Acceleron Aerospace Journal || AAJ.11.2106-2446
Volume 3, Issue 3, pp (487-494)
Student Paper
E-ISSN- 2583-9942 || DOI: 10.61359/11.2106-2446

Optimizing UAV-Assisted Relaying through Split Learning for Enhanced Distributed Inference in IoT-Based Ecological Monitoring

Samarth Pandey*

ORCID: 0009-0007-3087-2044

DBMS English School, Kadma, Jamshedpur, Jharkhand 831005

Abstract: The integration of unmanned aerial vehicles (UAVs) into Internet of Things (IoT) systems present a promising solution for enhancing connectivity and data transmission, particularly in remote and rural areas. I introduce an innovative approach that combines UAV-assisted relaying with split learning (SL) for distributed inference in IoT environmental monitoring applications. By leveraging UAVs as relays, this method addresses connectivity challenges and extends coverage in underserved regions, while SL optimizes data processing and privacy. The proposed system architecture enables UAVs to facilitate reliable data transmission between edge devices and central servers, while SL allows for efficient learning by partitioning the neural network across devices and UAVs. This approach provides adaptive server strategies based on channel conditions and performance metrics. Extensive simulations demonstrate that the proposed framework significantly enhances system adaptability, reduces latency, and improves accuracy even under adverse channel conditions. The integration of UAV relaying with SL offers flexibility in managing trade-offs between data quality, latency, and computational load, ensuring robust performance in challenging environments. This research contributes valuable insights into advancing distributed learning in IoT systems with potential applications in environmental monitoring, disaster response, and rural connectivity.

Table of Contents

1. Introduction

Pollution from industrial activities, transportation emissions, and waste disposal presents severe environmental challenges, raising substantial concerns about its impact [1]. Maintaining health and hygiene is crucial for both human sustainability and national progress, which depend on a clean and hazard-free environment. Effective monitoring of these factors is essential to ensure public health, particularly in rural and underdeveloped regions. Recent advancements have introduced IoT-based environmental monitoring systems, which utilize interconnected devices equipped with various sensors to collect real-time data on critical environmental parameters such as air quality, soil moisture, water quality, temperature, and humidity [1, 2]. These devices leverage wireless communication technologies—including Wi-Fi, cellular networks, LoRaWAN, and satellite connectivity—to transmit data to centralized servers or cloud platforms for storage, analysis, and further processing [2]. Unmanned aerial vehicles (UAVs) have gained extensive application across diverse sectors, including telecommunications, rescue operations, and surveillance [3, 4]. Their potential to facilitate end-to-end wireless communications, particularly in remote and rural areas where traditional cellular infrastructure is lacking, has been widely recognized [4]. UAVs can be employed as mobile access points, addressing data demand and congestion challenges anticipated in future wireless networks [5]. Unlike static infrastructure, UAV networks offer flexible deployment, allowing them to dynamically extend coverage as needed [6]. In this work, I propose a novel approach for distributed learning in IoT environmental monitoring scenarios. By integrating the split learning (SL) paradigm with UAV relaying within an IoT network, this approach enhances data transmission rates and ensures a balanced distribution of computational tasks among edge devices, UAVs, and central servers. Additionally, the system's adaptability is improved through the server's ability to determine the optimal transmission strategy based

*Student Scholar, DBMS English School, Kadma, Jamshedpur, Jharkhand 831005. **Corresponding Author: samarthpan945@gmail.com.**
** Received: 16-August-2024 || Revised: 25-August-2024 || Accepted: 30-August-2024 || Published Online: 30-August-2024.

on real-time channel conditions and performance metrics such as latency, throughput, and energy efficiency. Numerical simulations demonstrate that this distributed learning approach provides significant robustness to varying channel conditions while achieving high estimation accuracy.

2. Background

2.1 IoT Connectivity in Rural Areas

IoT connectivity is closely tied to the developmental status of a country [7]. In developed nations, rural areas are typically accessible via transportation networks and are supported by the electricity grid. However, mobile operators often face challenges in achieving satisfactory returns on investment when extending backhaul to these regions. In contrast, developing countries, especially impoverished areas, struggle with bridging the digital divide. Essential services like healthcare and education depend on connectivity, but inadequate transportation infrastructure isolates rural communities, and local power generation is common. Establishing backhaul in such areas often requires state subsidies to facilitate cost-effective solutions.

UAVs equipped with communication technology offer a promising solution for providing connectivity in rural and underserved regions [8]. These drones can act as flying base stations, creating temporary or permanent connectivity where traditional infrastructure is impractical or prohibitively expensive. By flying over remote areas, UAVs can establish wireless links between users and broader network infrastructures, extending backhaul links to underserved communities [9]. In regions with underdeveloped or non-existent transportation infrastructure, UAVs offer a flexible and efficient means of providing backhaul links [7]. Their swift deployment and adaptability make them particularly valuable in emergency situations or resource-limited areas [10].

2.2 Split Learning

Split learning (SL) [11, 12] represents a novel distributed learning paradigm that divides a neural network F (comprising L layers) into sequential layers distributed across multiple participants, such as edge devices and servers. In SL, edge devices securely share their training data with the server, which manages the training process and handles most of the computational work- load. This distributed approach accelerates convergence and alleviates band- width constraints [12].

SL separates model training and inference processes. During training, raw data remains on edge devices, preventing unnecessary data transmission across the network. The neural network can be represented as $F = (f_E, f_S)$.

where $f_E : R^N \quad R^M$ and $f_S : R^M \quad R^1$, with N and M denoting the dimensions of raw data and intermediate representations, respectively, and $M < N$. During activation, the edge device's sub-network produces an intermediate representation of raw data x as $z = f_E(x)$, which is then sent to the server for prediction $\hat{y} = f_S(z)$ (where f_S is the sub-network located on the server). This approach enables collaborative learning while preserving data privacy by iteratively exchanging model updates between the server and edge devices [12]. In split inference, SL enhances efficiency by utilizing pre- trained models distributed across multiple devices. Initial data processing occurs locally on edge devices, generating intermediate representations z, which are then transmitted to a centralized server for aggregation and final inference [12].

In the context of a rural IoT network, which includes edge devices, servers, and a relaying UAV, the neural network F is divided into three components: $F = (f_E, f_D, f_S)$, where $f_D : R^M \rightarrow R^M$ represents the UAV's sub-network.

2.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs), designed for sequence-based tasks, are well-suited for processing time series data due to their ability to discern temporal relationships [13]. A simple single-layer RNN is illustrated in Fig. 1, where the output from the previous time step, t 1, is fed into the current time step, t, allowing the network to retain past information. The computation for a single RNN cell is described by:

$$\boldsymbol{h_t = \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh}),}$$

where tanh is the hyperbolic tangent function, h_t and h_{t-1} are the hidden states at time steps t and $t-1$, respectively, and W_{ih}, W_{hh}, b_{ih}, and b_{hh} are the weights and biases to be learned, with x_t denoting the input at time t.

Basic RNN cells face challenges in learning long-range dependencies due to issues such as vanishing or exploding gradients. To address these limitations, Long Short-Term Memory (LSTM) cells were introduced **[14]**. LSTM cells include specialized memory blocks within the recurrent hidden layer, enhancing their ability to capture long-term dependencies. Each memory block consists of memory cells and gates, which manage the flow of information and retain temporal states.

Split Learning–Based RNNs: Integrating SL into LSTM networks initially posed challenges, leading researchers to explore alternative methods. Some studies have suggested using 1D-CNNs instead of LSTMs. Recent advancements have introduced efficient techniques for incorporating SL into LSTM architectures, directly embedding SL into LSTM models to address implementation challenges. This paper builds on the LSTMSPLIT algorithm **[19]**, which vertically splits the LSTM neural network, requiring at least two LSTM layers. The input sequence is stored at the edge device, with intermediate representations transmitted from the edge device's LSTM layer to the server's LSTM layer, and update gradients flowing in the opposite direction (Fig. 2). Another approach **[20]** involves distributing a single LSTM layer across multiple edge devices, partitioned into sub-networks trained individually, handling segments within multi-segment training sequences. Communication and parameter sharing among edge devices align with the federated learning paradigm.

3. UAV-Assisted Relaying in IoT

3.1 System Model

This section discusses a conventional IoT system that includes an edge device, a server, and a UAV acting as a wireless relay. The UAV serves as a mobile base station providing a backhaul link, as shown in Figure 2. Each system component has distinct computational capabilities, represented as C(fE) < C(fD) < C(fS), where C( ) denotes the computational complexity of the sub-networks.

Data Collection and Pre-Processing

The edge device gathers raw data, labeled as x, which includes sensor measurements and corresponding labels y. This data is pre-processed by the edge device's sub-network, producing an intermediate representation z = fE(x). This intermediate representation is essential for subsequent processing stages.

Communication and Processing Strategy

The server manages communication and inference processes based on channel conditions and performance metrics like error rate, latency, and communication overhead. The server decides whether to communicate directly with the edge device or involve the UAV's sub-network in processing. Channel conditions can vary significantly:

- **Edge Device to UAV (WED):** This link may experience varying quality.
- **Edge Device to Server (WES):** This link may also have fluctuating quality.
- **UAV to Server (WDS):** This link is expected to maintain consistently good quality throughout the network's operation.

Processing and Inference

Depending on the communication strategy:

- **Direct Communication:** If the edge device communicates directly with the server, the intermediate representation at the server side is $\hat{z}$ = WES(z).

- **Relay Through UAV:** If the UAV relays the communication, the intermediate representation after processing by the drone sub-network is zˆ = WDS(fD(WED(z))). The server then decides between two estimation strategies based on latency constraints:
- Full Server Network: yˆfull = fS(zˆ)
- Reduced Server Network: yˆFC = fˆS (zˆ), where only the output layer of the server sub-network is used.

Loss Function and Optimization

The loss function for model evaluation is:

$$\boldsymbol{L(y, \hat{y}full, \hat{y}FC) = MSE(y, \hat{y}full) + MSE(y, \hat{y}FC)},$$

where MSE represents the mean squared error over the training dataset Dtr. During backpropagation, gradients are calculated and transmitted from the server through the UAV to the edge device, reversing the neural network's operations, as shown by the blue arrows in Figure 2. All three sub-networks (fE, fD, fS) are optimized together using algorithms like stochastic gradient descent (SGD) or its variants, such as Adam.

Optimization Goal

The main objective is to define the most effective transmission strategy that minimizes the overall system error, measured by the MSE error between y and yˆ across all test examples. This involves considering channel conditions and latency constraints to select the optimal approach for communication and processing.

3.2 Channel Model

The wireless communication links between the edge device and server (WES), edge device and UAV (WED), and UAV and server (WDS) are modeled as conventional erasure channels. Each link is characterized by an erasure probability p, which affects the reliability of data transmission.

Channel Representation

Each link is represented as a binary vector q (0, 1)^M, where M is the length of the intermediate representation z. The channel introduces erasures by either retaining or removing individual symbols from z. Therefore, the received version of z at the server side is:

$$\hat{\boldsymbol{z}} = \boldsymbol{z} \odot \boldsymbol{q},$$

where ⊙ denotes element-wise multiplication. This model accounts for the potential loss of information during transmission and its impact on overall system performance.

Impact on Performance

The erasure probability p affects the quality of the intermediate representation received at the server. Higher erasure rates lead to greater distortion of z, impacting the accuracy of the final estimation. Effective strategies to mitigate channel distortions are crucial for maintaining high performance in IoT systems with UAV-assisted relaying.

4. Performance Evaluation

3.1 Training Setup

To evaluate the UAV-assisted relaying approach combined with split learning (SL) for distributed inference, a dataset specifically designed for environmental monitoring was used. The dataset focuses on pollution monitoring in the Danube River near Novi Sad, containing 3,264 instances. 70% of the data was allocated for training and 30% for testing. Each instance includes daily measurements from November 2013 to October 2022, covering eight water quality parameters: temperature, pH, electrical conductivity, dissolved oxygen, oxygen saturation, ammonium, and nitrite. The predictive modeling focused on forecasting dissolved oxygen levels. The prediction

is based on the last 20 measurements of dissolved oxygen and the measurements of the other seven parameters for the current day. After preprocessing and converting to time series, each dataset instance consists of 27 features (20 historical measurements of dissolved oxygen plus 7 other parameters) and one label (the current day's dissolved oxygen level). The data is normalized to the range of -1 to 1. The training follows the conventional SL method, with modifications for this specific scenario. Initially, raw data x is preprocessed on the edge device, and the intermediate representation is transmitted either directly to the server via WES or through the backhaul using a drone (WED and WDS). The drone performs additional processing of the intermediate representation.

The neural network is partitioned into sub-networks across the edge device (one LSTM layer), the drone (two LSTM layers), and the server (three LSTM layers followed by a fully connected (FC) layer). The number of LSTM hidden units H is set to match the length of the intermediate representation M, with H = M = 10. The FC layer at the server also has 10 neurons. Training uses a learning rate $\alpha = 0.01$, $\beta 1 = 0.9$, and $\beta 2 = 0.999$, with a batch size of 64. Stochastic gradient descent (SGD) with the Adam optimizer is employed. During training, channel conditions are fixed, following a proposed approach. However, during testing, the model's performance is assessed across varying erasure probabilities p. Specifically, the impact of varying training erasure probabilities ptr in WES and WED is explored while maintaining favorable conditions in the backhaul channel WDS (with a small erasure probability). Dropout layers simulate the erasure channel conditions during training, substituting all three wireless links in Figure 2 on a symbol basis. Dropout probabilities are adjusted to regulate channel erasures in simulations.

3.2 Numerical Results

System performance under varying channel conditions was investigated, considering significant distortions introduced by either WES or WED. During testing, the erasure probability for the more distorted channel is set to p1, while the less distorted channel is set to p2 = p1 * 0.3. The backhaul channel WDS remains constant with an erasure probability of 0.05 during both training and testing phases. Figure 3 illustrates the mean squared error (MSE) performances of fronthaul and backhaul systems. The backhaul (WED) introduces significant distortion due to a high erasure probability. Specifically, the training erasure probability for WED is 0.5, while for WES it is 0.1. During testing, p is set to 0.5 for WED and 0.15 for WES. Similar setups are evaluated with different configurations to demonstrate the proposed approach's robustness to distorted channels. These settings can generalize to more complex scenarios involving multiple drones and mobile edge devices. Results show that the combination of split learning and UAV-assisted relaying provides superior performance compared to direct server communication, particularly in highly distorted channels. The proposed approach effectively mitigates the impact of erasure channels, demonstrating its potential in real-world IoT deployments where wireless channels are unreliable.

5. Conclusion

This work introduces an innovative framework that combines distributed learning with UAV-assisted relaying for IoT environmental monitoring systems. The proposed architecture demonstrates significant adaptability to varying channel conditions, offering performance trade-offs that can be optimized by the server. By integrating the split learning (SL) paradigm, the framework efficiently balances the computational load among the edge device, UAV, and server. Future research will focus on incorporating additional factors into the server's decision-making process, such as latency and energy efficiency. This enhancement will increase the system's flexibility and responsiveness, further improving its performance in dynamic and challenging environments.

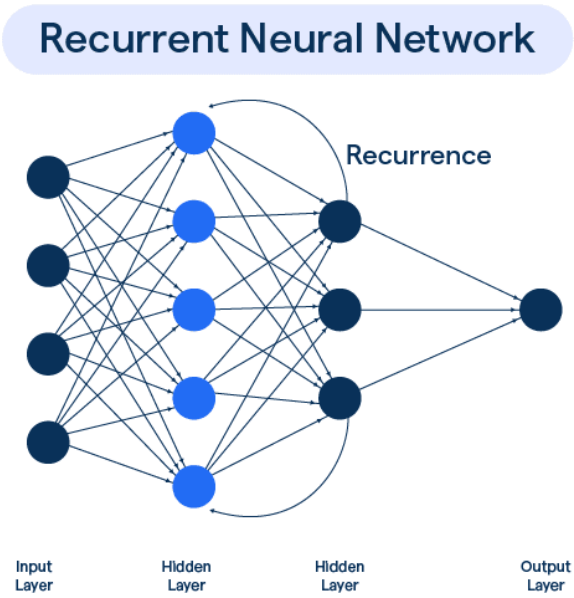

Figure-1 The structure of Recurrent Neural Network [Image Courtesy: Google]

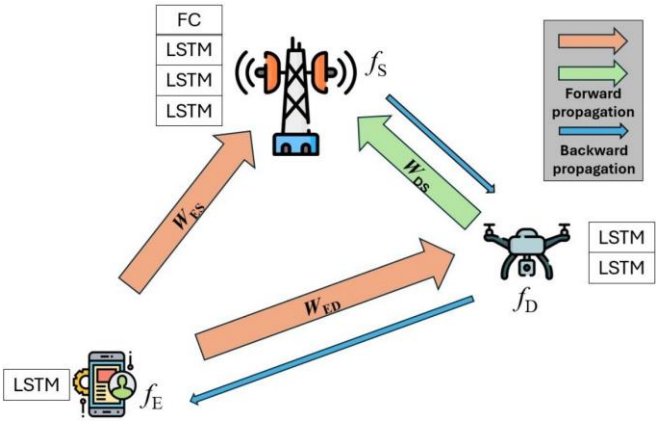

Figure-2 SL–based fronthaul/backhaul communication with different channel conditions [Image Courtesy: Google]

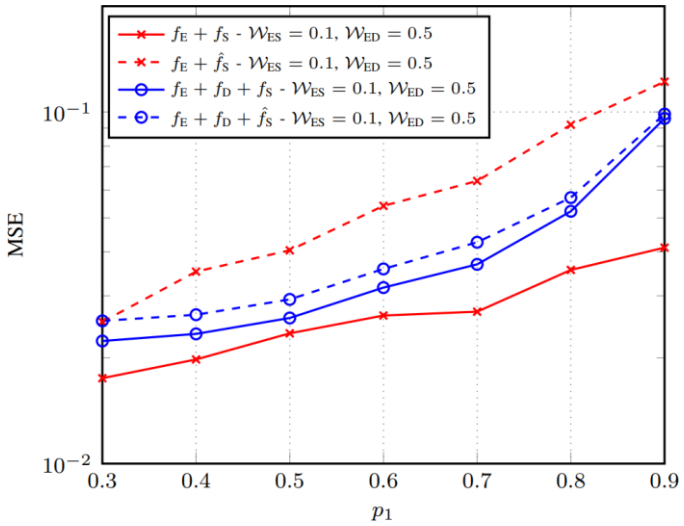

Figure-3 MSE vs. p1 Erasure Probabilities: Better Edge-to-Server Channel Conditions (ptr for WED greater than WES) with WDS = 0.05.

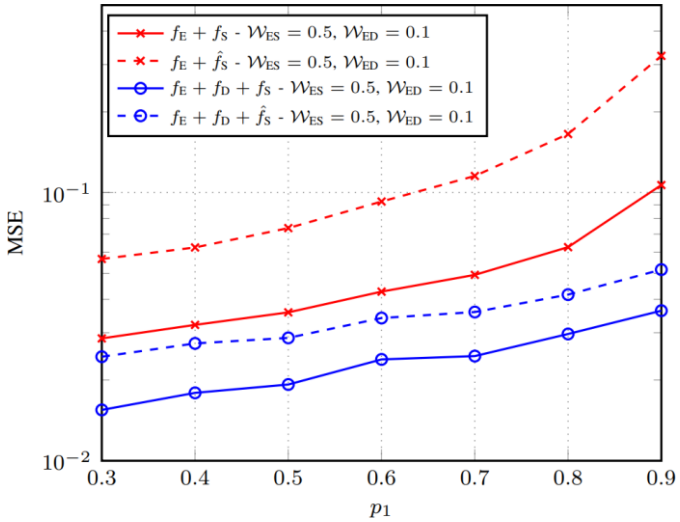

Figure-4 MSE as a Function of p1 Erasure Probabilities: Analysis of Improved Edge-to-Drone Channel Conditions (with ptr for WED smaller than WES) and WDS = 0.05.

6. References

[1] Ullo, S. L., & Sinha, G. (2020). Recent advancements in smart environmental monitoring through IoT and sensor technologies. *Sensors, 20*(11), Article 3113. https://doi.org/10.3390/s20113113.

[2] Mois, G., Folea, S., & Sanislav, T. (2017). Evaluation of three wireless sensor systems based on IoT for environmental observation. *IEEE Transactions on Instrumentation and Measurement, 66*(8), 2056–2064. https://doi.org/10.1109/TIM.2017.2670418.

[3] Mozaffari, M., Saad, W., Bennis, M., Nam, Y.-H., & Debbah, M. (2019). Introduction to UAVs in wireless networks: Use cases, challenges, and research directions. *IEEE Communications Surveys & Tutorials, 21*(3), 2334–2360. https://doi.org/10.1109/COMST.2019.2916180.

[4] Sabzehali, J., Shah, V. K., Fan, Q., Choudhury, B., Liu, L., & Reed, J. H. (2022). Optimizing the quantity, positioning, and connectivity of multi-UAV networks. *IEEE Internet of Things Journal, 9*(21), 21548–21560. https://doi.org/10.1109/JIOT.2022.3184006.

[5] Zeng, Y., Zhang, R., & Lim, T. J. (2016). Challenges and opportunities in wireless communications with unmanned aerial vehicles. *IEEE Communications Magazine, 54*(5), 36–42. https://doi.org/10.1109/MCOM.2016.7470933.

[6] Galkin, B., Kibilda, J., & DaSilva, L. A. (2018). Backhaul strategies for UAVs operating at low altitudes in urban settings. In *Proceedings of the IEEE International Conference on Communications (ICC)* (pp. 1–6). https://doi.org/10.1109/ICC.2018.8422781.

[7] Yaacoub, E., & Alouini, M.-S. (2021). Optimizing fronthaul and backhaul connectivity for IoT traffic in rural settings. *IEEE Internet of Things Magazine, 4*(1), 60–66. https://doi.org/10.1109/IOTM.2021.3051838.

[8] Zhang, L., & Ansari, N. (2019). Enhancing deployment and throughput of DBSs for uplink communication. *IEEE Open Journal of Vehicular Technology, 1*, 18–28. https://doi.org/10.1109/OJVT.2019.2957325.

[9] Fouda, A., Ibrahim, A. S., Guvenc, I., & Ghosh, M. (2018). In-band integrated access and backhaul for UAVs in 5G networks. In *Proceedings of the IEEE Conference on Vehicular Technology* (pp. 1–5). https://doi.org/10.1109/VTCFall.2018.8690565.

[10] Selim, M. Y., & Kamal, A. E. (2018). Rehabilitating post-disaster 4G/5G networks using drones: Addressing battery and backhaul challenges. In *Proceedings of the IEEE Globecom Workshops (GC Wkshps)* (pp. 1–6). https://doi.org/10.1109/GLOCOMW.2018.8644516.

[11] Gupta, O., & Raskar, R. (2018). Distributed training of deep neural networks across multiple agents. *Journal of Network and Computer Applications, 116*, 1–8. https://doi.org/10.1016/j.jnca.2018.04.012

[12] Vepakomma, P., Gupta, O., Swedish, T., & Raskar, R. (2018). Split learning in healthcare: Distributed deep learning while preserving patient privacy. arXiv:1812.00564. https://doi.org/10.48550/arXiv.1812.00564

[13] Husken, M., & Stagge, P. (2003). Classification of time series using recurrent neural networks. *Neurocomputing, 50*, 223–235. https://doi.org/10.1016/S0925-2312(02)00505-3.

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. **https://doi.org/10.1162/neco.1997.9.8.1735**.

[15] Abuadbba, S., Kim, K., Kim, M., Thapa, C., Camtepe, S. A., Gao, Y., ... & Nepal, S. (2020). Applying split learning to 1D CNN models for privacy-preserving training. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (Asia CCS)* (pp. 305–318). **https://doi.org/10.1145/3320269.3384718**.

[16] Gao, Y., Kim, M., Abuadbba, S., Kim, Y., Thapa, C., Kim, K., Camtepe, S. A., ... & Nepal, S. (2020). Comprehensive evaluation of federated and split learning for IoT systems. In *Proceedings of the IEEE 2020 International Symposium on Reliable Distributed Systems* (pp. 91–100). **https://doi.org/10.1109/SRDS51060.2020.00021**.

[17] Zhang, W., Zhou, T., Lu, Q., Yuan, Y., Tolba, A., & Said, W. (2020). FedSL: A communication-efficient federated learning approach with split layer aggregation. *IEEE Internet of Things Journal*. **https://doi.org/10.1109/JIOT.2020.3035348**.

[18] Koda, Y., Park, J., Bennis, M., Yamamoto, K., Nishio, T., Morikura, M., & Nakashima, K. (2020). Efficient multimodal split learning for predicting mmWave received power. *IEEE Communications Letters*, *24*(6), 1284–1288. **https://doi.org/10.1109/LCOMM.2020.2993112**.

[19] Jiang, L., Wang, Y., Zheng, W., Jin, C., Li, Z., & Teo, G. S. (2022). LSTMSPLIT: Effective SPLIT learning based on LSTM for sequential time-series data. arXiv:2203.04305. **https://doi.org/10.48550/arXiv.2203.04305**

[20] Abedi, A., & Khan, S. S. (2023). Federated split learning for sequential data in recurrent neural networks. *Multimedia Tools and Applications, 83*, 28891–28911. **https://doi.org/10.1007/s11042-022-13758-8**.

[21] Kingma, D. P., & Ba, J. L. (2015). Adam: A stochastic optimization method. In *Proceedings of the International Conference on Learning Representations (ICLR)* (pp. 1–41). **https://doi.org/10.48550/arXiv.1412.6980**.

[22] Itahara, S., Nishio, T., Koda, Y., & Yamamoto, K. (2022). Communication-oriented fine-tuning for resilient distributed inference in lossy IoT networks. *IEEE Access, 10*, 14969–14979. **https://doi.org/10.1109/ACCESS.2022.3156589**.

[23] O'Shea, T., & Hoydis, J. (2017). An overview of deep learning techniques for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, *3*(4), 563–575. **https://doi.org/10.1109/TCCN.2017.2769223**.

[24] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by reducing feature detector co-adaptation. arXiv:1207.0580. **https://doi.org/10.48550/arXiv.1207.0580**.

7. Conflict of Interest

The author declares no competing conflict of interest.

8. Funding

No funding was received to support this study.