



Article

Space–Air–Ground–Sea Integrated Network with Federated Learning

Hao Zhao ^{1,2} , Fei Ji ^{2,3,*}, Yan Wang ³, Kexing Yao ³ and Fangjiong Chen ³

¹ Department of Information and Communication Engineering, Guangzhou Maritime University, Guangzhou 510725, China; zhaohao@gzmtu.edu.cn

² The Key Laboratory of Marine Environmental Survey Technology and Application, Ministry of Natural Resources, Guangzhou 510300, China

³ School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China; eeyanw@mail.scut.edu.cn (Y.W.); eekexingyao@mail.scut.edu.cn (K.Y.); eefjchen@scut.edu.cn (F.C.)

* Correspondence: feiji@scut.edu.cn

Abstract: A space–air–ground–sea integrated network (SAGSIN) is a promising heterogeneous network framework for the next generation mobile communications. Moreover, federated learning (FL), as a widely used distributed intelligence approach, can improve advanced network performance. In view of the combination and cooperation of SAGSINs and FL, an FL-based SAGSIN framework faces a number of unprecedented challenges, not only from the communication aspect but also on the security and privacy side. Motivated by these observations, in this article, we first give a detailed state-of-the-art review of recent progress and ongoing research works on FL-based SAGSINs. Then, the challenges of FL-based SAGSINs are discussed. After that, for different service demands, basic applications are introduced with their benefits and functions. In addition, two case studies are proposed, in order to improve SAGSINs' communication efficiency under a significant communication latency difference and to protect user-level privacy for SAGSIN participants, respectively. Simulation results show the effectiveness of the proposed algorithms. Moreover, future trends of FL-based SAGSINs are discussed.

Keywords: federated learning; SAGSIN; heterogeneous; wireless communications; privacy



Citation: Zhao, H.; Ji, F.; Wang, Y.; Yao, K.; Chen, F. Space–Air–Ground–Sea Integrated Network with Federated Learning. *Remote Sens.* **2024**, *16*, 1640. <https://doi.org/10.3390/rs16091640>

Academic Editors: Lu Ma, Yiping Duan, Qin Lu, Gang Qiao, Jianmin Yang and Andrea Nascetti

Received: 29 March 2024

Revised: 27 April 2024

Accepted: 30 April 2024

Published: 4 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, artificial intelligence (AI), e.g., deep learning (DL), has enhanced applications by playing an excellent role in different industries, such as natural language processing, image classification, and even traditional communication systems [1]. Notably, DL, as an effective data-driven way, can achieve better performance than classical model-driven methods in some conditions. Meanwhile, sixth-generation (6G) wireless systems have been widely considered, such as the space–air–ground–sea integrated network (SAGSIN) architecture [2], which consists of AI-based, decentralized, and heterogeneous Internet of Things (IoT) systems [3,4]. Meanwhile, AI-enabled distributed approaches have played important roles in recent wireless networks. Therefore, the distributed learning framework has great potential in SAGSINs. However, due to limited communication resources and the privacy constraints around the data transmission from all wireless devices, it is impractical for all wireless devices to transmit all of their collected data to a data center for data analysis and inference. Fortunately, federated learning (FL), as a decentralized learning framework, enables multiple users to learn a shared model collaboratively without exchanging local training data [5], which not only reduces the consumption of communication resources but also prevents privacy information leakage. FL combined with the SAGSIN framework, including terrestrial networks, aerial networks, satellite communications, even marine communication networks, is full of opportunities.

1.1. SAGSIN

In the literature, several tutorials about multi-scenario integrated network have been delivered. Initially, the demand of a high-quality and reliable access at any time and anywhere with anyone was considered, and the proposed network was a space–air–ground–integrated network (SAGIN) [6]. Then, with increasing requirements for communication capabilities, many marine devices, both over the sea surface and under the sea surface, needed to be taken into account. SAGINs were further extended, and sea networks were added. Therefore, the SAGSIN was presented as a promising network architecture for 6G, with multi-layer networks, an open communication environment, and time-varying topological characteristics [2]. For example, in [7], driven by the 6G demand that will integrate both physical and digital worlds, an SAGSIN and related technologies are introduced, which pay more attention to the shape-adaptive antenna and radar-communication integration. In [8], a type of resource-friendly authentication method is proposed for a maritime communication network of SAGSINs based on the elliptic curve cryptography approach. In [9], an SAGSIN framework with edge and cloud computing is introduced, in order to provide a flexible hybrid computing service for maritime service. In addition to algorithms, platforms for simulation and experiments have also been studied. In [10], an SAGIN simulation platform was developed, which adopted centralized and decentralized controllers to optimize network functions, such as access control and resource scheduling. Moreover, in [11], a multi-tier communication network field experiment was carried out by Pengcheng Laboratory, which used a 5G network technology based on floating platforms in Jingmen, China for ocean activities. With time elapsing, research on artificial intelligence (AI)-enabled SAGSIN systems has also been triggered, especially distributed algorithms. In [12], an AI-based optimization for SAGINs is proposed in order to improve the network performance, and its evident advantages are illustrated. In conclusion, the network framework has been extended from SAGIN to SAGSIN, and the research has developed from a basic algorithm simulation to field experiments. More importantly, AI has gradually been embedded in the SAGSIN system and makes huge differences.

1.2. FL

Recently, some surveys have emerged which illustrate FL from different perspectives, including challenges, possible solutions, and future research. In [5], a broad overview of current approaches of FL is presented and several directions of future research are outlined. More specifically, the general idea of FL is introduced, and possible applications of 5G networks are discussed. Moreover, the challenges and open issues in FL under a wireless scenario are described in [13]. Moreover, to implement FL for edge devices using less reliance, a novel collaborative FL framework is presented in [14], and four collaborative FL performance metrics with wireless techniques, e.g., network formation and scheduling policy, are also introduced. In [15], the relationship of a global server and participants is modeled via a Stackelberg game to stimulate device participation in the FL procedure. To explore emerging opportunities of FL for the industrial network, open issues have been discussed, focusing on automated vehicles and collaborative robotics [16]. Furthermore, the preservation of data privacy via federated learning is mainly focused. The mechanisms of attack are introduced, and the corresponding solutions are discussed [17]. It is worth noting that recent developments in FL-aided training for physical-layer problems have been presented. Meanwhile, the challenges of model, data, and hardware complexity, have been emphasized with some potential solutions [18].

In general, FL and wireless networks are thought to have a bidirectional relationship [19]. On the one hand, FL can be used as an enhanced solution for communication applications. Taking physical-layer applications as an example, an FL-based channel estimation scheme was developed for both conventional and intelligent reflecting surface (RIS)-assisted massive multiple-input multiple-output (MIMO) systems [20], which did not require the overhead of data transmission. For symbol detection, a downlink fading symbol detector based on decentralized data was proposed [21]. Moreover, some distributed

optimization problem of communication resources, e.g., computation offloading [22], and network slicing [23] can also be solved by using federated learning. On the other hand, the characteristics and strategies of wireless communication are important factors which affect the performance of FL, such as power allocation, scheduling policies, and channel fading. Howard et al. paid attention to scheduling policies for FL in wireless networks. The effectiveness of three different scheduling policies, i.e., random scheduling, round robin, and proportional fair, were compared in terms of FL convergence rate and tractable expressions were derived for the convergence rate of FL in a wireless setting [24]. Moreover, considering the wireless resource allocation, a method to dynamically adjust and optimize was proposed to balance the tradeoff between maximizing the number of selected clients and minimizing the total energy consumption, by selecting suitable clients and allocating appropriate resources in terms of CPU frequency and transmission power [25]. Furthermore, the problem of joint power and resource allocation for ultra-reliable low-latency communication (URLLC) in vehicular networks was studied [26]. Furthermore, to minimize the FL convergence time and the FL training loss, a probabilistic user-selection scheme was proposed such that the BS was connected to the users whose local FL models had significant effects on the global FL model with high probabilities [27–29]. Moreover, FL has been tightly combined with advanced distributed technology, e.g., fog communications [30] and blockchains [31]. It is worth mentioning that the privacy and security of individual users still risk being threatened by attackers, even if raw data are not needed in the FL architecture. In [32], a dual differentially private federated learning (2DP-FL) algorithm was proposed, which was suitable for nonindependent identically distributed (non-IID) datasets and had flexible noise addition scheme to meet various needs. Recently, methods for preserving privacy and security in FL have also been widely studied [33].

1.3. Contributions

Motivated by the above-mentioned issues, we want to fill the gap in surveys of SAGSINs with FL. We hope that this paper will receive the attention of researchers in both SAGSIN and FL communities to design advanced FL approaches to address the open issue caused by SAGSINs. Our contributions are as follows.

- We summarize the SAGSIN and FL state of the art, respectively. Then, we also review FL-based SAGSIN applications. After that, we provide a detailed overview of FL-based SAGSINs.
- We detail the challenges and practical problems in FL-based SAGSINs, where heterogeneous networks, multiple transmission media, and heterogeneous data impact performance. We summarize its benefits, disadvantages, and future directions.
- To better understand the problems in SAGSINs, we present two typical cases. Aiming at a multi-scale delay problem of SAGSINs, we propose a delay-aware FL to minimize the time consumption of FL aggregation. On the other hand, for user-level privacy protection and transfer learning, we propose a noise-based FML (NbFML) based on a differential-privacy (DP) algorithm and an expression for the sensitivity of the FML aggregation operation is derived. Experiment results show that these algorithms are feasible.

In the following, we give a brief overview of FL and SAGSINs in Section 2. Section 3 introduces FL-enabled SAGSIN applications, including FL-based satellite networks, FL-based aerial networks, FL-based terrestrial networks, and FL-based marine networks, respectively. Next, we indicate prominent challenges in FL-enabled SAGSINs in Section 4, such as heterogeneous networks, wireless communication challenges, and privacy protections. Then, we give two case studies, aiming at solving the multi-scale delay aggregation of SAGSINs and protecting user-level privacy for SAGSINs in Section 5. After that, we point out some future directions of FL-based SAGSINs in Section 6. Finally, Section 7 summarizes this paper and give future directions.

2. An Overview

In this section, we first introduce the basic concept of FL. Then, the communication latency of SAGSINs is illustrated. After that, we present the composition of an SAGSIN.

2.1. FL Algorithm

Normally, an FL framework consists of one FL computing center and a community of N participants. We assume that each participant is honest and use their own data to train local models, and they upload their local model with its parameters to the computing center. We denote by θ_i the local parameters of the i th participant. We define the local dataset as D_i on the i th participant with $|D_i|$ samples. FL, as an effective way to achieve distributed learning, can be performed in three steps, including acquiring global parameters, local updating, and global aggregation. The detailed procedure is shown in Algorithm 1. First, in each learning round all participants are activated to download a global model from the computing center. Second, participants learn parameters using their own data by using the stochastic gradient descent (SGD) algorithm locally [34]. Third, the computing center collects participants' parameters to do aggregation. More specifically, the user can calculate gradients using dataset $D_{i,m}$ according to

$$g_{i,m}^t(D_{i,m}) = \nabla_{\theta^t} F_i(D_{i,m}, \theta^t), \quad (1)$$

where $\nabla_{\theta} F_i(D_{i,m}, \theta)$ is the gradient function at the i th node calculated under dataset $D_{i,m}$, and θ is the corresponding parameter. The procedure for the local update of the i th user can be defined as

$$\theta_i^{t+1} = \theta^t - \frac{1}{|D_i|} \sum_{m=1}^{|D_i|} \alpha g_{i,m}^t(D_{i,m}), \quad (2)$$

where $m \in 1, \dots, |D_i|$, and α is the learning rate. Mathematically, the aggregations of K users can be expressed as

$$\theta^{t+1} = \frac{1}{n} \sum_{i=1}^K w_i \hat{\theta}_i^{t+1}, \quad (3)$$

where w_i is the weighting coefficient of the i th participant. After the aggregation, the global parameters are updated centrally and are used in the next communication round.

Algorithm 1 FL

Require: Dataset $\{D_i\}_{i=1}^K$ of each participant and the communication rounds T for all nodes.

- 1: Initialize: the initial global parameter $\theta_i^t = \theta^t, t = 0$
 - 2: **for** $t = 1 : T$ **do**
 - 3: **for** each participant $i \in \{1, 2, \dots, K\}$ in parallel **do**
 - 4: Calculate the local gradients:
 - 5: $g_{i,m}^t(D_{i,m}) = \nabla_{\theta^t} F_i(D_{i,m}, \theta^t), m \in 1, 2, \dots, |D_i|;$
 - 6: Update the local parameters:
 - 7: $\theta_i^{t+1} = \theta^t - \frac{1}{|D_i|} \sum_{m=1}^{|D_i|} \alpha g_{i,m}^t(D_{i,m});$
 - 8: Send protected parameter $\hat{\theta}_i$ to the server;
 - 9: **end for**
 - 10: The server collects all the parameters $\{\hat{\theta}_i^{t+1}\}_{i=1}^K$, and updates $\theta^{t+1} = \frac{1}{n} \sum_{i=1}^K w_i \hat{\theta}_i^{t+1}$
 - 11: The server broadcasts the global parameters θ^{t+1}
 - 12: $t = t + 1$
 - 13: **end for**
- Ensure:** θ^T
-

2.2. Communication Latency

In the upload procedure, the time latency o_i of the i th participant transmitting its own parameters to the center can be expressed as

$$o_i = \frac{|\theta_i|}{r_i} + \tau_i, \quad (4)$$

where r_i denotes the communication rate of the i th user, τ_i represents the signal propagation delay, and $|\theta_i|$ means the total number of parameters, which are compressed and transmitted in bits to the center. Specifically, the propagation delay can be expressed as

$$\tau_i = \frac{\mathbb{L}_i}{v_i}, \quad (5)$$

where \mathbb{L}_i means the distance between the i th participant and the computing center, and v_i means the signal propagation velocity in corresponding media. In the same way, the download latency can also be calculated in the same way.

2.3. SAGSIN Scenario

In an SAGSIN, the participant can come from different networks. The framework of an SAGSIN is shown in Figure 1. From this figure, it is evident that the satellite network, aerial network, terrestrial network, and marine network have an inseparable relationship. In general, satellite networks contain a variety of satellites at different altitudes, e.g., geostationary orbit (GEO) satellites, medium-earth orbit (MEO) satellites, and low-earth orbit (LEO) satellites. Aerial networks usually contain aircraft, unmanned aerial vehicles (UAVs), airships, even balloons and they can provide flexible and fast communications for emergency conditions. Moreover, because of these features, aerial networks can serve as a bridge among satellite networks, terrestrial networks, and marine networks. As for terrestrial networks, they always contain cellular networks, vehicle networks, ad hoc networks, local area networks (LANs), and so on. Last but not least, sea networks are made up of marine networks and underwater networks. On the one hand, marine networks provide communication services for ships, buoys, and surface equipment, e.g., floating oil platforms and offshore wind power platforms. On the other hand, underwater networks employ acoustic, optical, and radio data to achieve underwater observations and information gathering. Furthermore, due to the harsh environment and expensive cost of wired communications, it is a very common phenomenon that underwater networks take advantage of autonomous underwater vehicles (AUV), remotely operated vehicles (ROV), and unmanned underwater vehicles (UUV) to form a self-organizing network.

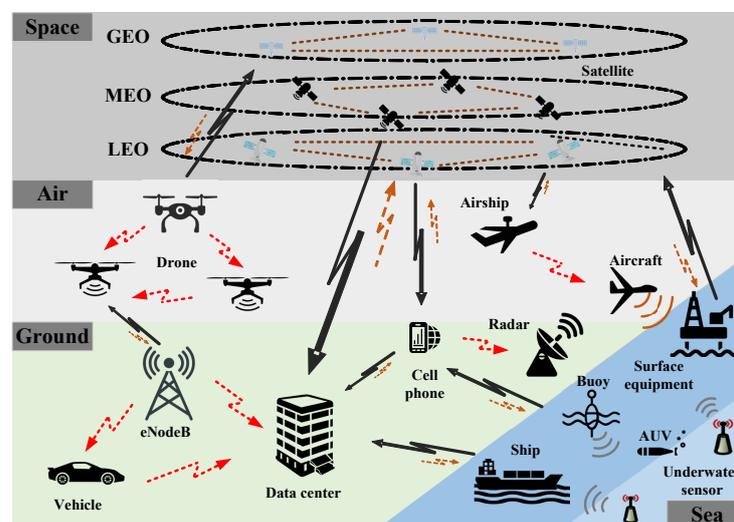


Figure 1. The framework of an SAGSIN.

3. FL-Enabled SAGSIN Applications

In this section, we present some applications of FL-based SAGSINs. According to different service demands, we divide recent FL-based SAGSINs into four aspects.

3.1. FL-Based Satellite Networks

A satellite network contains a collection of satellites according to the altitude, e.g., GEO satellites, MEO satellites, LEO satellites. Among these satellite applications, FL has been widely regarded as an effective solution for many problems, and many works have been presented. For example, considering the preciousness of resources and privacy of satellite networks, an algorithm for FL adaptation to satellite–terrestrial integrated networks was proposed [35]. In addition, aiming at solving inference based on satellite imaging, an asynchronous FL algorithm was proposed which obtained more robust performance against heterogeneous conditions than classical methods in LEO constellations [36]. Moreover, considering urban computing conditions with communication-heavy, high-frequency, and asynchronized characteristics, a hybrid FL framework named StarFL was proposed. To protect the security of users' data, the authors also gave a verification mechanism by utilizing Beidou satellites [37]. Furthermore, in view of the demand of massively interconnected systems with intelligent learning and reduced traffic in satellite communications, FL-based LEO satellite communication networks were proposed [38]. Meanwhile, the authors also reviewed LEO-based satellite communications and related learning techniques, and four possible ways of combining machine learning with satellite networks were discussed. Moreover, for natural resource management based on an integrated terrestrial–aerial–space network framework, an asynchronously FL method for forest fire detection was proposed without using explicit data exchange by using collected IoT data [39].

3.2. FL-Based Aerial Networks

Recently, plenty of aerial platforms, e.g., drones, aircraft, balloons, and airships, have been considered for future networks in order to provide better wireless communication services. The aerial station provides flexible communication services which can cover more area, including space–air communications, air–ground communications, and air–sea communications. In [40], image classification tasks were taken into account in UAV-aided exploration scenarios with a limited flight time and payload. A ground fusion center and multiple UAVs were employed, and FL was introduced to reduce the communication cost between the ground center and the UAVs. Moreover, considering how human health is impacted by air quality, an FL-based air quality forecasting and monitoring approach was proposed by taking advantage of an aerial–ground network. Specifically, the author also performed a numerical simulation on real-world data [41]. Moreover, unlike terrestrial networks, the maneuverability of UAV characteristics was considered and a jointly optimization for UAV location and resource allocation was proposed, which could minimize the energy consumption of terrestrial users [42]. Furthermore, to overcome the problem of battery limitation on the user's side, UAV-based wireless power transfer was proposed to enable sustainable FL-based wireless networks. In this regard, the UAV transmitted power efficiently via a joint optimization of propagation delay and bandwidth allocation, power control, and the UAV placement [43]. The authors developed an efficient algorithm for sustainable FL based on UAVs by leveraging the decomposition approach and a convex approximation.

3.3. FL-Based Terrestrial Networks

Terrestrial networks are one of the most important factors in modern life, including vehicular communications, edge computing, blockchain, and IoT services. Recently, some researchers employed FL in vehicular communications. For example, in view of data privacy in the training process of remote sensing data, an FL-based approach to identify vehicle targets in remote sensing images was proposed [44]. Moreover, considering the joint power and resource allocation (JPRA) problem in vehicular networks, a novel dis-

tributed approach based on FL was proposed to estimate the tail distribution of the queue lengths [26]. Specially, a Lyapunov optimization was proposed by considering the communication delays of FL, and the JPRA policies were derived by enabling URLLC for each vehicle. Moreover, for railway systems, considering classical methods lack effective trust and incentives, a blockchain-based FL framework was proposed to achieve asynchronous collaborative learning among distributed agents [45], where the management of the FL devices were realized by the blockchain technology. Furthermore, with the development of mobile communications, the techniques and theory of edge computing have grown fast. To optimize mobile edge computing, caching, and communication, deep reinforcement learning and FL were combined together and an “In-Edge AI” framework was proposed to intelligently utilize the collaboration between center and edge nodes, and a dynamic system-level optimization was presented to reduce the unnecessary communication consumption in [46]. In addition, FL has also been applied in the IoT. In [47], based on FL and 5G with decentralized computing and connectivity characteristics, by leveraging the cooperation of distributed devices, an FL methodology was proposed and verified using datasets generated from an Industrial IoT (IIoT) environment. In addition, a blockchain-enhanced secure data-sharing framework for distributed devices and privacy-preserved FL were proposed in the consensus procedure of a permissioned blockchain. The proposed algorithm was also tested on real-world IIoT datasets [48].

3.4. FL-Based Sea Networks

Seventy percent of the earth is covered by oceans. Nowadays, with the increasing human activity in the maritime area, it becomes more and more important to acquire oceanic information. Recently, FL, as an effective distributed way, has been paid more attention. Given the marine IoT is a new industry with edge computing and cloud computing characteristics, a secure sharing method of marine IoT data under an edge computing framework by using FL and blockchain was proposed, which guaranteed the privacy of local users [49]. Moreover, the marine IoT field is tied to the Internet of Ships (IoS) paradigm. In [50], to achieve fault diagnosis in the IoS, an adaptive privacy-preserving FL framework was presented, which used FL characteristics to organize various ships collaboratively. An adaptive aggregation method was designed to reduce computation and communication costs. The authors not only analyzed it theoretically but also presented some experiments, which proved the effectiveness of the FL-based IoS on a nonindependent and identically distributed (non-IID) dataset. Furthermore, with the progress of technology, underwater information is also receiving attention. Internet-of-Underwater-Things (IoUT) devices and FL technology have gradually been combined. Ref. [51] first gave a detailed overview of the IoUT, and AI technology was mentioned. Many scholars have applied distributed learning methods to oceanic things. For example, considering the unreliable links of underwater networks, it is difficult to gather all users’ data at a center and provide centralized training. Motivated by this, a multiagent deep reinforcement learning and FL-based framework was proposed and a multiagent deep deterministic policy gradient was used to solve distributed situations and unexpected time-varying problems [52]. Moreover, DL can be an effective way to deal with some mission-critical maritime tasks, such as ocean earthquake forecasting, underwater navigation, underwater communications. For instance, a federated meta-learning (FML) scheme was proposed to enhance underwater acoustic chirp communications leveraging an acoustic radio cooperative characteristic of a surface buoy. In addition, the FML convergence performance was analyzed [53]. Furthermore, a learning-aided maritime network and its key techniques including information sensing, transmission, and processing were presented [54]. Specifically, a research on reliable and low-latency underwater information transmission was introduced.

4. Challenges in FL-Enabled SAGSINs

SAGSINs have a lot of characteristics, e.g., heterogeneity, self-organization, and time-variability, which create difficulties for a federated training procedure. Following recently

developments, we detail the challenges of FL-based SAGSINs from four aspects, according to the network's hierarchical model: heterogeneous data challenges and wireless communication challenges at the physical layer, heterogeneous challenges at the network layer, and privacy challenges at the application layer.

4.1. Heterogeneous Data Challenges

Due to the large dynamics of SAGSINs and participants' behaviors, the collected training data on local devices become heterogeneous, not only on different devices, but also on the same device at different times. Among different devices, the data may be nonindependent and identically distributed (non-IID), which causes FL to experience severe performance degradation. Unlike IID data, well-trained parameters on a local dataset are an unreliable approximation for the global model. Therefore, an FL-based SAGSIN needs a novel algorithm to deal with non-IID problems, including a convergence analysis and a balance of local computations and global communications. On the other hand, because users are active, the data generated on local devices change, and it is difficult to guarantee that the distribution of new data will not change, especially for real-time tasks. In order for a trained model to perform new tasks quickly, meta learning-based transfer learning is a good choice. Therefore, researchers have paid more attention to FML frameworks in recent years, as a solution to real-time tasks [55]. To overcome the challenges raised by constrained resources and a limited quantity of local data, a platform-aided FML architecture was studied in [56] by using the knowledge transfer from prior tasks, achieving fast and continual edge learning. Moreover, in order to adapt to different tasks quickly, an FML approach to jointly learn good meta-initialization for both backbone networks and gating modules was proposed in [57], by using the model similarity across learning tasks at different nodes. Furthermore, considering the scarce radio spectrum and the limited battery capacity of IoT devices, an FML algorithm with a non-uniform device selection scheme was developed in [58] to accelerate the convergence, which can control the resource allocation and energy consumption when deploying FML in practical wireless networks.

4.2. Wireless Communications Challenges at the Physical Layer

Communication is the key procedure for FL aggregation. The different communication devices, methods, environments bring a huge challenge to FL-based SAGSINs. How to balance the limited communication resources for potentially millions of devices is a worthy problem. For example, the scheduling policy and power control mechanism of SAGSIN wireless communications need to be more elaborate. Meanwhile, the control of the wireless communication bit error rate (BER) should be reconsidered because the faults caused by bit errors may make DL or FL devices have better accuracy performance. In other words, FL-based SAGSINs have BER tolerance characteristics. From this point of view, the power, modulation, and coding of the transmitter can be adjusted to a relatively suitable solution in order to improve the whole system performance.

4.3. Heterogeneous Challenges at the Network Layer

Obviously, the terrestrial network is the core of SAGSINs, which is extended to space, air, and sea areas step by step. An SAGSIN provides multiple network services which guarantee normal activities for all participants. Each individual network of an SAGSIN has its merits and demerits from different aspects, e.g., bandwidth, coverage, and communication quality; therefore, joint information transmission networks are urgently needed. Meanwhile, it makes an SAGSIN become a heterogeneity network with large-scale, three-dimensional, and multi-layer characteristics. The heterogeneous nature creates a lot of difficulties for FL communication procedures, such as spectrum management, network control, power management, and routing management. In particular, in an FL-based SAGSIN scenario, there are multiple propagation conditions in its wireless transmission process, such as air-to-ground, ground-to-air, and sea-to-ground conditions. Since participants come from different networks with various communication distances, the communication

latency of each participant is different. For example, when devices in different networks are assumed to upload their parameters for data fusion to a computational center, the latency is not only affected by computation time but is also influenced by communication time. Specifically, for an underwater device and satellite facility, the large latency is caused by the long propagation time or lower propagation speed in special areas. The long time delay of the communication round should not be neglected. Moreover, asynchronous FL is another high-efficiency way to cope with different communication latency values, which require an appropriate asynchronous scheduling policy.

4.4. Privacy Challenges at the Application Layer

FL requires participants to upload updated model parameters and share them with other participants after each communication round. Although distributed learning can protect the privacy of individual users, the user information still can be extracted by analyzing the uploaded parameters. This leaves FL with a serious privacy risk. An attacker can act as a participant in model training and carry out a refactoring attack, an inference attack, or a steal attack. Therefore, privacy-preserving-based distributed frameworks have been paid more attention in recent decades. Moreover, it is necessary to design algorithms which have low computational complexity, efficient communications, and robustness characteristics, and the accuracy of the FL system also needs to be guaranteed at the same time. Recently, differential privacy (DP) has been introduced into DL-based systems as an effective solution to prevent client information leakage by adding artificial noise [59]. For distributed systems, the user-level DP plays a very good role in privacy protection. The authors in [60] proposed a local DP (LDP)-compliant stochastic gradient descent (SGD) algorithm, which was suitable for a variety of machine learning scenarios. Moreover, reference [61] studied data aggregation for distribution estimation with the LDP mechanism, which protects the user's local privacy.

5. Case Study

In this section, we present two applications with numerical simulations to give more detailed illustrations of FL-based SAGSINs. First, for the multi-scale latency problem in SAGSINs, we propose a delay-aware FL framework in order to reduce the aggregation consumption. Second, considering the real-time judgment of FL and privacy preservation, we propose an NbFML algorithm to protect user-level privacy. In conventional FL, it is assumed that the propagation delay of each participant is basically the same. However, in SAGSIN scenarios, this assumption is not well justified since participants come from different communication networks with different communication distances and mediums among involved participants. It makes the SAGSIN have a multi-scale delay characteristic. Therefore, the exaggeration of participants is affected by individual time delays.

5.1. Delay-Aware FL

The multi-scale feature is described in Figure 2. For simplicity, we only took the uplink communication as an example and the downlink procedure can be calculated in the same way. Theoretically, the consumption time of each aggregation is decided by the maximum latency among all participants, which can be calculated by

$$\tau = \max_i \{\tau_1, \tau_2, \dots, \tau_K\}. \quad (6)$$

5.1.1. Delay-Aware FL

Motivated by this, we propose a delay-aware based aggregation method, in order to effectively reduce communication time under the same communication epoch. The delay-aware FL procedure is shown in Algorithm 2.

In general, the more machines participate in training, the better the system performs. Moreover, to speed up the system's convergence, all devices are usually employed for training at the beginning of the FL stage. Especially for scenarios where data collection

is difficult, i.e., underwater scenarios and space scenarios, where a limited number of AI-based devices can be deployed. In this case, it is excellent for all equipment involved in the training process. Moreover, given the limited bandwidth and energy, when the number of devices is too large, only some of the devices are used for training at each epoch. Due to the restricted communication resources, the center only picks K participants out of N uniformly in each communication round. The random scheduling ratio is $G = K/N$. Participants N are divided into B groups according to different latency levels. Then, in the communication round, each group performs the learning algorithm sequentially. After that, participants send well-trained local parameters to the center to perform aggregation. In this condition, the time consumption of delay-aware FL is only determined by the maximum latency in the corresponding group.

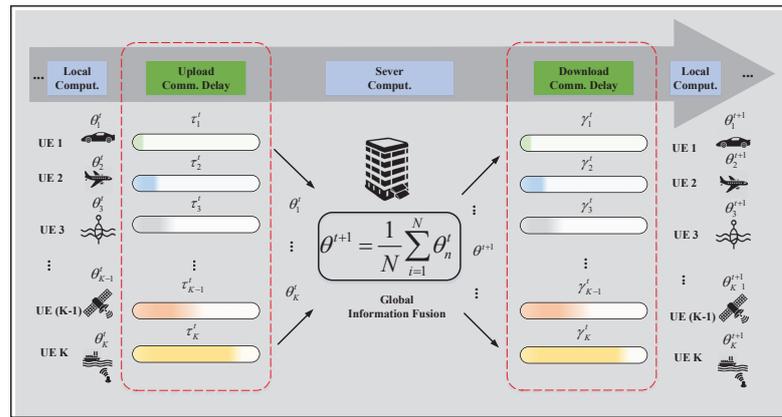


Figure 2. Multi-scale delay characteristic of FL-based SAGSIN.

Algorithm 2 Delay-aware FL

Require: Dataset $\{D_i\}_{i=1}^K$ at each UE, communication rounds T , scheduling ratio G , parameter w , threshold C , and DP parameters (ϵ, δ) for all nodes.

- 1: Initialize: $\theta_i^t = \theta^t$ and $t = 0$
- 2: **for** $t = 1 : T$ **do**
- 3: Clustering N users into B groups according to different scales of communication latency
- 4: **for** each group $P_b \in \{P_1, \dots, P_B\}$ **do**
- 5: **for** each participant $i \in P_b$ **do**
- 6: Calculate the local gradients:
- 7: $g_{i,m}^t(D_{i,m}) = \nabla_{\theta^t} F_i(D_{i,m}, \theta^t)$;
- 8: Update the local parameters:
- 9: $\theta_i^{t+1} = \theta^t - \frac{1}{|D_i|} \sum_{m=1}^{|D_i|} \alpha g_{i,m}^t(D_{i,m})$;
- 10: Send protected parameter $\hat{\theta}_i$ to the server;
- 11: **end for**
- 12: **end for**
- 13: The server collects all the parameters $\{\hat{\theta}_i^{t+1}\}_{i=1}^K$, and updates $\theta^{t+1} = \frac{1}{N} \sum_{i=1}^K w_i \hat{\theta}_i^{t+1}$
- 14: The server broadcasts the global parameters
- 15: $t = t + 1$
- 16: **end for**

Ensure: θ^T

5.1.2. Numerical Simulation

We assumed that the terminal device had high performance and the computation capacity of each participant was the same. Note that the parameter of the training model could be transmitted in one communication round because the parameter could be com-

pressed, and the communication bandwidth was sufficient. The communication parameters of the SAGSIN are shown in Table 1 and include the theoretical communication distance, propagation speed, and the corresponding communication time delay of different SAGSIN devices. In the simulation, 100 participants were employed which contained 10% space devices, 15% aerial devices, 70% terrestrial devices, and 5% oceanic devices. The communication delay was randomly set according to Table 1 with a dynamic delay with a 5% fluctuation. The devices were divided into $B = 4$ groups where the latency was sorted from small to large. The time consumption comparisons between random scheduling-based aggregation and delay-aware-based aggregation with scheduling ratios $G = 0.1$, $G = 0.2$, and $G = 0.3$ are shown in Figure 3. From this figure, we can see that the delay-aware strategy had a lower time consumption than the random mechanism. In particular, the more available uplink channels the system had, the more obvious the obtained effects. Therefore, by fine grouping according to the delay, the total time consumption of the aggregation was sharply reduced. Therefore, the delay-aware FL is a time-efficient approach for SAGSINs.

Table 1. Communication parameters of the SAGSIN.

Device	Comm. Distance	Propagation Speed	Comm. Delay
GEO	~35,000 km	3×10^8 m/s	~ 0.117 s
MEO	~8000 km	Idem	~27 ms
LEO	~800 km	Idem	~3 ms
HAP	~20 km	Idem	~66.6 μ s
Airplane	~10 km	Idem	~33.3 μ s
eVTOL	~1 km	Idem	~3.33 μ s
UAV	~150 m	Idem	~ 0.5 μ s
Vehicle	~100 m	Idem	~ 0.33 μ s
Buoy	~500 m	Idem	~ 1.65 μ s
Shallow water	~50 m	1.5×10^3 m/s	~30 ms
Deep water	~1500 m	Idem	~1 s
Deep ocean	~5000 m	Idem	~3 s

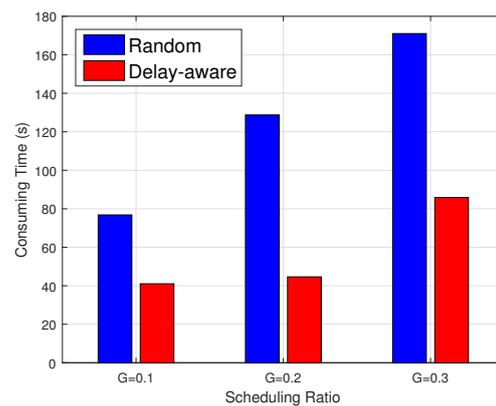


Figure 3. Time consumption vs. scheduling ratio of SAGSIN.

5.2. User-Level Privacy-Preserving-Based FML

Considering the federated participants in SAGSINs with a real-time decision demand and unfamiliar users, FL with a transfer learning function and privacy protection are well needed. In this part, we introduce DP into an FML scheme and propose the NbFML algorithm, which can provide privacy protection for local users by adding noise perturbations at the user's side. We propose a user-level local DP algorithm, called noise-based FML (NbFML), to protect the user's privacy. We analyze the user-level DP to provide a useful option for the balance between privacy protection and accuracy levels.

5.2.1. Differential Privacy Principle

First, we describe the concept of the DP algorithm. We assume (ϵ, δ) -DP can preserve the privacy of the distributed system. ϵ_i means the distinguishable bound between two neighboring datasets D_i and D'_i in a database. The δ_i describes the probabilities that these two neighboring datasets D_i and D'_i cannot be bounded by e^ϵ using a DP mechanism.

Definition 1 ((ϵ, δ) -DP). A randomized mechanism $\mathcal{M}: \mathcal{X} \rightarrow \mathcal{R}$ with a domain \mathcal{X} and range \mathcal{R} satisfies (ϵ, δ) -DP, if for all measurable sets $\mathcal{S} \in \mathcal{R}$ and for any two adjacent databases $D_i, D'_i \in \mathcal{X}$,

$$\Pr[\mathcal{M}(D_i) \in \mathcal{S}] \leq e^\epsilon \Pr[\mathcal{M}(D'_i)] + \delta. \quad (7)$$

In this part, the Gaussian mechanism was chosen as the privacy-protection mechanism \mathcal{M} , which employs the L_2 norm to measure the sensitivity, namely

$$\mathcal{M}(x) = s(x) + \mathcal{N}(0, \sigma^2 \mathbf{I}), \quad (8)$$

where \mathbf{I} is the identity matrix, and the sensitivity of $s(x)$ is defined as

$$\Delta s = \max_{D_i, D'_i} \|s(D_i) - s(D'_i)\|_2. \quad (9)$$

5.2.2. FML Framework

Let us consider a classical FML algorithm, which consists of one server and N participants. Each neural network of participant $i \in \mathbb{S}$ can be denoted by f_θ with $\theta \in \mathbb{R}^d$. The dataset with $(x_i^1, y_i^1), \dots, (x_i^j, y_i^j), \dots, (x_i^{J_i}, y_i^{J_i})$ of each participant can be denoted by J_i , where $|J_i|$ is the dataset, and (x_i, y_i) is the sample of this dataset. x_i is the input of the DL network and y_i denotes the output. The objective function can be defined as $l(\theta, (x_i, y_i))$. The loss of each user is

$$L_i(\theta, J_i) \triangleq \frac{1}{|J_i|} \sum_{(x_i^j, y_i^j) \in |J_i|} l(\theta, (x_i^j, y_i^j)), \quad (10)$$

where w_i can be calculated by the local data size according to $w_i = \frac{|J_i|}{\sum_{i \in \mathbb{S}} |J_i|}$. For the meta algorithm, given the model parameters of the i th participant, the user can update its parameters by one-step learning according to a gradient descent based on the support dataset D_i^S ,

$$\phi_i(\theta) = \theta - \alpha \nabla_\theta L(\theta, D_i^S), \quad (11)$$

where α is the learning rate. By evaluating the loss $L(\phi_i, D_i^S)$ in (10) based on the query dataset D_i^Q , the loss function of the FML system can be expressed as

$$\min_{\theta} \sum_{i \in \mathbb{S}} w_i L_i(\phi_i(\theta), D_i^Q). \quad (12)$$

5.2.3. NbFML Algorithm

Algorithm 3 describes our proposed NbFML and the framework can be seen in Figure 4. In order to protect the privacy of each user, the local (ϵ, δ) -DP protection is introduced into FML to train a serviceable model. The initial global parameter is denoted by $\theta^{(0)}$. In our NbFML algorithm, the server first broadcasts the targeted privacy level and the initial parameters to each participant.

On the user side, it first updates according to the UE local support data D_i^S . For the NbFML algorithm, given the model parameters, participants can update their own parameters with a few learning steps of gradient descent by utilizing D_i^S and (11). Then, the loss $L(\phi_i, D_i^S)$ is evaluated and θ_i is updated by using query data D_i^Q locally according to

$$\theta_i^{t+1} = \theta_i^t - \beta \nabla_{\theta} L(\phi_i(\theta), D_i^Q), \quad (13)$$

where β is the learning rate for the target dataset. The meta-learning process in NbFML is inspired by the model-agnostic meta-learning [62] but multiple local training epochs are allowed. After that, if the node is chosen by the server, it sends θ_i^{t+1} to the server. In this process, the model can be transferred from source to target by using local meta-learning. In this way, the user's online training can fix real-time data problems.

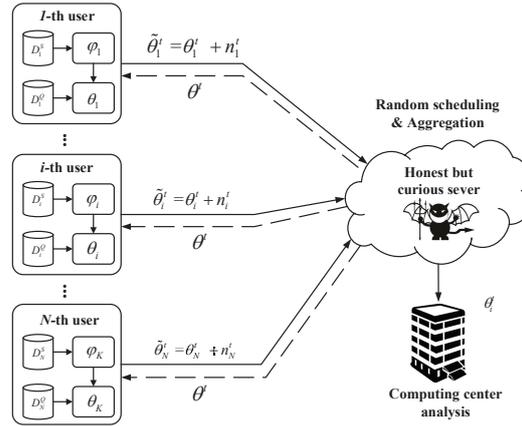


Figure 4. The training process of NbFML, where the training of the local UE model is carried out by meta learning.

Algorithm 3 NbFML

Require: Dataset $\{D_i\}_{i=1}^K$ of each user, communication rounds T , parameter θ , threshold C , and DP parameters (ϵ, δ) for all nodes.

- 1: Initialize: $\theta_k^t = \theta^t$ and $t = 0$
 - 2: **for** $t = 1 : T$ **do**
 - 3: **for** each user $i \in \{1, 2, \dots, K\}$ in parallel **do**
 - 4: **for** $t_{local} = 1$ to T_0 **do**
 - 5: Sample $i \in D_i^S$ uniformly at random, and update the local gradients:
 - 6: $g_{i,m}^t(D_{i,m}^S) = \nabla_{\theta^t} L_i(D_{i,m}^S, \theta^t)$;
 - 7: Update the local parameters:
 - 8: $\phi_i^t = \theta_i^t - \frac{1}{|D_i^S|} \sum_{m=1}^{|D_i^S|} \alpha g_{i,m}^t(D_{i,m}^S)$;
 - 9: Sample $i \in D_i$ uniformly at random, and update the local parameter using D^Q
 - 10: $g_{i,m}^t(D_{i,m}^Q) = \nabla_{\theta^t} L_i(D_{i,m}^Q, \phi_i^t)$;
 - 11: Clip the local gradients:
 - 12: $\hat{g}_{i,m}^t = g_{i,m}^t(D_{i,m}^Q) / \max(1, \frac{\|g_{i,m}^t(D_{i,m}^Q)\|_2}{C})$;
 - 13: obtain θ_i^{t+1} based on
 - 14: $\theta_{i,m}^{t+1} = \phi_{i,m}^t - \frac{1}{|D_i^Q|} \sum_{m=1}^{|D_i^Q|} \beta \hat{g}_{i,m}^t(D_{i,m}^Q)$;
 - 15: **end for**
 - 16: Calculate σ_i using DP parameters (ϵ, δ) ;
 - 17: Add noise $\hat{\theta}_i^t = \theta_i^t + n_i^t$;
 - 18: Send protected parameter $\hat{\theta}_i^t$ to the server;
 - 19: **end for**
 - 20: The server collects all the parameters $\{\hat{\theta}_i^{t+1}\}_{i=1}^K$, and updates $\theta^{t+1} = \frac{1}{N} \sum_{i=1}^K w_i \hat{\theta}_i^{t+1}$
 - 21: The server broadcasts the global parameters θ^{t+1}
 - 22: $t = t + 1$
 - 23: **end for**
- Ensure:** θ^T

On the server side, the server sets a threshold T for the aggregation time in advance. We consider that the FML process has been completed when the aggregation time reaches T . Then, the server collects all the local parameters set, aggregates local parameters with different weights to update the global parameters θ^t , and sends it to the user. The local perturbations generated by NbFML can confuse malicious attackers so that they cannot steal the i th UE information from the uploaded parameters $\tilde{\theta}^t$. After the model aggregation is completed, the user receives the aggregated parameters θ through the broadcast channels.

To update model parameters, we use the negative log-likelihood loss. The loss for the i th user can be expressed as $L_i(\theta, J_i)$. The loss function is

$$L_i(\theta, J_i) \triangleq -\log(p_i). \tag{14}$$

Here, p_i is the softmax output, and it can be rewritten as

$$p_i = \frac{e^{f_k}}{\sum_{j \in \mathcal{S}} e^{f_j}}, \tag{15}$$

where f is a vector containing the class scores for a single sample, that is to say, the network output, and f^k is an element for a certain class k in all j classes.

For the uplink case, we can use a clipping method to bound the parameters into a controllable range C , which can be denoted by $\theta_i \leq C$. Moreover, we assume that the batch size and local support set have the same quantity of data. Hence, we can define the i th user's local training process as

$$\begin{aligned} s_{\mathcal{U}}^{\mathcal{D}_i} &\triangleq \phi_i = \arg \min_{\phi} F_i(\phi, \mathcal{D}_i) \\ &= \frac{1}{|\mathcal{D}_i|} \sum_{j=1}^{|\mathcal{D}_i|} \arg \min_{\phi} F_i(\phi, \mathcal{D}_{i,j}), \end{aligned} \tag{16}$$

where $\mathcal{D}'_{i,j}$ means the j th sample of the i th dataset, and \mathcal{D}_i and \mathcal{D}'_i are two adjacent datasets.

Lemma 1. For the meta-learning-based FL process, the sensitivity for \mathcal{D}_i after the aggregation operation $\Delta s_{\mathcal{U}}^{\mathcal{D}_i}$ should satisfy

$$\Delta s_{\mathcal{U}}^{\mathcal{D}_i} = \frac{2(1 + \alpha H)Cp_i}{m}. \tag{17}$$

Lemma 1 tells us that the sensitivity of the FML aggregation operation can be affected by the threshold value, data quantity, random sampling probability, and learning rate of the meta-learning process.

Proof. First, the sensitivity of $s_{\mathcal{U}}^{\mathcal{D}_i}$ can be defined as

$$\begin{aligned} \Delta s_{\mathcal{U}}^{\mathcal{D}_i} &= \max_{\mathcal{D}_i, \mathcal{D}'_i} \|s_{\mathcal{U}}^{\mathcal{D}_i} - s_{\mathcal{U}}^{\mathcal{D}'_i}\| \\ &= \max_{\mathcal{D}_i, \mathcal{D}'_i} \left\| \frac{1}{|\mathcal{D}_i|} \sum_{j=1}^{|\mathcal{D}_i|} \arg \min_{\theta} F_i(\theta, \mathcal{D}'_{i,j}) \right. \\ &\quad \left. - \frac{1}{|\mathcal{D}'_i|} \sum_{j=1}^{|\mathcal{D}'_i|} \arg \min_{\theta} F_i(\theta, \mathcal{D}'_{i,j}) \right\| \\ &= \frac{1}{|\mathcal{D}'_i|} \max \|\phi_i - \phi'_i\|. \end{aligned} \tag{18}$$

We recall that the optimal model parameters are denoted by $\phi^* - \alpha \nabla L(\theta^*)$, i.e, ϕ^* can be obtained through a few gradient updates by using meta-learning model θ_c . To evaluate

the sensitivity of two adjacent datasets D_i and D'_i , we next calculate the gap between ϕ and ϕ' , which are learned from two adjacent datasets.

For convenience, let us denote $\tilde{\phi} = \theta^* - \alpha \nabla L(\theta^*)$.

$$\begin{aligned} \|\phi_i - \phi_i^*\| &\leq \|\theta_i - \theta_i^* - \alpha(\nabla L(\theta) - \nabla L(\theta^*))\| \\ &\leq \|\theta_i - \theta_i^*\| + \alpha \|\nabla L(\theta) - \nabla L(\theta^*)\| \\ &\leq (1 + \alpha H) \|\theta_i - \theta_i^*\|. \end{aligned} \quad (19)$$

Moreover, according to the triangle inequality, we have

$$\begin{aligned} \|\phi_i - \phi'_i\| &= \|\phi_i - \phi_i^* - (\phi'_i - \phi_i^*)\| \\ &\leq \|\phi_i - \phi_i^*\| + \|\phi'_i - \phi_i^*\|. \end{aligned} \quad (20)$$

Hence, we know that

$$\Delta s_{\mathcal{U}}^{\mathcal{D}_i} = \frac{2(1 + \alpha H)C}{m}. \quad (21)$$

Based on the previous formula, the sensitivity can be expressed as

$$\begin{aligned} \Delta s_D^{\mathcal{D}'_i} &= \max_{D_i, D'_i} \|p_i \theta_i(\mathcal{D}_i) - p_i w_i(\mathcal{D}'_i)\| \\ &= p_i \max_{D_i, D'_i} \|\phi_i - \phi'_i\| \\ &= p_i \Delta s_D^{\mathcal{D}'_i}. \end{aligned} \quad (22)$$

By combining (21) with (22), the proof is completed. \square

5.2.4. Numerical Experiment

In order to show the effectiveness of NbFML, we performed extensive simulations on famous image datasets. These include large-scale datasets of small images popular with DL methods. We used MNIST as the support dataset. We used MNIST-m as the query dataset, which contains the original set over patches randomly extracted from color photos from BSDS500 [63]. In our experiments, for each client, the local data contained 80% support set and 20% query set, as we consider the generalization to new industry users as a crucial property of FML. The baseline model in our experiments was a four-layer fully connected neural network.

The proposed NbFML and classical noise-based FL (NbFL) convergence performance values are compared in Figure 5, where models were trained on MNIST and validated on MNIST-m. In the simulation, the protection levels were set to $\epsilon = 50$ and $\epsilon = 100$, and each user had 800 training samples. The global communication round was $T = 40$, and the local epoch was $T_0 = 5$. From Figure 5, we find that with the increase in protection level ϵ , the accuracy of both NbFL and NbFML decreased. This is because the increased protection level ϵ caused more noise to be added to the parameters. The stricter the privacy protection level, the lower the accuracy of user model training. That is to say, the system should trade off accuracy and privacy protection level of users to achieve a desired state according to the actual situation. Moreover, we see that the proposed NbFML had excellent performance, while the classical NbFL accuracy decreased since it had no ability to do transfer learning. The proposed NbFML utilizes the meta-learning method, which is able to tune the network to the labeled data in the target dataset. Hence, the NbFML can transfer its parameters from the source dataset to the target dataset fast. NbFML has a wide application, especially in FL-based SAGSINs with the demand of real-time decision systems.

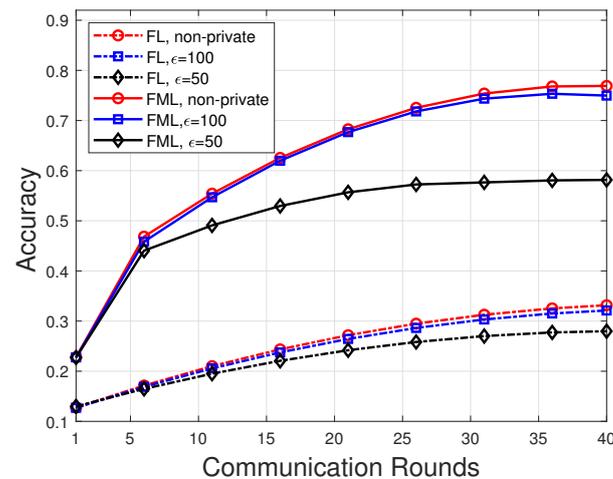


Figure 5. Accuracy vs. communication rounds with different protection levels, where models were trained on MNIST and validated on MNIST-m.

6. Future Direction

Undeniably, FL is a promising research area. With the development of physical-layer technologies in all manner of ways, such as holographic teleportation, intelligent reflecting surfaces, quantum communication, visible-light communication, underwater acoustic communication, and molecular communication, it will become more and more important to pay attention to the effectiveness and reliability of FL communications. Therefore, FL aggregation with different communication technologies should be further designed, in order to reduce the total communication epochs and reduce the transmitted information size at each epoch. Meanwhile, green communications also need to be considered in SAGSINs, especially in massive network traffic and dynamic network environments. Moreover, in 6G communication scenarios, the energy efficiency of FL should be a concern because of charging constraints and battery limitations. Moreover, based on advanced communications, heterogeneous characteristics, and dynamic characteristics, the system robustness and convergence analysis of FL-based SAGSINs can be further considered, especially with non-IID data. Last but not least, platform implementation remains a problem in SAGSINs. Currently, there are only a few experiments on the whole SAGSIN platform. Following the demands of information acquisition and fusion, a novel SAGSIN platform is also very desirable.

7. Conclusions

In this paper, we first presented a brief overview of FL and SAGSINs. Then, we gave a survey of recent applications on FL-based SAGSINs, including satellite networks, aerial networks, terrestrial networks and marine networks. Next, we illustrated existing challenges in FL-enabled SAGSINs, such as heterogeneous networks, wireless communication challenges, privacy threats, and data isomerism. In addition, two cases were analyzed. One provided a more specific aggregation algorithm for multi-scale delay, and the other aimed at protecting privacy in SAGSINs at the user level. Finally, we pointed out future directions, and we concluded this paper.

Author Contributions: Conceptualization, H.Z., F.J. and Y.W.; methodology, H.Z., F.J. and Y.W.; software, H.Z., Y.W. and K.Y.; validation, H.Z., Y.W. and K.Y.; formal analysis, H.Z., Y.W. and K.Y.; investigation, H.Z., Y.W. and K.Y.; resources, H.Z., Y.W. and K.Y.; data curation, H.Z., Y.W. and K.Y.; writing—original draft preparation, H.Z., F.J. and F.C.; writing—review and editing, H.Z., K.Y. and F.C.; visualization, H.Z.; supervision, F.J. and F.C.; project administration, F.J. and F.C.; funding acquisition, F.J. and F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Key Laboratory of Marine Environmental Survey Technology and Application, Ministry of Natural Resources, P. R. China under Grant MESTA-2023-B001, in part by the National Natural Science Foundation of China under Grant 62192712, and in part by the Stable Supporting Fund of National Key Laboratory of Underwater Acoustic Technology under Grant JCKYS2022604SSJS007.

Data Availability Statement: Raw data were generated using the Matlab simulator.

Acknowledgments: The authors would like to thank the anonymous reviewers for their careful assessment of our work.

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

References

1. O’Shea, T.; Hoydis, J. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [[CrossRef](#)]
2. Guo, H.; Li, J.; Liu, J.; Tian, N.; Kato, N. A Survey on Space-Air-Ground-Sea Integrated Network Security in 6G. *IEEE Commun. Surv. Tuts.* **2022**, *24*, 53–87. [[CrossRef](#)]
3. Zawish, M.; Dharejo, F.A.; Khowaja, S.A.; Raza, S.; Davy, S.; Dev, K.; Bellavista, P. AI and 6G Into the Metaverse: Fundamentals, Challenges and Future Research Trends. *IEEE Open J. Commun. Soc.* **2024**, *5*, 730–778. [[CrossRef](#)]
4. Yang, B.; Liu, S.; Xu, T.; Li, C.; Zhu, Y.; Li, Z.; Zhao, Z. AI-Oriented Two-Phase Multifactor Authentication in SAGINs: Prospects and Challenges. *IEEE Consum. Electron. Mag.* **2024**, *13*, 79–90. [[CrossRef](#)]
5. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [[CrossRef](#)]
6. Liu, J.; Shi, Y.; Fadlullah, Z.M.; Kato, N. Space-Air-Ground Integrated Network: A Survey. *IEEE Commun. Surv. Tuts.* **2018**, *20*, 2714–2741. [[CrossRef](#)]
7. Hong, T.; Lv, M.; Zheng, S.; Hong, H. Key Technologies in 6G SAGS IoT: Shape-Adaptive Antenna and Radar-Communication Integration. *IEEE Netw.* **2021**, *35*, 150–157. [[CrossRef](#)]
8. Khan, M.A.; Alzahrani, B.A.; Barnawi, A.; Al-Barakati, A.; Irshad, A.; Chaudhry, S.A. A Resource Friendly Authentication Scheme for Space-air-ground-sea Integrated Maritime Communication Network. *Ocean. Eng.* **2022**, *250*, 110894. [[CrossRef](#)]
9. Xu, F.; Yang, F.; Zhao, C.; Wu, S. Deep Reinforcement Learning Based Joint Edge Resource Management in Maritime Network. *China Commun.* **2020**, *17*, 211–222. [[CrossRef](#)]
10. Cheng, N.; Quan, W.; Shi, W.; Wu, H.; Ye, Q.; Zhou, H.; Zhuang, W.; Shen, X.; Bai, B. A Comprehensive Simulation Platform for Space-Air-Ground Integrated Network. *IEEE Wirel. Commun.* **2020**, *27*, 178–185. [[CrossRef](#)]
11. Bin, L.; Dianhui, Z.; Xiaoming, H.; Feng, W. Field Experiment of Key Technologies Oriented to Space-Air-Ground-Sea Integrated Network. *Peng Cheng Lab. Commun.* **2020**, *1*, 24–28.
12. Kato, N.; Fadlullah, Z.M.; Tang, F.; Mao, B.; Tani, S.; Okamura, A.; Liu, J. Optimizing Space-Air-Ground Integrated Networks by Artificial Intelligence. *IEEE Wirel. Commun.* **2019**, *26*, 140–147. [[CrossRef](#)]
13. Niknam, S.; Dhillon, H.S.; Reed, J.H. Federated Learning for Wireless Communications: Motivation, Opportunities, and Challenges. *IEEE Commun. Mag.* **2020**, *58*, 46–51. [[CrossRef](#)]
14. Chen, M.; Poor, H.V.; Saad, W.; Cui, S. Wireless Communications for Collaborative Federated Learning. *IEEE Commun. Mag.* **2020**, *58*, 48–54. [[CrossRef](#)]
15. Khan, L.U.; P; ey, S.R.; Tran, N.H.; Saad, W.; Han, Z.; Nguyen, M.N.; Hong, C.S. Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism. *IEEE Commun. Mag.* **2020**, *58*, 88–93. [[CrossRef](#)]
16. Savazzi, S.; Nicoli, M.; Bennis, M.; Kianoush, S.; Barbieri, L. Opportunities of Federated Learning in Connected, Cooperative, and Automated Industrial Systems. *IEEE Commun. Mag.* **2021**, *59*, 16–21. [[CrossRef](#)]
17. Li, Z.; Sharma, V.; Mohanty, S.P. Preserving Data Privacy via Federated Learning: Challenges and Solutions. *IEEE Consum. Electron. Mag.* **2020**, *9*, 8–16. [[CrossRef](#)]
18. Elbir, A.M.; Papazafeiropoulos, A.K.; Chatzinotas, S. Federated Learning for Physical Layer Design. *IEEE Commun. Mag.* **2021**, *59*, 81–87. [[CrossRef](#)]
19. Shome, D.; Waqar, O.; Khan, W.U. Federated Learning and Next Generation Wireless Communications: A Survey on Bidirectional Relationship. *arXiv* **2021**, arXiv:2110.07649.
20. Elbir, A.M.; Coleri, S. Federated Learning for Channel Estimation in Conventional and RIS-Assisted Massive MIMO. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 4255–4268. [[CrossRef](#)]
21. Mashhadi, M.B.; Shlezinger, N.; Eldar, Y.C.; Gündüz, D. Fedrec: Federated Learning of Universal Receivers Over Fading Channels. In Proceedings of the 2021 IEEE Statistical Signal Processing Workshop (SSP), Rio de Janeiro, Brazil, 11–14 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 576–580.
22. Ji, Z.; Chen, L.; Zhao, N.; Chen, Y.; Wei, G.; Yu, F.R. Computation Offloading for Edge-Assisted Federated Learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9330–9344. [[CrossRef](#)]

23. Messaoud, S.; Bradai, A.; Ahmed, O.B.; Quang, P.T.A.; Atri, M.; Hossain, M.S. Deep Federated Q-Learning-Based Network Slicing for Industrial IoT. *IEEE Trans. Ind. Informat.* **2021**, *17*, 5572–5582. [[CrossRef](#)]
24. Yang, H.H.; Liu, Z.; Quek, T.Q.S.; Poor, H.V. Scheduling Policies for Federated Learning in Wireless Networks. *IEEE Trans. Commun.* **2020**, *68*, 317–333. [[CrossRef](#)]
25. Yu, L.; Albelaihi, R.; Sun, X.; Ansari, N.; Devetsikiotis, M. Jointly Optimizing Client Selection and Resource Management in Wireless Federated Learning for Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 4385–4395. [[CrossRef](#)]
26. Samarakoon, S.; Bennis, M.; Saad, W.; Debbah, M. Distributed Federated Learning for Ultra-Reliable Low-Latency Vehicular Communications. *IEEE Trans. Commun.* **2020**, *68*, 1146–1159. [[CrossRef](#)]
27. Chen, M.; Poor, H.V.; Saad, W.; Cui, S. Convergence Time Optimization for Federated Learning Over Wireless Networks. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 2457–2471. [[CrossRef](#)]
28. Yang, J.; Liu, Y.; Chen, F.; Chen, W.; Li, C.; Asynchronous Wireless Federated Learning with Probabilistic Client Selection, *IEEE Trans. Wireless Commun.* **2023**, early access. [[CrossRef](#)]
29. Lin, X.; Liu, Y.; Chen, F.; Ge, X.; Huang, X.; Joint Gradient Sparsification and Device Scheduling for Federated Learning, *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 1407–1419. [[CrossRef](#)]
30. Zhao, Z.; Feng, C.; Yang, H.H.; Luo, X. Federated-Learning-Enabled Intelligent Fog Radio Access Networks: Fundamental Theory, Key Techniques, and Future Trends. *IEEE Wirel. Commun.* **2020**, *27*, 22–28. [[CrossRef](#)]
31. Nguyen, D.C.; Ding, M.; Pham, Q.V.; Pathirana, P.N.; Le L.B.; Seneviratne, A.; Li, J.; Niyato, D.; Poor, H.V. Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges. *IEEE Internet Things J.* **2021**, *8*, 12806–12825. [[CrossRef](#)]
32. Xiong, Z.; Cai, Z.; Takabi, D.; Li, W. Privacy Threat and Defense for Federated Learning With Non-i.i.d. Data in AIoT. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1310–1321. [[CrossRef](#)]
33. Ma, C.; Li, J.; Ding, M.; Yang, H.H.; Shu, F.; Quek, T.Q.; Poor, H.V. On Safeguarding Privacy and Security in the Framework of Federated Learning. *IEEE Netw.* **2020**, *34*, 242–248. [[CrossRef](#)]
34. Lin, X.; Liu, Y.; Chen, F.; Huang, Y.; Ge, X. Stochastic gradient compression for federated learning over wireless network, *China Commun.* **2024**, *21*, 230–247.
35. Li, K.; Zhou, H.; Tu, Z.; Wang, W.; Zhang, H. Distributed Network Intrusion Detection System in Satellite-Terrestrial Integrated Networks Using Federated Learning. *IEEE Access* **2020**, *8*, 214852–214865. [[CrossRef](#)]
36. Huang, A.; Liu, Y.; Chen, T.; Zhou, Y.; Sun, Q.; Chai, H.; Yang, Q. StarFL: Hybrid Federated Learning Architecture for Smart Urban Computing. *ACM Trans. Intell. Syst. Technol.* **2021**, *12*, 1–23. [[CrossRef](#)]
37. Razmi, N.; Matthiesen, B.; Dekorsy, A.; Popovski, P. Ground-Assisted Federated Learning in LEO Satellite Constellations. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 717–721. [[CrossRef](#)]
38. Chen, H.; Xiao, M.; Pang, Z. Satellite-Based Computing Networks with Federated Learning. *IEEE Wirel. Commun.* **2022**, *29*, 78–84. [[CrossRef](#)]
39. Fadlullah, Z.M.; Kato, N. On Smart IoT Remote Sensing over Integrated Terrestrial-Aerial-Space Networks: An Asynchronous Federated Learning Approach. *IEEE Netw.* **2021**, *35*, 129–135. [[CrossRef](#)]
40. Zhang, H.; Hanzo, L. Federated Learning Assisted Multi-UAV Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14104–14109. [[CrossRef](#)]
41. Liu, Y.; Nie, J.; Li, X.; Ahmed, S.H.; Lim, W.Y.B.; Miao, C. Federated Learning in the Sky: Aerial-Ground Air Quality Sensing Framework With UAV Swarms. *IEEE Internet Things J.* **2021**, *8*, 9827–9837. [[CrossRef](#)]
42. Jing, Y.; Qu, Y.; Dong, C.; Shen, Y.; Wei, Z.; Wang, S. Joint UAV Location and Resource Allocation for Air-Ground Integrated Federated Learning. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
43. Pham, Q.-V.; Zeng, M.; Ruby, R.; Huynh-The, T.; Hwang, W.-J. UAV Communications for Sustainable Federated Learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3944–3948. [[CrossRef](#)]
44. Xu, C.; Mao, Y. An Improved Traffic Congestion Monitoring System Based on Federated Learning. *Information* **2020**, *11*, 365. [[CrossRef](#)]
45. Hua, G.; Zhu, L.; Wu, J.; Shen, C.; Zhou, L.; Lin, Q. Blockchain-Based Federated Learning for Intelligent Control in Heavy Haul Railway. *IEEE Access* **2020**, *8*, 176830–176839. [[CrossRef](#)]
46. Wang, X.; Han, Y.; Wang, C.; Zhao, Q.; Chen, X.; Chen, M. In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning. *IEEE Netw.* **2019**, *33*, 156–165. [[CrossRef](#)]
47. Savazzi, S.; Nicoli, M.; Rampa, V. Federated Learning With Cooperating Devices: A Consensus Approach for Massive IoT Networks. *IEEE Internet Things J.* **2020**, *7*, 4641–4654. [[CrossRef](#)]
48. Lu, Y.; Huang, X.; Dai, Y.; Maharjan, S.; Zhang, Y. Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT. *IEEE Trans. Ind. Informat.* **2020**, *16*, 4177–4186. [[CrossRef](#)]
49. Qin, Z.; Ye, J.; Meng, J.; Lu, B.; Wang, L. Privacy-Preserving Blockchain-Based Federated Learning for Marine Internet of Things. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 159–173. [[CrossRef](#)]
50. Zhang, Z.; Guan, C.; Chen, H.; Yang, X.; Gong, W.; Yang, A. Adaptive Privacy Preserving Federated Learning for Fault Diagnosis in Internet of Ships. *IEEE Internet Things J.* **2022**, *9*, 6844–6854. [[CrossRef](#)]
51. Qiu, T.; Zhao, Z.; Zhang, T.; Chen, C.; Chen, C.L.P.; Underwater Internet of Things in Smart Ocean: System Architecture and Open Issues. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4297–4307. [[CrossRef](#)]

52. Kwon, D.; Jeon, J.; Park, S.; Kim, J.; Cho, S. Multiagent DDPG-Based Deep Learning for Smart Ocean Federated Learning IoT Networks. *IEEE Internet Things J.* **2020**, *7*, 9895–9903. [[CrossRef](#)]
53. Zhao, H.; Ji, F.; Li, Q.; Guan, Q.; Wang, S.; Wen, M. Federated Meta-Learning Enhanced Acoustic Radio Cooperative Framework for Ocean of Things. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 474–486. [[CrossRef](#)]
54. Hou, X.; Wang, J.; Fang, Z.; Zhang, X.; Song, S.; Zhang, X.; Ren, Y. Machine-Learning-Aided Mission-Critical Internet of Underwater Things. *IEEE Netw.* **2021**, *35*, 160–166. [[CrossRef](#)]
55. Lin, S.; Yang, G.; Zhang, J. Real-time edge intelligence in the making: A Collaborative learning framework via federated meta-learning. *arXiv* **2020**, arXiv:2001.03229.
56. Yue, S.; Ren, J.; Xin, J.; Lin, S.; Zhang, J. Inexact-ADMM Based Federated Meta-Learning for Fast and Continual Edge Learning. In Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, Shanghai China, 26–29 July 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 91–100.
57. Lin, S.; Yang, L.; He, Z.; Fan, D.; Zhang, J. MetaGater: Fast Learning of Conditional Channel Gated Networks via Federated Meta-Learning. In Proceedings of the 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 4–7 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 164–172.
58. Yue, S.; Ren, J.; Xin, J.; Zhang, D.; Zhang, Y.; Zhuang, W. Efficient Federated Meta-Learning Over Multi-Access Wireless Networks. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 1556–1570. [[CrossRef](#)]
59. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 308–318.
60. Wang, N.; Xiao, X.K.; Yang, Y.; Zhao, J.; Hui, S.C.; Shin, H.; Shin, J.; Yu, G. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 638–649.
61. Wang, S.; Huang, L.S.; Nie, Y.W.; Zhang, X.Y.; Wang, P.Z.; Xu, H.L.; Yang, W. Local differential private data aggregation for discrete distribution estimation. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *30*, 2046–2059. [[CrossRef](#)]
62. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic Meta-learning for Fast Adaptation of Deep Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
63. Ganin, Y.; Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1180–1189.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.