section 4** where the experimental setup and the results from the setup are presented. The final, **Section 5** summarizes the outcomes and discusses future directions.

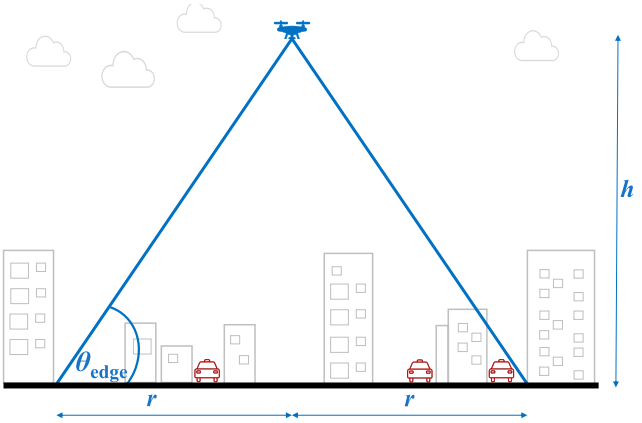

FIGURE 3 | Illustration of the drone position and geometry, in the communications setting.

2 SYSTEM MODEL

The setup is depicted in **Figure 2**, where we show the orchestrator block that sends and receives the models through wireless connections, while simultaneously broadcasts the unsupervized video surveillance footage at a constant data rate for all vehicles inside the MA. We assume that each vehicle acts as an ideal supervisor for the objects which are represented both in the broadcasted video and their sensor feed. Given some deadline of completion T, the learner needs to return its locally learnt model to the drone-orchestrator. After receiving the model, the orchestrator aggregates the K models, after which it can also evaluate the contribution of each learner separately. Each learner k has a contribution, that the contribution estimator estimates to be $G_{k,i}$, for some FL cycle/round i. Finally, the orchestrator contains a resource allocator module that based on the aforementioned information can readjust the wireless resources for the next round, in a way that it improves the FL process.

2.1 Federated Learning

The FL process starts when the orchestrator sends its weights to all K learners, where each learner $k \in \mathcal{K} = \{1, 2, .., K\}$ is present in the MA. The goal of FL methods (Konečnỳ et al., 2016) is to coordinate the optimization of a single global learning objective $\min_{\omega} f(\omega)$, where the function $f()$ is calculated across the whole network at each round i as:

$$f(\omega) = \sum_{k}^{K} p_k F_k(\omega) = \mathbb{E}[F_k(\omega)], \tag{1}$$

where ω are the instantaneous value of the local model weights, $F_k(\omega)$ is the local optimization function at each node, $p_k \geq 0$ and $\sum_k p_k = 1$ is the averaging weight when aggregating. In a single FL round $i \in \mathbb{Z}^+$, a server, i.e. the DTM-orchestrator, has a global model with weights $\omega_{g,i}$. On round i each kth learner receives the model and computes $\tau_{k,i}$ epochs of solving the local optimization function $F_k()$, with data batches of size B. Each batch represents a sample of items that have been sensed and collected from that learner's surroundings. The distributed training process produces a new set of weights $\omega_{k,i}$ at each k that totals to K different ML models. Hence, cycle i concludes when all $\omega_{k,i}$ are aggregated to a signle set of weights $\omega_{g,i+1}$, that serve as the collective model for the next iteration. The two most prominent approaches to solve the FL problem are Fedavg (Konečnỳ et al., 2016) and Fedprox (Li et al., 2018) and differentiate mainly in the local optimization problem $F_k()$ at each device.

Using stochastic gradient descent (SGD) as a local solver $F_k()$, federated averaging (FedAvg) locks the amount of local epochs for each device to a fixed value. As such, each learner is fixed on computing the same $F_k()$ with the same learning rate of SGD for the same amount of epochs. For the successful operation of this system, it is essential to tune the optimization hyperparameters properly including the amount of epochs. The tradeoff in FedAvg becomes one of computation and communication since computing more local epochs reduces communication overhead at the expense of diversifying the local objectives as each system converges to a local optima given their portion of the non-IID data.

Due to the expected heterogeneity in the network of learners in the proposed FL implementation, we use the FedProx algorithm. The benefit of FedProx is that it can converge and provide good general models even under partial work and very dissimilar amounts of $\tau_{k,i}$. This is done by introducing a proximal term $\left\|\omega - \omega_{g,i}\right\|$ that alleviates the negative impact of the heterogeneity as:

$$F_k\left(\omega; \omega_{g,i}\right) = L_k(\omega) + \frac{\mu}{2}\left\|\omega - \omega_{g,i}\right\|, \tag{2}$$

where ω is the instantaneous value of the local model weights at the local optimizer, $L_k(\omega)$ is a local cost function for the estimation losses, μ is a hyperparameter controlling the impact of the proximal term. The role of the proximal term here is that it prevents the local optimiser from straying far from the global model at round i. Moreover, we can control the local optimization problem to vary from a FedAvg ($\mu = 0$) to FedProx ($\mu > 0$). We

note that even when using Fedprox, too much local work causes the local optimizers to diverge from the global objective (Li et al., 2018). Finally, using **(2)** for minimizing the local sub-problem $\min_{\omega} F_k(\omega; \omega_{g,i})$ the FL converges to a solution even in the presence of heterogeneity and non-IID data distribution (Li et al., 2018). Therefore, we use the FedProx algorithm to allow for full flexibility in data and processing heterogeneity, in combination with the resource allocation module that follows.

2.2 Allocation of Wireless Resources

Though the work of (Chen et al., 2020a) covers a detailed cellular model for FL connectivity, drone provided connectivity is generally uniform and can be designed to be predominantly line of sight (Babu et al., 2020). As we illustrate in **Figure 3** the drone height h and the projected coverage on the ground with radius r impact the elevation angle at the edge of the MA, θ_{edge}. The steepness of the elevation can be derived from the evironmental parameters while also accounting for the directivity of the antenna mounted on the drone, as in (Donevski and Nielsen, 2020), and the service reliability that needs to be achieved (Donevski et al., 2021b).

Since our goal of a DMT implementation is to improve the worst case performance of autonomous traffic, we also model the communications system through θ_{edge} as a worst case design parameter. θ_{edge} is decided upon deployment as it plays an important role of controlling the likelihood of establishing line of sight with the ground vehicles at the edge of the cell as in:

$$\mathrm{P}_{\mathrm{DLoS}} = \frac{1}{1 + a\exp\left(-b\left(\theta_{\text{edge}} - a\right)\right)}, \tag{3}$$

where a and b are constants defined by the propagation topology of the environment, as given by (Al-Hourani et al., 2014). Through θ_{edge} in **(3)** a system designer controls not only the probability of detecting a CO but also the average quality of the communications channel at the edge of the MA as:

$$\Lambda = L_{\mathrm{LoS}} \cdot \mathrm{P}_{\mathrm{DLoS}} + L_{\mathrm{NLoS}} \cdot (1 - \mathrm{P}_{\mathrm{DLoS}}), \tag{4}$$

where L_{LoS} and L_{NLoS} are the pathloss coefficients when LOS is established or lost, respectively. As such, we arrive to the average rate for the user located at the edge of the cell by:

$$R_{\text{avg}} = \log_2\left(1 + \frac{P_{\text{tx}}}{N\Lambda}\right), \tag{5}$$

where P_{tx} is the transmission power, and N is the noise power. As FL model transmissions usually take several seconds depending on the size of the model, we omit small scale fading as an impactful factor in the analysis and assume that the drone provided links are symmetrical in both directions and offer each learner k a rate of $\frac{W}{K} \cdot R_{\text{avg}}$, where W is the total bandwidth dedicated for the FL model passing. W may be represented as discrete resource blocks or a band of spectrum that is left over after portioning part of it for the purpose of video broadcasting. Like this, R_{avg} acts as a lower bound guarantee for the amount of time spent learning at each ground device.

As the size of the processing batch is fixed to B, each device k is tasked with an equal number of floating point operations (FLO) for each epoch, and computes $\tau_{k,i}$ epochs. However, for each learner k we introduce a coefficient f_k that represents the learners' computational power with regards to the model size, and is a unit of amount of epochs computed per unit time. Having full information on f_k is generally trivial since it depends on the processing capabilities of the learner, which should be publicly available in the device specifications.

Given an equal bandwidth allocation to all devices, the total number of epochs is a linear function of f_k. This results in the following equation for $\tau_{k,i}$:

$$\frac{\tau_{k,i}}{f_k} = T - \frac{KD}{WR_{\text{avg}}}, \tag{6}$$

where, D is the total amount of data that needs to be sent in both directions within the deadline of T. We convert the problem to a step-wize nomenclature that gives the relationship between each learner, independent of the length of T but as a relative inter-learner metric:

$$\begin{aligned} \tau_{k,i} - \tau_{l,i} &= Tf_k - \frac{KDf_k}{WR_{\text{avg}}} - Tf_l + \frac{KDf_l}{WR_{\text{avg}}}, \\ \tau_{k,i} - \tau_{l,i} &= T\left(f_k - f_l\right) - \left(f_k - f_l\right)\frac{KD}{WR_{\text{avg}}}, \\ \frac{\tau_{k,i} - \tau_{l,i}}{f_k - f_l} &= T - \frac{KD}{WR_{\text{avg}}}, \end{aligned} \tag{7}$$

where $\forall k, l \in \mathcal{K}, l \neq k$. We then perform the substitution:

$$\begin{aligned} \alpha &= \frac{\tau_{k,i} - \tau_{l,i}}{f_k - f_l},\ \forall k, l \in \mathcal{K}, l \neq k, \\ T &= \alpha + \frac{KD}{WR_{\text{avg}}}, \end{aligned} \tag{8}$$

where α is the nominal time reserved for learning, and it is directly influenced by the amount of FLOPs required to compute one epoch. This simplifies to:

$$\begin{aligned} \frac{\tau_{k,i}}{f_k} &= \alpha + \frac{KD}{WR_{\text{avg}}} - \frac{KD}{S_{k,i}WR_{\text{avg}}}, \\ \frac{\tau_{k,i}}{f_k} &= \alpha + \frac{KD}{WR_{\text{avg}}}\left(\frac{S_{k,i} - 1}{S_{k,i}}\right), \end{aligned} \tag{9}$$

where $S_{k,i} \geq 0$ and $\Sigma_k^K S_{k,i} = K$ is the bandwidth allocation for learner k in round i, represented as the portion of the average spectrum $\frac{W}{K}$ occupied (i.e. $S_{k,i} = K$ is the full spectrum, and $S_{k,i} = 1$ is the average spectrum). We continue with the substitution:

$$\frac{KD}{WR_{\text{avg}}} = \beta, \tag{10}$$

where β is the portion of time spent transmitting within one round. As per β, it is obvious that it is much more important to investigate the ratio of data load on the channel instead of solely focusing on the achieved rate R_{avg}. Moreover, the time spent learning at each device becomes more significant the more we

load the resources, in both number of learners and the size of the model. This results in the final representation of epochs computed for learner k as a function of the bandwidth allocated to them:

$$\tau_{k,i} = f_k\alpha + f_k\beta\left(\frac{S_{k,i}-1}{S_{k,i}}\right), \tag{11}$$

Given a no-drop policy (each learner must complete at least one epoch $\tau \geq 1$), the lower bound on $S_{k,i}$ becomes:

$$S_{\min} = -\frac{\beta f_k}{1-\alpha f_k-\beta f_k}, \tag{12}$$

and the extreme upper bound of $S_{k,i}$ is therefore:

$$S_{\max} = K + \sum_{l}^{K-1} \frac{\beta f_l}{1-\alpha f_l-\beta f_l}, \forall l \in \mathcal{K}, l \neq k. \tag{13}$$

The behavior of the resource function for a single $\tau_{k,i}$ when adjusting β and $S_{k,i}$ within the bounds of **12**, **13**, is:

$$S_{\min} \leq S_{k,i} \leq S_{\max}, \tag{14}$$

The entire communications setup is reducible to the analysis of combinations of α and β, as both parameters directly determine the impact that resource allocation has on the system. Moreover, the parameter β modifies the impact of resource allocation for each learner, where systems with high β values stand to benefit the most, while low β values indicate near instantaneous model transfers which cannot be influenced by modifying the bandwidth. On the other hand, α is a system design hyperparameter that indicates the amount of epochs computed within a single round, by an average learner, and it is fully customizable before or even during operation.

3 ANALYSIS

Our goal is to improve the learning of a particular class among the network of FL devices, that may represent a CO, without harming the overall accuracy of the system. Thus, each round i we exploit our control over the wireless resources and optimize the bandwidth allocated to each device $S_{k,i}$. The vector representation of the bandwidth allocation for each round becomes $\mathrm{S}_i = (S_{1,i}, S_{2,i} \ldots S_{K,i})$. In the same way, the number of epochs computed in round i and the contribution estimations are reformulated into vectors: $\mathrm{T}_i = (\tau_{1,i}, \tau_{2,i} \ldots \tau_{K,i})$ and $\mathrm{G}_i = (G_{1,i}, G_{2,i} \ldots G_{K,i})$ respectively, where $G_{k,i}$ is an estimate of the contribution of learner k based of its learning performance in the past. Due to the rapidly changing environment around each learner, we cannot assume having information about the size or distribution of the data stored at each learner. Therefore, we can assume a function of utility from both aforementioned parameters $h_{\mathrm{X}}(\tau_i, \mathrm{G}_i)$, where X is a placeholder for the name of the approach. Given this function, the optimization problem of maximizing the utility X can be defined as:

$$\begin{aligned} \max_{\mathrm{S}_i} \quad & h_{\mathrm{X}}(\mathrm{T}_i, \mathrm{G}_i), \\ & \sum_{k}^{K} S_{k,i} = K, \\ & \tau_{k,i} \in \mathbb{Z}^+, \\ & (11), (12), (13), (14). \end{aligned} \tag{15}$$

Extracting the direct impact of $G_{k,i}$ and τ_i onto the future accuracy of the model, and under non-IID data distribution, is non-trivial and hence requires that we form several heuristic functions for $h_{\mathrm{X}}()$ to be tested on an experimental setup. Therefore we compare three different solutions for (15) by swapping the utility function $h_{\mathrm{X}}()$ with the ones named as $X \in \{\mathrm{MAX}, \mathrm{AAS}, \mathrm{ACT}\}$. The first two versions of the optimization problem (MAX and AAS) apply a static method that computes utility only as a function of the epochs that will be computed for that round for each learner. The third approach (ACT) is a novel reactive method, that extracts the utility of a learning round as a product of the estimated contribution by each learner and the epochs that will be computed by that learner. The details for each method follow below.

3.1 Static Resource Allocation Measures

The naive way of improving the convergence in a heterogeneous setting is maximizing the total amount of work done by all learners as in:

$$h_{\mathrm{MAX}}(\mathrm{T}_i, 0) = \sum_{k}^{K} \tau_{k,i}. \tag{16}$$

This optimization criteria maximizes the epochs computed across the whole network given the limited radio resources. Since **Eq. 16** implies asyncronous amount of work performed among the learners, it may not be considered as a potential maximization metric when using classical FedAvg implementations. However, since we use FedProx as a local optimizer, this is a sufficient naive solution that represents an exploitative behavior from the orchestrator.

Furthermore, given the work on asynchronous FL and the issues of diverse computational hardware in the network (Xie et al., 2019; Mohammad et al., 2020) we identify maximum *staleness* (Donevski et al., 2021a) as an important criterion toward the precision of the model. We define this as the maximal difference between the fastest and slowest learner:

$$s = \max\left(\left|\tau_{k,i} - \tau_{l,i}\right|\right) \forall k, l \in \mathcal{K}, l \neq k. \tag{17}$$

Nonetheless, minimizing staleness does not extract the full potential of our setup. Therefore, as in (Donevski et al., 2021b) we convene s and the average of the anticipated epochs to a more balanced heuristic metric, named Average Anchored Staleness (AAS) as an optimization metric:

$$h_{\mathrm{AAS}}(\mathrm{T}_i, 0) = \frac{1}{K}\sum_{k}^{K} \tau_{k,i} - s. \tag{18}$$

AAS gives a good general overview that is data-agnostic, without the need to assume the impact of data at some

particular learner and solely on spatial and computational performance. Like this, AAS provides a resource allocation objective function that serves an equally balanced amount of learning and *staleness*.

3.2 Contribution Estimation for Reactive Resource Allocation

In the case of DTMs, the considered vehicle supervisors/learners can find themselves in the presence of vastly different objects, and the data they sense changes constantly while they operate. Given the aforementioned, the contribution of each learner is hard to estimate especially in the presence of noisy samples. Hence, we assume that separating the important CO information ahead of time is impossible and only consider reactive approaches such as incentive mechanisms. To use incentive mechanisms we must assume that the validation dataset that is present at the orchestrator has equal representation of all classes. Hence, based on such validation data we can pass the weights ω through an evaluation function $E(\omega)$ which can be based on accuracy or loss evaluations of the model **(we choose accuracy)**. To calculate the contribution for each round i we define:

$$G_{k,i} = \frac{E\left(\omega_{g,i}\right) - E\left(\omega_{g\setminus\{k\},i}\right)}{\Sigma_k^K \left|E\left(\omega_{g,i}\right) - E\left(\omega_{g\setminus\{k\},i}\right)\right|}, \tag{19}$$

where $\omega_{g\setminus\{k\},i}$ is a model aggregator that constructs a new model that is an aggregate of all recieved models except the one of k. Hence the difference in accuracy between the fully aggregated model and the $\omega_{g\setminus\{k\},i}$ (Nishio et al., 2020) gives the added value (the uniqueness) of the learning done by learner k. Like this, the contribution estimator is capable of discovering the overall contribution from each learner for that round, without the capability of sampling for contributions on each detection class separately, or discern which object is underrepresented or is the CO. This is a central feature of our method, since we aim to improve CO learning without tailoring the solution to discern which class is the CO.

We note that the $\omega_{g\setminus\{k\},i}$ function needs to be called for each learner in order to produce K different contribution estimations. In addition to having to compute an additional parameter, there is one extra set of weights that needs to be aggregated for the calculation of $\omega_{g\setminus\{k\},i}$ for all other learners, thus making the complexity of the estimator scale as a square of the number of nodes in system K. Even though the computational complexity of this technique can escalate in big FL implementations, in the architecture that we propose there should be several active learners inside the MA. Thus, even aside the limited computational power on the drone, the estimator module should not experience lengthy computational times.

Following the first round, each device k provides its model to the DTM-orchestrator. After which, the aggregator provides the first aggregate model weights $\omega_{g,i}$. The resource allocator module in the orchestrator receives the contributions for each of the participating learners and hence can decide to adjust the resources based on $G_{k,i}$. Since $G_{k,i}$ is an estimation of the contributions for the past round, the goal is to maximize the total contribution of the upcoming round by introducing the following optimization function:

$$h_{\mathrm{ACT}}(\mathrm{T}_i, \mathrm{G}_i) = \sum_k^K \tau_{k,i} g\left(G_{k,i}\right), \tag{20}$$

where $g()$ is a utility function that scales the contributions to match the impact of the number of computed epochs. Introducing a utility function is necessary to properly scale each learner's impact since $-1 \le G_{k,i} \le 1$ and $\tau_{k,i} \in \mathbb{Z}^+$. Since in an average scenario $\mathbb{E}[\tau_{k,i}] = \alpha\mathbb{E}[f_k]$, and $\mathbb{E}[f_k] = 1$ we scale our utility function as per the average epochs computed for that round as $g(G_{k,i}) = \alpha^{G_{k,i}}$. The bounds of the function become $1/\alpha \le g(G_{k,i}) \le \alpha$, and the nominal non-contributive learners produce $g(0) = 1$. Thus the heuristic exponential optimization function for the reactive solution can be calculated as the contribution corrected maximum epochs computed as in:

$$h_{\mathrm{ACT}}(\mathrm{T}_i, \mathrm{G}_i) = \sum_k^K \tau_{k,i} \alpha^{G_{k,i}}. \tag{21}$$

In the case of constantly equal contributions from all learners, the heuristic maximization criteria is reduced to the epoch maximization problem defined in **(16)**. With h_{ACT} defined as in **(21)** we maintain the problem within the bounds of mixed integer linear programming since the utility is applied only to $G_{k,i}$ that remains constant for the whole round i.

TABLE 2 | The non-IID distribution of data among learners, and their computational coefficients f_k.

Learner	Classes stored (out of 0–9)	f_k
1	3, 4, 5, 6	0.15
2	0, 1, 2, 3, 4	0.7
3	4, 6, 7, 8, 9	1.0
4	0, 1, 2, 6, 7, 8, 9	1.3
5	0, 1, 2, 7, 8, 9	1.3
6	0, 1, 2, 7, 8, 9	1.0
7	3, 4, 6	0.7

4 RESULTS

4.1 Experimental Setup

For a set of learners that are scattered along the MA, our goal is to as closely as possible generate an experimental setup that simulates a realistic learner given the system model in **Section 2**. Since each learner has a very short amount of time to do the learning for the DTM, we approach the data as fleeting (stored very briefly) and concealed (cannot be known beforehand). Due to the complexity and the issues of reliably simulating the FL performance for full scale traffic footage, we test the performance of the proposed methods through simple and easily accessible

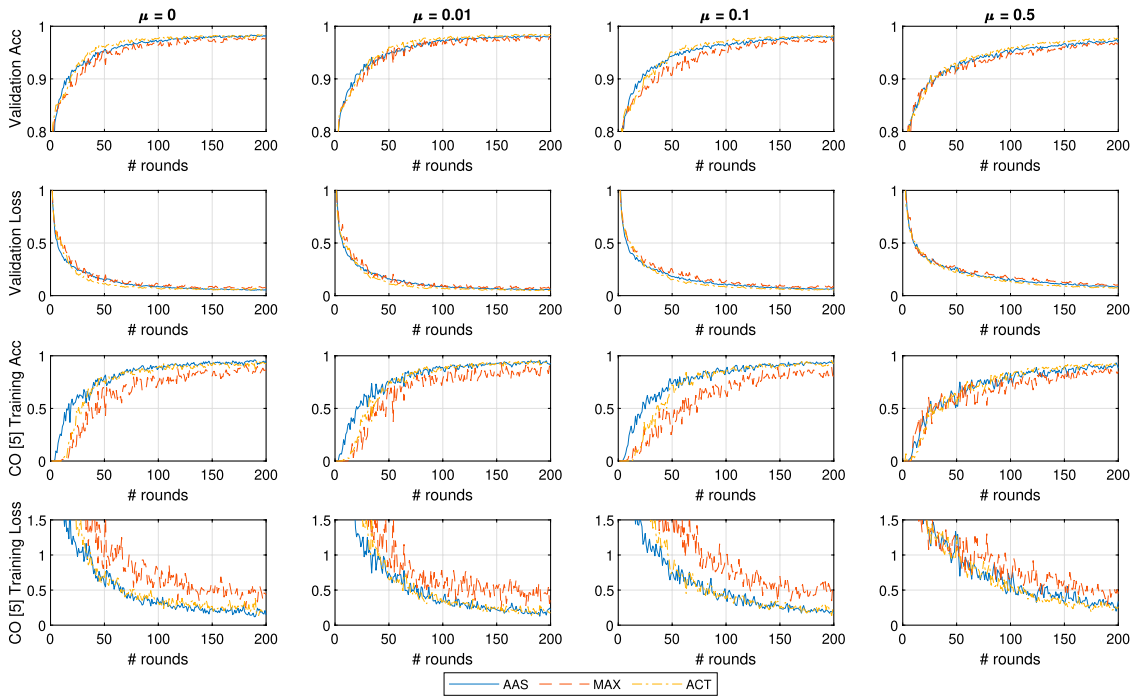

FIGURE 4 | General and CO-specific accuracy and loss results obtained when testing all three methods in combination with FedProx using the MNIST dataset.

computer vision datasets. Each testing scenario was built using either the MNIST dataset (LeCun et al., 1995) of handwritten digits, or the FMNIST (Xiao et al., 2017) dataset consisting of 10 different grayscale icons of fashion accessories.

As we expect that each vehicle contains strongly non-IID data we create a custom data distribution among $K = 7$ learners as shown in **Table 2**. In addition, the processing power for computing a certain amount of epochs per millisecond f_k for each learner, is distributed as: two standard vehicles ($f_k = 1$), two premium vehicles ($f_k = 1.3$), and two budget vehicles ($f_k = 0.7$); with the addition of one straggler that contains an older technology ($f_k = 0.15$). At each epoch the learner samples a single batch of $B = 16$ randomly selected values from the stored data (as per **Table 2**). Like this, the training data changes constantly, to mimic the changing environment of the vehicular scenario. This makes this FL testing scenario unique in that the number of epochs computed also reflects the amount of data sampled from the environment.

In the described setting, the class-number 5 (sixth class counting from zero) assumes the role of a CO. In addition to the CO, class-number three is another non-CO class that is not too common and appears at only three learners. This is an over-exaggerated situation of having the CO data hidden at one node that is also a straggler. We expect this to be a realistic reflection of data in drone orchestrated FLs as nodes carry only a small amount of supervisory data for each class due to the fact that they stumble upon important objects randomly.

For detection, we implement a small convolutional neural network (CNN), common for the global and local models implemented in python tensorflow (Abadi et al., 2015). In more detail, the CNN has only one 3×3 layer of 64 channels using the rectifier linear unit (ReLU), that goes to a 2×2 polling layer. A dense, fully connected neural network (NN) layer of 64

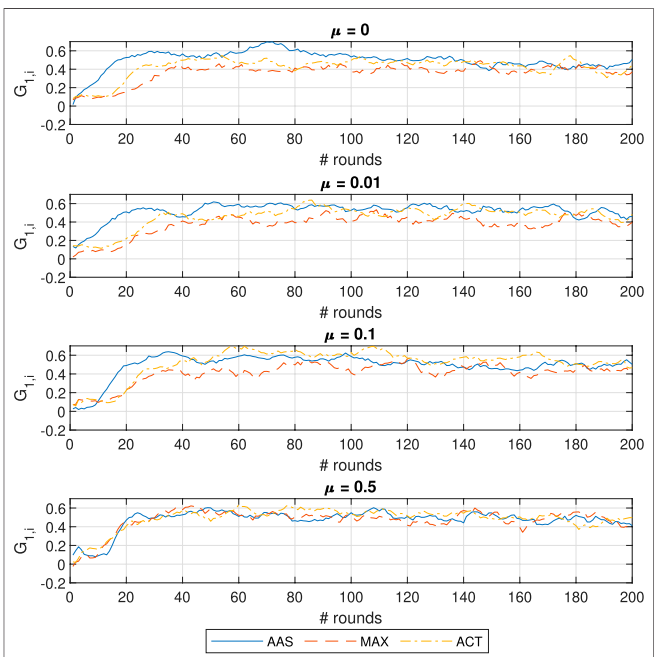

FIGURE 5 | Contribution evolution for learner $k = 1$ in the case of using the MNIST dataset.

ReLU activated neurons receives the polled outputs of the convolutional layer, which is then fully connected to a NN layer of 10 soft-max activated neurons, one for each of the 10 categories of the NIST dataset. The local optimizer at each learner is given by the FedProx calculation in **Eq. 2**, where the cost function $L_k()$ is a categorical cross-entropy loss function, and the learning rate performed well when fixed to $\gamma = 0.1$. The communication phase coefficient was considered in milliseconds and chosen as $\beta = 100$ considering our CNN model with a size of 2.5 Mb that needs to be transmitted to all seven learners, over a single $W = 80$MHz 802.11ax channel. Finally, in the reference frame of milliseconds, the cycle duration coefficient was set to $\alpha = 100$ in favor of allowing for higher flexibility when scaling the bandwidth allocation.

4.2 MNIST Testing

We proceed with the testing of all three approaches for five different values of the proximal importance hyperparameter $\mu \in \{0, 0.01, 0.1, 0.5\}$, as guided by the recommended values in (Li et al., 2018). μ values larger than 0.5 failed to produce productive results and only harmed the convergence outlook. The testing lasts for 200 rounds on the aforementioned CNN model. Aside the three shown FL implementations, we also implement a classical ML with only one learner that contains all the data. We do this to extract the performance ceiling of the NN approach, which is 98% for the validation accuracy and 0.0602 validation loss paired with training accuracy of 98.85% and training loss 0.0423.

In **Figure 4** we can notice a limited impact of changing the μ parameter of FedProx, most likely due to the small amount of learners and not as significant straggler impact. This is expected given that (Li et al., 2018) claim strong superiority over FedAvg in the cases of very large portions of stragglers. Interestingly, μ does not have a strong positive impact on the learning performance even in the case of MAX, and therefore, a system designer would most likely introduce a weak proximal term of $\mu = 0.01$. Additionally, using the ACT approach provides superior convergence, and in combination with $\mu = 0.01$ achieves the best overall accuracy. In addition to this, the ACT and $\mu = 0.01$ combination also keeps up with the performance of AAS with regards to the CO class after the first several rounds of convergence.

To better investigate the behavior of the ACT approach we illustrate the evolution of the estimated contributions for learner $k = 1$ in **Figure 5**, where $G_{1,i}$ is based on the performance of the learner estimated from the previous learning round as in **Eq. 19**. The overall conclusion here is that we achieve CO learning without tailoring the solution to discern which class is the CO. This is possible as the calculation of $G_{k,i}$ is focused around the uniqueness of the dataset at each learner. Here we can notice that increasing the strength of the proximal parameter through setting higher μ values equalizes the contributions between all three methods, particularly in the first 40 rounds. Moreover, when $\mu = 0.5$ the contributions are stabilized and vary very little once the initial phase of 40 rounds.

Most notably, the accuracy of AAS suffers significantly when $\mu = 0.5$ which results in a performance that is equally matched to the MAX approach when detecting the CO. It is thus evident that a strong FedProx implementation harms total system accuracy, and above all, diminishes the impact of the using resource allocation. Finally, we conclude that the task of learning MNIST is too simplistic for our assumed scenario of traffic monitoring, and thus we continue with testing the FMNIST dataset in the following subsection.

4.3 FMNIST Testing

Since modeling common tasks of computer vision on MNIST is a very easy task we repeat the test on the FMNIST dataset. This dataset consists of 10 classes of fashion accessories in equal distribution as the MNIST dataset (a training set of 60,000 examples and a test set of 10,000 examples) and as in the case of MNIST consists of 28×28 grayscale images. The dataset classes are: (0) T-shirt/top, 1) Trouser, 2) Pullover, 3) Dress, 4) Coat, 5) Sandal, 6) Shirt, 7) Sneaker, 8) Bag, and 9) Ankle boot; where each item is taken from a fashion article posted on Zalando. Compared to the number MNIST, in FMNIST the intensity of each voxel plays a much bigger role and is scattered across larger parts of the image. We consider the FMNIST dataset as a computer vision task that sufficiently replicates the problem of detecting 10 different types of vehicles, in a much more simplistic context that is furthermore easily replicable.

In **Figure 6**, we show the learning performance in the same setting and $\mu \in \{0, 0.01, 0.1, 0.5\}$, across 200 rounds of training. It is most obvious that the overall accuracy has dropped quite a lot from the 98% in the MNIST case to 88% in the best case scenario of ACT with $\mu = 0.01$ for the FMNIST. Most notably the largest difference is that the increased difficulty of the learning problem introduces a lot more noise in the learning process, particularly for the CO class. Due to this, when using no FedProx ($\mu = 0$) AAS does a good job at accelerating the learning process in the first 20 rounds until it is overtaken by ACT. Even though the combination of ACT with $\mu = 0.01$ shows the best overall accuracy on the validation data, the accuracy of detecting the CO class with ACT never truly reaches the performance of AAS.

Finally, we conclude that even though $\mu = 0.1$ and $\mu = 0.5$ were eligible in the MNIST run, the overall increased complexity of FMNIST harms the accuracy outlook in both, but with the most severe impact on AAS. This experimental run therefore inspired us to investigate the issue of underfitting, and we proceed with testing FMNIST performance with a deeper model.

4.4 Deeper FMNIST Testing

In this testing scenario we expand the small convolutional neural network by adding another 3×3 layer of 64 channels using ReLU activators as a first layer. In **Figure 7** we show the outcomes of the testing, where the overall accuracy of the system has been improved to 90%. However, the larger model acted as an equalizer across all three approaches and in the case of $\mu = 0$ generally gave equal performance both in convergence time and overall accuracy. It is important to also look at the validation loss following the round $i = 150$ as it starts to diverge for both ACT and MAX approaches. This did not directly map into the accuracy of the detection, but nonetheless is a first sign of possible overfitting and eventual divergence.

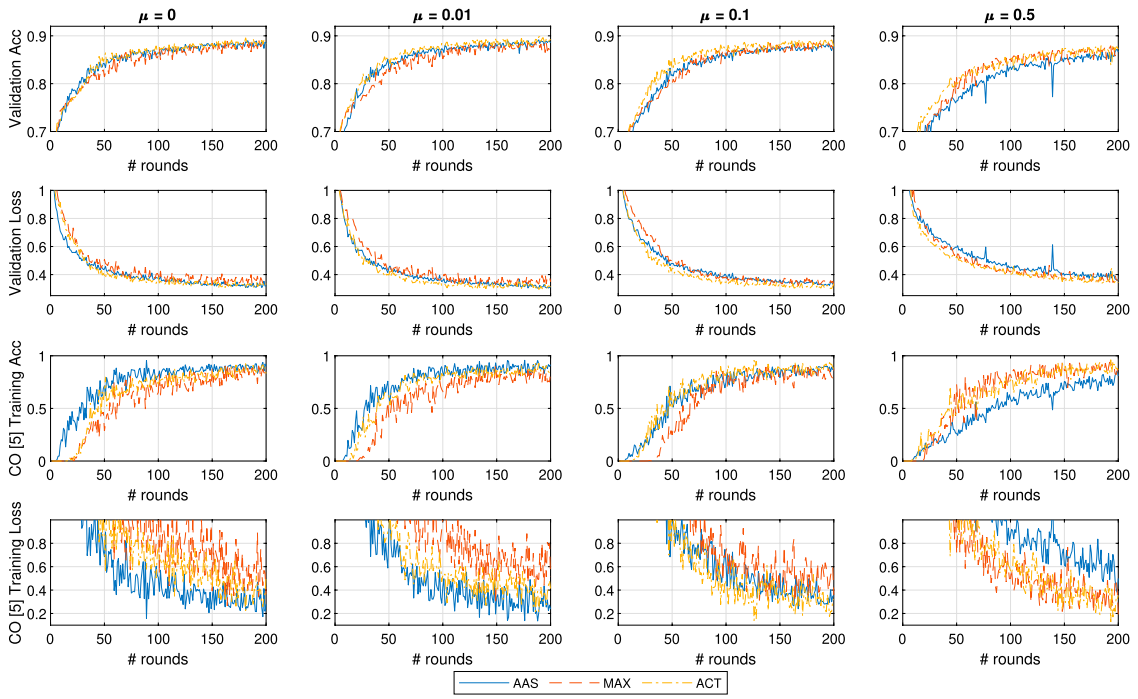

FIGURE 6 | General and CO-specific accuracy and loss results obtained when testing all three methods in combination with FedProx using the FMNIST dataset.

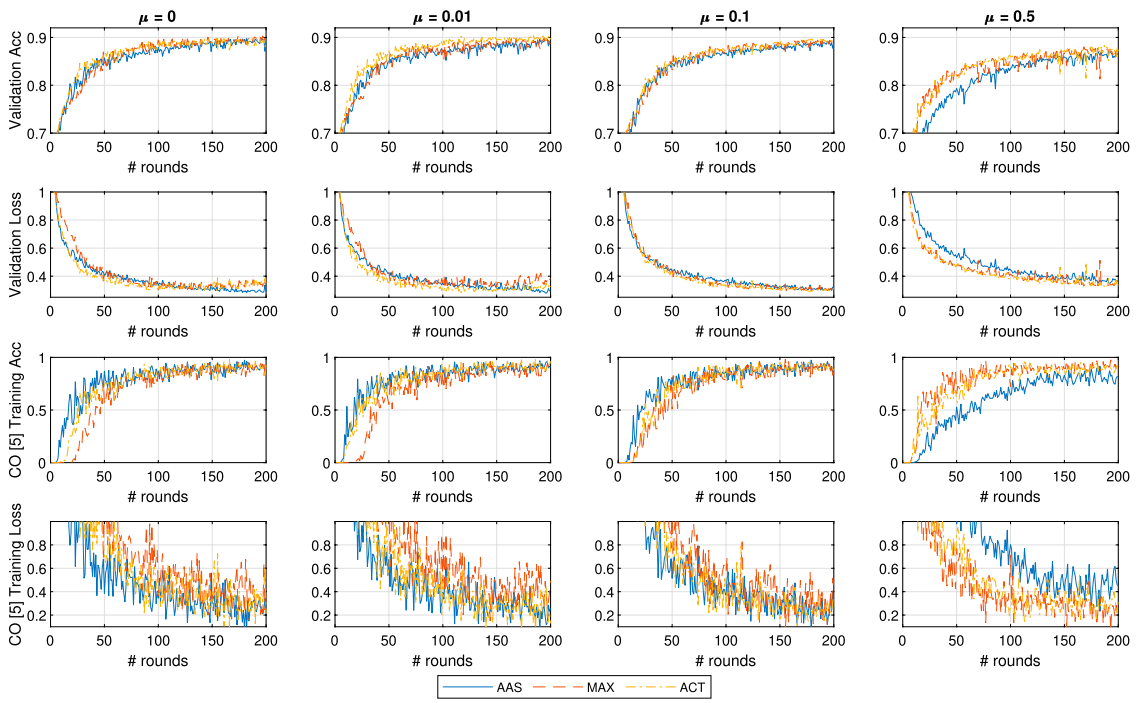

FIGURE 7 | General and CO-specific accuracy and loss results obtained when testing, with an extra CNN layer, all three methods in combination with FedProx using the FMNIST dataset.

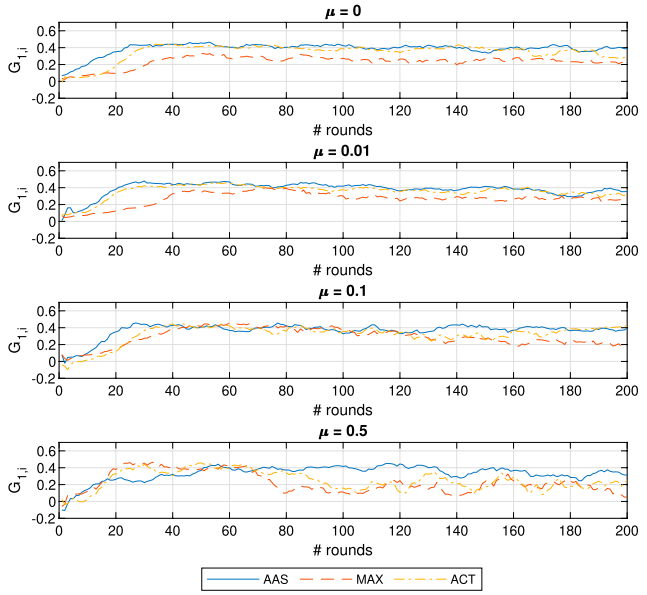

FIGURE 8 | Contribution evolution for learner $k = 1$ in the case of using the FMNIST dataset with an extra CNN layer.

With the deep model, this effect is diminished for the case of ACT with $\mu = 0.01$, and manages to reach the best convergence time along with overall accuracy from all tested implementations. This accuracy is also paired with improved detection of the CO that exactly matches the AAS approach. As such the ACT with $\mu = 0.01$ is both the best overall learning solution, but also the best CO detector.

It is also interesting to notice that the MAX approach does well with overall accuracy, particularly when compared to the inferior performance in the previous testing sets. Nonetheless, MAX is still inferior to both other approaches when it comes to detecting the CO class. Finally, we focus on the results on $\mu = 0.5$. When the proximal term has such a strong impact on the learning, all three approaches show inferior overall performance by 4-5 percentage points with regards to the best performing $\mu = 0.01$. However, it is interesting to see that the impact is by far most severe on the AAS approach, even reducing the CO detection performance. Additionally, MAX gives the best result when it comes to learning the CO behavior for $\mu = 0.5$. Opposed to the behavior back in the MNIST testing, here AAS suffers from the increased complexity of the task, and in combination with a very strong proximal term reduces the overall learning of detection. This makes it is easy to conclude that a strong proximal term reduces the effect of resource allocation efforts.

We seek to discover the culprit for the inferiority of AAS in CO discovery when $\mu = 0.5$ by plotting the contributions of learner $k = 1$ in **Figure 8**. Looking at the contribution evolution in case $\mu = 0.5$ we extrapolate that AAS aims to keep the learner relevant while the reduced amount of learning across the whole network harms the potential contribution of all other nodes. This leads us to the final conclusion of this experiment which is that the ACT based approach is extremely versatile in providing good CO detection and accuracy even in the cases of $\mu = 0$, a properly assigned μ, and overly restricted FedProx implementation.

4.5 Testing Fleeting FMNIST

The final test with the experimental setup is constructed such that we introduce stress in the learning process by introducing temporary losses in the supervision process. This is done by introducing a likelihood that a learner k loses access to a detection class. This would be representative of a learner losing LOS of the object was able to supervise, and is therefore modeled as a two state markov model (such as the Gilbert Elliot (Boban et al., 2016)) that has a good and a bad state. Hence each supervisor has $p = 0.9$ chance to maintain supervision for that class (stay in the good state), and $1 - p = 0.1$ probability to lose supervision capability (and move to the bad state). If the vehicle loses supervision capabilities for that class, it has $r = 0.5$ probability to maintain that state (remain in the bad state) or $1 - r = 0.5$ probability to regain supervision of that class. The values for the state transitions in the Gilbert-Elliot model were chosen with the experimental setup in mind so that not too much data is lost with regards to the previous testing setups. These testing parameters were provisioned arbitrarily, because higher values would make the learning process very lengthy imposing unrealistic testing times for our experiment, but still provide a lot of stress to the learning system.

Hence, to compensate for the smaller dataset, we let the simulations run for 250 rounds, and focus only on $\mu \in \{0.01, 0.1\}$. The fleeting data is provided from the same seed and the Gilbert Elliot model starts from the good state for every possible detection combination. In **Figure 9** we show the performance of all approaches on the aforementioned setup. Comparing this to the previous testing setup, we notice that the overall accuracy dropped by 1 percentage point for $\mu = 0.01$ and 2 percentage points when $\mu = 0.1$ due to the increased stress in the learning process. It is also apparent that both ACT and MAX show signs of overfitting – the diverging lines in the validation loss – which is improved when using $\mu = 0.1$, at the cost of reducing the overall system accuracy by an additional 1 percentage point.

Focusing on $\mu = 0.01$, all methods achieve nearly the same overall accuracy, since the learning of the computer vision task is bottlenecked by the presence of the data. However, AAS is superior in CO detection and it shows slightly inferior convergence time for overall accuracy (i.e. around the 50 round mark). In addition to this, AAS is the most data sensitive approach and experiences the largest overall accuracy dips in situations where many detection classes are in the bad state (such as around the 55th round and the 127th round). Finally, to better observe the noisy training data, we plot a 10-point moving average in **Figure 10**. Here we notice the in the common training scenario AAS and ACT perform rather equally when learning hidden information. However, in the presence of fleeting data, the ACT performance becomes very noisy and becomse slightly inferior than AAS with regards to CO learning performance. Nonetheless, as already mentioned, this CO learning performance of the AAS approach comes at a slight cost of general detection performance, in both fleeting and normal setting.

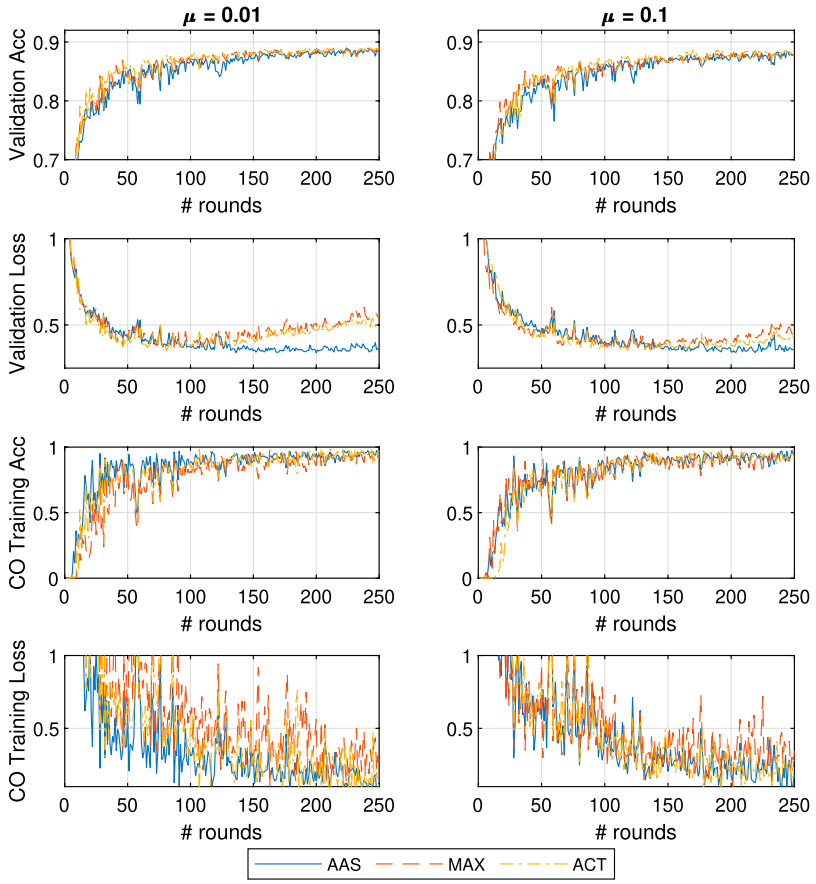

FIGURE 9 | General and CO-specific accuracy and loss results obtained when testing, with an extra CNN layer, all three methods in combination with FedProx using the FMNIST in the case of fleeting data.

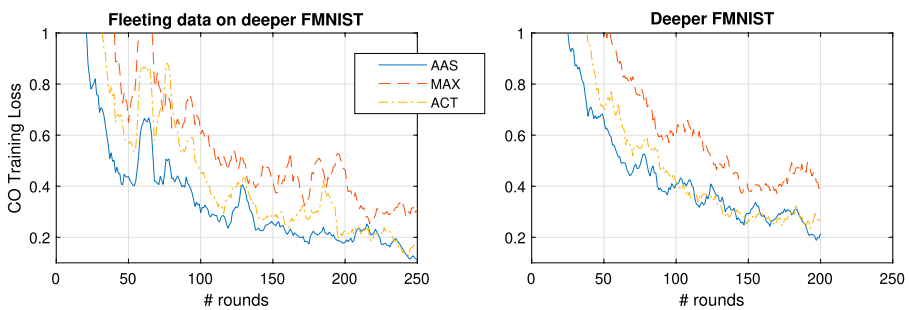

FIGURE 10 | 10-point moving average of CO training loss for $\mu = 0.01$ of the fleeting data vs. normal data sampling in the deeper FMNIST testcase.

4.6 Key Takeaways

We condense several takeaways that were derived from all four experimental runs. The initial and most important conclusion is that the concepts of resource allocation and FedProx are at odds in the case of FL implementations. In more detail, the goal of FedProx is to reduce the impact of each learner individually while resource allocation methods strive to improve the overall performance by exploiting or compensating the heterogeneity of the system. Hence the impact of resource allocation methods is diminished when strengthening the role of the proximal term. Nonetheless, in the many tests a safe balance between both μ and resource allocation ensure good learning behavior. As such, we recommend that all future works consider perturbed gradient descent implementations, such as FedProx, when dealing with non-IID data in heterogeneous FL.

Additionally, in the initial testing of our setup we noticed that testing on MNIST is not sufficient to provide reasonable results for the implementations, due to how trivial the task of recognizing digits is. Moreover, FL implementations, such as the proposed drone implementation, are based in the distributed learning of complex tasks and require deeper NN models. In such cases, it was evident that increasing the total amount of computed epochs benefits the convergence time of the system with potentially harmful effects in CO detection accuracy. Moreover, deeper model implementations did not behave well under strong proximal terms.

As a consequence to this, learning hidden data can be addressed by equalizing the contributions by using AAS or by introducing strong proximal terms. However, the strong proximal terms have potential to slow down the convergence time for all nodes. Hence, the safest implementation to achieving the best combination of convergence time, overall accuracy and CO learning rate is using the ACT approach with a weak proximal term.

Finally, in a case where the data is fleeting, using a $\mu > 0$ was crucial to reach stable learning performance. In this setting, the low availability of data acted as a lower bound for all learning implementations, but most importantly harms the convergence time performance of AAS. This is understandable since AAS was the approach that cumulatively computed the least amount of epochs at each round. On the other hand, the ACT approach maintained superior performance to both static approaches by maintaining good CO detection performance and great convergence times.

Finally, we extrapolate that defining a proper μ is cardinal. However, the hyperparameter needs to be defined ahead of the deployment of the system. As such, since we would not have access to the training data, the feasibility of implementing AAS is uncertain especially for situations where the presence of data changes quickly. This gives another strong motivation for using reactive measures based on contributions and incentive calculations, such as ACT.

5 CONCLUSION

In this paper we investigated the learning process in a novel Federated Learning (FL) architecture, where a DTM acts as an orchestrator and traffic participants act as supervisors on its model. Such an implementation expects impairments on the learning process due to unbalanced and non-IID data scattered across heterogeneous learners that have variable computational equipment. We therefore test the ability of two static methods (AAS and MAX), and one incentive based reactive (ACT) resource allocation method to improve the speed of learning CO classes and maintaining good overall model accuracy. The validity of the methods was tested with an experimental FL implementation that uses the novel FedProx algorithm to learn from the MNIST and FMNIST datasets. The testing was conduced across combinations of different FedProx strength, CNN model depth, and fleeting data. From the testing we conclude that both reactive (ACT) resource allocation and FedProx are essential to securing model accuracy. In more detail, due to the inability to anticipate the distribution of the data across the learners, the use of ACT ensures proper operation of the FL implementation. In accord, the combination of properly set FedProx with an ACT implementation provided faster convergence times, better accuracy, but most importantly it matched the AAS method in learning to recognize the CO. Such behavior was consistent across most runs given the varying task complexity, model size, and data presence. The goal of future works would be to look into more advanced proactive approaches, especially for the presence of imperfect data supervision.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

ID: investigation, writing; JN and PP: writing, review, editing, resources, funding acquisition, supervision, and project administration.

FUNDING

The work was supported by the European Union's research and innovation program under the Marie Sklodowska-Curie grant agreement No. 812991 "PAINLESS" within the Horizon 2020 Program.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Al-Hourani, A., Kandeepan, S., and Jamalipour, A. (2014). "Modeling Air-To-Ground Path Loss for Low Altitude Platforms in Urban Environments," in Proc. of IEEE Global Communications Conference, Austin, TX, USA, October, 2898–2904.

Aledhari, M., Razzak, R., Parizi, R. M., and Saeed, F. (2020). Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Access* 8, 140699–140725. doi:10.1109/ACCESS.2020.3013541

Amiri, M. M., and Gündüz, D. (2020). Federated Learning over Wireless Fading Channels. *IEEE Trans. Wireless Commun.* 19, 3546–3557. doi:10.1109/TWC.2020.2974748

Babu, N., Ntougias, K., Papadias, C. B., and Popovski, P. (2020). "Energy Efficient Altitude Optimization of an Aerial Access Point," in Proc. of IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, UK, September, 1–7. doi:10.1109/PIMRC48278.2020.9217265

Boban, M., Gong, X., and Xu, W. (2016). "Modeling the Evolution of Line-Of-Sight Blockage for V2v Channels," in Proc. 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, Canada, September, 1–7. doi:10.1109/VTCFall.2016.7881090

Chavdarova, T., Baqué, P., Bouquet, S., Maksai, A., Jose, C., Bagautdinov, T., et al. (2018). "Wildtrack: A Multi-Camera Hd Dataset for Dense Unscripted Pedestrian Detection," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition-Fall), Salt Lake City, UT, USA, June, 5030–5039.

[Dataset] Chen, M., Liu, Y., Shen, W., Shen, Y., Tang, P., and Yang, Q. (2020a). *Mechanism Design for Multi-Party Machine Learning*.

Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V., and Cui, S. (2020b). A Joint Learning and Communications Framework for Federated Learning over Wireless Networks. *IEEE Trans. Wireless Commun.* 20 (1), 269-283. doi:10.1109/TWC.2020.3024629

Donevski, I., Babu, N., Nielsen, J. J., Popovski, P., and Saad, W. (2021a). Federated Learning with a Drone Orchestrator: Path Planning for Minimized Staleness. *IEEE Open J. Commun. Soc.* 2, 1000-1014. doi:10.1109/OJCOMS.2021.3072003

Donevski, I., and Nielsen, J. J. (2020). "Dynamic Standalone Drone-Mounted Small Cells," in Proc. of European Conference on Networks and Communications, Dubrovnik, Croatia, June (Dubrovnik, Croatia: EuCNC), 342–347. doi:10.1109/EuCNC48522.2020.9200918

Donevski, I., Nielsen, J. J., and Popovski, P. (2021b). "Standalone Deployment of a Dynamic Drone Cell for Wireless Connectivity of Two Services," in Proc. of IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, April (Nanjing, China: TBP).

Kang, J., Xiong, Z., Niyato, D., Xie, S., and Zhang, J. (2019a). Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory. *IEEE Internet Things J.* 6, 10700–10714. doi:10.1109/jiot.2019.2940820

Kang, J., Xiong, Z., Niyato, D., Yu, H., Liang, Y.-C., and Kim, D. I. (2019b). "Incentive Design for Efficient Federated Learning in mobile Networks: A Contract Theory Approach," in Proc. of 2019 IEEE VTS Asia Pacific Wireless Communications Symposium, Singapore, August (Singapore: APWCS), 1–5. doi:10.1109/VTS-APWCS.2019.8851649

Khan, L. U., Pandey, S. R., Tran, N. H., Saad, W., Han, Z., Nguyen, M. N. H., et al. (2020). Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism. *IEEE Commun. Mag.* 58, 88–93. doi:10.1109/MCOM.001.1900649

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). *Federated Learning: Strategies for Improving Communication Efficiency*. arXiv preprint arXiv:1610.05492.

LeCun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., et al. (1995). Learning Algorithms for Classification: A Comparison on Handwritten Digit Recognition. *Neural networks: Stat. Mech. perspective* 261, 2.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal. Process. Mag.* 37, 50–60. doi:10.1109/MSP.2020.2975749

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2018). *Federated Optimization in Heterogeneous Networks*. arXiv preprint arXiv:1812.06127.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. (2019). Fair Resource Allocation in Federated Learning. In Proc. of International Conference on Learning Representations, Addis Ababa, Ethiopia, April, 2020.

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., et al. (2020). Federated Learning in mobile Edge Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* 22, 2031–2063. doi:10.1109/COMST.2020.2986024

Mohammad, U., Sorour, S., and Hefeida, M. (2020). *Task Allocation for Asynchronous mobile Edge Learning with Delay and Energy Constraints*. arXiv preprint arXiv:2012.00143.

Mohri, M., Sivek, G., and Suresh, A. T. (2019). *Agnostic Federated Learning*. arXiv preprint arXiv:1902.00146.

Mozaffari, M., Saad, W., Bennis, M., Nam, Y.-H., and Debbah, M. (2019). A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. *IEEE Commun. Surv. Tutorials* 21, 2334–2360. doi:10.1109/COMST.2019.2902862

Nielsen, T. A. S., and Haustein, S. (2018). On Sceptics and Enthusiasts: What Are the Expectations towards Self-Driving Cars?. *Transport policy* 66, 49–55. doi:10.1016/j.tranpol.2018.03.004

Niknam, S., Dhillon, H. S., and Reed, J. H. (2020). *Federated Learning for Wireless Communications: Motivation, Opportunities and Challenges*.

Nishio, T., Shinkuma, R., and Mandayam, N. B. (2020). *Estimation of Individual Device Contributions for Incentivizing Federated Learning*. arXiv preprint arXiv:2009.09371.

Pandey, S. R., Tran, N. H., Bennis, M., Tun, Y. K., Manzoor, A., and Hong, C. S. (2020). A Crowdsourcing Framework for On-Device Federated Learning. *IEEE Trans. Wireless Commun.* 19, 3241–3256. doi:10.1109/TWC.2020.2971981

Popovski, P., Nielsen, J. J., Stefanovic, C., Carvalho, E. d., Strom, E., Trillingsgaard, K. F., et al. (2018). Wireless Access for Ultra-reliable Low-Latency Communication: Principles and Building Blocks. *Ieee Netw.* 32, 16–23. doi:10.1109/mnet.2018.1700258

SAE(2016). Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *SAE Int.*, J3016. doi:10.4271/j3016_201609

Savazzi, S., Nicoli, M., Bennis, M., Kianoush, S., and Barbieri, L. (2021). *Opportunities of Federated Learning in Connected, Cooperative and Automated Industrial Systems*. arXiv preprint arXiv:2101.03367.

She, C., Liu, C., Quek, T. Q. S., Yang, C., and Li, Y. (2019). Ultra-reliable and Low-Latency Communications in Unmanned Aerial Vehicle Communication Systems. *IEEE Trans. Commun.* 67, 3768–3781. doi:10.1109/tcomm.2019.2896184

Shi, W., Zhou, H., Li, J., Xu, W., Zhang, N., and Shen, X. (2018). Drone Assisted Vehicular Networks: Architecture, Challenges and Opportunities. *IEEE Netw.* 32, 130–137. doi:10.1109/mnet.2017.1700206

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. (2017). "Federated Multi-Task Learning," in Proc. of Advances in neural information processing systems, 4424–4434.

Tran, N. H., Bao, W., Zomaya, A., Nguyen, M. N. H., and Hong, C. S. (2019). "Federated Learning over Wireless Networks: Optimization Model Design and Analysis," in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, Paris, France, April, 1387–1395. doi:10.1109/INFOCOM.2019.8737464

Wang, Y., Su, Z., Zhang, N., and Benslimane, A. (2021). Learning in the Air: Secure Federated Learning for Uav-Assisted Crowdsensing. *IEEE Trans. Netw. Sci. Eng.*, 1. doi:10.1109/TNSE.2020.3014385

Xiao, H., Rasul, K., and Vollgraf, R. (2017). *Fashion-mnist: A Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv preprint arXiv:1708.07747.

Xie, C., Koyejo, S., and Gupta, I. (2019). *Asynchronous Federated Optimization*. arXiv preprint arXiv:1903.03934.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. (2019). Federated Machine Learning. *ACM Trans. Intell. Syst. Technol.* 10, 1–19. doi:10.1145/3298981

Yang, Z., Chen, M., Wong, K.-K., Poor, H. V., and Cui, S. (2021). *Federated Learning for 6g: Applications, Challenges, and Opportunities*. arXiv preprint arXiv:2101.01338.

Yaqoob, I., Khan, L. U., Kazmi, S. A., Imran, M., Guizani, N., and Hong, C. S. (2019). Autonomous Driving Cars in Smart Cities: Recent Advances, Requirements, and Challenges. *IEEE Netw.* 34, 174–181.

Zeng, T., Semiari, O., Mozaffari, M., Chen, M., Saad, W., and Bennis, M. (2020). "Federated Learning in the Sky: Joint Power Allocation and Scheduling with UAV Swarms," in Proc. of the IEEE International Conference on Communications (ICC), Next-Generation Networking and Internet Symposium, Dublin, Ireland, June, 1–6.

Zhang, H., and Hanzo, L. (2020). Federated Learning Assisted Multi-Uav Networks. *IEEE Trans. Veh. Technol.* 69, 14104–14109. doi:10.1109/TVT.2020.3028011

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.