Semantic Communication in Satellite-borne Edge Cloud Network for Computation Offloading

Guhan Zheng, Qiang Ni, *Senior Member, IEEE*, Keivan Navaie, *Senior Member, IEEE*, and Haris Pervaiz, *Member, IEEE*

***Abstract*—The low earth orbit (LEO) satellite-borne edge cloud (SEC) and machine learning (ML) based semantic communication (SemCom) are both enabling technologies for 6G systems facilitating computation offloading. Nevertheless, integrating SemCom into the SEC networks for user computation offloading introduces semantic coder updating requirements as well as additional semantic extraction costs. Offloading user computation in SEC networks via SemCom also results in new functional challenges considering, e.g., latency, energy, and privacy. In this paper, we present a novel SemCom-assisted SEC (SemCom-SEC) framework for computation offloading of resource-limited users. We then propose an adaptive pruning-split federated learning (PSFed) method for updating the semantic coder in SemCom-SEC. We further show that the proposed method guarantees training convergence speed and accuracy. This method also improves the privacy of the semantic coder while reducing training delay and energy consumption. In the case of trained semantic coders in service, for the users processing computational tasks, the main objective is to minimise the users' delay and energy consumption, subject to sustaining users' privacy and fairness amongst them. This problem is then formulated as an incomplete information mixed integer nonlinear programming (MINLP) problem. A new computational task processing scheduling (CTPS) mechanism is also proposed based on the Rubinstein bargaining game. Simulation results demonstrate the proposed PSFed and game theoretical CTPS mechanism outperforms the baseline solutions reducing delay and energy consumption while enhancing users' privacy.**

***Index Terms*—Satellite-borne edge cloud, SemCom, computation offloading, delay, energy consumption, privacy.**

I. Introduction

A. Background

MULTI-ACCESS edge computing (MEC) is emerging as one of the key techniques for next-generation wireless communication systems [2]. MEC enables the development of Internet of Things (IoT) applications and improves network performance and quality of service (QoS) [3]. MEC brings cloud services closer to the users at the network edge, e.g., base stations (BSs), and roadside units (RSUs) providing them with abundant computational resources. Therefore, users can offload their computationally intensive tasks to the MEC for faster processing.

Nevertheless, users located in remote areas or disaster zones might not be able to connect to terrestrial edge cloud network infrastructures. Alternatively, such under-served users may offload their computationally intensive tasks to remote core cloud servers via Geosynchronous Equatorial Orbit (GEO) or Medium Earth Orbit (MEO) satellites. In addition to the costs, the corresponding propagation latency to and from the satellite platforms however impedes the delay requirements of these users. Using Low Earth Orbit (LEO) satellites can partly address this issue by providing lower propagation latency as their orbits are much closer to the ground compared to GEO and MEO satellites. Comparing to GEO and MEO, constellations of LEO satellites also provide low-cost, high-throughput services and extensive radio coverage. To further reduce the propagation delay, the satellite-borne edge cloud (SEC) setting was proposed, where the offloaded processing is conducted on board the LEO satellite, hence reducing the propagation delay by a factor of 2 [4], [5].

Adopting SEC for users in remote areas or disaster zones has been recently investigated in [6] and [7]. The authors in [6], and [7] mainly focused on developing offloading decisions that minimise offloading delay or energy consumption for cases where users have direct radio links to the satellites. (e.g., in C-Band). An alternative access scenario is proposed in [8], where the user transmits to the SEC indirectly through an intermediary terrestrial-station-terminal (TST). In this approach, the user transmission to the TST is on a C-band radio link and TST communicates to the SEC through a K-band radio link. Wang et al. [9] also proposed a dual-edge cloud network, where the edge servers are placed on both BSs and LEO satellites. In this approach, a BS acts as a TST to assist users with computation offloading to the SEC. Similarly, [10] proposed an energy-efficient strategy for terrestrial users to offload computing tasks to the SEC via TSTs. Tang et al. [11] further investigated the impact of the core cloud on users' offloading decisions. They then proposed a minimal energy consumption computing offloading decision method, where users access SEC directly.

B. Challenges: SEC for user offloading

The approaches mentioned above frequently confine their investigations to a singular connectivity scenario between users and the SEC. In essence, by concentrating solely on specific performance aspects, such as energy consumption or latency,

Manuscript received 8 July 2023; revised 15 November 2023; accepted 15 December 2023. This work was supported in part by the Western O-RAN Deployment (ONE WORD) Project. Part of this work was presented at the IEEE ICC 2022 DDINS workshop [1]. (Corresponding author: Qiang Ni.)

G. Zheng, Q. Ni and K. Navaie are with the School of Computing and Communications, Lancaster University, LA1 4WA, UK (Email: g.zheng2, q.ni, k.navaie@lancaster.ac.uk).

H. Pervaiz is with the School of Computer Science and Electronic Engineering (CSEE), University of Essex, CO4 3SQ, UK (Email: haris.pervaiz@essex.ac.uk).

potential privacy concerns and associated risks to users are disregarded. This poses inherent risks to users. For example, prioritizing latency without considering energy consumption and privacy may lead to a user in the desert swiftly losing the ability to communicate, with this information potentially accessible by a third party. To address this issue, in this paper, we investigate SEC incorporating various access modalities, task processing entities, latency, energy consumption, and privacy of users.

Moreover, in the majority of instances, offloading substantial computing tasks to the SEC demands an exceptionally high transmission rate and substantial throughput. Consequently, alongside considerations of latency, energy efficiency, and data privacy, the computation offloading to SEC encounters a fundamental constraint—the inherent limitation of accessible radio spectrum. Hence, it is imperative to devise techniques that markedly enhance the spectrum efficiency of these systems, all the while upholding the quality of service (QoS) in the offloading process. A promising approach to address this issue is semantic communication (SemCom) based on machine learning (ML) [12].

SemCom leverages ML techniques for information transmission. A goal-oriented semantic encoder, powered by ML, selectively extracts semantic information from the transmitted or offloaded content. Rather than transmitting raw data, only the essential semantic information is conveyed, later decoded by the ML-based semantic decoder. This approach significantly enhances spectrum efficiency by balancing the communication load against the computational load through machine learning. Moreover, it mitigates the impact of unstable radio links, such as variable path loss due to weather conditions commonly observed in high-frequency satellite links. SemCom thus plays a pivotal role in the significant enhancement of the performance and speed of offloading. The integration of SemCom and SEC for computation offloading presents a promising solution to address the challenges of task offloading in the next generation of wireless communications.

C. Challenges: SemCom for SEC

Integrating SemCom and SEC for computation offloading requires a carefully designed architecture. Such an architecture needs to consider various possible task-processing entities (satellites and terrestrial cloud) and various user access methods (direct and indirect) to the SEC network. Furthermore, goal-oriented ML-based SemCom coders need to be updated in real-time according to new transmission content [13].

In the SEC network, updating the semantic coder presents several emerging challenges, e.g., mobility of SEC, low tolerance of service interruption and energy consumption, and privacy. However, the existing distributed learning frameworks designed for SemComs in generic networks (e.g., [14]–[16]) do not seamlessly translate to the SEC network. For instance, Xie and Qin [14] introduced a pruned lite ML model tailored for distributed semantic coders. Their approach focuses on refining models over edges rather than updating goal-oriented coders in a federated training approach. Qin et al. [16] proposed a general SemCom framework involving users and terrestrial base station edge clouds. In [15], the SemCom framework also includes users and terrestrial base station edge clouds, with the distinction that users in [15] must provide information to base stations for semantic extraction. However, these frameworks suffer from prolonged service interruptions, increased energy consumption, and heightened privacy risks within SEC networks. Furthermore, these methods only engage users and the edge cloud. In SEC offloading scenarios, the SemCom for offloading framework necessitates the active participation of all parties including users, terrestrial-station-terminal, satellites, and terrestrial clouds. The aforementioned research underscores the critical need to develop efficient distributed learning methods for updating semantic coders in SemCom SEC networks.

In addition to the above, SemCom alters the transmission paradigm of SEC networks by increasing the computational load while reducing the communication load. Users are therefore required to develop optimal computational task strategies in case trained semantic coders are utilised for computation offloading. Such strategies need to be developed taking into account not only scenarios specific to SemCom in the SEC, but also operational factors that have not been considered in the existing SEC offloading research. Such factors include using both access modalities, the task processing entities, latency, energy consumption and privacy.

D. Contributions

To tackle the above-mentioned challenges, in this paper, we propose a novel SemCom-assisted SEC (SemCom-SEC) framework for terrestrial users' computation offloading. In our proposed method, we split the SemCom service into in-maintenance (i.e., semantic coders need updating) and in-service (i.e., trained semantic coders are utilised for computation offloading) scenarios. For the in-maintenance scenario, we investigate real-time updating of deployed semantic coders in SemCom-SEC. A pruning-split federated learning (PSFed) approach is then proposed to update semantic coders considering offloading QoS while privacy-preserving. For the in-service scenario, we study the computational task processing challenge of terrestrial users in the new SemCom paradigm. We then propose a new computational task processing scheduling (CTPS) mechanism based on the Rubinstein bargaining game to minimise the users' processing delay and energy consumption while preserving users' privacy. The main contributions of this paper are summarised as follows:

- We integrate the SemCom and SEC networks and propose a novel SemCom-SEC framework enabling task offloading for under-served users. Diverging from current SemCom frameworks, which exclusively factor in users and terrestrial edge clouds, the envisioned framework extends its reach by deploying semantic coders on both the TSTs and satellites. Furthermore, SemCom-SEC accommodates a variety of user task-processing approaches and access modalities. Computational tasks for users can occur locally, at SEC, or in the core cloud server.

Additionally, users possess the flexibility to access LEO satellites either directly or through the semantic encoder-equipped TST.

- We then propose a PSFed approach for semantic coder updating for the SemCom-SEC framework enabling computation offloading. PSFed adaptively "splits" and "prunes" the semantic coders for federated aggregation subject to various users' personalised conditions. In contrast to the conventional "split" and "prunes" models, the semantic coder model components remain intact after updating. PSFed reduces the consumption of training communication resources and improves the privacy of the trained encoder while enhancing the training convergence speed and model accuracy.
- We introduce an innovative CTPS mechanism, distinct from previous studies that only address partial performance considerations. Our approach takes a comprehensive stance, jointly addressing user privacy, delay, energy consumption, and fairness to tackle the novel challenge of incomplete information task processing scheduling in SemCom-SEC. The CTPS operates in two steps: firstly, a game-theoretic model is crafted to transform this mixed-integer nonlinear programming (MINLP) problem from incomplete information, stemming from privacy concerns, into a complete information problem. In the second step, the converted complete information MINLP problem is decomposed and solved through the application of the Lagrangian dual decomposition method.

The rest of the paper is organised as the following. Section II presents the system model of the proposed SemCom-SEC framework. In Section III and Section IV, we then investigate the unique challenges and corresponding solutions for SemCom in-maintenance and in-service scenarios, respectively. The performance of the proposed PSFed and CTPS are then evaluated and analysed by simulations in Section V. Finally, conclusions are drawn in Section VI.

II. System Model

In this section, the system model of the proposed SemCom-SEC is introduced. We then provide the computing, communication, path loss and semantic coder training model.

A. System description

Consider the SemCom-SEC (Fig.1), where terrestrial users are located in areas without having access to terrestrial edge service. Users can offload computation-intensive tasks to LEO SEC. In practice, an LEO satellite constellation is similar to a cellular network operating above the ground [17]. Whereas the space cellular network is on the move, while ground users are relatively stationary.

We consider both types of approaches for users to access the SEC for computation offloading [8]. Users can communicate with LEO satellites directly through a C-band user-satellite radio link. Furthermore, they are also allowed to indirectly access the SEC through a TST via a C-band link to TST, and a Ka-band link between TST and SEC. The terrestrial C-band user-TST link spectrum resources are utilised in an orthogonal frequency division multiple access (OFDMA) setting to optimise the utilisation of terrestrial radio resources [10].

To improve the spectrum efficiency and QoS of SEC networks, semantic coders are deployed on the TSTs and LEO satellites for transmitting offloaded tasks over Ka-band. This is due to TSTs being primarily responsible for transmitting significant amounts of tasks to satellites and requiring extremely high spectral efficiency. Furthermore, their service area is fixed and the content to assist in task offloading (e.g., scene perception task, augmented reality task) only minimally varies. The mobility of the users causes the fact that the offloading content is often variable. For instance, the content of the transmission when offloading a scene perception task varies depending on the scene. The content-oriented semantic coders need to be constantly updated as the user moves. We thus consider factors such as utilisation, and reliability, for which goal-oriented SemCom is most appropriate for the TST-satellite link in SEC networks. Moreover, due to the dynamic nature of the system and the limited storage resources of LEO satellites, it is not viable to store semantic decoders for all TSTs on the route. The semantic coders are therefore stored on the TST. Similarly, for economic and satellite storage resources considerations, at least the trained decoder of TSTs should be the same for the same transmission task [18]. The TST delivers the related semantic decoders to the corresponding satellite when it needs to perform SemCom.

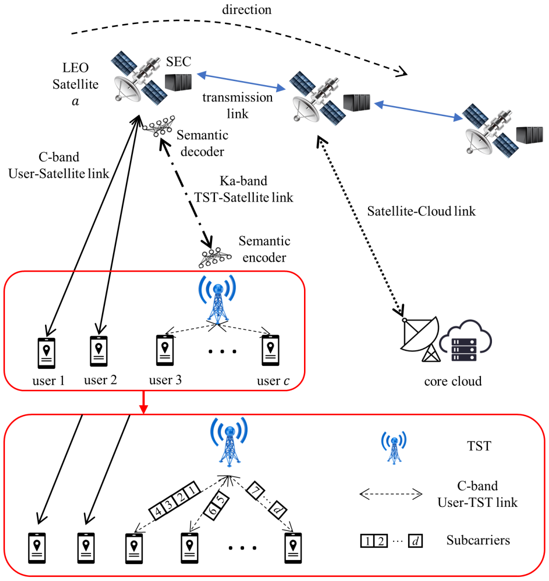


Fig. 1: The proposed SemCom-SEC framework.

Furthermore, LEO satellites can alternatively connect to the cloud servers on the terrestrial network via Ka-band backhaul links to provide cloud service for users.

In this model, a user may process indivisible computational tasks in either of the following five scenarios: 1) computing locally; 2) offloading the tasks to SEC over the user-satellite link; 3) offloading the tasks to the SEC via TST; 4) offloading the tasks to terrestrial cloud over the user-satellite link; 5) offloading the tasks to the terrestrial cloud via TST-satellite link.

B. Computiong models

Denote the set of LEO satellites as $\mathcal{A} = \{1, 2, ..., a, ..., A\}$ and set of TSTs as $\mathcal{B} = \{1, 2, ..., b, ..., B\}$. A TST b is on the terrestrial and provides service to C users within the coverage as a small cell in which the set of users in TST b's service range is denoted by $\mathcal{C} = \{1, 2, ..., c, ..., C\}$. We consider each terrestrial user c to have indivisible computational sensitive tasks with the size in bits of $m_c \in \{m_1, m_2, ..., m_c, ..., m_C\}$, and the CPU cycles needed to execute one bit of tasks is δ. The local computation task latency of the user c can be given by

$$t_c^{LC} = \frac{\delta m_c}{f_c}, \tag{1}$$

where f_c is user c's CPU-cycle frequency with the unit cycles/s. The energy required to calculate locally is hence expressed as [1]:

$$E_c^{LC} = p_c^{LC} t_c^{LC} = \varepsilon f_c^3 \frac{\delta m_c}{f_c} = \varepsilon \delta m_c f_c^2, \tag{2}$$

where $p_c^{LC} = \varepsilon f_c^3$ is the power needed to be computing locally and ε is the energy factor related to the electronics [19].

Similarly, if user c chooses to offload the tasks to SEC or the terrestrial cloud, the computational latency can be obtained by

$$t_c^{SEC} = \frac{\delta m_c}{f_a}, \tag{3}$$

$$t_c^{Cloud} = \frac{\delta m_c}{f_{Cloud}}, \tag{4}$$

where f_a and f_{Cloud} are the CPU-cycle frequency of the LEO satellite a being offloaded to and terrestrial cloud, respectively. Similar to [11] and [20], we assume that all LEO satellites have similar computing capabilities.

C. Communication models

There are two options for each user to access LEO satellites, i.e., directly access the LEO satellite or via a semantic encoder deployed on the TST. The total bandwidth of the C-band user-TST link is divided into D_0 orthogonal sub-carriers based on OFDMA manner [10]. The transmission rate of the user c to the TST b on a sub-carrier d_0 in this link is

$$r_{c,d}^{cb} = B_{d_0}^{cb} \log_2(1 + \frac{p_{c,d_0}^{cb} g_{c,d_0}^{cb}}{\sigma_0^2}), \tag{5}$$

where $B_{d_0}^{cb}$, p_{c,d_0}^{cb} and g_{c,d_0}^{cb} are bandwidth, transmission power and the channel gain on sub-carrier d_0 in the user-TST link, separately. Further, in (5), σ_0^2 is the noise power in this link. Hence, the transmission delay from user c to TST b is

$$t_c^{cb} = \frac{m_c}{\sum_{d_0=1}^{D_0} x_{d_0}^{cb} r_{c,d_0}^{cb}}, \tag{6}$$

where $x_{d_0}^{cb} \in 0, 1$ is the allocation indicator of user-TST over the C-band. In the case of a sub-carrier d_0 in C-band is allocated to user c to offload the tasks, $x_{d_0}^{cb} = 1$; otherwise, $x_{d_0}^{cb} = 0$. Therefore, the transmission energy is

$$E_c^{cb} = t_c^{cb} \sum_{d_0=1}^{D_0} x_{d_0}^{cb} p_{c,d_0}^{cb}. \tag{7}$$

If user c chooses to access satellite a directly, due to the ultra-long propagation distance, the propagation delay is not negligible and the round-trip propagation delay is

$$t_c^{proa} = \frac{2h}{c_l}, \tag{8}$$

where h is the distance between user c and satellite a, c_l is the speed of light. We assume the users in the same TST, this TST and terrestrial cloud have the same distance to the satellite a. Moreover, path loss should be considered when transmitting over long distances. We are not concentrating on the path loss in the user-TST link because they communicate in a small cell range and haven't got a significant impact on the transmission delay. The transmission rate from the user c to satellite a thus can be denoted by

$$R_c^{ca} = B_c^{ca} \log_2(1 + \frac{p_c^{ca} g_c^{ca}}{\sigma_0^2 PL_c^{ca}}), \tag{9}$$

where B_c^{ca}, p_c^{ca} and g_c^{ca} are bandwidth, transmission power, and channel gain from the user c to satellite a, respectively. Furthermore, PL_c^{ca} is the path loss. Note that the path loss affects the channel hence the channel gain. Nevertheless, to better demonstrate the advantages of SemCom, similar to [21], we present the path loss separately in the formula to facilitate subsequent analysis. Normally, the path loss PL for the satellite channels mainly consists of free-space path loss PL_f and atmospheric (rainfall) loss PL_r [21]. Hence, we assume the total path loss $PL = PL_f + PL_r$. We will specify these losses later. We then have the transmission delay and energy consumption when user c accesses the SEC a directly, which are given by

$$t_c^{ca} = \frac{m_c}{R_c^{ca}}, \tag{10}$$

$$E_c^{ca} = t_c^{ca} p_c^{ca}. \tag{11}$$

In contrast to users, the transmission process from TST b to satellite a integrates SemCom. It thus increases the computing delay while significantly decreasing the data required to be transmitted. The transmission rate of TST can be expressed as:

$$R_b^{ba} = B_b^{ba} \log_2(1 + \frac{p_b^{ba} g_b^{ba}}{\sigma_0^2 PL_b^{ba}}), \tag{12}$$

where B_b^{ba}, PL_b^{ba}, P_b^{ba} and g_b^{ba} are bandwidth, path loss, transmission power and the channel gain in TST b-satellite a link, respectively. In addition, since antennas of TSTs have good directivity, they can communicate with multiple LEO satellites via Ka-band and the corresponding interference can be ignored [10], [22], [23]. Therefore, the transmission delay of all users' tasks are transmitted from TST b to satellite a is

$$t_c^{ba} = \frac{\sum_{j=1}^{F} \psi m_j}{R_b^{ba}} + \frac{\sum_{j=1}^{F} m_j}{R_{SemCom}^{ba}}, \tag{13}$$

where F is the number of users allocated to offloading the task to satellite a and $F \in \mathcal{C}$. Furthermore, ψ is the compression ratio and the R^{ba}_{SemCom} is the rate of semantic extraction and semantic parsing, i.e., computing delay during data transmission.

Since the computation task calculation result is often much smaller than the offloaded data, it is reasonable to ignore the backhaul transmission delay (see also [24] and [25]. Moreover, estimating the number of subcarriers provided by satellite a to user c is difficult due to the large number of satellite service users. We assume that the satellite transmits user data to the ground cloud with a constant transmission rate R^a_c similar to [11]. The transmission delay between satellite and cloud t^{Cloud}_a thus equals m_c/R^a_c. Due to the mobility of satellites, the distance from the satellite to the terrestrial cloud is difficult to precisely inform users, we thus use h to estimate the distance between the satellite and the terrestrial cloud. The propagation delay where user c chooses to offload to the terrestrial cloud is

$$t^{proC}_c = 2t^{proa}_c = \frac{4h}{c_l}. \tag{14}$$

D. Path loss model

As mentioned in Section II-C, the path loss for the terrestrial-satellite channel is mainly free-space path loss PL_f and atmospheric (rainfall) loss PL_r. Free-space path loss is a basic power loss that increases depending on the communication distance. In dB, PL_f is [26]

$$PL_f(\text{dB}) = 92.44 + 20\log(h) + 20\log(f), \tag{15}$$

where h is the communication distance unit in km, and f is the operating frequency with the unit of GHz.

Atmospheric loss is a type of signal absorption and scattering due to meteorological causes, i.e., mainly related to rainfall. The rain attenuation is described by [27]

$$PL_r(dB) = \xi L_E, \tag{16}$$

where ξ is the frequency-dependent parameter unit in dB/km and L_E is the effective path length unit in km. We first introduce the calculation method of ξ as:

$$\xi = k(R_{0.001})^v, \tag{17}$$

where $R_{0.001}$ is the rainfall rate, unit in mm/h. Further, k and v are coefficients given as:

$$k = [k_H + k_V + (k_H - k_V)cos^2(\omega)cos(2\tau)]/2, \tag{18}$$

$$v = [k_H v_H + k_V v_V + (k_H v_H - k_V v_V)cos^2(\omega)cos(2\tau)]/2, \tag{19}$$

where $\tau = \pi/4$ for circular polarization and ω is the elevation angle between terrestrial transmitter and satellite. Moreover, k_H, k_V, v_H, and v_V are coefficients related to operating frequency f and can be found out the specific value from [28].

L_E, is therefore

$$L_E = L_R v_{0.001}, \tag{20}$$

where L_R is the distance parameter related to rainfall height and $v_{0.001}$ is the adjustment factor. We have

$$v_{0.001} = \frac{1}{1 + \sqrt{sin(\omega)}(\frac{31(1-e^{-(\frac{\omega}{1+\chi})})\sqrt{LR^{\xi}}}{f^2} - 0.45)}, \tag{21}$$

where χ equals 36-—latitude— in the case of latitude less than 36^o, or equals 0. In most scenarios

$$L_R = \frac{h_R - h_s}{sin(\omega)} \tag{22}$$

where h_R is the rain height relative to the mean sea level and h_s is the altitude of the terrestrial transmitter, all units in km.

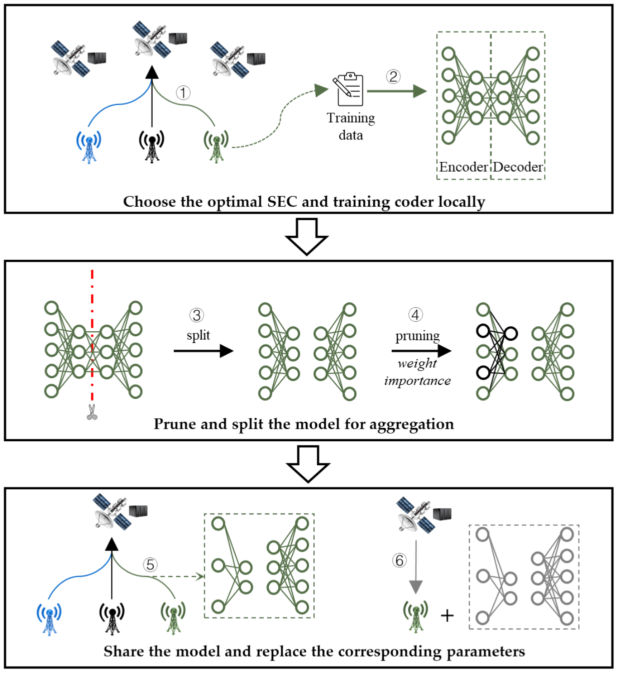

①
Training data
②
Encoder
Decoder
Choose the optimal SEC and training coder locally
③
split
④
pruning
weight importance
Prune and split the model for aggregation
⑤
⑥
+
Share the model and replace the corresponding parameters

Fig. 2: The schematic of the proposed PSFed in one communication round. The workflow contains the following 6 steps: ① TSTs choose optimal SEC for federated aggregation jointly; ② local training on private data; ③ the TST's coder model is split into the encoder and decoder part; ④ the TSTs prune the encoder model according to parameter importance; ⑤ each TST uploads the model for federated aggregation; ⑥ the TSTs download the personalised models and replace the corresponding parameters.

E. Semantic coder training model

In general distributed learning frameworks based on FedAvg [29], the training process requires multiple distributed participants and a federated aggregation node. Participants train their ML models locally and upload them to the federated aggregation node at fixed communication rounds. The federated aggregation node aggregates all the models and then returns the aggregated model to the participants for further training. This enables participants to update the model without sharing private training data. The goal of FL is to collaboratively train a global coder model among multiple TSTs while keeping TSTs' local data private. We set the $X_b = \{x^b_{in}\}^{s_b}_{b=1}$ as the data set of the TST b, where x_{in}^b is the in-th input sample and s_b is the size of the data set. The objective of FedAvg can be denoted by

$$\min_{\Theta} \frac{1}{B} \sum_{b=1}^{B} L_b(\theta_b), \tag{23}$$

where θ_b is the coder model parameter of the TST b and $\Theta = \theta_1, \theta_2, ..., \theta_b$. Further, $L_b(\theta_b)$ is the loss function of the TST b trained by X_b. We utilise the mean squared error (MSE) loss as the loss function in this paper. We have

$$L_b(\theta_b) = \frac{1}{s_b} \sum_{in=1}^{s_b} L_{MSE}(\theta_b; x_{in}^b, \widehat{x_{in}^b}), \tag{24}$$

where $\widehat{x_{in}^b}$ is the fitting output and L_{MSE} is the MSE loss.

III. Updating the Semantic Coders

Employing general FL frameworks for SemComs, TSTs need to upload encoder and decoder models to the SEC to implement federated aggregation after one communication round of training. Therefore, the federated model must be sent back to TSTs for the next communication round of training. However, uploading and downloading all coder models by TSTs would cause long-term interruptions of the offloading-assisted service, significant energy consumption, and lead to privacy leakage of entire coder models. Previous studies, e.g. [30] show that when reconstructing an ML model, increasing the number of parameters increases the accuracy of the model following a logarithmic function. In SemCom, the accuracy of the SemCom coder represents the accuracy of the received data. Therefore, the privacy of the coder model/parameter is closely tied to the accuracy. We can adopt a general parameter privacy leakage metric as in [31] and assess model parameter leakage by

$$\Theta_b(\theta_b) = \chi \log_2(1 + e^{1 - \frac{N_b + 1}{n_b}}), \tag{25}$$

where χ is the weight parameter, N_b is the total number of parameters at the encoder model and n_b is the number of parameters transmitted. In practice, Θ_b adopts a value in [0,1], where $\Theta_b = 0$ indicates that there is no privacy leakage, while a $\Theta_b = 1$ indicates fully compromised privacy where the same information can be decoded from the leaked model as the original model.

By increasing the number of training epochs the parameters of the training model become closer to the final trained model. Therefore, the model obtained from more training epochs is more important relative to the model obtained from previous training epochs before training is finished. In other words, the private information contained in the parameters is increased over time. More important parameters bear higher sensitivity in terms of privacy. Therefore, we rewrite the privacy leakage for TST b's encoder training as:

$$\Theta_b(\theta_b) = \sum_{r=1}^{R} W_r \chi \log_2(1 + e^{1 - \frac{\sum_i^{N_b} I_i n_{b,i} + 1}{\sum_i^{N_b} I_i n_{b,i}}}), \tag{26}$$

where r is the communication rounds and R is the total rounds. Also, W_r is the model importance weight of training round r. Similarly, I_i is a weight parameter denoting the importance transmitted parameter i.

In the proposed PSFed (Fig. 2), the goal is to collaboratively train semantic coder models among multiple TSTs while reducing network service interruptions, and energy consumption, and decreasing the degree of privacy leakage. Due to the high mobility of satellites, we note that all TSTs are not always within the same satellite service area. TSTs are therefore required to select the most appropriate satellite for each model aggregation round from the multiple satellites based on real-time circumstances. Taking into account TSTs' training delay and energy consumption jointly, the satellite selection algorithm is denoted by

$$\min_{x_a} \sum_{a=1}^{A} x_a(\alpha \max\{\frac{M_{b,r}}{R_b^{ba}} + \frac{2h^{ba}}{c_l}|b \in \mathcal{B}\} + \sum_{b=1}^{B} \beta p_b^{ba} \frac{M_{b,r}}{R_b^{ba}}), \tag{27a}$$

$$s.t. \quad \sum_{a=1}^{A} x_a = 1, \forall b \tag{27b}$$

$$x_a = \{0, 1\}, \tag{27c}$$

$$\sum_{r=1}^{R} \frac{M_{b,r}}{R_b^{ba}} \leq t_b^{'}, \forall b \tag{27d}$$

$$\max\{\frac{M_{b,r}}{R_b^{ba}} + \frac{2h^{ba}}{c_l}|b \in \mathcal{B}\} < t_a^{'}, \forall a \tag{27e}$$

where $\max\{\frac{M_{b,r}}{R_b^{ba}} + \frac{2h^{ba}}{c_l}|b \in \mathcal{B}\}$ is the training transmission and propagation delay, identified by the TST with the longest transmission and propagation time. Here, A is the number of accessible satellites of all TSTs, and h^{ba} is the distance between TST b and satellite a. Further, $\sum_{b=1}^{B} \beta p_b^{ba} \frac{M_{b,r}}{R_b^{ba}}$ is the total energy consumption of transmission from TSTs to a satellite. In (27a), α and β are weight parameters to balance the importance and unit of latency and energy consumption. Furthermore, p_b^{ba} is the transmission power of TST b to satellite a, and x_a is the federated decision for all TSTs. Constraint (27d) ensures that the transmission time of the TST for training the semantic model remains less than the maximum tolerable service interruption time. Also, $M_{b,r}$ is the coder model size in communication round r, $t_b^{'}$ is the maximum tolerable service interruption time and $t_a^{'}$ is the maximum service time of the satellite a in this region. The optimization problem in (27) is a simple 0,1 linear programming and hence can be easily solved.

During training in each communication round, we split the coder model into an encoder and a decoder. Only the decoder model needs entire federated aggregation. This is due to LEO satellites having limited storage capacity, it is not practical to use individual decoder models for each task of each TST. The semantic coders are therefore stored on the TST. For economic considerations, we argue that TSTs require a shared decoder model to be used. We then encourage TSTs to assess the importance of the encoder parameters during the local training phase. Inspired by continual learning [32], changes in parameters with different importance have a different impact

Algorithm 1 PSFed

```
Input: dataset $\{X_1, X_2, ..., X_b\}$, model size $\{M_1,M_2,...,M_b\}$ and total communication rounds $R$
Output: trained coder models $\{\theta_1, \theta_2, ..., \theta_b\}$
Initialize: the TSTs' model parameters and the importance weight of parameters SECs:
for each communication round $r \in R$ :
  $Y_b^{r+1}, \theta_b^{r+1} \longleftarrow TST\ update(\theta_b^r)$
  Update $\{\theta_{b,1}, \theta_{b,2}, ..., \theta_{b,N_b}\}$ according to $Y_b^{r+1}$ and $\theta_b^{r+1}$
end for
TSTs:
TST $b$ receives $\theta_b$ from the SEC
TSTs choose the optimal SEC for federated aggregation
for  each TST in parallel:
  for  each local training epoch:
    Loss $\longleftarrow \frac{1}{s_b} \sum_{in=1}^{s_b} L_{MSE}(\theta_b; x_{in}^b, \widehat{x_{in}^b})$
  end for
  foreach encoder parameter $i$:
    $I_i = -\frac{\partial L_b}{\partial \theta_{b,i}} \delta_{b,i}$
  end for
  Splitting coder model and pruning encoder model based on $I_i$ in the case of satisfying:
  $\begin{cases} \sum_{r=1}^{R} \frac{M_{b,r}}{R_b^{ba}} \leq t_b^{'} \\ \Theta_b(\theta_b^r) \leq \Theta_b^{'} \end{cases}$
  Obtain $\theta_b^r$ to be shared
  return: $\theta_b^r$
end for
```

on the output results. We thus evaluate parameter importance according to the implications of parameter changes on the loss function. We express the change in the loss by

$$L_b(\theta_b + \delta) - L_b(\theta_b) \approx \sum_{i=1}^{s_b} g_{b,i}\delta_{b,i}, \tag{28}$$

where g_i is the gradient and δ_i is the update of parameter i during this parameter assessment period of the TST b. Setting $g_i = \frac{\partial L_b}{\partial \theta_{b,i}}$ during online training, the parameter importance weight is

$$I_i = -\frac{\partial L_b}{\partial \theta_{b,i}} \delta_{b,i}. \tag{29}$$

Subsequently, to reduce the training communication cost, we prune the encoder models uploaded by TSTs according to parameter importance. Parameters with high importance contain most of the valid information [33] and therefore can provide further valid information to the aggregated model than lower-important parameters. The lower-importance parameters are thus encouraged to be pruned. The pruning here differs from the conventional ML studies. It is not the deletion of the training model parameters, but the non-transmission of the pruned parameters for federated aggregation. The corresponding SEC generates a global encoder model and a global decoder model based on the federated aggregation of the number of the received parameters. Once TST receives the global decoder model and personalised pruned global encoder model, it merely substitutes the local decoder and substitutes important parameters of the local encoder. It trains the individual local coder again based on the personal encoder model and the global decoder model in the next communication round of training.

Furthermore, the closer to the completion of the training, the higher the importance of the parameters. To further reduce the privacy leakage degree, our proposed PSFed progressively increases the pruning ratio according to the number of communication rounds. This is until the coder model is split and only the decoder model is federated aggregated. The more important privacy training models are thus kept local.

The objective of PSFed during training is to minimise the training loss, therefore,

$$\min_{\Theta, Y} \quad \sum_{b=1}^{B} L_b(y_b^1\theta_{b,1}, y_b^2\theta_{b,2}, ..., y_b^n\theta_{b,N_b}), \tag{30a}$$

$$s.t. \quad \sum_{r=1}^{R} \frac{M_{b,r}}{R_b^{ba}} \leq t_b^{'}, \forall b \tag{30b}$$

$$\max\{\frac{M_{b,r}}{R_b^{ba}} + \frac{2h^{ba}}{c_l} | b \in \mathcal{B}\} < t_a^{'}, \forall a \tag{30c}$$

$$\sum_{r=1}^{R} W_r \chi \log_2(1 + e^{1 - \frac{\sum_i^{N_b} I_i n_{b,i} + 1}{\sum_i^{N_b} I_i n_{b,i}}}) \leq \Theta_b^{'}, \forall b \tag{30d}$$

where $y_b^n \in [0, 1]$ is the aggregation weight vector of parameter i in TST b. It acts similar to the weighted average in FedAvg. Since each TST uploads a different number and location of parameters in the same model, the proportion of each parameter that is weighted is different. The y_b^n for various parameters also different and $Y = y_1, y_2, ..., y_b$. Further, $\Theta_b^{'}$ is privacy leakage consideration and $\Theta_b^{'}$ is the maximum tolerable leakage The procedure of the PSFed is demonstrated in Algorithm 1.

IV. The Semantic Coders in Service

In this section, the problem of users' computational task processing schedule for SemCom-SEC is presented first. We then detail the proposed CTPS.

A. Computational task processing

In service offloading decision-making, we consider the SemCom-SEC with C users severed by one TST b in A satellite coverage. Each user has five task processing choices, 1) local computing; 2) offloading the tasks to SEC directly; 3) offloading the tasks to SEC via the TST; 4) offloading the tasks to the terrestrial cloud only via the satellite; 5) offloading the tasks to the terrestrial cloud via the TST and the satellite. We firstly list the user c's cost functions in terms of processing delay and energy consumption for each option in order as follows based on Section II:

$$\Phi_{c1} = \alpha t_c^{LC} + \beta E_c^{LC}, \tag{31}$$

$$\Phi_{c2} = \alpha(t_c^{proa} + t_c^{ca} + t_c^{SEC}) + \beta E_c^{ca}, \tag{32}$$

$$\Phi_{c3} = \alpha(t_c^{proa} + t_c^{cb} + t_c^{ba} + t_c^{SEC}) + \beta E_c^{cb}, \tag{33}$$

$$\Phi_{c4} = \alpha(t_c^{proa} + t_c^{ca} + t_c^{Cloud} + t_a^{Cloud}) + \beta E_c^{ca}, \tag{34}$$

$$\Phi_{c5} = \alpha(t_c^{proC} + t_c^{cb} + t_c^{ba} + t_c^{Cloud} + t_a^{Cloud}) + \beta E_c^{cb}, \tag{35}$$

where Φ_c is the actual processing cost when the user c sizing a task. It is related to user task processing decisions, the transmission power, and the number of subcarriers allocated. In the above, t_a^{Cloud} is the transmission delay between satellite and cloud as mentioned in Section II-C. We also utilise $\gamma_{ic} = \{0, 1\}$ to represent the offloading decision of user c and $\gamma_{ic} \in \{\gamma_{1c}, \gamma_{2c}, \gamma_{3c}, \gamma_{4c}\}$. If user c chooses one processing strategy, the indicator for the corresponding strategy equals 1, otherwise equals 0. We argue that the optimal decision for a user is to minimise the latency and energy consumption of the processing tasks. Mathematically, the optimisation task processing strategy problem of user c thus can be formulated as a MINLP problem:

$$\begin{aligned}
\min_{\gamma_c, f_c, p^{cb}_{c,d_0}, m_{c,d_0}, p^{ca}_c} \sum_{a=1}^{A} \Phi_c &= (1 - \gamma_{1c} - \gamma_{2c} - \gamma_{3c} - \gamma_{4c})\Phi_{c1} \\
&+ \gamma_{1c}\Phi_{c2} + \gamma_{2c}\Phi_{c3} + \gamma_{3c}\Phi_{c4} + \gamma_{4c}\Phi_{c5}, && \text{(36a)} \\
s.t. \quad & f_{cloud} \geq f_a \geq f_{c,max} \geq 0, && \text{(36b)} \\
& \gamma_{1c}, \gamma_{2c}, \gamma_{3c}, \gamma_{4c} \in \{0, 1\}, && \text{(36c)} \\
& \gamma_{1c} + \gamma_{2c} + \gamma_{3c} + \gamma_{4c} \leq 1, && \text{(36d)} \\
& \sum_{d_0=1}^{D_0} x^{cb}_{d_0} p^{cb}_{c,d_0} \leq P_{c,max}, && \text{(36e)} \\
& P^{ca}_c \leq P_{c,max}, && \text{(36f)} \\
& x^{cb}_{d_0} \in \{0, 1\}, && \text{(36g)} \\
& \sum_{d_0=1}^{D_0} x^{cb}_{d_0} \leq D_0, && \text{(36h)} \\
& t^* < t_a^{'}. && \text{(36i)}
\end{aligned}$$

The constraint (36b) guarantees that edge and cloud have strong computing capability that is not less than users' maximum computing capability $f_{c,max}$. Constraints (36c) and (36d) show the relationship between $\gamma_{1c}, \gamma_{2c}, \gamma_{3c}$ and γ_{4c}. In constraints (36e) and (36f), $P_{c,max}$ is the maximum available transmission power of user c to TSTs or satellites. The constraint (36g) denotes the subcarrier allocation indicator. The constraint (36h) means that the number of allocated subcarriers should not exceed the total number of sub-carriers. The constraint (36i) is to ensure the optimal decision's transmission time t^* is less than the time $t_a^{'}$ available to access satellite a.

The problem in (36) is an MINLP problem with incomplete information due to privacy concerns. This is because users need the allocation of subcarriers to make decisions. Nevertheless, such information is relevant to decisions and privacy information (e.g., local computing capability and transmission power) from other users. This MINLP problem thus is computationally complex and hard to solve.

B. CTPS

In this paper, we propose a CTPS mechanism (see, Fig. 3) to minimise the delay and energy consumption of users to process computational tasks, while privacy-preserving and equitable. We assume all the participants are trustworthy It is divided into two steps. Firstly, it converts the optimisation task processing strategy problem with privacy considerations into a complete information problem based on the Rubinstein bargaining model [34] equitably. Subsequently, users develop the optimisation task processing strategies by solving the complete information MINLP problem of Eq. (36). We detail our CTPS mechanism as follows.

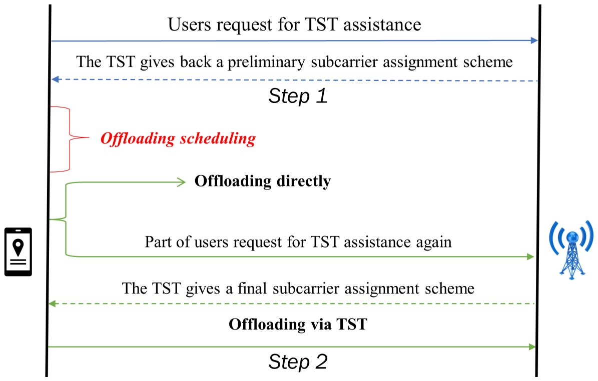

Users request for TST assistance
The TST gives back a preliminary subcarrier assignment scheme
Step 1
Offloading scheduling
Offloading directly
Part of users request for TST assistance again
The TST gives a final subcarrier assignment scheme
Offloading via TST
Step 2

Fig. 3: Proposed CTPS mechanism.

C. First step of the CTPS mechanism

We enable users to communicate/bargain with TST several times so that subcarriers are allocated fairly without privacy leakage based on the Rubinstein bargaining game. TST acts as the bidder and the user has the option to continue the game or leave the game. The gaming process is limited to two periods. In the first period, the users send the offloading request to the TST. Upon receiving users' offloading requests, without loss of generality and fairness, TST allocates the number of C-band sub-carriers based on the size of the tasks offloaded by users. Further, the transmission delay of the TST to the satellite and semantic extraction delay are also notified via this communication.

To achieve the game-perfect equilibrium, the cost function for user c to assess to continue participating in the game can be denoted by

$$\mu_c^{'} = \epsilon \iota \Phi_c^{'}, \;\; \Phi_c^{'} = \{\Phi_{c3}, \Phi_{c5}\}, \tag{37}$$

where $\iota \in (0, 1)$ is the bargaining discount factor that represents the revenue loss value for the second-period communication due to the bargaining process being time and energy-consuming. Further, $\epsilon \geq 1$ is the weight parameter to evaluate the further possible benefit by applying offloading again via the TST b, i.e., remaining engaged in the game. This is attributable to some users abandoning their requests for TST offloading due to not being allocated a satisfactory number of C-band subcarriers. The actual number of subscribers should eventually be greater than or equal to this allocation. Simultaneously, the strategies of various users also affect the user-satellite link interference for different users. In order to estimate the influence of interference, pricing is a frequently utilised method in the game theory employed studies [35]. We hence rewrite the part of the cost function for user c considering interference pricing as:

$$\mu_c^{''} = \Phi_c^{''} + \alpha\varrho m_c \varpi,\ \Phi_c^{''} = \{\Phi_{c2}, \Phi_{c4}\}, \tag{38}$$

where ϱ is the factor for the interference related to the number of users, transmission power, and channel gain. Further, $\varpi \in [0, 1]$ is the proportion to denote the anticipation rate of not performing local computing users, thus predicting the fraction of time in which interference is received.

Finally, the incomplete information MINLP problem is converted to a complete information MINLP problem. Users thus could develop the optimal processing decision based on allocated subcarriers and the calculation frequency or transmitting power in the second step.

D. Second step of the CTPS mechanism

In the second step, users make the decision based on the complete information MINLP problem of Eq. (36) to minimise the latency and energy consumption of the processing tasks. The maximum number of satellites expected to be accessible at the same time is extremely limited [23]. The decision problem Eq. (36) can be considered as $5 \cdot A$ independent subproblems, where 5 is five offloading decision subproblems and A is A satellite selection subproblems. In case of the local computing, the best user c's CPU-cycle frequency f_c is only related to local computing costs. We thus can express the f_c optimisation subproblem as:

$$\min_{f_c} \quad \Phi_{c1} = \alpha \frac{\delta m_c}{f_c} + \beta \varepsilon \delta m_c f_c^2, \tag{39a}$$

$$s.t. \quad (36b). \tag{39b}$$

We can express the first-order derivative of (39a) as: $-\alpha\frac{\delta m_c}{f_c^2} + 2\beta\varepsilon\delta m_c f_c$. Eq. (39a) monotonically increases in the constraint (39b), hence $f_c = f_{c,max}$.

In addition, in case the user needs to employ TSTs, the user needs to derive the optimal subcarrier task allocation strategy m_{c,d_0} and subcarrier transmission power p_{c,d_0}^{cb}. To model and optimise the transmission power, in CTPS, we assume each subcarrier in the same link accomplishes the transmission tasks at the same time for fully using spectrum resources in a synchronous manner based on previous studies [24], [36]. As the allocated subcarrier for user c is known, we set η to denote the number of allocated subcarriers. We can simplify the optimisation problem associated with TST as:

$$\min_{m_{c,d_0}, p_{c,d_0}^{cb}} \quad \sum_{d_0=1}^{D_0} (\frac{\alpha x_{d_0}^{cb} m_{c,d_0}}{\eta r_{c,d_0}^{cb}} + \frac{\beta p_{c,d_0}^{cb} x_{d_0}^{cb} m_{c,d_0}}{r_{c,d_0}^{cb}}), \tag{40a}$$

$$s.t. \quad (36e), (36g), (36h), \tag{40b}$$

$$\sum_{d_0=1}^{D_0} x_{d_0}^{cb} m_{c,d_0} = m_c. \tag{40c}$$

We only need to consider the situation that $x_{d_0}^{cb} = 1$. By relaxing constraints, we have the Lagrangian function for Eq. (40a) as:

$$\begin{aligned} L = &\sum_{d_0=1}^{D_0} x_{d_0}^{cb} (\frac{\alpha m_{c,d_0}}{\eta r_{c,d_0}^{cb}} + \frac{\beta p_{c,d_0}^{cb} m_{c,d_0}}{r_{c,d_0}^{cb}}) \\ &+ \varphi(\sum_{d_0=1}^{D_0} x_{d_0}^{cb} p_{c,d_0}^{cb} - P_{c,max}) + \lambda(m_c - \sum_{d_0=1}^{D_0} x_{d_0}^{cb} m_{c,d_0}), \end{aligned} \tag{41}$$

where φ and λ are the Lagrangian multipliers. The dual problem thus is $\min_{m_{c,d_0}, p_{c,d_0}^{cb}} L$. Then, we can observe that Eq. (41) can be further decomposed into D_0 independent subproblems, and the actual objective function in each d_0 subproblem can be denoted by

$$\min_{m_{c,d_0}, p_{c,d_0}^{cb}} L_{d_0} = \frac{\alpha m_{c,d_0}}{\eta r_{c,d_0}^{cb}} + \frac{\beta p_{c,d_0}^{cb} m_{c,d_0}}{r_{c,d_0}^{cb}} + \varphi p_{c,d_0}^{cb} + \lambda m_{c,d_0}. \tag{42}$$

For simplicity, we define

$$H_{d_0} = \frac{\alpha}{\eta r_{c,d_0}^{cb}} + \frac{\beta p_{c,d_0}^{cb}}{r_{c,d_0}^{cb}}. \tag{43}$$

According to Karush-Kuhn-Tucker conditions, taking the partial derivatives of L_{d_0} with respect to p_{c,d_0}^{cb} and m_{c,d_0}, respectively. We have

$$\begin{cases} \frac{\partial L_{d_0}}{\partial p_{c,d_0}^{cb}} = m_{c,d_0} \frac{\partial H_{d_0}}{\partial p_{c,d_0}^{cb}} + \varphi = 0 & (44a) \\ \frac{\partial L_{d_0}}{\partial m_{c,d_0}} = H_{d_0} - \lambda = 0 & (44b) \\ \varphi(\sum_{d_0=1}^{D_0} x_{d_0}^{cb} p_{c,d_0}^{cb} - P_{c,max}) = 0. & (44c) \end{cases}$$

Thus, we have

$$\begin{cases} \varphi = 0, \sum_{d_0=1}^{D_0} x_{d_0}^{cb} p_{c,d_0}^{cb} \leq P_{c,max}, & (45a) \\ \varphi > 0, \sum_{d_0=1}^{D_0} x_{d_0}^{cb} p_{c,d_0}^{cb} = P_{c,max}, & (45b) \end{cases}$$

where (45) is complementary slackness. For (45a), p_{c,d_0}^{cb} can be directly solved by (44) causing $m_{c,d_0} \neq 0$. After deriving the optimal p_{c,d_0}^{cb}, m_{c,d_0} can be easily solved as all subcarriers have the same subcarrier completion time. Only if the solution $\sum_{d_0=1}^{D_0} p_{c,d_0}^{cb} = P_{c,max}$, we need to consider Eq. (45b). In that case, the Lagrangian multipliers can be obtained by the subgradient method and further achieve the optimal p_{c,d_0}^{cb}, m_{c,d_0}. Moreover, as we utilise the Lagrangian dual decomposition method, the solution may have a duality gap. However, this gap should approach zero and can be ignored in practical systems as the number of subcarriers D_0 is large enough [10].

Algorithm 2 CTPS

```
Input: Tasks $m_c$ generation
Output: The computation offloading and resource allocation result $\gamma_c, f_C, p^{cb}_{c,d_0}, m_{c,d_0}, x^{cb}_{d_0}$
Initialize the optimal TST transmission power $p^{ba}_b$
Obtain necessary information $x^{cb}_{d_0}$ after first period game
Obtain the necessary information $x^{cb}_{d_0}$ after first period game
Calculate optimally $f_c$
Relax Eq. (40)
if $\varphi = 0$:
    $p^{cb}_{c,d_0} \longleftarrow \frac{\partial H_{d_0}}{\partial p^{cb}_{c,d_0}}$
    $m_{c,d_0} \longleftarrow \frac{m_c p^{cb}_{c,d_0}}{\sum_{d_0=1}^{D_0} x^{cb}_{d_0} p^{cb}_{c,d_0}}$
else:
    $p^{cb}_{c,d_0} \longleftarrow$ Eq. (44)
    $m_{c,d_0} \longleftarrow \frac{m_c p^{cb}_{c,d_0}}{P_{c,max}}$
end if
Find the maximum $\Phi_c$ and derive $\gamma_c$
if $\gamma_{c3} + \gamma_{c5} = 1$:
    Obtain the necessary information $x^{cb}_{d_0}$ after the second period game
    Obtain updated $p^{cb}_{c,d_0}$ and $m^{cb}_{c,d_0}$
end if
Find the maximum $\Phi_c$ and derive $\gamma_c$
```

TABLE I: The setting of the CAE

Encoder	Neuron num	Decoder	Neuron num
Conv+ReLU	512	transConv+ReLU	10
Conv+ReLU	256	transConv+ReLU	32
Conv+ReLU	128	transConv+ReLU	64
Conv+ReLU	64	transConv+ReLU	128
Conv+ReLU	32	transConv+ReLU	256
Conv+Sigmod	10	transConv+Sigmod	512

TABLE II: Rainfall coefficients

C-band	Value	Ka-band	Value
k_H	0.0001340	k_H	0.2403
k_V	0.0002347	k_V	0.2291
v_H	1.6948	v_H	0.9485
v_V	1.3987	v_V	0.9129

Therefore, users can make the decision based on the computation cost of various alternatives, without compromising privacy. Throughout the CTPS, the user is only communicated externally about the size of the tasks being processed. It also needs to be known by TST during the offloading process. Hence the CTPS protect the privacy of computing power, transmit power, etc. Further, the computational complexity is linearly related to D_0 and A, whereas both D_0 and A are finite. CTPS thus can be used in large-scale satellite networks. The CTPS and offloading decision process is summarised as Algorithm 2.

TABLE III: Simulation parameters

Parameters	Default values
The coverage radius of LEO satellites	280 km
Ka-band carrier frequency	30 GHZ
C-band carrier frequency	4.5GHZ
Number of C-band subcarriers	128
The maximum transmit power of each user	23dBm
Transmit power of TST	30 dBm
h	780km
δ	120
ε	10^{-26}
f_c	0.5×10^9 cycles/s
f_a	3×10^9 cycles/s
f_{Cloud}	10×10^9 cycles/s
α, β	0.5
ι, ϵ	1

V. Simulation Results

A. Simulation setting

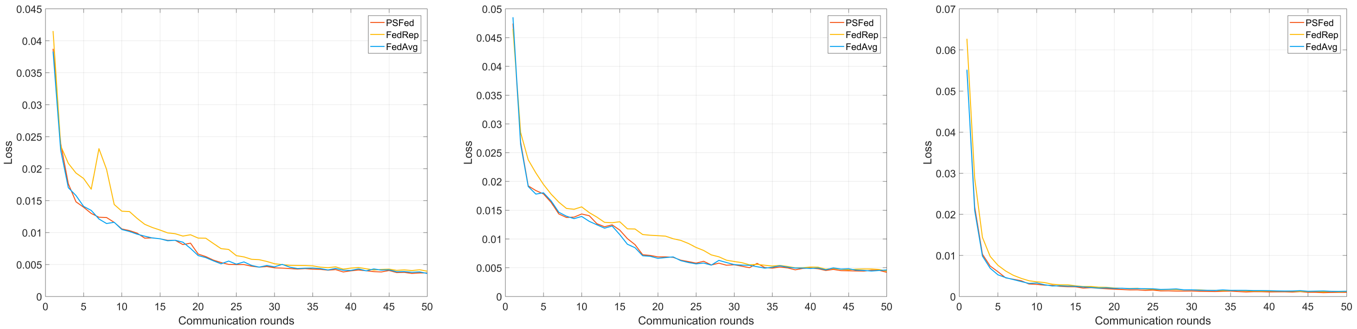
In this section, we evaluate the performance of the present PSFed and CTPS. In the simulations, if not specifically mentioned, we set the parameters as follows. The LEO satellites' coverage radius is 280 km and the vertical altitude is 780km based on the Iridium satellite system [37]. The frequencies of the C-band and the Ka-band are 4.5 GHz and 30 GHz separately based on 3GPP specifications [38]. We assume the number of C-band subcarriers is 128, the maximum transmission power of users is 23 dBm and the transmit power of each TST is 30 dBm [10]. The offloading task is assumed an image recognition task and the semantic coder is considered an autoencoder based on the convolutional autoencoder (CAE) similar to [39].

Communication rounds for the proposed PSFed to aggregate the semantic encoder are 20 rounds. The coder settings are listed in Table I. Furthermore, we set the number of CPU cycles for computing one bit δ as 120 cycles/bit, which is from the real applications [19]. We assume all users have the same CPU frequency f_c, and set it as 0.5×10^9 cycles/s. The computation capabilities of SEC on satellite a and the cloud server are 3×10^9 cycles/s and 10×10^9 cycles/s, respectively [11]. The energy factor ε is set as 10^{-26} [10].

Moreover, we assume weight parameters of latency and energy consumption are set as $\alpha = 0.5$ and $\beta = 0.5$, and weight parameters in bargain process ι and ϵ are all considered as 1. In addition, the atmospheric loss is adopted, and the related coefficients are shown in Table II [28]. The simulation parameters are also listed in Table III.

B. Performance evaluation of PSFed

Fig. 4 illustrates the convergence speed of the different frameworks under different transmission tasks. The TSTs' images are from CIFAR 10 [40], CIFAR 100 [41] and MNIST [42] image datasets and TSTs perform federated aggregation after every five local epochs. Based on the feasibility in SEC networks, we compare the proposed PSFed with the generalised learning approach for SemCom [15], [16], i.e., FL frameworks based on the FedAvg [29].



(a) CIFAR 10 dataset

(b) CIFAR 100 dataset

(c) MNIST dataset

Fig. 4: Convergence speed of various learning algorithms with different datasets.

Based on the existing FL methods that are potentially for SEC SemCom, FedRep [43] is also compared to demonstrate the effectiveness of our PSFed. The FedRep is based on the Fedavg but only aggregates part of the training model during each communication round. We set it to only aggregate SemCom decoder to adapt the SemCom-SEC. Moreover, we set the training sample to 5000 images per TST to reflect the differences between the frameworks more effectively. It can be observed that our PSFed achieves similar convergence rates to the FedAvg and is much better than the FedRep, regardless of the dataset. This is because our method aggregates important weights in the early stages of training and therefore accelerates convergence similarly to the FedAvg with all parameters aggregated.

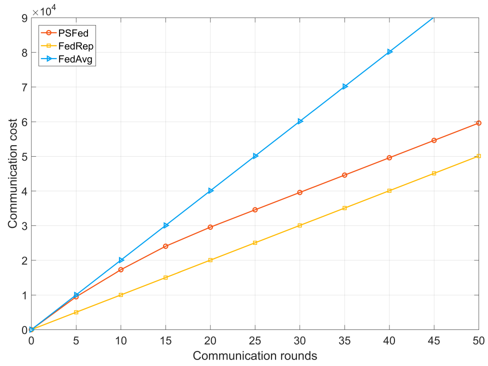


Fig. 5: Communication cost of various learning approaches.

In Fig. 5, we compare the total communication cost of PSFed, FedRep and FedAvg during training. We assume that each neuron transmitted consumes the same amount of communication resources. The communication cost is therefore defined as the number of neurons transmitted during communication. It is seen that the PSFed expenses are approximately the same communication cost as the FedAvg in the early stages of training. The growth then gradually slows down and increases at the same magnitude as the FedAvg after round 20. This is because the PSFed gradually decreases the number of weights aggregated by the encoder model.

It is also seen that in round 20, the number of aggregated weights for the encoder model is 0, the same as the FedRep, only the decoder model is aggregated. Therefore, the PSFed only consumes additional communication resources for the importance weight aggregation than the FedRep. Considering that the FedRep converges much more slowly than the proposed PSFed, the total communication resource consumption can be considered to be similar. However, in comparison to the FedAvg, the communication consumption of our PSFed decreases by 40.50% in round 50.

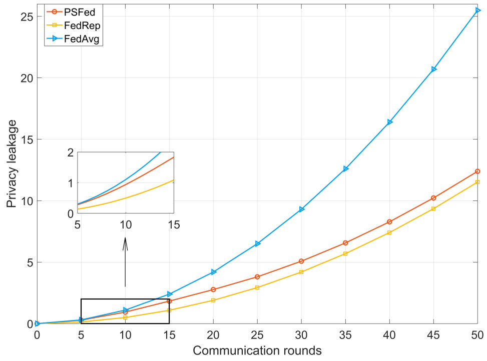


Fig. 6: Privacy leakage of various learning approaches.

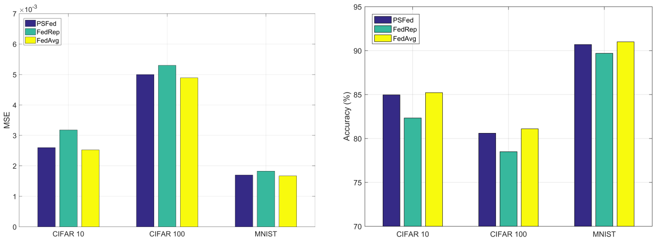
C. Performance evaluation CTPS

We evaluate the total model privacy leakage during training in Fig. 6 according to Eq. (26). We assume that the model in each communication round has the same importance and that each neuron is of equal importance. It can be observed that PSFed is initially similar to FedAvg leakage and subsequently follows the same growth trend as FedRep. This is equally due to the number of PSFed decreasing importance weight aggregations. After training, both the PSFed and the FedRep encoder models are saved locally. It is foreseeable that if the importance of each round of communication changes, the PSFed would be extremely close to the FedRep in terms of total privacy leakage. In addition, the privacy leakage of PSFed should widen the gap with FedAvg, even though the privacy leakage of our PSFed already decreases by 51.43% in round 50 in comparison to FedAvg in the same importance.

In Fig. 7(a), the accuracy of the different frameworks under different transmission tasks is shown. We evaluate the accuracy utilising Peak Signal-to-Noise Ratio (PSNR), a general metric for evaluating image transmission in SemCom [39]. We have

$$PSNR = 10 \log \frac{MAX^2}{MSE} (dB), \tag{46}$$

where MAX is the maximum value for a pixel and MSE is the mean squared deviation. Since different datasets have different MAX, we assume that the learning method with the smaller MSE has a higher accuracy. It is seen that the FedRep is significantly the least accurate with different datasets trained. The accuracy of PSFed is similar to FedAvg but slightly FedAvg higher. Because encoder models of both PSFed and FedRep are kept at the TST that are not aggregated when training is completed. Some aggregation information thus is lacking. However, the average training accuracy of the PSfed decreased by only 0.33% relative to the FedAvg due to the important weight aggregation acting as pre-training. Compared to the FedAvg, the accuracy loss of the PSfed deems acceptable given the significant communication cost and privacy concerns of the former.



(a) MSE

(b) Recognise accuracy matches after coding

Fig. 7: Accuracy of various learning algorithms with different datasets.

Fig 7(b) further demonstrates the effect of image transmission accuracy on offloading via different approaches. We employed commonly used ML models for image recognition to identify the accuracy of images before/after transmission. The accuracy here is the proportion of the received object/image recognition accuracy to the pre-transmission image recognition accuracy. It can be seen that with the same trend as Fig. 7(b) FedRep has the significantly lowest accuracy while our method is similar to FedAvg but slightly lower. Figs. 5, 6, and 7 collectively suggest that PSFed achieves the fastest convergence rate, the lowest communication cost, and a high accuracy rate.

Fig. 8 illustrates the impact of users in one TST coverage on the total cost. As users are not always able to offload tasks via the TST, the proposed CTPS is compared with the local computing, offloading to the SEC directly, offloading to the cloud directly and CTPS without the game. The task size for each user is randomly generated over a range of 5 kb-300kb and subjected to 200 times replications of the simulation. Fig.8 shows that the total cost grows with the number of users. This is because raising the number of users increases the corresponding number of computing tasks and thus the total cost of users. The total cost of the proposed CTPS always keeps the total cost to the minimum and the advantage increases as the number of users increases. In addition, in cases where the number of users is small, the proposed CTPS thus maintains almost the same processing cost as ”CTPS without game”. By increasing the number of users, TST becomes unable to satisfy all the requests and CTPS starts to show its advantage in reducing the cost. We expect this advantage to increase by further increasing the number of users. This is because the optimal reallocation of resources through our design game scheme increases the efficiency of network resource utilisation.

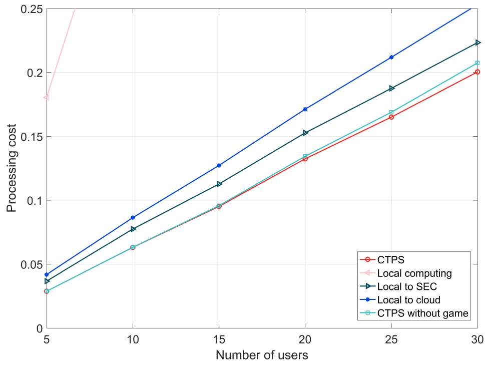


Fig. 8: The processing cost of the varying number of users.

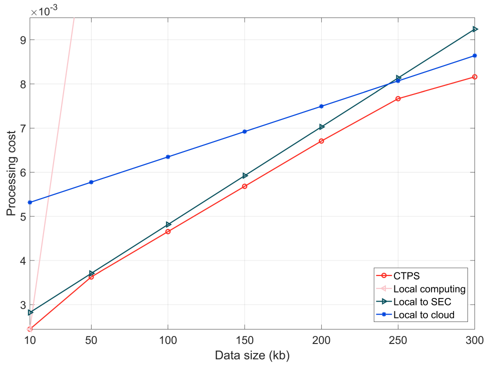


Fig. 9: The processing cost of a single user.

In Fig. 9, we show the offloading and computing cost of a single user versus the size of generating tasks. It is observed that the cost increases with the data size for all schemes. Our proposed mechanism always has a lower cost compared to the other three approaches. In case the data size is small (10 kb), our CPTS choose local computing as the optimal option. As the data size grows, the local computing latency and energy consumption increase, and CTPS chooses other minimum cost strategies, i.e., offload tasks to the SEC via the TST. After 250kb, the optimal value of our mechanism fluctuates. This is due to the data size being large enough, and the best strategy changes to offload tasks to the cloud via a TST. Therefore, the processing of the single-user tasks can be performed efficiently via our proposed processing strategy.

Fig. 10 demonstrates the importance of integrating SemCom into SEC networks in future communication environments. We set the user and the TST to maintain the same status to transmit to LEO satellites in different rainfall environments. It can be observed that as the rainfall probability increases, the task transmission cost of TST without SemCom exhibits a significant increase. Because the Ka-band frequency is extremely high and is strongly influenced by rainfall-induced path loss. In contrast, the processing costs for users transmitting via C-band are only slightly increasing. Since the C-band frequency is smaller than the Ka-band frequency and thus tolerates less path loss. Nevertheless, the TST configuration with the semantic encoder spends the least processing cost. Furthermore, the processing cost did not increase significantly with the increase in rainfall rate. This is because the latency of semantic extraction is not affected by the environment. The improved spectrum efficiency also reduces the impact of rainfall-induced path loss. Therefore, the integration of SemCom in SEC networks is necessary.

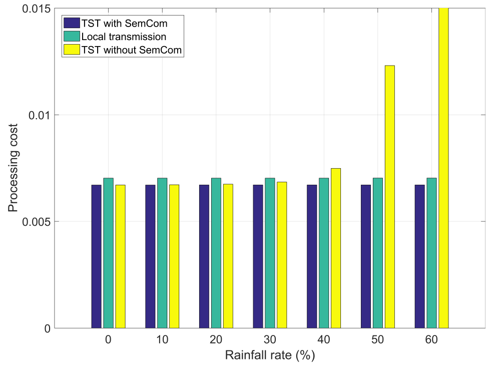


Fig. 10: The usefulness of SemCom in the network.

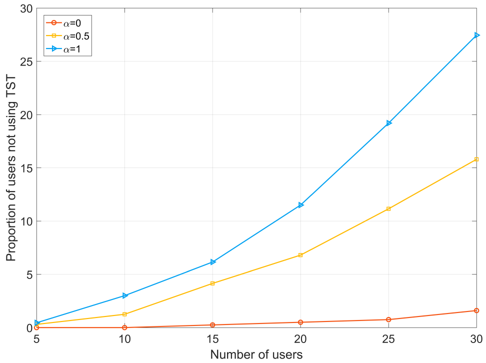


Fig. 11: Impact of α and β on strategy developing.

In Fig. 11, the influence of α and β on user strategies is investigated and the data size is from 5kb to 300kb simulated 50 times. The energy consumption weight β is always set as 0.5. We list the proportion of users that do not choose to offload via TST. It can be noticed that as the number of users increases, the unwillingness to offload increases due to the reduced number of subcarriers being allocated to them. However, users are always more reluctant to offload via TST in case the delay is more important (i.e., bigger α). These provide a criterion for the appropriate α and β to be chosen.

VI. Conclusion

In this paper, we investigated the integration of SemCom and SEC networks for terrestrial resource-limited users' computation offloading. We further proposed a novel SemCom-SEC framework for computation offloading. In addition, we examined the challenges that SemCom confronts in the proposed framework. For analysis, we then considered the challenges in two different scenarios. For the in-maintenance SemCom service, we proposed PSFed for the semantic coder update challenge. In the in-service SemCom service, we presented a game theoretical CTPS mechanism for task processing decision challenges of users. Compared with the general learning approach for semantic coder updating in SEC networks, simulation studies indicate that, on average, the proposed PSFed saves 40.50% of communication resources and further reduces privacy risk by 51.43%. Nevertheless, the training accuracy and convergence speed of PSFed and the general learning approach almost remain the same.

References

[1] G. Zheng, Q. Ni, K. Navaie, H. Pervaiz and C. Zarakovitis, "Efficient Pruning-Split LSTM Machine Learning Algorithm for Terrestrial-Satellite Edge Network," *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, Seoul, Korea, Republic of, 2022, pp. 307-311.

[2] P. Rahimi, C. Chrysostomou, H. Pervaiz, V. Vassiliou and Q. Ni, "Joint Radio Resource Allocation and Beamforming Optimization for Industrial Internet of Things in Software-Defined Networking-Based Virtual Fog-Radio Access Network 5G-and-Beyond Wireless Environments," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4198-4209, June 2022.

[3] Y. Xiao, G. Shi, Y. Li, W. Saad and H. V. Poor, "Toward Self-Learning Edge Intelligence in 6G," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 34-40, December 2020.

[4] Z. Zhang, W. Zhang and F. 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, Jan/Feb 2019.

[5] E. C. Strinati, S. Barbarossa, T. Choi, A. Pietrabissa, A. Giuseppi, E. De Santis, J. Vidal, Z. Becvar, T. Haustein, N. Cassiau, F. Costanzo, J. Kim, and I. Kim, "6G in the sky: On-demand intelligence at the edge of 3D networks (Invited paper)," *ETRI J.*, vol. 42, no. 5, pp. 643–657, Oct. 2020.

[6] Y. Wang, J. Yang, X. Guo and Z. Qu, "A game-theoretic approach to computation offloading in satellite edge computing", *IEEE Access*, vol. 8, pp. 12510-12520, 2019.

[7] 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 Netw.*, vol. 33, no. 1, pp. 70-76, Jan./Feb. 2019..

[8] B. Di, L. Song, Y. Li and H. V. Poor, "Ultra-Dense LEO: Integration of Satellite Access Networks into 5G and Beyond," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 62-69, April 2019.

[9] Y. Wang, J. Zhang, X. Zhang, P. Wang and L. Liu, "A computation offloading strategy in satellite terrestrial networks with double edge computing", *Proc. IEEE Conf. Commun. Syst.*, pp. 450-455, Dec. 2018.

[10] Z. Song, Y. Hao, Y. Liu and X. Sun, "Energy-Efficient Multiaccess Edge Computing for Terrestrial-Satellite Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14202-14218, 15 Sept.15, 2021.

[11] Q. Tang, Z. Fei, B. Li and Z. Han, "Computation Offloading in LEO Satellite Networks With Hybrid Cloud and Edge Computing," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9164-9176, 1 June1, 2021.

[12] Q. Lan et al., ”What is Semantic Communication? A View on Conveying Meaning in the Era of Machine Intelligence,” *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 336-371, Dec. 2021.

[13] Z. Qin, X. Tao, J. Lu, and G. Y. Li, “Semantic communications: Principles and challenges,” Jun. 2022, [online] Available: http://arXiv:2201.01389.

[14] H. Xie and Z. Qin, ”A Lite Distributed Semantic Communication System for Internet of Things,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142-153, Jan. 2021

[15] G. Shi, Y. Xiao, Y. Li and X. Xie, ”From Semantic Communication to Semantic-Aware Networking: Model, Architecture, and Open Problems,” *IEEE Communications Magazine*, vol. 59, no. 8, pp. 44-50, August 2021.

[16] Z. Qin, G. Y. Li and H. Ye, ”Federated Learning and Wireless Communications,” *IEEE Wireless Communications*, vol. 28, no. 5, pp. 134-140, October 2021.

[17] L. D. Earley, ”Communication in Challenging Environments: Application of LEO/MEO Satellite Constellation to Emerging Aviation Networks,” *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, 2021, pp. 1-8.

[18] G. Zheng, Q. Ni, K. Navaie, H. Pervaiz and C. Zarakovitis, ”A Distributed Learning Architecture for Semantic Communication in Autonomous Driving Networks for Task Offloading,” *IEEE Communications Magazine*, vol. 61, no. 11, pp. 64-68, November 2023.

[19] A. P. Miettinen and J. K. Nurminen, ”Energy efficiency of mobile clients in cloud computing”, *Proc. USENIX HotCloud*, pp. 4-11, Jun. 2010.

[20] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang and X. S. Shen, ”Software Defined Space-Air-Ground Integrated Vehicular Networks: Challenges and Solutions,” *IEEE Communications Magazine*, vol. 55, no. 7, pp. 101-109, July 2017.

[21] K. Tekbıyık, G. K. Kurt and H. Yanikomeroglu, ”Energy-Efficient RIS-Assisted Satellites for IoT Networks,” *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14891-14899, 15 Aug.15, 2022.

[22] J. Du, C. Jiang, H. Zhang, Y. Ren and M. Guizani, ”Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks”, *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2202-2217, Oct. 2018.

[23] R. Deng, B. Di, S. Chen, S. Sun and L. Song, ”Ultra-dense LEO satellite offloading for terrestrial networks: How much to pay the satellite operator?”, *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6240-6254, Oct. 2020.

[24] F. Wang, J. Xu, and Z. Ding, “Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems,” *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2450–2463, Mar. 2019.

[25] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, “NOMAassisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12244–12258, Dec. 2018.

[26] S. Fu, J. Gao, and L. Zhao, “Integrated resource management for terrestrial-satellite systems,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3256–3266, Mar. 2020.

[27] ITU-R, “Propagation data and prediction methods required for the design of Earth-space telecommunication systems,” International Telecommunication Union (ITU), Recommendation P.618-13, 12 2017. [Online]. Available: https://www.itu.int/dms pubrec/itu-r/rec/p/R-REC-P.618-13-201712-I!!PDF-E.pdf

[28] ITU-R, “Specific attenuation model for rain for use in prediction methods,” International Telecommunication Union (ITU), Recommendation P.838-3, 03 2005. [Online]. Available: https://www.itu.int/dms pubrec/itu-r/rec/p/R-REC-P.838-3-200503-I!!PDF-E.pdf

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 54, 2017, pp. 1273–1282.

[30] Z. Chen, T. -B. Xu, C. Du, C. -L. Liu and H. He, ”Dynamical Channel Pruning by Conditional Accuracy Change for Deep Neural Networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 799-813, Feb. 2021.

[31] R. Xing, Z. Su and Y. Wang, ”Intrusion Detection in Autonomous Vehicular Networks: A Trust Assessment and Q-learning Approach,” *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, 2019, pp. 79-83.

[32] F. Zenke, B. Poole and S. Ganguli, ”Continual learning through synaptic intelligence”, *Proc. 34th Int. Conf. Mach. Learn.*, pp. 3987-3995, 2017.

[33] X. Ma, J. Zhang, S. Guo and W. Xu, ”Layer-wised model aggregation for personalized federated learning”, *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 10092-10101, 2022.

[34] A. Rubinstein, “Perfect equilibrium in a bargaining model,” *Econometrica: Journal of the Econometric Society*, pp. 97–109, 1982.

[35] R. Deng, B. Di and L. Song, ”Pricing Mechanism Design for Data Offloading in Ultra-Dense LEO-Based Satellite-Terrestrial Networks,” *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1-6.

[36] Y. Pan, M. Chen, Z. Yang, N. Huang, and M. Shikh-Bahaei, “Energy efficient NOMA-based mobile edge computing offloading,” *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 310–313, Feb. 2019.

[37] K. Maine, C. Devieux and P. Swan, ”Overview of iridium satellite network”, *Proc. WESCON Conf. Rec. Microelectron. Commun. Technol. Producing Quality Products Mobile Portable Power Emerg.*, pp. 483, Nov. 1995.

[38] 3GPP TR, 38.811 (V0.3.0), ”Study on New Radio (NR) to Support non Terrestrial Networks (Release 15)”, Dec. 2017.

[39] E. Bourtsoulatze, D. Burth Kurka and D. Gündüz, ”Deep Joint Source-Channel Coding for Wireless Image Transmission,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567-579, Sept. 2019.

[40] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (Canadian Institute for Advanced Research).” [Online]. Available: http://www.cs.toronto.edu/kriz/cifar.html

[41] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009

[42] Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits, 1998. URL http://yann.lecun.com/exdb/mnist/.

[43] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, “Exploiting shared representations for personalized federated learning,” arXiv preprint arXiv:2102.07078, 2021.