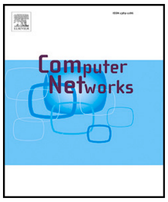
Computer Networks 251 (2024) 110625

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Proposal and investigation of a distributed learning strategy in Orbital Edge Computing-endowed satellite networks for Earth Observation applications

Francesco Valente [a], Francesco G. Lavacca [b], Tiziana Fiori [a], Vincenzo Eramo [a,*]

[a] Department of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Via Eudossiana 18, Rome, 00184, Italy
[b] Department of Human Science, Link Campus University, Via del Casal di San Pio V, Rome, 00165, Italy

ARTICLE INFO

Keywords:
Satellite networks
Distributed learning
Earth Observation

ABSTRACT

One of the key enabling solutions to in-orbit extract information from Earth Observation images is given by deep learning techniques. However, the accuracy of these algorithms is strictly related to the availability of large datasets of satellite images for training purposes. Limitations on the available transmission bandwidth in the orbital context may prevent the possibility to downlink all acquired images to a node where centralized training happens. Instead, Federated Learning (FL) could be fruitfully leveraged in this scenario, since it provides for each satellite to train a local model only with its own dataset, and then to share its trained model with a central server, which receives models trained by the different satellites and aggregates them into a new global model being eventually shared with all the satellites, and this repeats until convergence is reached. However, because communication with a node acting as a central parameter server may be still limited by short visibility time, the described process may need a long time because of limited communication windows, negatively impacting the time needed to reach model convergence. For this reason, we propose a communication strategy to support a completely distributed learning technique to train a deep learning model in-orbit, by leveraging the fact that satellites may form a network thanks to the potential availability of Inter-Satellite Links (ISLs) within and between orbital planes. Our proposal is different from a FL approach since we provide for each satellite to receive all the information needed to calculate an updated global model by itself, without leaning on a central parameter server. Numerical results show that distributed learning outperforms FL in number of learning rounds completed in the unit time, allowing for reaching validation accuracy convergence in a shorter time, as it has been verified on a land coverage classification task based on the EuroSAT dataset.

1. Introduction

Orbital Edge Computing [1] (OEC) is a solution providing for leveraging computational capacity available on board of satellites to process data directly in-orbit. This technique would enable a paradigm shift with respect to what happens today, providing for satellites gathering information and storing it in their memories until they fly over a ground station, when all stored information are downlinked for processing on the Earth. However, data downlink may require a high data rate, since a high amount of data shall be transmitted to the ground station during short visibility time, and this poses a problem for future satellites, since data rate is limited by the amount of power available on board, thus, by the dimension and masses of solar panels and batteries. Thanks to the ability to process data on board, OEC can benefit several applications, like mega-LEO constellations [2], Non-Terrestrial Networks (NTNs) [3, 4] and Space–Air–Ground Integrated Networks (SAGINs) devoted to 5G [5] and to 6G [6]. Furthermore, OEC can be fruitfully leveraged also in the context of Earth Observation (EO) constellations, since by processing acquired images on board, it is possible to downlink to ground only the information actually useful to the application [7–12]. For example, in an hypothetical flooding detection application, thanks to OEC it would be possible to determine on-board whether there is a flooding in a certain region or not, and only transmit this information on ground, instead of downlinking the full image to be then elaborated by the ground station. In particular, when OEC is available in a constellation where satellites constitute a communication network thanks to ISLs, this solution shows a reduction in operating cost [13], in time to make gathered information available on ground [13–16], and in energy to be used on ground stations to process data [17].

Under the application viewpoint, one of the key enabling solutions to extract information from acquired images, both on satellites and on ground, is given by deep learning techniques [7,18]. However, the accuracy of these algorithms is strictly related to the availability

[*] Corresponding author.
E-mail address: vincenzo.eramo@uniroma1.it (V. Eramo).

https://doi.org/10.1016/j.comnet.2024.110625
Received 21 December 2023; Received in revised form 10 June 2024; Accepted 1 July 2024
Available online 6 July 2024

of large datasets for training purposes [19]. In case of EO-related applications, these datasets involve the availability of a high number of satellite images on the device where model are trained. For example, in this scenario we would have to transfer all the acquired images to the training node (i.e., a ground station or a specific satellite), where model training is executed. However, this would again require for a high amount of bandwidth, i.e., high transmission data rate to transfer a high amount of data in a short visibility time. Another solution could be given by making satellite sharing their own datasets in such a way that each satellite can train the model by itself on a dataset given by the union of its own dataset and the ones appertaining to the other satellites in the constellation. However, this solution can be again limited by the available bandwidth and by the computational capacity available on-board, since training a model on a larger dataset requires an increased computational effort. Instead, Centralized federated learning can be fruitfully leveraged in this scenario, since this technique provides for each satellite to train a local model only with its own dataset, and then to share its trained model with a central server, which receives models trained by the different satellites and aggregates them into a new global model which is finally shared with all the satellites, and this repeats until convergence is reached [20]. It is important to underline that this solution is more appropriate in the orbital environment because only the models are shared instead of all datasets, and models have a reduced size with respect to datasets of satellite images. Furthermore, local training happens with a reduced amount of data (i.e., a single satellite dataset, being obviously smaller than the union of the datasets appertaining to all satellites). However, because communication with a ground station (or, in general, with a node acting as a central parameter server) is limited by short visibility time, local model gathering and consequent global model transmission to all satellites may need a long time because of limited communication windows, and this has a negative impact on the time needed to reach model convergence, which is strictly related to the completion of these model sharing rounds. Applications of decentralized federated learning solutions are not suited in satellite network environments; in fact their serial local training and the needed to serially transfer the local models between satellites make them slow in convergence time.

For this reason, in this work we propose and investigate a distributed learning solution where satellites share their own models among themselves, without leaning on a central parameter server. When each satellite has received the models related to all the other satellites, it autonomously aggregate them in a new global model, which will be locally trained. Then, new local models are again shared among satellites to start a new iteration, and this procedure goes on until reaching model convergence. In particular, the novel contribution of this work can be found in the following points:

- We propose a completely distributed learning-based strategy to train a deep learning model in orbit, by leveraging the fact that satellites may form a network thanks to the potential availability of ISLs within and between orbital planes. Our proposal is different from a traditional federated learning approach since our strategy does not rely on a central parameter server; it is also different from a decentralized federated because the local models can be in parallel trained and that leads to better performance in training convergence time.
- We evaluate the proposed solution in a real orbital scenario, in particular by focusing on the impact that constellation and application design-related parameters have on its performance, in comparison with state-of-the-art federated learning-based solutions. Results show that the proposed solution outperforms the federated learning ones in terms of weight distribution time; for instance this time is improved by 80% in a scenario with 12 satellites and 2 orbital planes.
- We also carry out the comparison in a real training case with data extrapolated by the EuroSAT dataset [21]. The comparison parameters are the typical ones that is the convergence time and the test accuracy; we obtain test accuracy of 80% with convergence time of 1,25 h and 1,69 h for the proposed and learning federated solutions respectively, that is to say that the distributed learning solution allows for a reduction by 26% of the convergence time with respect the federated learning solution.

The rest of the paper will be organized as follows. Related works are discussed in Section 2; we introduce the network and application model in Section 3; Section 4 is devoted to the definition of the distributed learning solution; Section 5 shows the numerical results to compare the proposed solution with state-of-the-art, federated-learning based benchmarks and, finally, Section 6 concludes the paper retracing the main results of the work.

2. Related works

In this work, we focus on the advantage that an appropriate leverage of networks made by satellites endowed with on-board processing capability can bring to in-orbit machine learning applications dedicated to Earth Observation. For this reason, the proposed research can be associated to literature dedicated to NTNs and SAGINs, as well as to edge computing in satellite constellations, with a special focus on studies on the EO applications and on how machine learning can be leveraged within EO constellations.

As far as NTNs and SAGINs are concerned, strategies to deal with latency appears to be one of the most explored topics in literature. For example, a solution to guarantee real-time communications in mega-constellations is illustrated in [22], whose authors propose a strategy based on appropriate flow allocation and on cloud or satellite relay servers. Instead, Zhang at al. [3] underline how latency, jitter, unstable routing and limited network reachability may pose an issue in the integration of LEO mega-constellations and ground networks, and for this reason they propose an optimal solution to integrate these networks with minimum latency and stable routing. Furthermore, several works propose applications of Artificial Intelligence in this context. For example, deep reinforcement learning is leveraged in [23] for traffic offloading purposes in a highly dynamic topology and traffic scenarios like orbital ones. Furthermore, AI applications to NTNs and SAGINs are particularly considered in the context of 6G. Thang et al. [24], leverages AI in the integration of NTNs and terrestrial networks enhancing the energy efficiency of maritime networks. Chen et al. [25], focuses on the performance evaluation of federated learning techniques in LEO constellations for 6G. Finally, the advantage of leveraging a satellite network in the EO missions context is underlined by [26], where a solution to improve timeliness of EO data by taking advantage of mega-constellations is proposed.

Moving on research related to in-orbit edge computing applications, two main research strands can be identified, related to satellite edge computing supporting either terrestrial mobile users or EO missions. As far as the first area of research is concerned, an investigation on how Mobile Edge Computing can be extended to SAGINs has been proposed in [27], with a particular focus on challenges, architecture and technologies needed to achieve this goal. Instead, scheduling strategies and architecture to leverage satellite edge computing in Internet of Things (IoT) have been studied in [28]. AI applications have been considered also in this context [29], where deep imitation learning is leveraged in a task offloading and caching strategy to optimize the task completion time and the satellite resource usage. As far as support of EO missions is concerned, European Space Agency Φ-Sat −1 Mission [7] demonstrated the feasibility of processing data on a satellite by means of AI techniques. However, several research works focus on the gains that can be obtained by combining satellite networks and OEC capabilities. In particular, authors of [13] proposed a strategy able to jointly allocate resource and place processing in networks of EO satellites having on-board processing capability to optimize the total operating cost to be paid for data transmission, storage and processing. The same authors proposed in [17] a heuristic-based strategy to minimize the energy consumption on Earth due to image processing by leveraging the possibility of in-orbit processing, given the limitations on bandwidth, memory, processing capacity and energy on board of satellites. Instead, an improvement in EO data timeliness can be obtained by means of load-balancing within EO constellations where satellites have the possibility to process data directly in-orbit [14].

On the same research strand, it is possible to identify a high interest in federated learning techniques in LEO satellites, which can be fruitfully applied in the EO mission context. In the satellite network environments, centralized federated learning solutions have been proposed and evaluated. The decentralized ones [30] have not had applications [20] because their ineffectiveness in this context, being based on both a serial local training and model transfer between satellites.

A comprehensive presentation of the state-of-the-art centralized federated learning strategies in mega-LEO constellations is given by [20], where three federated learning scenarios in orbital environment are discussed, depending on the availability of links between a node acting as a parameter server (where local models are aggregated to update a global model) and the satellites, as well as the communication opportunities within the constellation itself. In particular, the paper identifies the scenario where inter-orbital ISLs are available as the most promising for further research. The same authors proposed a deeper investigation of the three federated learning scenarios in further works. In particular, in case of sporadic connection possibilities to a parameter server, as it happens in case no ISL is available or only a ground station takes part to the federated learning process, an asynchronous federated learning strategy has been proposed in [31]. However, the convergence speed of such a scheme may be compromised by model staleness. As explored in [32], a solution to this problem can be given by leveraging the predictability in communication opportunities with the ground station, which allows for the proposal of a scheduling algorithm to optimally decide when the model parameter exchange between ground stations and satellites shall happen. Instead, in [33], authors focused on the case in which only intra-orbital ISLs are available, proposing a communication scheme enabling synchronous federated learning with a parameter server placed on the ground or in a satellite not appertaining to the constellation.

To the best of our knowledge, no study in literature proposed a distributed learning strategy in constellations of satellite endowed with on-board processing capability and interconnected by means of ISLs. A distributed learning approach has been evaluated [34] in the context of the prediction of processing capacities in network function virtualization (NFV). However, this solution is based on the fact that topology is static in time, while this is not true in case of orbital environment, where the availability of inter-orbital ISLs and links between satellites and ground stations changes in time because of the satellites and Earth relative motion. For this reason, the distributed learning strategy available in literature cannot be applied in the context of satellite constellations for EO, and an appropriate solution needs to be proposed and investigated.

3. Network modeling

The satellite network scenario is reported in Fig. 1.

The main parameters introduced in the paper are reported in Table 1. We consider to have a constellation of N_{Sat} satellites, equally distributed over N_{op} orbital planes. In this scenario, we will assume each orbit to be circular, with altitude h_p, inclination i_p and right ascension of the ascending node Ω_p, for $p \in [0, \dots, N_{op} - 1]$. We assume that the ith satellite, with $i \in [0, \dots, N_{Sat} - 1]$, appertain to the orbital plane $p_i = \lfloor i \cdot N_{op}/N_{Sat} \rfloor$ and occupies the position $\mathbf{r}_i(t)$ at time t in the ECI reference frame. Satellite motion repeats with a period depending on the altitude h_p of the orbit it occupies, i.e., $T_p = 2\pi\sqrt{\left(R_E + h_p\right)^3/\mu_E}$, with $p \in [0, \dots, N_{op} - 1]$, where R_E is the Earth's radius (assuming, without loss of generality, a perfectly spherical Earth) and μ_E is the Earth's gravitational constant.

Table 1
Network model parameters.

Set or parameter	Description
N_{sat}	number of satellites
N_{op}	number of orbital planes
h_p	orbit altitude
i_p	orbit inclination
Ω_p	orbit right ascension of the ascending node
i	index in $[0, \dots, N_{sat} - 1]$ representing a satellite
p_i	orbital plane occupied by the ith satellite, with $i \in [0, \dots, N_{sat} - 1]$
$\mathbf{r}_i(t)$	position vector of the ith satellite, with $i \in [0, \dots, N_{sat} - 1]$
h_p	altitude of satellites on the pth orbital plane, with $p \in [0, \dots, N_{op} - 1]$
T_p	period of motion of satellites on the pth orbital plane, with $p \in [0, \dots, N_{op} - 1]$
R_E	Earth's radius
μ_E	Earth's gravitational constant
G	antenna gain
c	speed of light
ν_{tx}	carrier frequency
P	transmission power
$R_{i,j}$	transmission data rate between ith and jth satellite, with $i, j \in [0, \dots, N_{sat} - 1],\ i \neq j$
B	transmission bandwidth
k_B	Boltzmann's constant
T_s	system noise temperature
N_{GS}	number of ground stations
g	index in $[0, \dots, N_{GS}]$ representing a ground station
$\mathbf{r}_{GS_g}(t)$	position vector of the gth ground station, with $g \in [0, \dots, N_{GS} - 1]$
T_{GS}	ground station rotation period, i.e., Earth's sidereal day
T	repeat cycle time
El_{min}	minimum elevation angle

We can define the distance between the ith and jth satellite at time t, with $i, j \in [0, \dots, N_{Sat} - 1], i \neq j$, with the following expression:

$$d(i, j, t) = |\mathbf{r}_i(t) - \mathbf{r}_j(t)| \tag{1}$$

The distance between two satellites at time t allows us to determine whether an ISL is available or not. In fact, in general an ISL between the ith satellite and the jth satellite at time t is available if the following condition is verified:

$$d(i, j, t) \le d_{max}(i, j) \tag{2}$$

where $d_{max}(i, j)$ represents the maximum distance between the ith and jth satellite at which communication is possible. Under the assumption of Additive White Gaussian Noise channel, this can be calculated as:

$$d_{max}(i, j) = \frac{G\ c}{4\pi\nu_{tx}} \sqrt{\frac{P}{\left(2^{\frac{R_{i,j}}{B}} - 1\right) k_B T_s B}} \tag{3}$$

where G is the antenna gain, c is the speed of light, ν_{tx} is the carrier frequency, P is the transmission power, $R_{i,j}$ is the transmission data rate between the satellites, B is the bandwidth, k_B is the Boltzmann's constant, T_s is the system noise temperature.

From Fig. 1 we notice that two types of ISLs can be distinguished in the network topology: intra-orbital and inter-orbital reported with red and green colors, respectively. The intra-orbital links connect satellites located in the same orbital plane; conversely the inter-orbital links connect satellites located in different orbital planes. Next we describe the characteristics of the two types of links.

Since all satellites on a same circular orbit move with the same angular velocity, the intra-orbital links does not change their distance

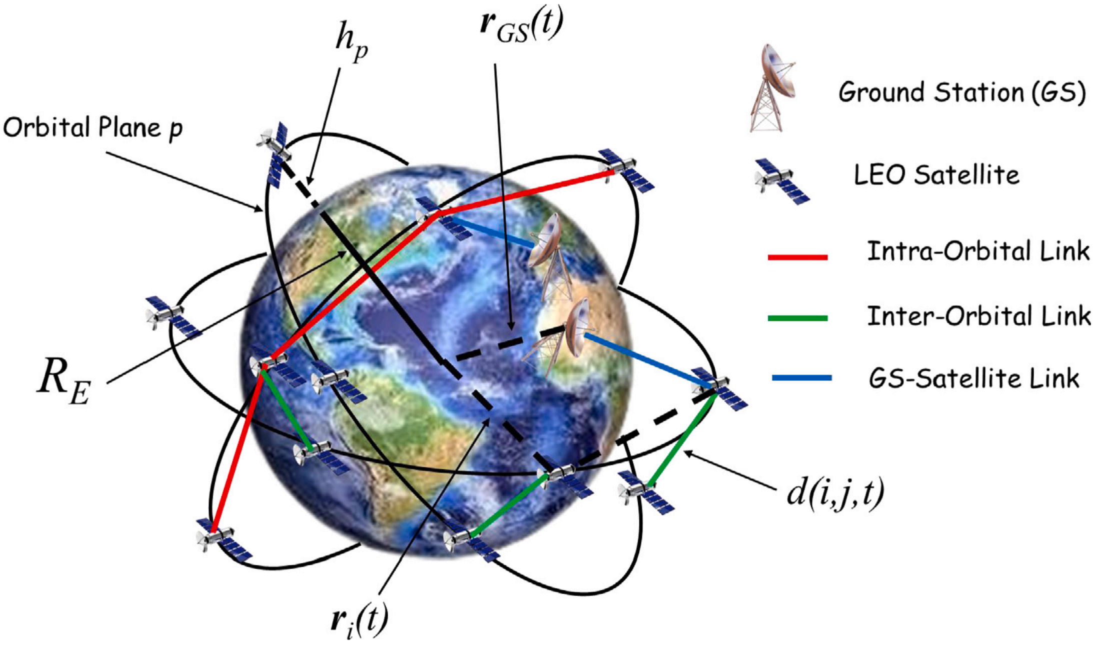


Fig. 1. Satellite network scenario.

over time and contribute to a stable network topology. The availability of intra-orbital link does not depend on the time and it can be verified more easily than from the application of the expression (2). From knowledge of orbital parameters and in particular angular velocity, we can easily identify simple conditions for the availability of intra-orbital links. They are available if and only if the following two conditions are verified for any couple of adjacent satellites in the orbit:

$$\begin{cases} 2\left(R_E + h_p\right)\sin\left(\frac{2\pi N_{op}}{N_{Sat}}\right) \leq d_{max}(i,j) \\ \left(R_E + h_p\right)\cos\left(\frac{2\pi N_{op}}{N_{Sat}}\right) > R_E, \end{cases} \tag{4}$$

for any $i \in [0, \dots, \dfrac{N_{Sat}}{N_{op}} - 1],\ j = (i+1) \mod \dfrac{N_{Sat}}{N_{op}}$

where the first condition ensures that the line of sight distance between two adjacent nodes is smaller than the maximum distance at which communication is possible, as defined in Eq. (3), while the second condition ensures that the line of sight does not intersect the Earth. If these are satisfied for any couple of satellites in the orbit, intra-orbital ISLs are always active, otherwise, they are always unavailable. Please notice that we are assuming intra-orbital ISLs to be potentially available only between a satellite and its adjacent nodes in the orbit.

The inter-orbital links change their distance because of the movement of the satellites on orbital planes with different inclinations. For this reason the availability of these links is intermittent but predictable according to the study over time of the availability expression (2) where $d_{max}(i,j)$ is expressed by (3) and $d(i,j,t)$ depend on the position vectors $\mathbf{r}_i(t)$ and $\mathbf{r}_j(t)$ and consequently by the orbital parameters. The predictability of link unavailability allows for a-priori assessment of its impact on the performance of the distributed procedure.

As far as ground segment is concerned, we consider N_{GS} ground stations on the Earth enabled to receive data from the constellation. Each ground station moves with the Earth, thus, its position in time $\mathbf{r}_{GS_g}(t)$, with $g \in [0, \dots, N_{GS}]$, changes periodically with a period equal to the sidereal day, $T_{GS} = 86164$ s. Thus, the entire system constituted by all satellites and ground stations is periodic, with a period $T = lcm\left(T_{GS}, T_0, \dots, T_{N_{op}-1}\right)$ where lcm denotes the least common multiple. We assume that communications between the ith satellite and the gth ground station are possible when the elevation angle of the satellite with respect to the ground station (i.e., the angle between the tangent plane to the Earth surface containing the ground station and the vector Earth center-satellite vector) is higher than a minimum elevation El_{min}. This translates in the following condition:

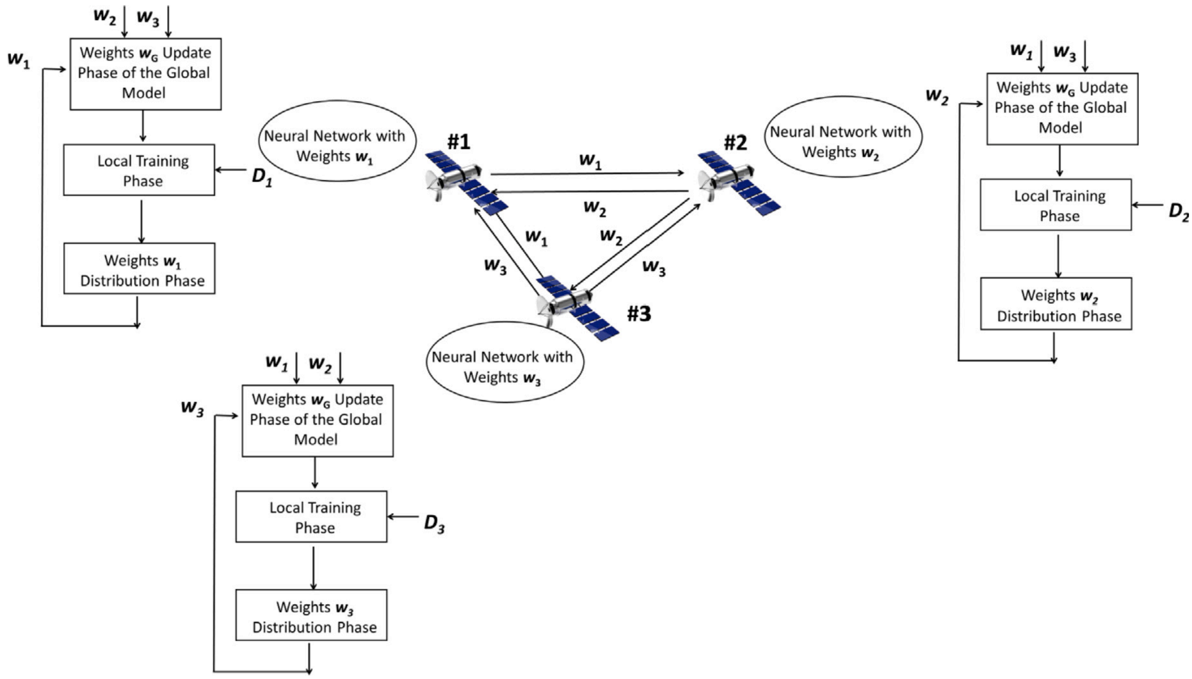
$$\frac{\pi}{2} - \arccos\left(\frac{\mathbf{r}_{GS_g}(t)\cdot\mathbf{r}_i(t)}{|\mathbf{r}_{GS_g}(t)||\mathbf{r}_i(t)|}\right) \geq El_{min}, \tag{5}$$
$$\forall i \in [0, \dots, N_{Sat} - 1],\ g \in [0, \dots, N_{GS} - 1]$$

where $\mathbf{r}_{GS_g}(t)\cdot\mathbf{r}_i(t)$ represents the scalar product between the position vector of the ground station and of the satellite, respectively, and El_{min} is in radiant. It is important to underline that, in case a satellite is able to communicate with more than a ground station at a time, we assume that it is connected with only one of them.

4. Proposed distributed learning solution

We propose a purely distributed learning solution where each satellite stores the global training model, train it locally with its data and distributes the weights towards all of the other satellites so that the global model can be updated. We show in Fig. 2 an example of the proposed solution in the case of three satellites where each of them stores the global training model whose the weights $\mathbf{w}_G$ are updated by averaging the weights $\mathbf{w}_i$ $(i = 1,2,3)$ of the local models of all of the satellites (weights update phase). The global model is trained with the local data $\mathbf{D}_i$ $(i = 1,2,3)$ collected in the satellite. Each satellite must exchange the weights $\mathbf{w}_i$ $(i = 1,2,3)$ of its neural network, sharing it with the other satellites via a weight distribution process (weight distribution phase). The satellites receive the weights of the various local models and update the weights of the global one with a merge of the weights.

The execution times of the three phases are reported in Fig. 3 where we report the repeat cycle time T that is the repetition time of the satellite network topology; in fact despite the rotation of the Earth and the movement of satellites around the Earth it is possible to prove that satellites reach the same position at instants that are a repeat cycle time T apart. From Fig. 3 we notice how the distributed learning algorithm is applied in N_r rounds in a repeat cycle time where in each round the weight update, local training and weight distribution phases are executed. We denote with σ_l, τ_l and δ_l the duration of the weight update, local training and weight distribution phases of the rth



w_2 w_3
w_1
Weights w_G Update Phase of the Global Model
Local Training Phase
D_1
Weights w_1 Distribution Phase
Neural Network with Weights w_1
#1
#2
#3
w_1
w_2
w_1
w_2
w_3
w_3
Neural Network with Weights w_2
Neural Network with Weights w_3
w_1 w_3
w_2
Weights w_G Update Phase of the Global Model
Local Training Phase
D_2
Weights w_2 Distribution Phase
w_1 w_2
w_3
Weights w_G Update Phase of the Global Model
Local Training Phase
D_3
Weights w_3 Distribution Phase

Fig. 2. An example of the proposed distributed algorithm with three satellites.

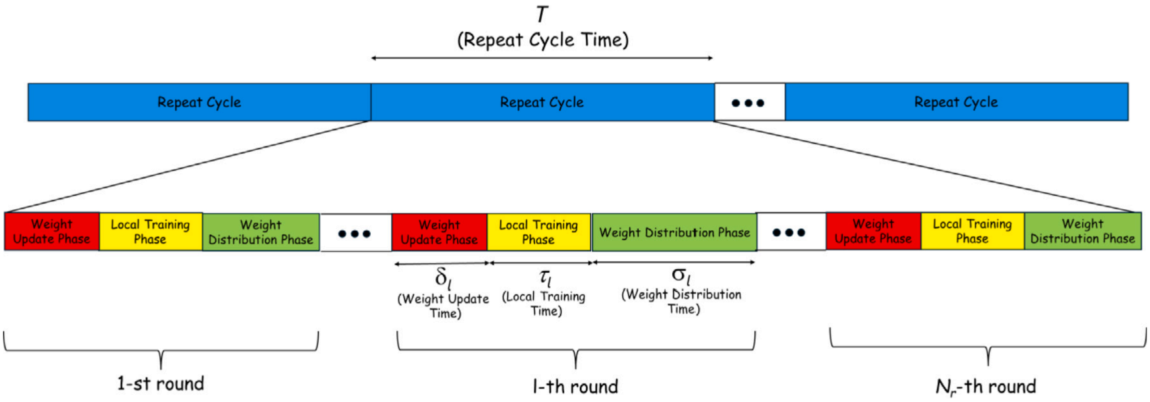


Fig. 3. Description of the weight distribution, weight update and local training phases in the distributed learning algorithm.

round respectively. Finally we assume that in each local training phase N_{ep} epochs are executed. We highlighted the difference between the proposed distributed learning solution and the synchronous centralized federated one [33]. This second solution is reported in Fig. 4 and it is based on: (i) a global training model implemented in a Ground Station; (ii) the distribution of the global training model from the GS to all of the satellites; (iii) the local training of the model in each satellite by using the collected data; (iv) the distribution of the local models from the satellites to the GS so that the GS can merge them in the global learning model with updated weights. From Fig. 4 we can notice how the centralized federated algorithm is based on the execution of N_{r_F} rounds where in each round are involved two weights distribution phases that, because of the intermittent network links, leads to an increase of the round time and consequently to an decrease in the training convergence time with respect to our proposed distributed learning solution.

Decentralized federated learning solutions have been also proposed in literature [30] in which the global model is trained in turn by the satellites. They are not considered in this paper because they require in each round both a serial local training and a transfer of the global model between satellites and for this reason they are very slow because of the intermittent links of the satellite network.

Next we describe the weight update and local training phase of our proposed distributed learning solution in Section 4.1. Finally the weight distribution phase is reported in Section 4.2. Two options of the proposed distributed learning procedure are illustrated in Section 4.3.

4.1. Weight update and local training phase

The main parameters of the learning model are reported in Table 2.

Thus, we assume that each ith satellite, with $i \in [0, \dots, N_{Sat}-1]$, has its own dataset D_i, containing a number of samples $|D_i|$, and a local model represented by the vector $\mathbf{w}_i^{r,e}$, containing the local values for weights at the eth learning epoch of the rth distributed learning round, with $r \in [0, \dots, N_r]$, $e \in [0, \dots, N_{ep}]$ and N_{ep} representing the maximum number of local learning epochs, i.e., the maximum number of times the local model goes through updates over the local dataset during the round. As previously stated, each rth round, with $r \in [1, \dots, N_r]$, starts with each ith satellite sharing its most updated version of locally trained model, i.e., the local model after N_{ep} epochs at the end of the previous round, denoted with $\mathbf{w}_i^{r-1,N_{ep}}$, with all the other satellites. We assume that $\mathbf{w}_i^{0,N_{ep}}$ is the local model on the ith satellite before any distributed learning round occurs. Once each satellite has received the

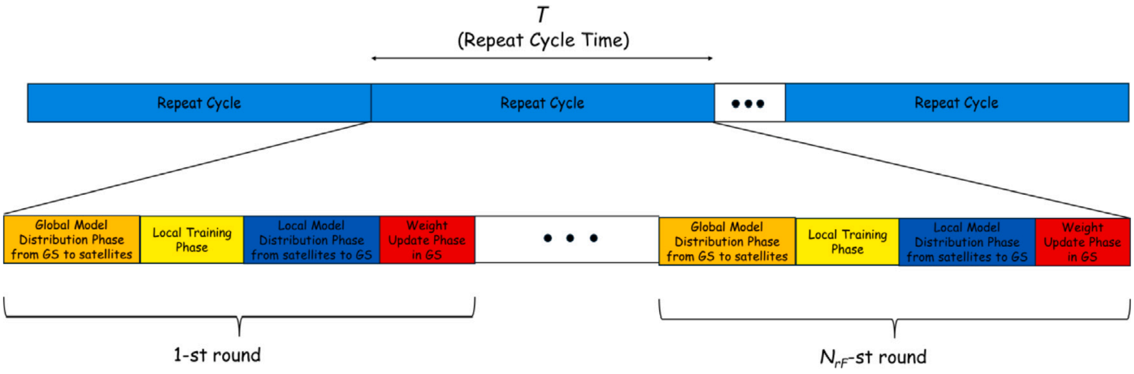

T
(Repeat Cycle Time)
Repeat Cycle
Repeat Cycle
Repeat Cycle
Global Model Distribution Phase from GS to satellites
Local Training Phase
Local Model Distribution Phase from satellites to GS
Weight Update Phase in GS
Global Model Distribution Phase from GS to satellites
Local Training Phase
Local Model Distribution Phase from satellites to GS
Weight Update Phase in GS
1-st round
N$_{rF}$-st round

Fig. 4. Phase Description in the centralized federated learning algorithm.

Table 2
Application model parameters.

Set or parameter	Description
N_r	number of distributed learning rounds
N_{ep}	number of local learning epochs
D_i	dataset of the ith satellite, with $i \in [0,\dots,N_{sat}-1]$
$\|D_i\|$	number of samples in the dataset of the ith satellite, with $i \in [0,\dots,N_{sat}-1]$
$\mathbf{w}_i^{r,e}$	local model on the ith satellite, at the eth local training epoch of rth distributed learning round, with $i \in [0,\dots,N_{sat}-1]$, $r \in [0,\dots,N_r]$, $e \in [0,\dots,N_{ep}]$
$\mathbf{w}_G^r$	global model at the beginning of rth distributed learning round, with $r \in [0,\dots,N_r]$
F_i	local loss function on the ith satellite, with $i \in [0,\dots,N_{sat}-1]$

local models of all the remaining ones, the global model can be locally obtained as:

$$\mathbf{w}_G^r = \sum_{i=0}^{N_{sat}-1} \frac{|D_i|}{\sum_{i=0}^{N_{sat}-1} |D_i|} \mathbf{w}_i^{r-1,N_{ep}} \tag{6}$$

and the local model on the ith satellite during the rth round before any local learning epoch (i.e., $e = 0$) is set as:

$$\mathbf{w}_i^{r,0} = \mathbf{w}_G^r, \ \forall i \in [0,\dots,N_{sat}-1] \tag{7}$$

Finally, each ith satellite trains its local model by applying the gradient descent technique considering only the local dataset for N_{ep} epochs. After local training, the local model will be:

$$\mathbf{w}_i^{r,N_{ep}} = \mathbf{w}_i^{r,0} - \eta \sum_{k=0}^{N_{ep}-1} \nabla F_i \left(\mathbf{w}_i^{r,k}\right), \ \forall i \in [0,\dots,N_{sat}-1] \tag{8}$$

with η representing the learning rate, and ∇F_i the gradient operator applied on the function F_i representing a local loss function, i.e., a loss function evaluated only on the samples of the ith satellite dataset.

4.2. Weight distribution phase

Next we describe the Weight Distribution Phase in which each satellite sends its weight to all other satellites. The main pseudo code of the proposed algorithm proposed in this phase is reported in Alg. 1. The main parameters and sets have been previously defined except for the event list E of events. In particular, E is an ordered list where events are ordered by their occurrence time, with the earliest event occupying the first position, i.e., being the $E[0]$ element of the list. Each event $\varepsilon \in E$ is characterized by an event type denoted by ε.type, whose value is *'link-on'* if the event represents the fact that two nodes reached a relative position such that they are close enough to allow data transmission, it is equal to *'link-off'* if the event represents two nodes reaching a relative position such that their distance is not enough to enable communication, or it is equal to *'transfer_completed'* to indicate that data transmission between two nodes is completed. Furthermore, each ε event is also associated with a list of involved nodes $\varepsilon.N$ and a time at which the event happens $\varepsilon.t$. Finally, in case the event is *'link-on'* or *'link-off'*, the event is also associated to a property $\varepsilon.LOT$ indicating the time at which the link will be unavailable in case the event type is *'link-on'*, or the time at which the link became available in case the event type is *'link-off'*; instead, in case the event type is *'transfer_completed'*, the event is associated to a property $\varepsilon.W$ representing the models memorized on both the nodes involved in the data transfer after the transmission ends. At the beginning, the event list contains only *'link-on'* and *'link-off'* events that can be obtained by orbital mechanics, propagating node positions in time and evaluating their distances as discussed in Section 3. In particular, we evaluate these events only during a repeat cycle T, since because of the periodicity of both satellite and Earth motion, what happens in a repeat cycle will be repeated the same in the following ones.

Moving to the discussion of the different algorithm steps described in Alg. 1, in Line 1 we initialize:

- a boolean auxiliary variable *exit* indicating whether the local model distribution phase is completed or not;
- an auxiliary variable $\Lambda_{i,j}$ whose value is zero if at the current event time no link is available between the ith and jth node, with $i, j \in [0,\dots,N_{sat}-1]$, and is different from zero when the link is available, with the specific value indicating the time at which link will become unavailable;
- an auxiliary list M_i for each ith node, with $i \in [0,\dots,N_{sat}-1]$, containing local models currently kept in memory on the ith node.

In particular, before the algorithm starts, we assume no link is already available, since no link-on event has been considered, yet, and we assume each node has in its memory its own local model only, since no data transfer happened, yet.

After auxiliary variable initialization, next steps will be repeated either until event list is not empty or the variable *exit* is *True*. In particular, we first extract the earliest event in time ε from the event list and remove it from the list (Lines 3–4). Then, we set auxiliary variables t, i, j to be equal to the event time, and to the first and second nodes involved in the event, respectively (Lines 5–6). Following steps depend on the event type. In case of *'link-on'* event, since a new link is available, we set $\Lambda_{i,j}$ to be equal to the link-off time (Line 8). Furthermore, we assume that as soon as a link becomes available, the two connected nodes try to share the models in their memories following Alg. 2, which will be discussed in detail further on. In case model transmissions are possible, a transfer completed event is added to the event list (Line 9). Instead, in case of *'transfer_completed'* event, we first update the lists of received models on both nodes involved in the exchange (Line 11) and then we check if each satellite in the constellation has local models of all satellites in its memory (Lines 12–19. In case all satellites have all the local models, auxiliary variable *exit* is set to *True*. Furthermore, we assume that as soon as a node has received local models, it tries to share them with other nodes, again leveraging Alg. 2 (Lines 20–24).

Algorithm 1: Distributed Learning Algorithm

```
   Input: $E$, $N_{sat}$, $R_{i,j}$ $\forall i,j \in [0,\dots,N_{sat}-1]$
1  Initialize: $exit \leftarrow False$, $\Lambda_{i,j} \leftarrow 0$, $M_i \leftarrow \{\mathbf{w}_i^{0,N_{ep}}\}$, $\forall i,j \in [0,\dots,N_S-1]$;
2  while $E \neq \emptyset \wedge exit \neq False$ do
3      $\varepsilon \leftarrow E[0]$ //Extract earliest event;
4      $E \leftarrow E - \{\varepsilon\}$ //Remove the extracted event from the event list;
5      $t \leftarrow \varepsilon.t$ //Extract the event time;
6      $i \leftarrow \varepsilon.N[0]$, $j \leftarrow \varepsilon.N[1]$ //Extract linked nodes;
7      if $\varepsilon$.type is 'link-on' then
8          $\Lambda_{i,j} \leftarrow \varepsilon.LOT$ //Set the link-off time;
9          Add Transfer Complete Event($E$, $i$, $j$, $M_i$, $M_j$, $R_{i,j}$, $\Lambda_{i,j}$)
             //Apply Alg.2;
10     else if $\varepsilon$.type is 'transfer_completed' then
11         $M_i \leftarrow M_i \cup \{\varepsilon.W\}$, $M_j \leftarrow M_j \cup \{\varepsilon.W\}$ //Update the list of
             received models on both the sharing nodes;
12         for $n \in [0,\dots,N_{sat}-1]$ do
13             if $|M_n| == N_{sat}$ then
14                 $exit \leftarrow True$ //All of the satellites have received all local
                     models;
15             else
16                 $exit \leftarrow False$ //At least a satellite has not received all
                     local models, yet;
17                 break;
18             end
19         end
20         for $n \in \{i,j\}$ do
21             for $m \in [0,\dots,N_{sat}-1]$ do
22                 Add Transfer Complete Event($E$, $i$, $j$, $M_n$, $M_m$, $R_{n,m}$,
                     $\Lambda_{n,m}$) //Apply Alg.2;
23             end
24         end
25     else if $\varepsilon$.type is 'link-off' then
26         $\Lambda_{i,j} \leftarrow 0$ //Set the link as unavailable;
27     end
28 end
```

Let us now discuss steps of Alg. 2. As introduced before, it evaluates if it is possible to complete a data transfer between two nodes and, in positive case, it adds a transfer completed event in the event list. In particular, we consider that in the communication between the ith and jth node, i sends to j all models that are in M_i but not in M_j, and vice versa. For this reason, we first evaluate the number of models to be exchanged by means of the expression in Line 1, as well as the transmission and propagation delay (Lines 2–3). Then, if the number of models to be exchanged is different from zero and the link is still available when the potential transmission is completed (Line 4), we add a *'transfer_completed'* event to the event list, having the time at which transfer finishes as event occurrence time, i and j as involved nodes, and the union of M_i and M_j as the models memorized on both i and j after transfer is completed (Lines 5–8). Finally, the new event is added to the event list (Line 9).

Algorithm 2: Add Transfer Complete Event

```
   Input: $E$, $w_s$, $i$, $j$, $t$, $M_i$, $M_j$, $R_{i,j}$, $\Lambda_{i,j}$
1  $n_w \leftarrow |M_i| + |M_j| - 2|M_i \cap M_j|$ //Number of local models to be
     exchanged;
2  $\tau_t \leftarrow w_s \cdot n_w / R_{i,j}$ //Calculate the transmission time;
3  $\tau_p \leftarrow 2 \cdot d(i,j,t)/c$ //Calculate the propagation time;
4  if $n_w \neq 0 \ \wedge t + \tau_t + \tau_p \leq \Lambda_{i,j}$ then
5      $\tilde{\varepsilon}$.type $\leftarrow$ 'transfer_completed' //Set new event type as transfer
         completed;
6      $\tilde{\varepsilon}.t \leftarrow t + \tau_t + \tau_p$ //Time of the new transfer completed event;
7      $\tilde{\varepsilon}.N \leftarrow [i,j]$ //Nodes involved in the new transfer completed event;
8      $\tilde{\varepsilon}.W \leftarrow M_i \cup M_j$ //Weights on each involved node after new transfer
         completed event;
9      $E \leftarrow E \cup \{\tilde{\varepsilon}\}$ //Add the new transfer completed event in the event list;
10 end
   Output: $E$
```

As far as the complexity of the proposed strategy is concerned, it is possible to notice that it is mainly related to the length of the event list, to the maximum number of rounds and to the complexity of adding a *'transfer_completed'* event to the list. In particular, by implementing the event list as an heap queue, adding an element to the queue has a complexity $\mathcal{O}(\log n)$, where n is the length of the event list. By looking at the loop starting in Line 2, we notice that, in order to complete a round it is necessary that each satellite has received local models from all the other satellites. In the worst case, this is accomplished by means of direct communications between each couple of satellites, leading to $N_{sat}(N_{sat}-1)/2$ *'transfer_completed'* events for each round. For this reason, the complexity of the strategy can be given by $\mathcal{O}\left(N_{sat}^2 \log n\right)$. However, in the proposed strategy the event list length may increase during each cycle of the loop, due to the inclusion of new *transfer_completed* events. An upper bound to the event list length can be estimated by considering that, in the worst case local model transfer between each couple of satellites can be completed only if there is an active link between them and consequently a number of $N_{sat}(N_{sat}-1)/2$ *'link-on'* (and, potentially, *'link-off'*) events shall be also included in the list. For this reason, an upper bound to the event list length can be given by $(N_{sat}(N_{sat}-1)/2)^2$. Consequently, the complexity of the proposed strategy can be expressed by $\mathcal{O}\left(N_{sat}^2 \log\left(N_{sat}^4\right)\right)$.

In order to better clarify the presented communication strategy, let us introduce the example shown in Fig. 5. In particular, we consider a constellation of three satellites, i.e., $N_{sat} = 3$. Initial event list represented by link-on and link-off events is known because of orbital mechanics, and we assume these events to be represented by the initial and final extremes of the colored horizontal bars in Fig. 5, respectively. At time t_0, each node has only its own local model in memory, which we assume to have been trained, as a matter of example, for $N_{ep} = 1$ epoch, i.e., $M_i = \{\mathbf{w}_i^{0,1}\}$, $\forall i \in [0,\dots,2]$. However, at t_0 a link between nodes 0 and 1 becomes available, thus, the two nodes tries to share the models in their memories, i.e., node 0 try to send its model to node 1, and vice versa. Assuming this information transfer needs a time $\Delta_{t,0}$ such that from t_0 to $t_1 = t_0 + \Delta_{t,0}$ the link between 0 and 1 is always active, a transfer completed event occurs at t_1. Furthermore, at t_1 we have $M_0 = M_1 = \{\mathbf{w}_0^{0,1}, \mathbf{w}_1^{0,1}\}$ and $M_3 = \{\mathbf{w}_3^{0,1}\}$. Both nodes 0 and 1 will try to share their updated model lists with other nodes at t_1, but since no link is available, no further transfer happens. Next event occurs at t_2, when a link between nodes 1 and 2 becomes available. Thus, node 1 tries to share the models it has in its memory, i.e., both $\mathbf{w}_0^{0,1}$ and $\mathbf{w}_1^{0,1}$ with 2, and vice-versa. Again, assuming this information transfer needs a time $\Delta_{t,1}$ such that the link between nodes 1 and 2 remains active from t_2 to $t_3 = t_2 + \Delta_{t,1}$, transfer is possible and a transfer completed event happens at t_3. Thus, at this time $M_0 = \{\mathbf{w}_0^{0,1}, \mathbf{w}_1^{0,1}\}$ and $M_1 = M_2 = \{\mathbf{w}_0^{0,1}, \mathbf{w}_1^{0,1}, \mathbf{w}_2^{0,1}\}$ and both nodes 1 and 2 try to share their updated model lists with other nodes. In particular, since at t_3 a link between nodes 0 and 1 is active and will be still available during the full time span $\Delta_{t,2}$ needed to transmit $\mathbf{w}_2^{0,1}$ (the only local model being in node 1 memory but not in node 0 one), a transmission completed event occurs at $t_4 = t_3 + \Delta_{t,2}$. Since at this time we have that $M_0 = M_1 = M_2 = \{\mathbf{w}_0^{0,1}, \mathbf{w}_1^{0,1}, \mathbf{w}_2^{0,1}\}$, the local model distribution phase is over and the local learning phase can begin.

4.3. Options of the distributed learning procedure

We will consider two schemes in the proposed distributed learning strategy. In the first one, hereafter named "DL w/ GS", we assume that ground stations can contribute to model distribution as relay nodes, i.e., when a satellite A flies over a ground station, it shares the models it has in its memory with the ground, in such a way that when another satellite B flies over a ground station, it can receive the models memorized in A directly from the ground, without having

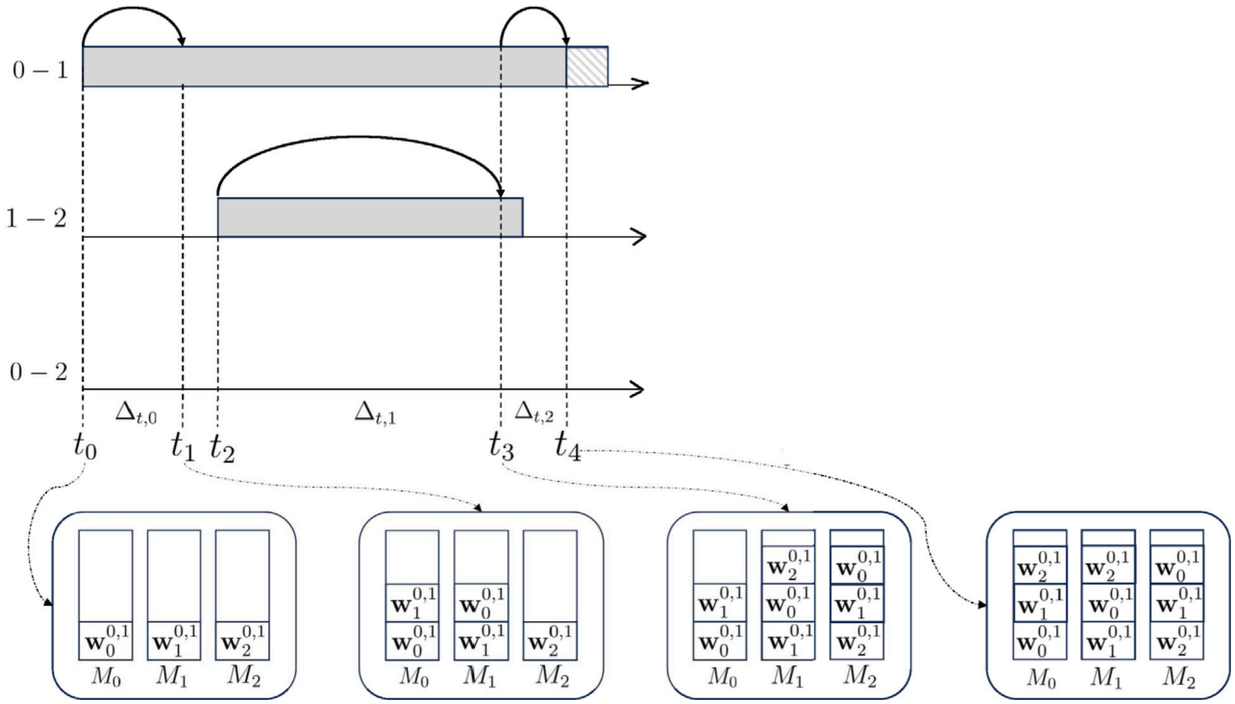


Fig. 5. Example of the proposed strategy with 3 satellites. Gray horizontal bars indicates the availability of a link between a couple of satellites in time, thus, their limits represent link-on and link-off events. Transfer completed events are represented by curved arrow ends. Dashed bars indicates events that are not included in the new round event list. At the bottom, memories of nodes at t_0, t_1, t_3 and t_4 are shown.

to wait to communicate with A. Please notice that ground stations only act as models relay, and they do not have to receive all local models from satellites to calculate the global model, as it happens in federated learning solutions. Furthermore, we assume that ground stations are all interconnected in such a way that communicating with one of them is equivalent to communicate with all of them. For this reason, if a ground station receives models from a satellite, these are immediately available on any other ground station, too. Instead, the second distributed learning strategy considered, named "DL w/o GS", does not provide for involving ground stations as relay nodes, and satellites can share models only by means of in-orbit communications.

5. Numerical results

We evaluate the performance of the proposed distributed learning solution with respect to federated learning-based one [33]. The only option "DL w/o GS" illustrated in Section 4.3 will be investigated in the case studies in which only one GS is considered because the advantages of the "DL w/ GS option is negligible in this case. In the scenario with more than one GS we will report the results for both the options "DL w/ GS" and "DL w/o GS". As benchmark solution we will consider the centralized federated learning solution illustrated in Section 4 while, because the decentralized federated learning solutions suffer for the intermittency of satellite links, they will not considered in the comparison. As far as centralized federated learning-based schemes considered as benchmarks are concerned, we consider a strategy leveraging both intra-orbital and inter-orbital ISLs ("FL w/ all ISLs"), a strategy where no inter-orbital communication is possible because of lack of inter-orbital ISLs ("FL w/o inter-orbital ISLs"), and a strategy providing for satellites having no ISLs ("FL w/o ISLs"), i.e., satellites cannot communicate with each other. All the Federated Learning solutions provide for having ground stations as parameter servers which, at each round, receive all the local models and aggregate them into a single global model which is transmitted to all satellites at the beginning of the following round. Furthermore, in this work we will assume that no aggregation of a partial number of local models happens on nodes neither in case of distributed learning-based solutions nor in federated learning-based ones.

Table 3
Proposed and benchmark learning solutions in relation to the use of GS as model rely node, intra-orbital ISL and inter-orbital ISL.

Learning solution	GS as relay node	Intra-orbital ISL	Inter-orbital ISL
"DL w/ GS"	Yes	Yes	Yes
"DL w/o GS"	No	Yes	Yes
"FL w/ all ISLs"	No	Yes	Yes
"FL w/o inter-orbital ISLs"	No	Yes	No
"FL w/o ISLs"	No	No	No

We report in Table 3 the investigated learning solutions in relation to the use of GS as model rely node, intra-orbital ISL and inter-orbital ISL.

Next we report the experimental setup and the results in Sections 5.1 and 5.2 respectively.

5.1. Experimental setup

The comparison of the proposed solution and the benchmark ones is carried out as a function of the choice of the satellite network topology in terms of number of orbital planes and number of satellites in each orbital plane. In this way we will be able to assess the impact of the inter-orbital link unavailability on the effectiveness of the learning procedure.

The parameters values used in the experiments are reported in Table 3. Some of them are varying such as the number of satellites N_{sat}, the number of orbital planes N_{op}, the number of model parameters (i.e., weights and biases) N_{mp}, the duration of the local learning phase τ_l, the number of ground stations N_{GS} and their location. Please notice that, since model parameters are usually expressed in *float32* format, it easy to obtain the model size in bit given N_{mp}, since $w_s = 32 \cdot N_{mp}$ bit. The values of each parameter will be specified for each of the following analysis. Instead, the remaining parameters will be the same for each analysis. In particular, we will assume to have a Walker constellation,

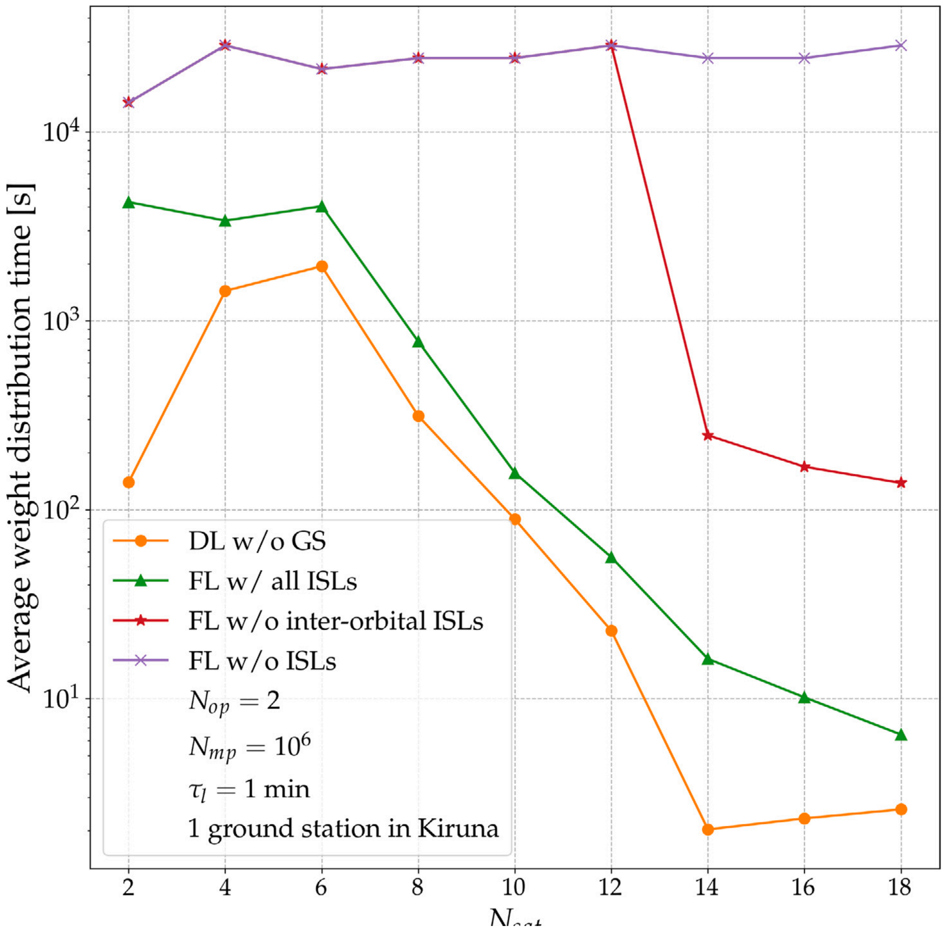


Fig. 6. Average weight distribution time for different distributed learning-based and federated learning-based strategies, obtained by varying the number of satellites and by fixing the number of orbital planes $N_{op} = 2$, the number of model parameters $N_{mp} = 10^6$, the local learning time $\tau = 1$ min, and placing a single ground station in Kiruna (Sweden).

Table 4
Parameters values.

Parameter	Values
N_{sat}	$[2, \dots, 18]$
N_{op}	$[1, \dots, 5]$
h_p $p \in [0, \dots, N_{op}]$	712,84 Km
i_W	98.24 deg
T	2 sidereal days
N_{mp}	$[10^5, 10^6, 10^7, 10^8, 10^9]$
τ_l	[1 min, 10 min, 20 min, 30 min, 40 min, 50 min, 60 min]
P	10 W
G	34.31 dBi
R_{ISL}	200 Mbps
R_{GS}	200 Mbps
v_{tx}	26 GHz
T_s	290 K
B	500 MHz
El_{min}	5 deg

with circular orbits having altitude $h_p = 712.84$ km, inclination i_W 98.24 deg, repeat cycle time $T = 2$ sidereal days, transmission data rate on both ISLs and links to the ground stations $R_{ISL} = R_{GS} = 200$ Mbps, transmission power $P = 10$ W, antenna gain $G = 34.31$ dBi, transmission frequency $v_{tx} = 26$ GHz, system noise temperature $T_s = 290$ K, bandwidth $B = 500$ MHz, minimum elevation angle $El_{min} = 5$ deg (see Table 4).

The following performance indexes will be considered for the comparison of the various solutions:

- The average weight distribution time, that is time needed to distribute the weights in a round as illustrated in Figs. 3 and 4; notice how this index allows to prove the effectiveness of the proposed solution regardless the training dataset considered.
- The convergence time and validation accuracy evaluated in the case of an image classification application trained on the EuroSAT [21] dataset.

5.2. Results

First analysis focuses on the impact of the number of satellites in an orbital plane on the average weight distribution time. For this analysis, we consider the number of satellites to be $N_{sat} \in [2, 4, \dots, 18]$, equally distributed over $N_{op} = 2$ orbital planes, and we consider to have only a ground station placed in Kiruna (Sweden), a typical choice for orbits with the chosen inclination, a local learning time on satellites $\tau_l = 1$ min, and model having a number of parameters $N_{mp} = 10^6$. Results in Fig. 6 show that any distributed learning-based strategy outperforms any federated learning-based strategy, regardless the number of satellites. However, it can be noticed that the average weight distribution time when a distributed learning-based strategy is applied first increases with the number of satellites, then it starts decreasing, and finally it increases again. This is due to the fact that by increasing the number of satellites, we increase the number of models that each satellite has to receive to calculate the global model, thus, we increase the number of models to be shared. At the same time, by increasing

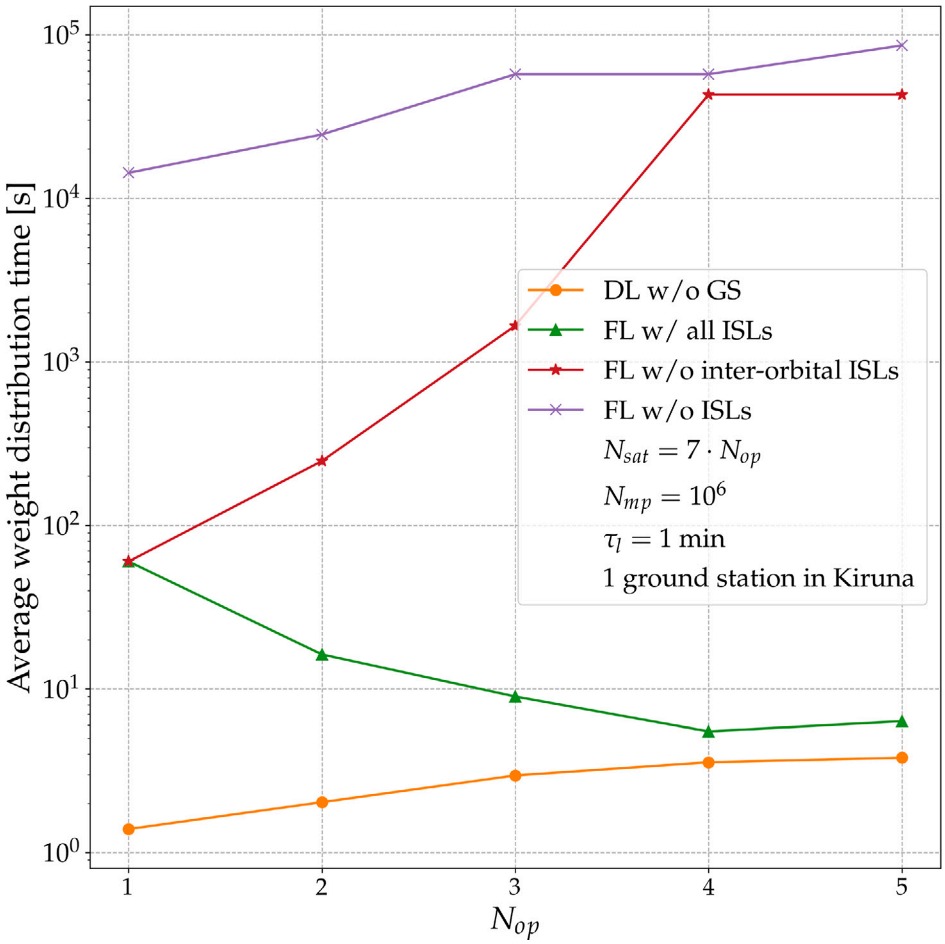


Fig. 7. The average weight distribution time for different distributed learning-based and federated learning-based strategies, obtained by varying the number of orbital planes N_{op}, by considering 7 satellites on each orbital planes, $N_{mp} = 10^6$ model parameters, local learning time $\tau = 1$ min, and placing a single ground station in Kiruna (Sweden).

the number of satellites without changing the number of orbital planes, each orbit has an increased number of satellites, and this increases the communication possibilities within satellite couples. However, for $2 < N_{sat} \leq 6$, the increase in communication opportunities is not enough to allow the sharing of an increased number of models in the same time span as in case $N_{sat} = 2$, thus, the average weight distribution time increases. Instead, for $N_{sat} > 6$, the increased number of communications between satellites is such that, even though the number of models to be shared increases, a complete sharing is achievable in a smaller time than in case $N_{sat} = 6$, thus, the average weight distribution time decreases. In particular, for $N_{sat} \geq 10$, in the same amount of time it is even possible to share an increased number of models with respect to the case $N_{sat} = 2$, thus, we obtain an average weight distribution time lower than the one obtained when $N_{sat} = 2$. However, for $N_{sat} > 14$, the average weight distribution time starts increasing again. This is due to the fact that, as it can be easily verified by means of expressions in (4), the intra-plane ISLs become always active for $N_{sat} \geq 14$, while they are never active when $N_{sat} < 14$. Thus, for $N_{sat} \geq 14$, as soon as a satellite receives models, it transmits them to satellites in the same orbital plane, with a delay only depending on the amount of models to be transferred and on the propagation time, without having to wait for an intra-plane ISLs to become available. Thus, the model sharing is completed shortly after any couple of satellites appertaining to different orbital planes is able to communicate. However, the average weight distribution time is minimized exactly when $N_{sat} = 14$, since for a higher number of satellites, even though inter-plane communications happen slightly earlier because of the increased number of satellites and, consequently, of the inter-plane communication opportunities, this does not compensate for the increase in the amount of data to be exchanged because of the increased number of models to be shared. Thus, for $N_{sat} > 14$, there is a mild increase in the average weight distribution time. Moving to federated learning-based strategies, it can be noticed that the best performance is obtained when all ISLs are leveraged, but the average weight distribution time is higher than the one obtained in case of distributed learning strategies, because in federated learning schemes first all models have to reach the ground, where the global model is centrally determined, and then the ground station has to uplink the updated global model to all satellites. This also explains why, by increasing the number of satellites, we have an overall decrease in the average weight distribution time, since the higher the number of satellites is, the higher the number of communication opportunities with the ground is. Finally, it can be noticed that “FL w/o inter-orbital ISLs” and “FL w/o ISLs” have the same behavior for $N_{sat} \leq 12$, while the latter leads to an decrease in the average weight distribution time for $N_{sat} > 12$. This is due to the fact that, as discussed before, for $N_{sat} \leq 12$, no intra-orbital ISL is available, while they are always available for $N_{sat} \geq 14$. Obviously, the worst performance in terms of the average weight distribution time is obtained with the “FL w/o ISLs” solution, since by means of this strategy we have to wait for all satellites to fly over the ground to transmit their models, and then, after the global model has been aggregated on the ground, we have to wait again for all satellites to fly over the ground to receive it.

In Fig. 7, we analyze the impact of the number of orbital planes, chosen to be $N_{op} \in [1, \dots, 5]$, on the average weight distribution time when the different strategies previously introduced are applied. In particular, in this analysis we consider again a number of model

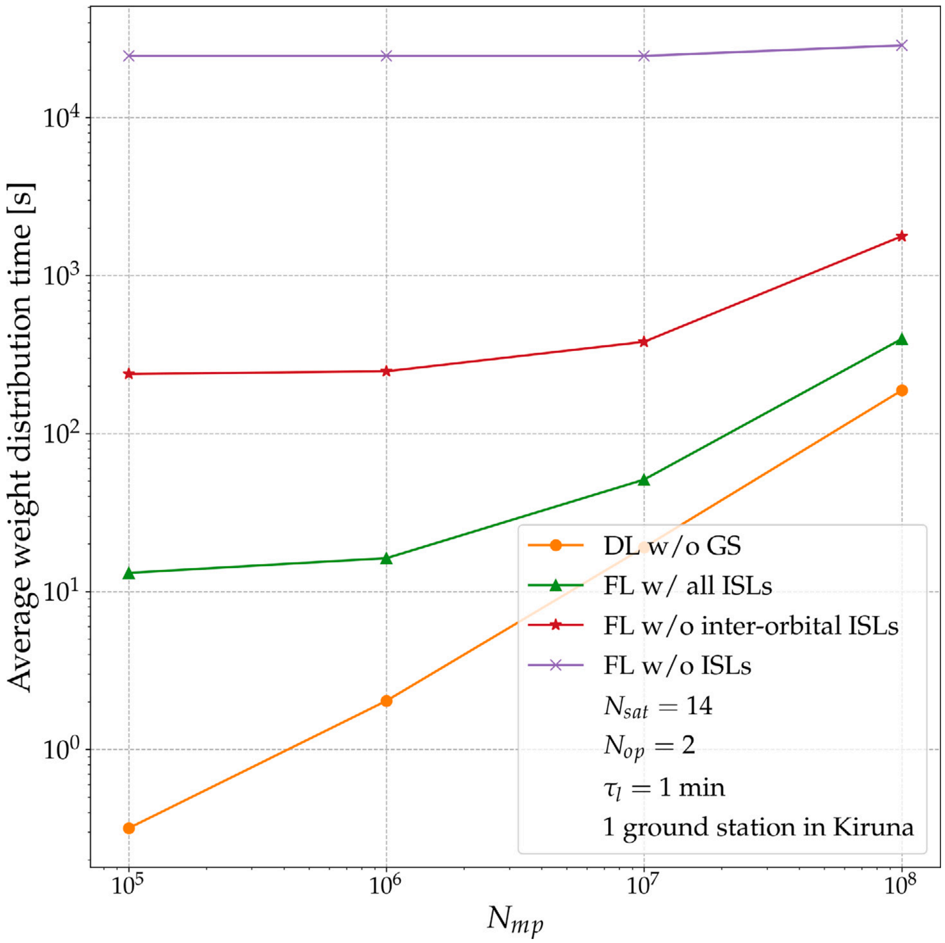


Fig. 8. The average weight distribution time for different distributed learning-based and federated learning-based strategies, obtained by varying the number of model parameters and by fixing the number of satellites $N_{sat} = 14$, the number of orbital planes $N_{op} = 2$, the local learning time $\tau = 1$ min, and placing a single ground station in Kiruna (Sweden).

parameters $N_{mp} = 10^6$, a learning time $\tau_l = 1$ min and a single ground station placed in Kiruna. Instead, the number of satellites will be equal to $N_{sat} = 7N_{op}$, in such a way that each orbital plane has 7 satellites and intra-orbital ISLs are always active, as previously discussed. Again, distributed learning-based solutions outperform the federated learning-based ones. However, it can be noticed that by increasing the number of orbital planes, the average weight distribution time increases in case distributed learning-based strategies are applied, while it decreases up to a maximum in case “FL w/ all ISLs” is considered. The behavior of the distributed learning solutions is due to the fact that, even though by increasing the number of orbital planes we increase the communication opportunities among satellites on different orbital planes, again this increase is not enough to allow for the sharing of an increased number of models to conclude a learning round in the same time span, leading to an overall increase in the average weight distribution time. However, this increase becomes milder by increasing the number of orbital planes, since this reduces the distances between couples of satellites on different orbits, increasing the number of communication opportunities until this is high enough to allow for the sharing of an increased number of models in almost the same time span. Instead, by looking at “FL w/ all ISLs” solution, it is possible to notice that the average weight distribution time decreases for $N_{op} \leq 4$, and increases when $N_{op} > 4$. This is due to the fact that, by increasing the number of orbital planes, we both increase the communication opportunities among satellites appertaining to different orbital planes, but we also increase the communication opportunities among satellites and ground. This leads to the decrease in the average weight distribution time for $N_{op} \leq 4$. However, when $N_{op} = 4$, any couple of satellites can communicate, regardless the occupied orbital plane, because intra-orbital ISLs are always active and the position of orbital planes is such that at any time there is at least a satellite of an orbital plane being able to communicate with a satellite of any other plane. Furthermore, for this number of orbital planes, at any time there is at least a satellite being able to communicate with the ground. It follows that any satellite in the constellation can communicate with the ground station at any time. Obviously, this property will be still valid by increasing the number of orbital planes, and the number of communication links between orbital planes will even increase with N_{op}. However, for $N_{op} > 4$, we have to share with the ground a higher number of models because of the increased number of satellites, and this requires a higher amount of time because of the higher amount of data to be transmitted, leading to an increase in the average weight distribution time with respect to the case $N_{op} = 4$, since the increased number of communication opportunities is not high enough to allow for the increased number of models to be transmitted in the same time span. Finally, in case no inter-orbital ISL is leveraged, like in “FL w/o inter-orbital ISLs” and “FL w/o ISLs” solutions, there is no advantage in increasing the number of orbital planes, and the average weight distribution time increases when N_{op} increases because of the increased number of satellites having to communicate with the ground to complete a learning round.

We also analyze the impact of the number of model parameters on the average weight distribution time. In particular, we considered

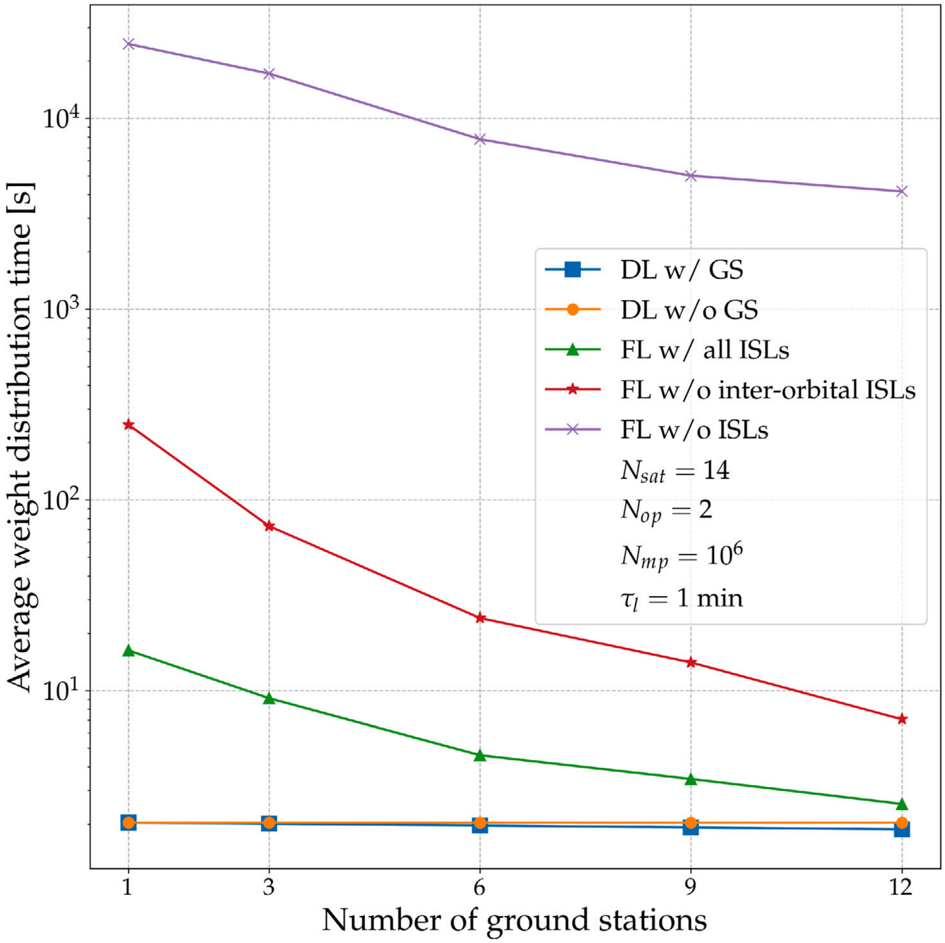


Fig. 9. The average weight distribution time for different distributed learning-based and federated learning-based strategies, obtained by varying the number of ground stations and by fixing the number of satellites $N_{sat} = 14$, the number of orbital planes $N_{op} = 2$, the number of model parameters $N_{mp} = 10^6$, and the local learning time $\tau = 1$ min.

$N_{mp} \in \{10^5, 10^6, 10^7, 10^8\}$, we fixed the number of satellites $N_{sat} = 14$, distributed over $N_{op} = 2$ orbital planes, we assumed a local learning time $\tau_l = 1$ min and a single ground station placed in Kiruna. Please notice that, even though the local learning time is actually dependent on the number of model parameters, it also depends on the available computational capacity. For this reason, we left the local learning time as an analysis parameter that will be investigated further on. Results shown in Fig. 8 allow to conclude that distributed learning-based strategies outperform the federated learning-based ones for any value of the number of model parameters. However, the average weight distribution time increases when N_{mp} increases. This is due to the fact that, by increasing the number of parameters, we increase the amount of data to be transmitted, and, consequently, the time needed to accomplish the data transfer, making each learning round longer. However, in case of distributed learning, since we do not have to wait to first transfer all local models to the ground and then receive the updated global model from the Earth, we have an overall shorter duration of model distribution phase, which allows for completing a round in a shorter time and, consequently, to have a lower average weight distribution time.

Results in Fig. 9 give insight on the impact of the number of ground stations on the average weight distribution time for the different considered strategies. In this case study we report the results for both the options "DL w/ GS" and "DL w/o GS". In particular, we set the number of satellites $N_{sat} = 14$, the number of orbital planes $N_{op} = 2$, the number of model parameters $N_{mp} = 10^6$, the local learning time $\tau_l = 1$ min, and we consider an increasing number of ground stations, mainly provided by Amazon Web Services [35], grouped as follows:

- in case of a single ground station, we consider it to be placed in Kiruna (Sweden);
- in case of three ground stations, we consider them to be placed in Kiruna (Sweden), Matera (Italy) and Kourou (French Guyana);
- in case of six ground stations, we add ground stations placed in Hawaii, Punta Arenas (Chile) and Singapore to the previous group;
- in case of nine ground stations, we add ground stations placed in Ohio (USA), Cape Town (South Africa) and Sidney (Australia) to the previous group;
- in case of twelve ground stations, we add ground stations placed in Oregon (USA), Bahrein and Seoul (South Korea) to the previous group.

From Fig. 9 it is possible to notice that the average weight distribution time is lower in all solutions providing for leveraging ground stations, i.e., "DL w/ GS" and the three federated learning-based strategies when the number of ground stations increases, as a consequence of the fact that there are more communication possibilities with ground stations when their number increases. Furthermore, it is important to underline that since we consider the ground stations being interconnected, as soon as a model is available on one of them, it will be

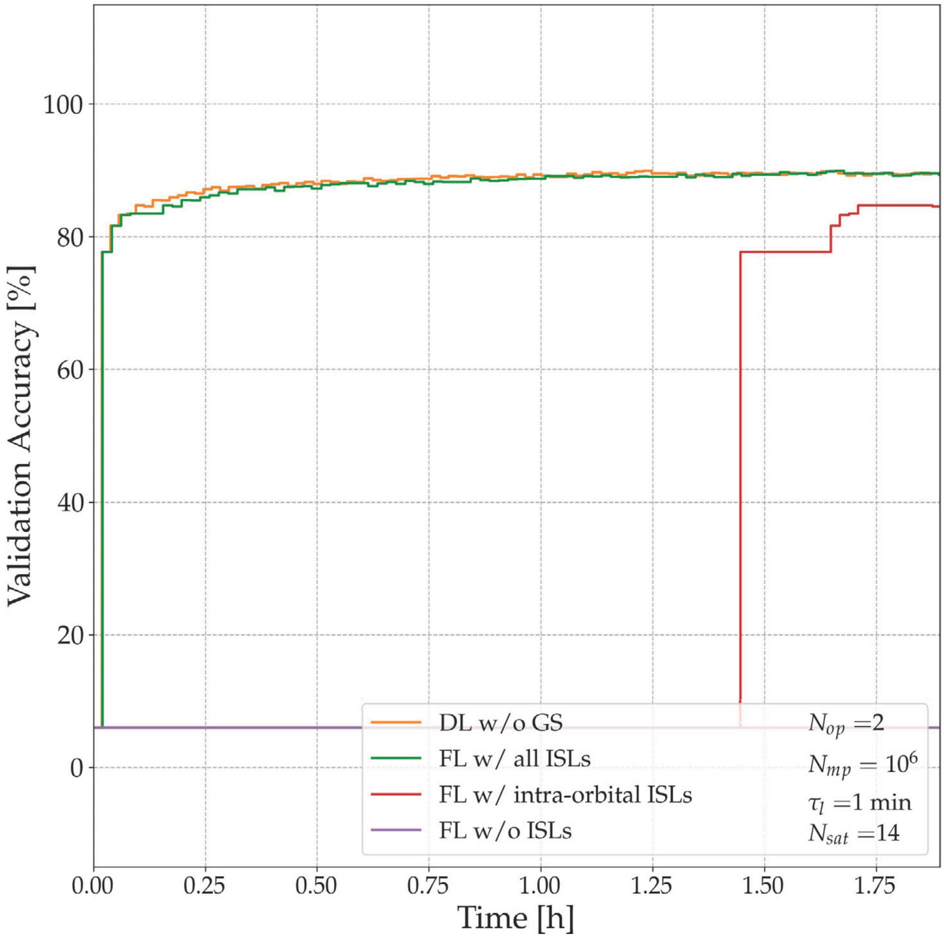


Fig. 10. Validation accuracy in training a VGG16-based satellite image classification model on the EuroSAT dataset by applying different distributed learning-based and federated learning-based strategies, obtained by setting the number of satellites $N_{sat} = 14$, the number of orbital planes $N_{op} = 2$, the number of model parameters $N_{mp} = 10^6$, the local learning time $\tau = 1$ min and a ground station in Kiruna (Sweden).

immediately available on each of them, facilitating the model sharing within the constellation, since a model can be transferred between two satellites not being able to directly communicate but simultaneously flying over two different ground stations. This reflects in the fact that the “DL w/ GS” strategy allows for weight distribution time lower than the “DL w/o GS”, and the “FL w/ all ISLs” solution achieves almost the same performance as the distributed learning-based ones for a high number of ground stations.

Finally, we evaluate the time of convergence of the validation accuracy when the different strategies are applied. For this analysis we consider to have $N_{sat} = 14$ satellites distributed over $N_{op} = 2$ orbital planes, a local learning time $\tau_l = 1$ min and a single ground station placed in Kiruna. We also set the maximum number of learning rounds to be $N_r = 100$ and the number of local learning epochs $N_{ep} = 1$. We consider a land cover classification task based on the EuroSAT dataset [21], made of 27000 64 × 64 images, taken by Sentinel-2. Images are classified with respect to 10 classes (AnnualCrop, Forest, HerbaceousVegetation, Highway, Industrial, Pasture, PermanentCrop, Residential, River, SeaLake), depending on the represented scene. We are aware of the fact that training a classification model requires labeled data, and this may be not the case when considering images acquired from satellites to be used for training without previously transmitting them to the ground. However, since the focus of this paper is on the communication strategy underlying the learning algorithm, we only want to provide some insight on the performance of the communication scheme by focusing on a general machine learning task, as it happens also in other works [20,31–33,36], since this insight may be also extended to more sophisticated machine learning techniques, like self-supervised learning, which are beyond the scope of this work. Furthermore, it is important to underline that, given the model to be trained, the chosen strategy does not influence the accuracy obtained when training converges, but only the time needed to reach validation convergence. Following [37], we consider VGG16 as classification model, pre-trained on ImageNet dataset, to which a regular densely-connected neural network of 2048 units with ReLU activation function, a dropout layer with 0.2 drop rate, and a regular densely-connected neural network of 10 units with softmax activation function are added. We will assume that only these added layers will be trained, thus, the number of model parameters to be exchanged will be equal to $N_{mp} = 4216842$. Differently from [37], we assume that the model is not trained on a central node, but each satellite trains its own local model during local training phase. In particular, each satellite will have a different dataset, since satellites fly over different areas. However, after some orbits, each satellite will have flown over a high amount of different areas, thus, we suppose that in datasets of each satellite we have samples for all classes, but the distribution of samples on each orbital plane with respect to the different classes is different. In particular, we randomly split the initial dataset to separate a 20% of samples for validation. For each image in the training set, we generate a randomly rotated version and a noisy version of it, and we add the

Table 5
Time to converge.

Strategy	Time to converge
DL w/o GS	1.25
FL w/ all ISLs	1.68
FL w/ intra-orbital ISLs	4.13
FL w/o ISLs	558 (23.3 days)

two new images to the training set for augmentation purposes. In order to obtain the training sets associated to each satellite, for each class we split the samples between the two orbital planes in random proportion, and the samples for each class associated to each orbital plane are equally and randomly associated to each satellite on the orbital plane. This also allows to obtain non-IID training sets on the satellites. We thus evaluate the accuracy of the global model (i.e., of the model obtained by aggregating the locally trained models) on the validation set for each learning round. Since we know how long each learning round lasts when the different strategies are applied, it is easy to report the validation accuracy in time, as shown in Fig. 10. We also evaluate the time to converge as the time at which validation accuracy reaches a value that is not improved in the next 10 learning rounds. Values of the time to converge are summarized in Table 5.

From presented results, it can be noticed that by using strategies providing for the use of inter-orbital ISLs, like in case of distributed learning-based and “FL w/ all ISLs” schemes, validation accuracy converges much faster than in case of strategies where inter-orbital ISLs are not used. This is due to the fact that, as previously discussed, by leveraging inter-orbital ISLs we increase the communication opportunities among satellites. Furthermore, distributed learning-based strategies, thanks to a reduced duration of the model distribution phase, allows to reach convergence in a shorter time than any federated learning-based solution.

6. Conclusions

In this work, we proposed and evaluated a distributed learning solution in the context of EO constellations with satellites forming a network by means of ISLs. This solution differs from federated learning one in the fact that there is no central node which has to receive the local models to aggregate them in an updated version of the global model, since we assume that satellites share local models with each other until each satellite has received the local models of the others, in order to locally calculate the updated global model. Numerical results show that distributed learning outperforms federated learning in number of learning rounds completed in the unit time by increasing the number of satellites, of orbital planes, of model parameters, of ground stations and by increasing the time needed to accomplish local learning. This translates in a faster test accuracy convergence, as evaluated in a land coverage classification task based on the EuroSAT dataset.

CRediT authorship contribution statement

Francesco Valente: Writing – original draft, Investigation. **Francesco G. Lavacca:** Supervision, Conceptualization. **Tiziana Fiori:** Validation. **Vincenzo Eramo:** Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] B. Denby, B. Lucia, Orbital edge computing: Nanosatellite constellations as a new class of computer system, in: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS ’20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 939–954, http://dx.doi.org/10.1145/3373376.3378473.

[2] P. Cassará, A. Gotta, M. Marchese, F. Patrone, Orbital edge offloading on Mega-LEO satellite constellations for equal access to computing, IEEE Commun. Mag. 60 (4) (2022) 32–36, http://dx.doi.org/10.1109/MCOM.001.2100818.

[3] Y. Zhang, Q. Wu, Z. Lai, H. Li, Enabling low-latency-capable satellite-ground topology for emerging LEO satellite networks, in: IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, 2022, pp. 1329–1338, http://dx.doi.org/10.1109/INFOCOM48880.2022.9796886.

[4] Z. Jia, M. Sheng, J. Li, D. Zhou, Z. Han, VNF-based service provision in software defined LEO satellite networks, IEEE Trans. Wireless Commun. 20 (9) (2021) 6139–6153, http://dx.doi.org/10.1109/TWC.2021.3072155.

[5] F. Rinaldi, H.-L. Maattanen, J. Torsner, S. Pizzi, S. Andreev, A. Iera, Y. Koucheryavy, G. Araniti, Non-terrestrial networks in 5G & beyond: A survey, IEEE Access 8 (2020) 165178–165200, http://dx.doi.org/10.1109/ACCESS.2020.3022981.

[6] G. Araniti, A. Iera, S. Pizzi, F. Rinaldi, Toward 6G non-terrestrial networks, IEEE Netw. 36 (1) (2022) 113–120, http://dx.doi.org/10.1109/MNET.011.2100191.

[7] G. Giuffrida, L. Fanucci, G. Meoni, M. Batič, L. Buckley, A. Dunne, C. van Dijk, M. Esposito, J. Hefele, N. Vercruyssen, G. Furano, M. Pastena, J. Aschbacher, The Φ-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–14, http://dx.doi.org/10.1109/TGRS.2021.3125567.

[8] G. Mateo-Garcia, J. Veitch-Michaelis, L. Smith, S.V. Oprea, G. Schumann, Y. Gal, A.G. Baydin, D. Backes, Towards global flood mapping onboard low cost satellites with machine learning, Sci. Rep. 11 (1) (2021) 1–12, http://dx.doi.org/10.1038/s41598-021-86650-z.

[9] H. Chen, M. Xiao, Z. Pang, Satellite-based computing networks with federated learning, IEEE Wirel. Commun. 29 (1) (2022) 78–84, http://dx.doi.org/10.1109/MWC.008.00353.

[10] N. Razmi, B. Matthiesen, A. Dekorsy, P. Popovski, On-board federated learning for dense LEO constellations, in: ICC 2022 - IEEE International Conference on Communications, 2022, pp. 4715–4720, http://dx.doi.org/10.1109/ICC45855.2022.9838619.

[11] L. He, J. Wang, F. Wang, E. Lansard, C. Yuen, Energy-efficient data offloading for earth observation satellite networks, 2024, pp. 1–7, http://dx.doi.org/10.48550/arXiv.2401.06419, Arxiv.

[12] L. He, K. Guo, H. Gan, L. Wang, Collaborative data offloading for earth observation satellite networks, IEEE Commun. Lett. 26 (5) (2022) 1116–1120, http://dx.doi.org/10.1109/LCOMM.2022.3151657.

[13] F. Valente, V. Eramo, F.G. Lavacca, Optimal bandwidth and computing resource allocation in low earth orbit satellite constellation for earth observation applications, Comput. Netw. 232 (2023) 109849, http://dx.doi.org/10.1016/j.comnet.2023.109849.

[14] Y. Wang, J. Che, N. Wang, L. Liu, N. Wu, X. Zhong, X. Han, Load-balancing method for LEO satellite edge-computing networks based on the maximum flow of virtual links, IEEE Access 10 (2022) 100584–100593, http://dx.doi.org/10.1109/ACCESS.2022.3207293.

[15] T. Kim, J. Kwak, J.P. Choi, Satellite edge computing architecture and network slice scheduling for IoT support, IEEE Internet Things J. 9 (16) (2022) 14938–14951, http://dx.doi.org/10.1109/JIOT.2021.3132171.

[16] M. Jia, L. Zhang, J. Wu, Q. Guo, X. Gu, Joint computing and communication resource allocation for edge computing towards huge LEO networks, China Commun. 19 (8) (2022) 73–84, http://dx.doi.org/10.23919/JCC.2022.08.006.

[17] F. Valente, F.G. Lavacca, V. Eramo, A resource allocation strategy in earth observation orbital edge computing-enabled satellite networks to minimize ground station energy consumption, in: NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, 2023, pp. 1–6, http://dx.doi.org/10.1109/NOMS56928.2023.10154220.

[18] K.S. Basavaraju, N. Sravya, S. Lal, J. Nalini, C.S. Reddy, F. Dell’Acqua, UCDNet: A deep learning model for urban change detection from bi-temporal multispectral sentinel-2 satellite images, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–10, http://dx.doi.org/10.1109/TGRS.2022.3161337.

[19] N.E. Khalifa, M. Loey, S. Mirjalili, A comprehensive survey of recent trends in deep learning for digital images augmentation, Artif. Intell. Rev. (2022) 1–27, http://dx.doi.org/10.1007/s10462-021-10066-4.

[20] B. Matthiesen, N. Razmi, I. Leyva-Mayorga, A. Dekorsy, P. Popovski, Federated learning in satellite constellations, IEEE Netw. (2023) 1–16, http://dx.doi.org/10.1109/MNET.132.2200504.

[21] P. Helber, B. Bischke, A. Dengel, D. Borth, Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, in: IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018, pp. 204–207, http://dx.doi.org/10.1109/IGARSS.2018.8519248.

[22] Z. Lai, W. Liu, Q. Wu, H. Li, J. Xu, J. Wu, Spacertc: Unleashing the low-latency potential of mega-constellations for real-time communications, in: IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, 2022, pp. 1339–1348, http://dx.doi.org/10.1109/INFOCOM48880.2022.9796887.

[23] F. Tang, H. Hofner, N. Kato, K. Kaneko, Y. Yamashita, M. Hangai, A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN), IEEE J. Sel. Areas Commun. 40 (1) (2022) 276–289, http://dx.doi.org/10.1109/JSAC.2021.3126073.

[24] S. Saafi, O. Vikhrova, G. Fodor, J. Hosek, S. Andreev, AI-aided integrated terrestrial and non-terrestrial 6G solutions for sustainable maritime networking, IEEE Netw. 36 (3) (2022) 183–190, http://dx.doi.org/10.1109/MNET.104.2100351.

[25] H. Chen, M. Xiao, Z. Pang, Satellite-based computing networks with federated learning, IEEE Wirel. Commun. 29 (1) (2022) 78–84, http://dx.doi.org/10.1109/MWC.008.00353.

[26] Z. Lai, Q. Wu, H. Li, M. Lv, J. Wu, OrbitCast: Exploiting mega-constellations for low-latency earth observation, in: 2021 IEEE 29th International Conference on Network Protocols, ICNP, 2021, pp. 1–12, http://dx.doi.org/10.1109/ICNP52444.2021.9651919.

[27] Y. Qiu, J. Niu, X. Zhu, K. Zhu, Y. Yao, B. Ren, T. Ren, Mobile edge computing in space-air-ground integrated networks: Architectures, key technologies and challenges, J. Sensor Actuator Netw. 11 (4) (2022) http://dx.doi.org/10.3390/jsan11040057.

[28] T. Kim, J. Kwak, J.P. Choi, Satellite edge computing architecture and network slice scheduling for IoT support, IEEE Internet Things J. 9 (16) (2022) 14938–14951, http://dx.doi.org/10.1109/JIOT.2021.3132171.

[29] S. Yu, X. Gong, Q. Shi, X. Wang, X. Chen, EC-SAGINs: Edge-Computing-Enhanced Space–Air–Ground-Integrated Networks for Internet of Vehicles, IEEE Internet Things J. 9 (8) (2022) 5742–5754, http://dx.doi.org/10.1109/JIOT.2021.3052542.

[30] L. Yuan, L. Sun, P. Yu, Z. Wang, Decentralized Federated Learning: A Survey and Perspective, 2023, pp. 1–17, http://dx.doi.org/10.48550/arXiv.2306.01603, Arxiv.

[31] N. Razmi, B. Matthiesen, A. Dekorsy, P. Popovski, Ground-assisted federated learning in LEO satellite constellations, IEEE Wirel. Commun. Lett. 11 (4) (2022) 717–721, http://dx.doi.org/10.1109/LWC.2022.3141120.

[32] N. Razmi, B. Matthiesen, A. Dekorsy, P. Popovski, Scheduling for ground-assisted federated learning in LEO satellite constellations, in: 2022 30th European Signal Processing Conference, EUSIPCO, 2022, pp. 1102–1106, http://dx.doi.org/10.23919/EUSIPCO55093.2022.9909556.

[33] N. Razmi, B. Matthiesen, A. Dekorsy, P. Popovski, On-board federated learning for dense LEO constellations, in: ICC 2022-IEEE International Conference on Communications, IEEE, 2022, pp. 4715–4720.

[34] T. Catena, V. Eramo, M. Panella, A. Rosato, Distributed LSTM-based cloud resource allocation in network function virtualization architectures, Comput. Netw. 213 (2022) 109111, http://dx.doi.org/10.1016/j.comnet.2022.109111.

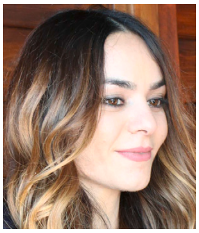
[35] Amazon, Amazon Web Services Ground Station Locations, 2023, Available at https://aws.amazon.com/ground-station/locations/. (Accessed 11 January 2023).

[36] N. Razmi, B. Matthiesen, A. Dekorsy, P. Popovski, On-board federated learning for satellite clusters with inter-satellite links, IEEE Trans. Commun. (2024) http://dx.doi.org/10.1109/TCOMM.2024.3356429, 1–1.

[37] N. Sonune, Land cover classification with EuroSAT dataset, 2023, Available at https://www.kaggle.com/code/nilesh789/land-cover-classification-with-eurosat-dataset (Accessed 16 September 2023).

Francesco Valente received the Bachelor's degree in Aerospace Engineering (cum laude) in 2018 and the Master's degree in Space and Astronautical Engineering (cum laude) in 2020, both from Sapienza University of Rome. In November 2020, he started his Ph.D. in Information and Communication Technologies at Sapienza University of Rome, Department of Information Engineering, Electronics, and Telecommunications. His research has involved collaboration on projects with the Italian Space Agency (ASI) and the aerospace industry. His current interests encompass Orbital Edge Computing, Artificial Intelligence applied to Network Function Virtualization, and Time-Sensitive Networking, including Deterministic Ethernet.

Francesco Giacinto Lavacca received the Laurea (M.Sc.) (cum laude) degree in electronic engineering in 2013 and the Ph.D. degree In Information Technology in 2017 from the Sapienza University of Rome (Italy). In 2018 he joined Fondazione Ugo Bordoni as researcher in the TLC group. Now, he is a Researcher at the DIET Department in Sapienza, where he started his postdoctoral research activities. His current research interests include all optical networks and switching architectures, 5G networks planning and design, Network Function Virtualization, and Time-Triggered and Deterministic Ethernet. In 2016, he has been a Visiting Researcher with the College of Computing, Georgia Institute of Technology, Atlanta, GA, USA. Furthermore, he was involved in the framework of national and international projects, like AAA (Advanced Avionic Architecture) and NMLV (Nano Micro Launch Vehicle) with ASI (Italian Space Agency) and ESA (European Space Agency) respectively.



Tiziana Fiori received the BS degree in Aerospace Engineering, and the MS degree in Space and Astronautical Engineering from Sapienza University of Rome, Italy, in 2018 and 2021, respectively. During 2021 she received a junior fellowship for the research activity of "Study of real-time Ethernet technologies for TLC networks of space launchers", and a temporary fellowship researcher for the project NIBBIO "Development of an experimental Real Time Ethernet test bed", in collaboration with the Italian Space Agency (ASI) from Sapienza University of Rome, Italy. In November 2021 she joined the Department of Information Engineering, Electronics and Telecommunications of Sapienza University of Rome as a Ph.D. student in Information and Communication Technologies (ICT) and Her research interests include real-time Ethernet technologies, space launcher networks, message scheduling. She is a member of the community N2Women.

Vincenzo Eramo received his Laurea degree in Electronics Engineering in 1995 and his Dottorato di Ricerca (Ph.D. degree) in Information and Communications Engineering in 2001, both from University of Roma La Sapienza. From June 1996 to December 1996 he was a researcher at the Scuola Superiore Reiss Romoli. In 1997, he became a researcher at the Fondazione Ugo Bordoni, in the Telecommunication Network Planning group. From November 2002 to October 2005 was an assistant professor and from November 2006 to June 2010 was an Aggregate Professor in the INFOCOM Department of Sapienza University of Rome. Vincenzo Eramo is currently an Associate Professor in the Department of Engineering of Information, Electronics, and Telecommunications. His research activities involve national and international projects, notably as the scientific coordinator for Sapienza University of Rome in two Networks of Excellence, E-PhotoONe+ and BONE, funded by the European Commissions (FP6 and FP7) during 2006–2007 and 2008–2011 respectively. These projects focused on studying Optical Networks. He served as Associate Editor for multiple IEEE journals, including IEEE Transactions on Computer (2011–2015), IEEE Communication Letters (since 2014), IEEE Transactions on Sustainable Computing (since 2016), and IEEE Open Journal of the Communications Society (since 2019). Additionally, he is Associate Editor for Optical Fiber Technology (Elsevier) since 2020 and ACM Computing Surveys since 2021. He was winner of the Exemplary Editor Award 2016 and 2017 of IEEE Communications Letter and the Exemplary Editor Awards 2021 of IEEE Open Journal of the Communications Society.