Olive Branch Learning: A Novel Federated Learning Framework for Space-Air-Ground Integrated Network

Qingze Fang, Shuai Yu and Xu Chen
School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
Email: fangqz@mail2.sysu.edu.cn, {yushuai, chenxu35}@mail.sysu.edu.cn

2021 International Conference on Space-Air-Ground Computing (SAGC) | 978-1-6654-0011-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/SAGC52752.2021.00015

***Abstract*—Space-air-ground integrated network (SAGIN) can provide global data collection to support various data-driven artificial intelligence (AI) services, especially in some remote areas where terrestrial communication facilities are unavailable (e.g., desert and isolated islands). However, the hierarchical data collection mode of SAGIN faces the challenges of redundant data collection, ineffective data transmission and privacy concerns. To this end, we first propose a novel configurable federated learning approach in SAGIN environment, namely Olive Branch Learning (OBL). Specifically, it consists of a ring structure LEO constellation in the space layer, and two-tier star structures imposed on the air nodes and IoRT devices in the air layer and the ground layer, respectively. Then, an efficient clustering and mapping algorithm is designed for OBL by taking into account the model bias of the air nodes as well as their geographic distributions. The objective is to minimize the time cost required to train the global model to achieve the preset accuracy. Extensive experimental results based on realistic learning datasets demonstrate the superior performance of our algorithm over the benchmark policies.**

***Keywords*-Space-air-ground integrated network, Internet of remote things, federated learning, edge computing.**

I. Introduction

With the rapid development of aerospace, deep-sea exploration and unmanned technologies, traditional Internet of things (IoT) devices can reach to more remote areas, such as isolated islands, desert hinterland and deep sea. In recent years, a new research field, Internet of remote things (IoRT) [1] appears and attracts attentions from both the industry and academia, since the environmental monitoring data collected by IoRT devices usually provides important information of the world. However, the remote areas are usually lack of terrestrial communication facilities (e.g., 5G base station). Thus, it is hard for the IoRT devices to forward their collected data to a centralized server for further processing. In addition, in the 6G era, the data samples can be used for AI models training to support a wide range of data-driven intelligent applications and services, such as remote sensing [2] and environmental monitoring [3]. Transmitting massive data directly from the IoRT devices to a centralized cloud server usually causes network congestion, long response time and privacy issues.

Fortunately, the space-air-ground integrated network (SAGIN) [4] and satellite edge computing (SEC) [5] can facilitate the IoRT data collection, transmission and processing by leveraging the global coverage, processing capacity and low transmission delay of low earth orbit (LEO) satellites. In SAGIN and SEC assisted IoRT, LEO satellites form a constellation to provide global coverage with low transmission delay. Although the LEO constellation can guarantee the global coverage, it is still hard for IoRT devices to directly communicate with LEO satellites in consideration of resource limits of IoRT devices in remote areas and the long communication distance between IoRT devices and satellites. To this end, air nodes (e.g., unmanned aerial vehicles, airships, and balloons, in the air layer of SAGIN) can help to collect the data from IoRT devices and forward to LEO satellites, since the air nodes flying at a low altitude can provide flexible ubiquitous accessibility [1].

Nevertheless, training AI models for intelligent applications through the SAGIN assisted IoRT still faces the following challenges. First, the network environment to perform AI model training is non-trivial. On the one hand, devices in SAGIN (i.e., LEO satellites, air nodes and IoRT devices) have limited and heterogeneous computation capabilities. On the other hand, the wireless connections between the devices are usually bandwidth-limited and intermittent. For example, LEO satellites only enable to connect to a ground station in a certain period of time (i.e., satellite coverage time, around 600 seconds) [6]. The complex network environment makes it difficult for massive IoRT devices to synchronize in the learning process. Second, the data on IoRT devices in heterogeneous environments is usually not Independently and Identically Distributed (non-i.i.d.) [7], causing the statistical heterogeneity problem in distributed optimization and may lead to low model accuracy and slow convergence speed. Last but not least, the direct transmission of data samples from the IoRT devices may cause privacy or security concerns while also incurring significant communication overhead. Thus, it makes sense to exchange trained models

This work was supported in part by the National Science Foundation of China (No. U1711265, No. 61972432, No. 62002397); the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No. 2017ZT07X355); the Pearl River Talent Recruitment Program (No. 2017GC010465); Guangdong Basic and Applied Basic Research Foundation (No. 2019A1515010030).

Corresponding Author: Shuai Yu.

978-1-6654-0011-4/21/$31.00 ©2021 IEEE
DOI 10.1109/SAGC52752.2021.00015

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on February 15,2023 at 13:23:41 UTC from IEEE Xplore. Restrictions apply.

instead of raw data among the devices.

As a state-of-the-art distributed learning method, federated learning [8] has a great potential to deal with the above challenges. It provides a distributed learning framework where IoRT devices can train their own models using their own data, while the upper layer nodes pull models from IoRT devices and perform model aggregation to obtain a global model. In addition, sharing models instead of original data helps to preserve privacy.

In this paper, we promote a tailored design of federated learning in the space-air-ground integrated network to achieve data-driven AI solutions for 6G. The main contribution of this paper are summarized as follows:

- We propose a novel configurable federated learning framework in the SAGIN environment, namely Olive Branch Learning (OBL). Note that, most of existing works [9], [10] only deploy classic federated learning on the aerial-ground integrated network. To the best of our knowledge, OBL is the first federated learning framework for the SAGIN.
- To deal with the topology dynamics of LEO satellites, as well as the non-i.i.d. data features on the IoRT devices, we also introduce a clustering and mapping algorithm for OBL to alleviate the negative affection of non-i.i.d. data on IoRT devices, and reduce the time overhead of model training.
- Evaluation results based on realistic learning datasets show that our algorithm is superior to the other two methods in terms of model accuracy and time cost.

The rest of this paper is organized as follows. In Section II, we introduce the related works most relevant to this paper. Section III illustrates the OBL framework, and presents the system model. The problem formulation is presented in Section IV. The proposed clustering and mapping algorithm is expounded in Section V. The performance evaluation is conducted in Section VI. At last, Section VII concludes this paper.

II. Related Works

Due to the ubiquitous and global-area coverage of space-air-ground integrated network (SAGIN), it is considered to be one of the key technologies of 6G [4]. SAGIN has a hierarchical network architecture which integrates satellite constellation in the space, air nodes (e.g., UAVs) in the air and end devices (e.g., IoT devices) in the ground. The above networks can work independently or inter-operationally. In addition, it is widely believed that 6G will be established on ubiquitous end devices to realize data-driven Artificial Intelligence (AI) in massive-scale networks (e.g., SAGIN) [8], [11]. As an emerging decentralized machine learning approach, federated learning (FL) has attracted much attentions in recent years. In the training phase of FL, participating devices collaboratively train a global model by leveraging their local data, thus, only model updates are transmitted. Due to complexity, highly dynamic nature and inherent heterogeneity of the SAGIN, how to efficiently deploy federated learning functions into SAGIN should be carefully considered. Existing works on the deployment focus on the air and ground segments of SAGIN. For example, Liu et al. [9] propose a federated learning-based air quality sensing framework in aerial-ground integrated network. In the framework, UAVs collaboratively monitor air quality index (AQI) and train the data locally, without sharing the raw data. The main objective is to predict AQI scale distribution with privacy protection. Zhang et al. [10] present a UAV-aided image classification approach for area exploration. In their scheme, federated learning is considered for training collected images in each UAV locally. Then, the UAVs send their local model to a ground server for model aggregation via a fading wireless channel. Through multiple rounds of training, a global model is achieved with the objective of reducing the communication cost between the UAVs and the ground server. All of these works focus on the deployment of federated learning in the air-ground integrated network but pay no attention to the space segment, an essential component of SAGIN. To the best of our knowledge, we are the first to propose a federated learning framework of SAGIN.

III. Framework Description and System Model

In this section, we first present the proposed OBL framework, followed by a description of our system model.

A. Olive Branch Learning Framework Description

We consider a federated learning based model training scenario in space-air-ground integrated network, as shown in Fig. 1, which comprises three layers: space layer, air layer, and ground layer. Satellites in the space layer are evenly distributed in the equatorial low earth orbit and air nodes in the air layer are evenly distributed over the equator. The IoRT devices in the ground layer utilize their local data to train their own model, while nodes in the space layer and the air layer serve as aggregation nodes.

The overall learning steps are described as follows. At the beginning of each global round, the satellites broadcast their models to all the IoRT devices, and each IoRT device trains its model with its local data. Then, the air nodes collect models of IoRT devices in their respective coverage areas for aggregation, that is, weight the models based on the amount of their training data, and perform a weighted average on the models to obtain an aggregated model [8]. After that, air nodes send their models to satellites, and each satellite aggregates the received models from its associated air nodes in the same way. This step is the end of the satellite round, and the satellites send their models back to the IoRT devices at the beginning of next satellite round. It is worth mentioning that the data distribution on the IoRT device is always non-i.i.d., and the aggregated model of each air

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on February 15,2023 at 13:23:41 UTC from IEEE Xplore. Restrictions apply.

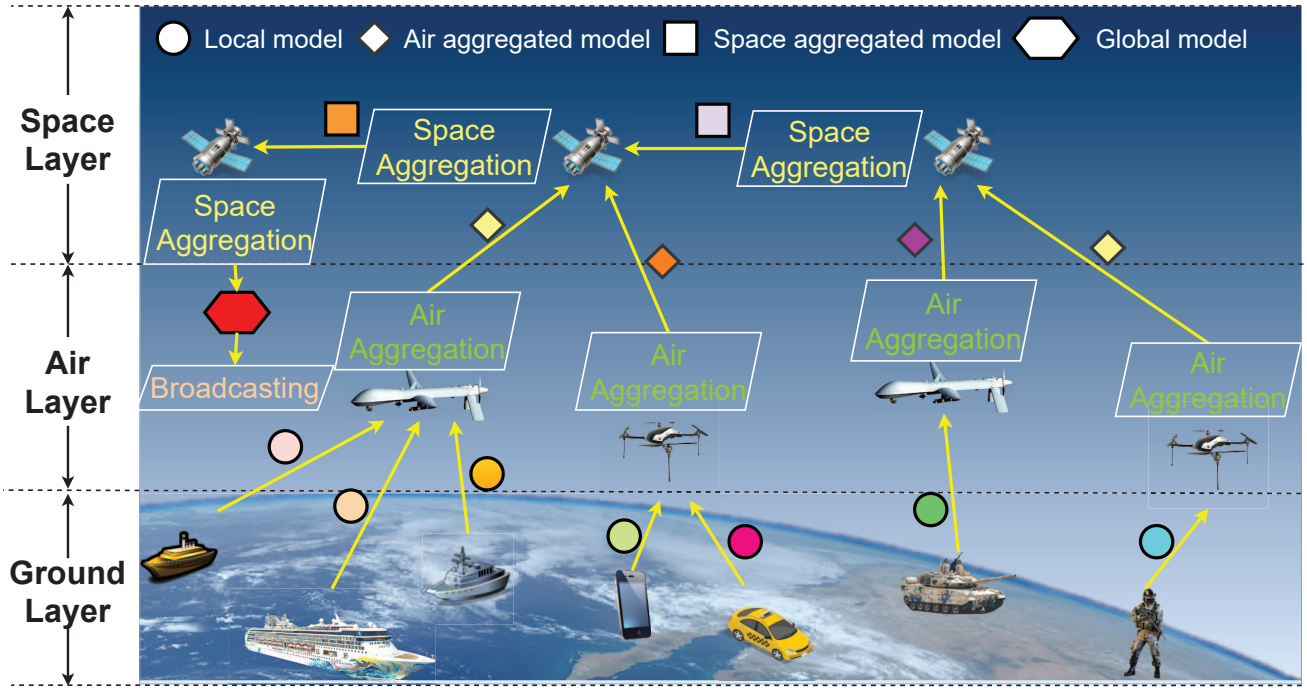


Figure 1. AI model training based on federated learning in space-air-ground integrated network.

node is biased towards the data owned by the IoRT devices covered by the air node, so it is significant to decide which air nodes send their models to the same satellite. After several satellite rounds, all the satellites cooperate to obtain a global model. Once all satellites have obtained a global model, the current global round ends, and the next global round starts.

Note that the satellites in the same orbit naturally form a ring structure for inter-satellite communications [12], and the Ring Allreduce algorithm [13] can be used for model aggregation among satellites. Whereas in the air and ground layer, network devices (i.e., air nodes and IoRT devices) prone to send models to upper-layer nodes that they access, thus forming star structures. The structure of our federated learning framework in SAGIN is shown in Fig. 2. Since the structure is very similar to the olive branch, we name it Olive Branch Learning (OBL). We believe that this structure is suitable for SAGIN. If we organize the satellites into a star structure, then one of the satellites is needed to serve as the master satellite to receive models from other satellites for aggregation, which increases the multi-hop communication time cost between satellites. It is unrealistic to organize IoRT devices or air nodes into a ring structure due to their large scale and highly-scattered distributions.

B. System Model

Assume that there are N_S LEO satellites and N_A air nodes, where each air node covers N_G IoRT devices on the ground without overlap. In other words, there are $N_A \cdot N_G$ IoRT devices in total. Suppose that the set of IoRT devices covered by air node A_i is $\{G_1, G_2, ..., G_j, ..., G_{N_G}\}$, and each IoRT device G_j has its own dataset D_{G_j}. The number of distinct class labels of all IoRT devices' dataset is N_C, and the class distribution vector of dataset D_{G_j} is $\boldsymbol{p}_{G_j} = (p_{G_j,1}, p_{G_j,2}, ..., p_{G_j,N_C})$, where $\sum_{n=1}^{N_C} p_{G_j,n} = 1$. Since air nodes aggregate the models of IoRT devices with the corresponding dataset size as their weights, the class distribution vector of the training data of air node A_i's model can be denoted as

$$\boldsymbol{p}_{A_i} = \frac{\sum_{j=1}^{N_G} |D_{G_j}| \cdot \boldsymbol{p}_{G_j}}{\sum_{j=1}^{N_G} |D_{G_j}|}, \tag{1}$$

where $|D_{G_j}|$ denotes the number of samples in dataset D_{G_j}. This can be used to decide which air nodes send their models to the same satellite.

IV. Problem Formulation

Based on the class distribution vectors of all the air nodes, we can decide which satellite each air node sends the model to. Our primary goal in this study is to minimize the time cost of training a model to achieve a preset accuracy. We will also consider the energy overhead optimization in a future work. The time cost can be divided into two parts, namely communication time cost and computation time cost. The communication time cost can be expressed as

$$T_{comm} = N_{AG} \cdot T_{AG} + N_{SA} \cdot T_{SA} + N_{SS} \cdot T_{SS}, \tag{2}$$

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on February 15,2023 at 13:23:41 UTC from IEEE Xplore. Restrictions apply.

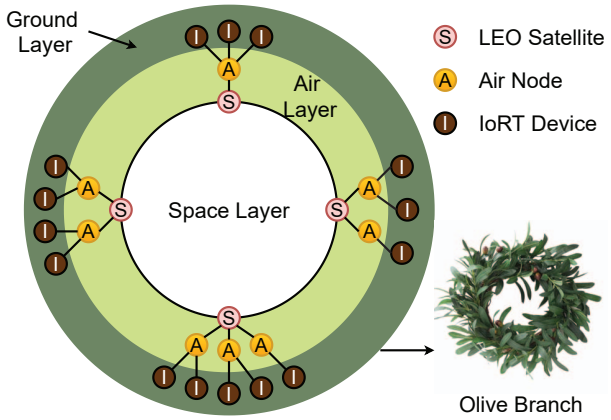


Figure 2. Structure overview of Olive Branch Learning.

where N_{AG}, N_{SA} and N_{SS} denote the number of communications between air nodes and IoRT devices, between satellites and air nodes, and between satellites, respectively. T_{AG}, T_{SA} and T_{SS} denote the end-to-end delay of model transmission between air nodes and IoRT devices, between satellites and air nodes, and between satellites, respectively. The end-to-end delay of model transmission contains transmission delay and propagation delay. The transmission delay can be obtained by calculating the quotient of model parameters size $|M|$ and the link bandwidth B, which is represented as

$$T_{XY} = T_{XY}^{trans} + T_{XY}^{prop} = \frac{|M|}{B_{XY}} + T_{XY}^{prop}. \tag{3}$$

The computation time cost mainly includes the model training time cost and the model aggregation time cost. The model training time cost depends on the floating point operations (FLOPs) of model, the number of data samples used in one local training epoch $|D_{ep}|$, the number of local training epoch N_{ep}, and the floating point operations per second (FLOPS) of the IoRT device $FLOPS_{train}$. The model aggregation time cost is related to the size of model parameters $|M|$, the number of models N_M, and the FLOPS of the node responsible for aggregation $FLOPS_{agg}$. The formula of computation time cost T_{comp} is defined by

$$T_{comp} = N_{train} \cdot T_{train} + N_{agg} \cdot T_{agg}, \tag{4}$$

$$T_{train} = \frac{FLOPs \cdot |D_{ep}| \cdot N_{ep}}{FLOPS_{train}}, \tag{5}$$

$$T_{agg} = \frac{|M| \cdot N_M}{FLOPS_{agg}}, \tag{6}$$

where N_{train} and N_{agg} denote the number of model training time and model aggregation time, respectively. T_{train} and T_{agg} denote the model training time cost and the model aggregation time cost, respectively.

As mentioned before, our goal is to minimize the time cost of training a model to achieve the preset accuracy. Since the class distribution vector of each air node is different due to the non-i.i.d nature, we can partition the air nodes into N_S clusters based on the class distribution vectors, and assign a satellite to each cluster, then each air node sends its model to the satellite its cluster belongs to. However, if we only consider the class distribution vectors, we may group the air nodes with very different class distribution vectors into one cluster, which can facilitate the convergence of the global model, but the geographic distance between two air nodes in the same cluster may become so far that it takes a large amount of time to transmit their models to the satellite their cluster belongs to. A lot of relay satellites are needed to help forward the models, and the number of communications between satellites N_{SS} will become huge, thereby increasing the time cost. Therefore, our proposed clustering algorithm in Section V will take both the class distribution and the geographic distance between air nodes into account.

After dividing the air nodes into N_S clusters, we need to assign a satellite to each cluster. We denote the clusters as $C = \{C_1, C_2, ..., C_{N_S}\}$, the satellites as $S = S_1, S_2, ..., S_{N_S}$, and the bijective function from clusters to satellites as $f : C \to S$. The objective function is given as

$$\min_{C,f} T_{comm} + T_{comp} \quad \text{s.t.} \quad acc_M \geq \alpha, \tag{7}$$

where acc_M denotes the test set accuracy of global model, and α denotes the preset test set accuracy the global model should achieve.

V. Algorithm Design

A. The Proposed Clustering and Mapping Algorithm

To solve the problem stated above, we propose the clustering and mapping algorithm for OBL, which heuristically solves the optimization problem in (7). For the convenience of explanation, we suppose that the number of air nodes is evenly divided by the number of satellites. The proposed algorithm is expounded as follows.

1) *Step 1: Geographical distance based division.* Satellites are divided into several parts equally and without overlapping, and each part includes N_{geo} satellites, which are continuous in orbit. Since every air node is covered by satellites, the number of air nodes covered by each satellite is $2N_A/N_S$, and the number of overlapping air nodes covered by every two adjacent satellites is N_A/N_S. To simplify the description of geographical distance based division, we can only consider the first half of air nodes covered by each satellite, so the number of air nodes corresponding to each satellite is N_A/N_S. Therefore, the air nodes are also divided into several parts $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_{N_S/N_{geo}}\}$ equally and without overlapping, and each part contains $N_{geo} \cdot N_A/N_S$ air nodes.
2) *Step 2: Clustering based on class distribution.* After dividing the air nodes into N_{geo} parts, the geographic

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on February 15,2023 at 13:23:41 UTC from IEEE Xplore. Restrictions apply.

distance between air nodes in each part is limited by the boundary of the part, so we only consider the class distribution of air nodes when clustering. The similarity of class distribution of air nodes can be estimated through the Euclidean distance between class distribution vectors of air nodes. We first use the k-means clustering algorithm to divide the air nodes into N_A/N_S groups according to the similarity of air nodes' class distribution. Then, we initialize N_S empty clusters. For each cluster, we sample one air node from each group without replacement and add it to the cluster. If a group becomes empty, we sample air node from other groups. Eventually we obtain N_S clusters, each cluster has N_A/N_S air nodes, and the air nodes within one cluster are with large differences in class distribution.

3) *Step 3: Cluster-to-satellite mapping.* Since we divide air nodes into several parts in Step 1, each cluster can only select the satellites of the part it belongs to. Therefore, the number of options for cluster-to-satellite mappings in each part is $N_{geo}!$, which is relatively small. To minimize the forward times required to send models from air nodes to satellites, we traverse all the options for cluster-to-satellite mappings and take the mapping with the least forward times as the final result.

The pseudo-code of the clustering and mapping algorithm is presented in Algorithm 1. The time complexity of the algorithm is $O(N_A^2/N_S + N_S \cdot N_{geo}!)$. In practical application, N_{geo} is an adjustable parameter to balance a trade-off between the geographical distance and the class distribution. In reality, the class distribution of devices that are geographically close may be similar, so set N_{geo} to a small value may not be conductive to the diversity of data. On the contrary, if we assign a large value to N_{geo}, e.g., N_S, the data diversity can be satisfactory, but the geographic distance between two air nodes in the same cluster may become very far. Therefore, it is essential to decide the value of N_{geo} carefully.

Algorithm 1 Clustering and Mapping Algorithm

Input: the number of satellites N_S, the number of air nodes N_A, the number of satellites in each geographical distance based division N_{geo}

Output: the set of clusters $C = \{C_1, C_2, ..., C_{N_S}\}$, the bijective function from clusters to satellites $f : C \to S$

Initialize an empty set of clusters $C = \{\}$
Divide the air nodes into N_S/N_{geo} parts equally and without overlapping $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_{N_S/N_{geo}}\}$ based on geographical distance, where the air nodes in each part have access to the same satellite
for each part $\mathcal{P}_i$ **do**
    Use k-means clustering algorithm to divide the air nodes into N_A/N_S groups $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_{N_A/N_S}\}$ according to the similarity of air nodes' class distribution
    Create an empty cluster C_{new}
    for each group $\mathcal{G}_i$ **do**
        if $\mathcal{G}_i$ is not empty **then**
            Sample one air node from $\mathcal{G}_i$ without replacement, and add it to C_{new}
        else
            Sample one air node from another non-empty group without replacement, and add it to C_{new}
        end if
    end for
    Add C_{new} to C
end for
Traverse all the options for cluster-to-satellite mappings, and find the mapping f with the least forward times

VI. Performance Evaluation

A. Settings

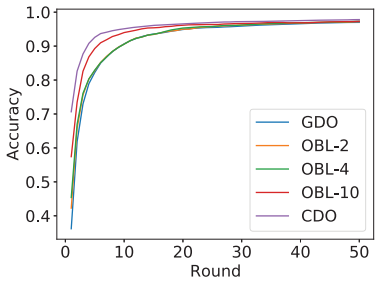
We evaluate our proposed clustering and mapping algorithm in this section. We present controlled experiments where two other methods are used for comparison with our algorithm. The specific settings are listed as follows:

▷ *Network settings.* We consider a space-air-ground integrated network with 20 low earth orbit (LEO) satellites, 100 air nodes, and 200 IoRT devices. The LEO satellites are evenly distributed in an equatorial orbital plane at an altitude about 330 km, and the air nodes are evenly distributed at a height of 100 m above the equator. The connectivity data between satellites and air nodes is obtained from Satellite Tool Kit (STK) simulation software. The propagation delay between an air node and its access satellite is 5 ms, between an IoRT device and its access air node is 5 ms, and between two neighboring satellites is 14 ms. The bandwidth of link between an air node and its access satellite is 600 Mbps, between an IoRT device and its access air node is 1000 Mbps, and between two neighboring satellites is 300 Mbps. We set the propagation delay and the bandwidth of link referring to [14], [15]. The computing capabilities of the IoRT devices, the air nodes, and the satellites are the same as that of a Jetson TX2 module, i.e., 1.33 TFLOPS.

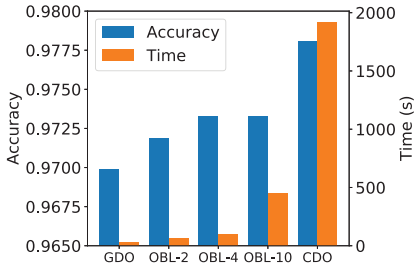
▷ *Training settings.* For the federated learning task, we focus on the image classification task and use the popular model training datasets MNIST, Fashion-MNIST, and CIFAR-10 to evaluate the performance of our proposed algorithm. The specific explanation for each dataset are as follows:

1) **MNIST** is a classic hand-written digit classification dataset with 10 classes. The learning model used for MNIST is convolution neural network (CNN) with 2 convolutional layers and 2 fully connected layer, whose number of trainable parameters is 21840.

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on February 15,2023 at 13:23:41 UTC from IEEE Xplore. Restrictions apply.

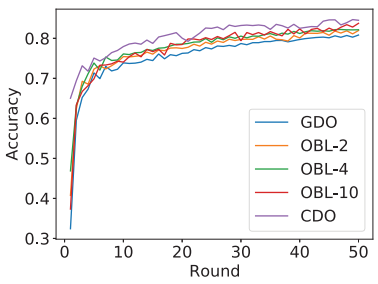


(a) Test accuracy varies with global round.

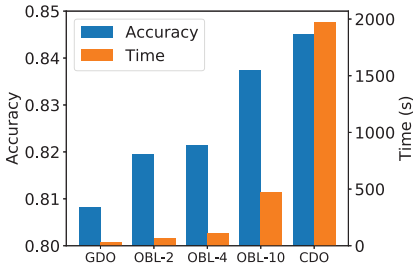


(b) Test accuracy and time cost after 50 global rounds of training.

Figure 3. Results on MNIST.

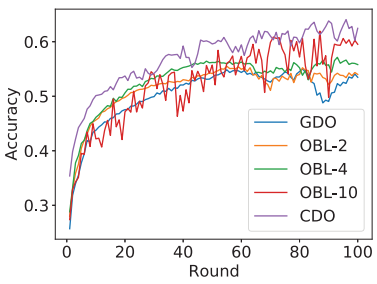


(a) Test accuracy varies with global round.

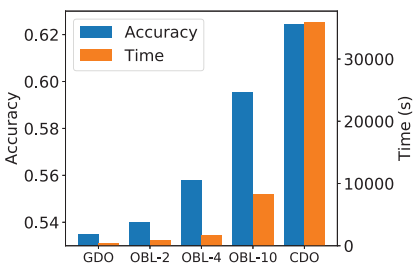


(b) Test accuracy and time cost after 50 global rounds of training.

Figure 4. Results on Fashion-MNIST.



(a) Test accuracy varies with global round.



(b) Test accuracy and time cost after 100 global rounds of training.

Figure 5. Results on CIFAR-10.

2) **Fashion-MNIST** is a dataset of article images with 10 classes. The model used for Fashion-MNIST is convolution neural network (CNN) with 2 convolutional layers and 2 fully connected layer, whose number of trainable parameters is 421642.
3) **CIFAR-10** is an object recognition dataset with 10 classes. The model used for CIFAR-10 is convolution neural network (CNN) with 9 convolutional layers, with one Group Normalization layer between every two convolutional layers. The number of trainable parameters is 1369738.

Each IoRT device possesses its own dataset with the same size but different class distribution. To simulate the non-i.i.d. scenario, we assign each IoRT device samples of 2 classes, and the class distribution of devices that are geographically close is similar.

▷ *Comparison settings.* We set different N_{geo} values to explore its influence on the experimental results. To verify the effectiveness of the proposed algorithm, we compare our algorithm and another two methods in terms of the time cost for the model to reach the preset accuracy, i.e., 97% for the MNIST dataset, 82% for the Fashion-MNIST dataset, and 55% for the CIFAR-10 dataset. Applying other algorithms to SAGIN is left for future work. The comparison methods are listed as follows:

1) Geographic Distance Only (GDO): The air nodes only send the models to their access satellites, which does not rely on relay satellites to help forward the models.
2) Class Distribution Only (CDO): The air nodes with large differences in class distribution vectors send their models to the same satellite, which may require relay satellites to help forward the models.

B. Experiment Results

We implement our framework and algorithm in Python using PyTorch. Fig. 3, Fig. 4, and Fig. 5 show the results of our experiments, where OBL-2, OBL-4, and OBL-10 denote OBL with $N_{geo} = 2$, $N_{geo} = 4$, and $N_{geo} = 10$, respectively. As illustrated in Fig. 3a, Fig. 4a, and Fig. 5a, the test accuracy of models increases rapidly in the early stages of learning, and stabilizes in the later stages of learning.

Fig. 3b, Fig. 4b, and Fig. 5b show the test accuracy and time cost of our proposed algorithm and two comparison methods at the end of training. For each dataset, as the value of N_{geo} in OBL increases, the test set accuracy of the model increases, while the time cost also increases. This is because the larger the value of N_{geo}, the larger the data diversity in each cluster, which is more conducive to improve the accuracy of the model. However, a larger value of N_{geo} means a larger number of model forwarding time, thus increasing the time cost. Besides, it can be seen that the more complex the dataset is used, the lower the accuracy of the model and the higher the time cost.

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on February 15,2023 at 13:23:41 UTC from IEEE Xplore. Restrictions apply.

We notice that the test accuracy of OBL-10 and CDO methods fluctuates with the number of rounds, which may because the data diversity introduces instability to the training process. Even so, the final models of OBL-10 and CDO methods are still superior to other methods in terms of test accuracy.

It is worth noting that the test accuracy of the final model of the GDO method trained on the MNIST dataset, the Fashion-MNIST dataset and the CIFAR-10 dataset is 96.98%, 80.82%, and 53.48%, respectively, which does not exceed our preset accuracy. In addition, the time cost of the CDO method to train a model for 50 rounds on the MNIST dataset, the Fashion-MNIST dataset and the CIFAR-10 dataset is 59, 59, and 111 times that of the GDO method, respectively, which is unacceptable.

Overall, the models of our proposed algorithm with $N_{geo} = 4$ can achieve acceptable accuracy, i.e., 97.33% for the MNIST dataset, 82.14% for the Fashion-MNIST dataset, and 55.81% for the CIFAR-10 dataset, and the time cost of 50 rounds of training is relatively small, i.e., 2.96, 3.27, and 5.22 times that of the GDO method for the MNIST dataset, the Fashion-MNIST dataset, and the CIFAR-10 dataset, respectively, which verifies the effectiveness of our algorithm.

VII. Conclusion

In this paper, we propose a federated learning framework for SAGIN, called Olive Branch Learning (OBL). In OBL framework, the IoRT devices that are far away from each other can collaborate to train powerful models utilizing their own local data with their privacy protected. The nodes in space layer and air layer serve as aggregation node, which can help the IoRT devices share their models. In order to weaken the impact of non-i.i.d. data on IoRT devices and reduce the time cost of training a global model, we introduce a clustering and mapping algorithm for OBL, which takes both the geographic distance and data distribution into account. Experiment results demonstrate that models trained using our proposed framework and algorithm can achieve satisfactory accuracy while reducing the time cost.

References

[1] Z. Jia, M. Sheng, J. Li, D. Niyato, and Z. Han, "Leo-satellite-assisted uav: Joint trajectory and data collection for internet of remote things in 6g aerial access networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9814–9826, June 2021.

[2] J. Wang, Y. Zhong, Z. Zheng, A. Ma, and L. Zhang, "Rsnet: The search for remote sensing deep neural networks in recognition tasks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 2520–2534, March 2021.

[3] S. H. Khan, X. He, F. Porikli, and M. Bennamoun, "Forest change detection in incomplete satellite images with deep neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5407–5423, Sep. 2017.

[4] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2714–2741, Fourthquarter 2018.

[5] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: Improving qos of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, January 2019.

[6] Y. Seyedi and S. M. Safavi, "On the analysis of random coverage time in mobile leo satellite communications," *IEEE Communications Letters*, vol. 16, no. 5, pp. 612–615, May 2012.

[7] L. Yang, Y. Lu, J. Cao, J. Huang, and M. Zhang, "E-tree learning: A novel decentralized model learning framework for edge ai," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 290–11 304, July 2021.

[8] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6g communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, Sep. 2020.

[9] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated learning in the sky: Aerial-ground air quality sensing framework with uav swarms," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9827–9837, June 2021.

[10] H. Zhang and L. Hanzo, "Federated learning assisted multi-uav networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 14 104–14 109, Nov 2020.

[11] N. Kato, Z. M. Fadlullah, F. Tang, B. Mao, S. Tani, A. Okamura, and J. Liu, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 140–147, 2019.

[12] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2442–2473, Fourthquarter 2016.

[13] A. Gibiansky. (2021) Bringing hpc techniques to deep learning. [Online]. Available: https://andrew.gibiansky.com/blog/machine-learning/baidu-allreduce/

[14] C. Li, Y. Zhang, R. Xie, X. Hao, and T. Huang, "Integrating edge computing into low earth orbit satellite networks: Architecture and prototype," *IEEE Access*, vol. 9, pp. 39 126–39 137, 2021.

[15] J. Wei, J. Han, and S. Cao, "Satellite iot edge intelligent computing: A research on architecture," *Electronics*, vol. 8, no. 11, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/11/1247

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on February 15,2023 at 13:23:41 UTC from IEEE Xplore. Restrictions apply.