Efficient Pruning-Split LSTM Machine Learning Algorithm for Terrestrial-Satellite Edge Network

Guhan Zheng*, Qiang Ni*, Keivan Navaie*, Haris Pervaiz*, and Charilaos Zarakovitis†
*School of Computing and Communications, Lancaster University, UK
†National Centre for Scientific Research Demokritos, Greece
Email:{g.zheng2, q.ni, k.navaie, h.b.pervaiz}@lancaster.ac.uk, c.zarakovitis@iit.demokritos.gr

***Abstract*—The recent advances in low earth orbit (LEO) satellite-borne edge cloud (SEC) enable resource-limited users to access edge servers via a terrestrial station terminal (TST) for rapid task processing capability. However, the dynamic variation in the TST transmit power challenges the served users to develop optimal computing task processing decisions. In this paper, we propose an efficient pruning-split long short-term memory (LSTM) learning algorithm to address this challenge. First, we present an LSTM algorithm for TST transmit power prediction. The proposed algorithm is then pruned and split to decrease the computing workload and the communication resource consumption considering the limited computing resource of TSTs and served users' quality of service (QoS). Finally, an algorithm split layer selection method is introduced based on the real-time situation of the TST. The simulation results are shown to verify the effectiveness of the proposed pruning-split LSTM algorithm.**

I. Introduction

The sixth-generation (6G) wireless network is anticipated to provide high data rate, low latency, low communication cost, and ubiquitous service [1]. This allows for the rapid development and popularity of the Internet of Things (IoT) [2]. However, traditional terrestrial networks are often difficult to fully cover remote regions or disaster areas to meet the demands for "ubiquitous service". Fortunately, in recent years, the development of low earth orbit (LEO) satellites technologies provide an alternative solution for underserved and unserved users. It has low cost, high throughput, and wide coverage to provide communication service to users with low communication latency [3]. Many countries and companies have already initiated LEO satellite projects, such as SpaceX Starlink [4] and OneWeb [5]. It is therefore foreseeable that in the near future, the integrated terrestrial-satellite network (TSN) will make communication services truly ubiquitous.

On another front, the popularity and development of IoT devices have spawned numerous computation-intensive applications. But dealing with these computation-intensive tasks is difficult due to IoT devices' limited computing power and battery life. To support the rising number of large-scale IoT applications [6], edge computing [7], which places cloud service closer to the users and provides abundant computing resources, is becoming a promising technology [8]. Users can offload computational tasks to edge cloud servers for fast processing. For those users without terrestrial network communication infrastructure support, computational tasks can only be offloaded to the cloud server for processing via the LEO satellite network. However, the high propagation latency makes it hard to satisfy those users' real-time requirements. Edge cloud servers can therefore be placed on LEO satellites to improve edge cloud service range and reduce users' computation latency [3], [9].

Today, the LEO satellite-borne edge cloud (SEC) has attracted much academic attention. There are mainly two types of methods for terrestrial users to offload computing tasks to LEO satellites: 1) Devices offloading to SEC directly over the C-band; 2) Offloading over the Ka-band with the assistance of terrestrial-station-terminals (TSTs) [10]. *Tang et al.* [11] considering terrestrial cloud presented a computing offloading decision problem, where suggested alternating direction method of multipliers (ADMM) algorithm is used to approximate the optimal solution. *Wang et al.* [12] proposed a TSN model and an offloading strategy based on game theory, in which users can choose computing tasks locally or offload tasks to SEC. Further, as same as [11] and [12], users were suggested directly offloading tasks to satellites via Ka-band and nonorthogonal multiple access were employed to improve the transmission rate [13]. However, limited by users' energy consumption and the number of satellite links, it is not practical for all users to offload directly [14]. Consequently, user offloading via TST assistance received more attention. For example, [3], [14]- [17] optimize users' computing offloading strategies considering offloading via TSTs, thus reducing latency and energy consumption of users.

However, existing research ignores the fact that TSTs are also IoT devices. Constrained by the tough environment in which they are deployed, their computing resources and energy are limited, and they cannot guarantee optimal QoS at all times. In particular, the movement of satellites, complex climate dynamics, and energy conditions result in fluctuations in transmit power. Previous research treated the transmit power as fixed, which may lead to the QoS being lower than expected, causing users' task completion strategies to be not optimal. It hence becomes an issue of how to anticipate TSTs' next-moment service capacity and inform it to the users to make users' reasonable offloading decisions. Further, when they expect to offload tasks to the SEC, it can cause service disruption, due to the radio resources taken up. This becomes a challenge for TSTs to ensure timely service to users while optimizing the latency and energy of their own computing tasks.

To tackle these challenges, our work considers a TSN, where users can choose offload tasks to the SEC via TSTs. Furthermore, a new pruning-split long short-term memory (LSTM) algorithm is proposed to predict the TST transmit power. Specifically, we employ LSTM deep learning approach to predict the TSTs' transmit power, and reduce the computational workload of the algorithm by pruning. Moreover, the algorithm can be split according to the actual scenario, so that TSTs can choose to compute this algorithm locally and in the SEC together. This minimises the latency and power consumption of TST's power prediction within tolerable service interruption time. Finally, the predicted information is informed to the users served and optimizing their computing offload strategies.

The rest of this paper is organized as follows. Section II introduces the system model of the TSN. In section III we described the proposed pruning-split LSTM algorithm. Section IV presents the simulation results. Finally, we conclude the paper in Section V.

II. System Model

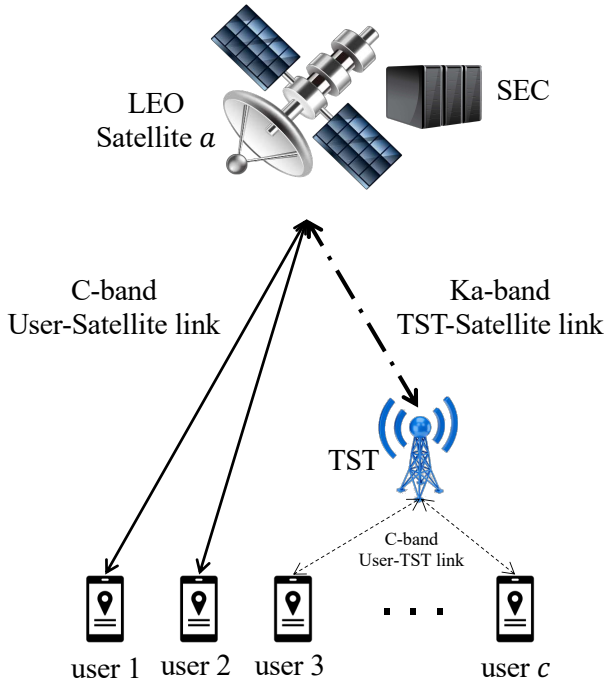

Fig. 1: System model

We consider a TSN (as illustrated in Fig. 1), where users are in areas without the support of ground edge service and can access SEC service via TST or connect directly. Denote the set of LEO satellites as $\mathcal{A} = \{1, 2..., a, ..., A\}$ and the set of TSTs in LEO satellite a coverage is $\mathcal{B} = \{1, 2..., b, ..., B\}$. The TST b provides service to C users within the coverage as a small cell which the set of users denoted by $\mathcal{C} = \{1, 2..., c, ..., C\}$. We consider each terrestrial user and TSTs has indivisible tasks [18] with the size in bits of $m \in [1, 2, \ldots, M]$, and the CPU cycle needed to execute one bit of task is δ. One task computing latency when user c computing task locally can be given by [19]

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

where f_c is user c's CPU-cycle frequency with the unit cycles/s. The energy required to calculate this task locally is expressed as

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

where $p_c^{LC} = \varepsilon f_c^3$ is the power needed to compute locally and ε is the energy factor related on chip architecture [20].

Similarly, when user c chooses to offload the task to SEC for faster computation, the computational latency can be obtained by

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

where f_a is the CPU-cycle frequency of edge server in LEO satellite a.

In addition, when the user c chooses to offload the task to the SEC, it has to tolerate transmission and propagation delays. In this paper, we only consider the scenario that users offload to SEC via TSTs. Thus, the communication delay caused by offloading can be denoted by

$$t_c^{TS} = \frac{2h}{c} + \frac{m_c}{R_b} + \frac{m_c}{R_c}, \tag{4}$$

where $\frac{2h}{c}$ is the propagation delay between TST b and LEO satellite a. h is the distance between TST b and LEO satellite a. c is the speed of light. Due to the small coverage of TSTs, the propagation delay between the user c and the TST b is negligible. Further, $\frac{m_c}{R_b}$ is the TST transmission delay to satellite a. $\frac{m_c}{R_b}$ is the transmission delay user c to TST b. R_b and R_c are the transmission rates of TST and the user, respectively. According to Shannon Theory, the transmission rate R_b and R_c can be expressed as

$$R_b = B_b \log_2 (1 + \frac{p_b g_b}{\sigma_b^2}), \tag{5}$$

$$R_c = B_c \log_2 (1 + \frac{p_c g_c}{\sigma_c^2}), \tag{6}$$

where B_b, p_b and g_b are bandwidth, transmission power and the channel gain on TST b-satellite a link, respectively. Further, σ_b^2 is the additive white Gaussian noise (AWGN) power in this link. Similarly, B_c, p_c, g_c and σ_c^2 are bandwidth, transmission power, the channel gain and AWGN power on user c-satellite a link, separately. The transmit power therefore greatly influences the transmission rate. Since latency and energy consumption should be considered jointly when user c is making a task computing processing decision. The change in transmit power impacts the system QoS and the user's offloading decision. In case the transmit power value used by the user to make the decision differs from the variable actual transmit power value, it is likely to result in the user

decision not being the actual optimum. Therefore, a scheme is needed to anticipate the TST transmit power in near future in advance and give guidance to users to reduce the possibility of increased user latency and energy consumption due to non-optimal decisions made by users.

III. Proposed Pruning-Split LSTM Algorithm

In this section, we propose our pruning-split LSTM algorithm to address the previously mentioned challenges. The proposed algorithm consists of 3 steps: 1) offline training and pruning; 2) online split; 3) deployment. The algorithm workflow is shown in Figure 2.

A. Offline Training and Pruning

To improve the QoS of the system, we need to design a transmit power prediction mechanism based on the available energy, supplementary energy, communication conditions of the TST b and so on. In this paper, we employ the LSTM model, an evolved algorithm of recurrent neural network (RNN), to predict the future transmit power through historical trajectories. This is because its main feature compared to other algorithms is that it can learn not only the current data but also past data sets in order to make future predictions taking into account the relevant context [21]. It is therefore very effective in sequential variable prediction such as transmit power prediction in our paper.

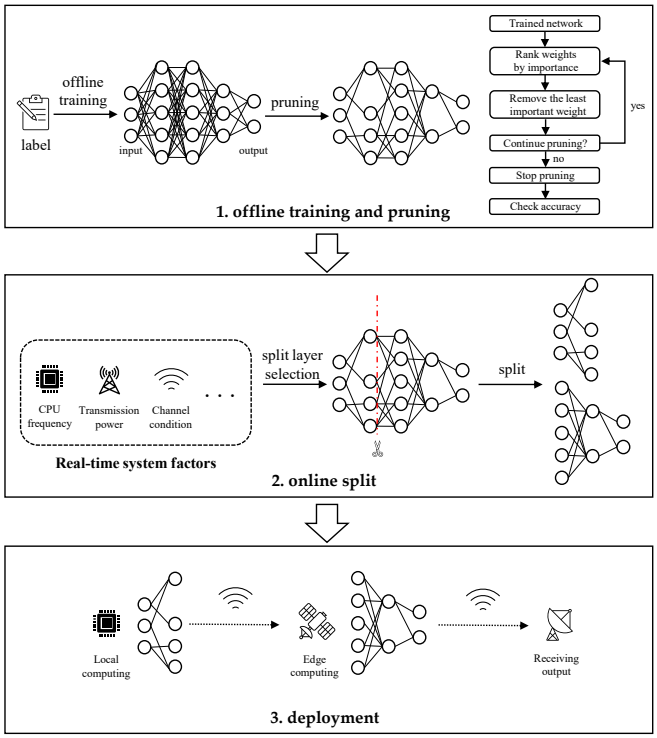

Fig. 2: Workflow of proposed Pruning-Split LSTM

The LSTM model first learns offline with the realistically acquired labels. Note that the LSTM algorithm is computationally intensive, it is usually hard for an IoT device to complete this model alone. To reduce the computing workload of this algorithm, after training the model, we use the size of the weights as the pruning criterion [22]. In each layer, a given percentage of lower magnitude weights will be considered insignificant and zeroed out. The pruning range is the whole test network, and the pruned model must achieve the allowed minimum accuracy A. As pruning reduces the number of weights, the model becomes sparse, thus decreasing the computational workload.

B. Online Split

Because of the limited computation capability of the TST, performing computing tasks on its own may result in high latency and energy consumption. But offloading to SEC may degrade the QoS of the system by taking up too many communication resources. Hence, the proposed algorithm can be split online into two computing tasks depending on the TST conditions. The first half of the algorithm could be chosen to be computed locally and the second half offloaded to the SEC. The first half is not chosen to be computed at the SEC in order to avoid leaking the original data and to protect TST's privacy. The ground station can therefore perform a better computing strategy by choosing at which layer to split the model. This minimises the latency and energy consumption in transmit power prediction while satisfying acceptable service interruption time.

In order to find the optimal splitting point, we need to know the following profiles: 1) weights and output data size for each layer in the pruned model; 2) maximum interruption time tolerance ς (ms); 3) system factors such as TST's CPU frequency and wireless channel condition. We assume that the LSTM network splits at layer i and there are I layers in the prediction model. When TST can offload the model after layer i to the SEC on satellite a for calculation within the tolerable time ς, similar to the users, the delay t_i for computing and transmission can be expressed as

$$t_i = \begin{cases} \frac{\delta M_{bi}}{f_b}, & \text{if } i = 0 \\ \frac{\delta M_{bi}}{f_b} + \frac{\delta M_{b(I-i)}}{f_a} + \frac{2h}{c} + \frac{M_{b(I-i)}}{R}, & \text{if } i > 0 \end{cases}, \tag{7}$$

where $M_{b(I-i)}$ is the output size of i-th layer and weights size of the remaining layers, f_b is the TST b's CPU-cycle frequency. Furthermore, R is the transmission rate. We can also have the energy consumption as

$$e_i = \begin{cases} \varepsilon \delta M_{bi} f_b^2, & \text{if } i = 0 \\ \frac{p_b M_{T(I-i)}}{R} + \varepsilon \delta M_{bi} f_b^2, & \text{if } i > 0 \end{cases}, \tag{8}$$

where p_b is the real-time transmission power.

Similar to the users, the TST requires minimal latency and energy consumption for the design of the task computing strategy. Thus, the selection of split point should minimise t_i and e_i, i.e., $\min\ \alpha t_i + \beta e_i$, when $\frac{M_{b(1-i)}}{R} \leqslant \varsigma$. The weight parameters α and β are used to weigh up the importance of delay versus energy consumption. Further, we define $\alpha t_i + \beta e_i$ as the cost, which has no units, and choose the value when

Algorithm 1 Pruning-split LSTM for transmitter power prediction

initialize: layers I in LSTM, allowed lowest accuracy A
training LSTM model and get the model accuracy $A^{'}$
if $A^{'} > A$
pruning to scale
  for i=1,2...I
    if $A_i > A$
      $t_i \longleftarrow (6)$
      $e_i \longleftarrow (7)$
    end if
  end for
  if t_i and e_i exist
    determining the weighting α, β of latency and energy consumption respectively
    $\min \ (\alpha t_i + \beta e_i),\ i = 1, 2, ...I$
    $s.t. \ \frac{M_{b(1-i)}}{R} \leqslant \varsigma$
  return $\min \ (\alpha t_i + \beta e_i)$
  else
  return NULL
  end if
else
return NULL
end if

the time unit is second and the power unit is watt. As this calculation is a simple linear computation, the number of model layers is also very limited. It is possible to iterate according to the sequence of the layer to find the optimal result, with much less complexity than power prediction.

C. Deployment

In the deployment phase, the first half of the split LSTM model is computed locally. The results, together with the second half of the pruned weights, are then offloaded to the SEC for computing. The entire algorithm process is shown in Algorithm 1.

IV. Numerical Evaluation

In this section, we evaluate the performance of the proposed algorithm. The algorithm used in simulations has three hidden layers, each with a weights' pruning ratio of 70%. In the simulations, the LEO satellite vertical altitude is 780 km based on Iridium satellite system [23], which is a classical LEO satellite system. We set the frequency of Ka-band as 30 GHz, and the maximum transmit power of each TST is 2 W [14]. Furthermore, we set the requirement of CPU cycles for computing one bit, δ being 120 cycle/bit and energy factor ε being 10^{-26}, according to the real applications [20]. The computation capability of SEC on satellite a and TST b are assumed to be $3 \times 10^9 cycles/s$ and $0.3 \times 10^9 cycles/s$, respectively [11]. The weight parameters of latency and energy consumption are set as $\alpha = 1$ and $\beta = 1$. The simulation parameters are also listed in Table I.

TABLE I: SIMULATION PARAMETERS

Parameters	Default Values
Ka-band carrier frequency	30 GHZ
Maximum transmit power of TST	2 W
Number of LSTM hidden layers	3
Number of neurons per layer	256
h	780km
δ	120 cycle/bit
ε	10^{-26}
f_a	$3 \times 10^9 cycles/s$
f_b	$0.3 \times 10^9 cycles/s$
α	1
β	1

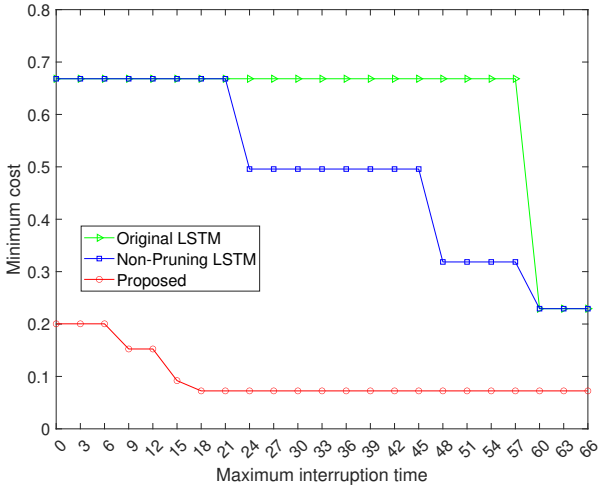

Fig. 3: Cost performance in different TST's service interrupt delay tolerance

In Fig. 3, the minimum latency and energy consumption cost versus the different TST's service interrupt delay tolerance is demonstrated, where the proposed pruning-split LSTM is compared with the original LSTM algorithm and non-pruning but split LSTM algorithm. We assume that a floating-point number takes up four bytes and the unit of time is millisecond. It can be found that the TST cost is decreasing as the tolerable time increases. This is because the TST has more time to offload this task to the SEC for computation thus reducing latency and energy consumption. Besides, the TST can obtain less cost by splitting the model in some fixed tolerance time. Furthermore, via the pruning and split, the TST computing cost can be reduced to a minimum.

Fig. 4 compares the successfully finished ratio of users' tasks for fixed transmission power and pruning-split LSTM at different times (2 p.m.-5 p.m.). We assume that the TST transmission is based on orthogonal frequency division multiplexing (OFDM) with carriers assigned to the user. When a user chooses TST-assisted offloading but does not receive the expected number of allocated subcarriers for transmission, the offloading strategy ultimately chosen by the user is not optimal

for the actual situation, is unsuccessful. It can be observed that the success rate of our proposed pruning-split LSTM algorithm is close to reality. As the energy supplement available to the TST gradually decreases with time, the transmitting power also decreases. It causes the gap between actual transmission power and fixed power increasingly large and thus reduces the success rate.

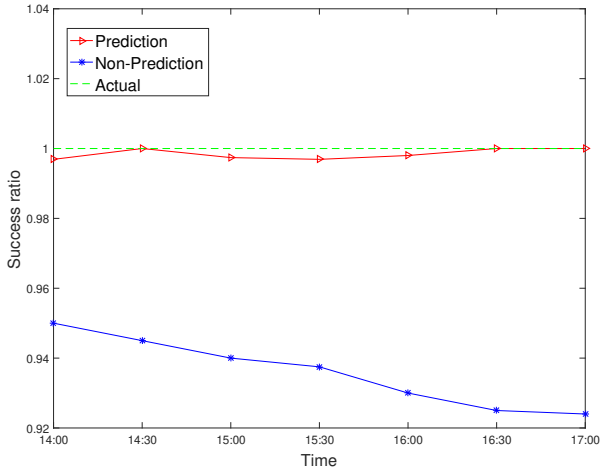

Fig. 4: Successfully finished ratio of tasks

V. Conclusion

In this paper, we investigated transmit power prediction problem for TST in a TSN. Considering the special position of the TST in the system, a pruning-split LSTM machine learning algorithm is proposed. It predicts the transmit power of the TST while considering the delay and energy consumption of the TST to calculate this algorithm. The simulation results show the effectiveness of the proposed algorithm in transmitting power prediction and reducing computational consumption.

Acknowledgement

This work was supported in part by the EU H2020 SANCUS project under the grant number GA952672.

References

[1] M. H. Alsharif, A. H. Kelechi, M. A. Albreem, S. A. Chaudhry, M. S. Zia, and S. Kim, “Sixth generation (6G) wireless networks: Vision, research activities, challenges and potential solutions,” *Symmetry*, vol. 12, no. 4, p. 676, Apr. 2020.

[2] W. Gao, Z Zhao, G. Min, Q. Ni, and Y. Jiang, ”Resource Allocation for Latency-aware Federated Learning in Industrial Internet of Things,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8505-8513, Dec 2021.

[3] 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.

[4] V. L. Foreman, A. Siddiqi and O. D. Weck, ”Large satellite constellation orbital debris impacts: Case studies of OneWeb and SpaceX proposals”, *Proc. AIAA SPACE Astronaut. Forum Exposit.*, pp. 5200, Sep. 2017.

[5] T. Azzarelli, ”OneWeb global access”, *Global Conf. Space Inf. Soc.*, pp. 1-24, Jun. 2016.

[6] P. Rahimi, C. Chrysostomou, H. Pervaiz, V. Vassiliou, Q. Ni, ”Joint Radio Resource Allocation and Beamforming Optimization for Industrial IoT in SDN-based Virtual Fog-RAN 5G-and-beyond Wireless Environments,” *IEEE Transactions on Industrial Informatics*, accepted in press, Oct 2021.

[7] H. Tian, X. Xu, L. Qi, X. Zhang, W. Dou, S. Yu and Q. Ni, ”CoPace: Edge Computation Offloading and Caching for Self-Driving with Deep Reinforcement Learning”, *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13281-13293, Dec 2021.

[8] P. Porambage, T. Kumar, M. Liyanage, J. Partala, L. Loven, M. Ylianttila, and T. Seppanen, ”Sec-EdgeAI: AI for edge security vs security for edge AI,” *Proc. 1st 6G Wireless Summit*, At Levi, Finland, 2019. Accessed: May 16, 2019.

[9] 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.

[10] 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.

[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] 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, 2020.

[13] X. Liu, X. B. Zhai, W. Lu and C. Wu, ”QoS-Guarantee Resource Allocation for Multibeam Satellite Industrial Internet of Things With NOMA,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2052-2061, March 2021.

[14] 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.

[15] 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.

[16] R. Deng, B. Di and L. Song, ”Ultra-Dense LEO Satellite Based Formation Flying,” *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3091-3105, May 2021.

[17] B. Mao, F. Tang, Y. Kawamoto and N. Kato, ”Optimizing Computation Offloading in Satellite-UAV-Served 6G IoT: A Deep Learning Approach,” *IEEE Network*, vol. 35, no. 4, pp. 102-108, July/August 2021.

[18] J. Ren, G. Yu, Y. Cai and Y. He, ”Latency optimization for resource allocation in mobile-edge computation offloading”, *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506-5519, Aug. 2018.

[19] C. You, K. Huang, H. Chae and B.-H. Kim, ”Energy-efficient resource allocation for mobile-edge computation offloading”, *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397-1411, Mar. 2017.

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

[21] S. Hochreiter and J. Schmidhuber, ”Long short-term memory”, *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, 1997.

[22] M. Zhu and S. Gupta, “To prune, or not to prune: exploring the efficacy of pruning for model compression,” *ICLR Workshop*, 2018.

[23] 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.