




Please cite the Published Version

Li, Zihong, Hong, Yang , Bashir, Ali Kashif , Al-Otaibi, Yasser D  and Wu, Jun  (2023) Software-Defined GPU-CPU Empowered Efficient Wireless Federated Learning With Embedding Communication Coding for Beyond 5G. IEEE Open Journal of the Communications Society, 4. pp. 990-1000. ISSN 2644-125X

DOI: <https://doi.org/10.1109/OJCOMS.2023.3266444>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Version: Published Version

Downloaded from: <https://e-space.mmu.ac.uk/634335/>

Usage rights:  [Creative Commons: Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)

Additional Information: This is an open access article which first appeared in IEEE Open Journal of the Communications Society

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

Software-Defined GPU-CPU Empowered Efficient Wireless Federated Learning With Embedding Communication Coding for Beyond 5G

ZIHONG LI¹ (Graduate Student Member, IEEE), YANG HONG¹ (Graduate Student Member, IEEE),
ALI KASHIF BASHIR^{1,2,3} (Senior Member, IEEE), YASSER D. AL-OTAIBI^{1,4} (Senior Member, IEEE),
AND JUN WU¹ (Senior Member, IEEE)

¹Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan

²Department of Computing and Mathematics, Manchester Metropolitan University, M15 6BH Manchester, U.K.

³Woxsen School of Business, Woxsen University, Hyderabad 502345, India

⁴Department of Information Systems, Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Jeddah 21589, Saudi Arabia

CORRESPONDING AUTHOR: J. WU (e-mail: junwu@aoni.waseda.jp)

This work was supported in part by JSPS KAKENHI under Grant 23K11072; in part by the National Natural Science Foundation of China under Grant U21B2019 and Grant 61972255; and in part by the Institutional Fund Projects under Grant IFPIP: 50-830-1443.

ABSTRACT Currently, with the widespread of the intelligent Internet of Things (IoT) in beyond 5G, wireless federated learning (WFL) has attracted a lot of attention to enable knowledge construction and sharing among a huge amount of distributed edge devices. However, under unstable wireless channel conditions, existing WFL schemes exist the following challenges: First, learning model parameters will be disturbed by bit errors because of interference and noise during wireless transmission, which will affect the training accuracy and the loss of the learning model. Second, traditional edge devices with CPU acceleration are inefficient due to the low throughput computation, especially in accelerating the encoding and decoding process during wireless transmission. Third, current hardware-level GPU acceleration methods cannot optimize complex operations, for instance, complex wireless coding in the WFL environment. To address the above challenges, we propose a software-defined GPU-CPU empowered efficient WFL architecture with embedding LDPC communication coding. Specifically, we embed wireless channel coding into the server weight aggregation and the client local training process respectively to resist interruptions in the learning process and design a GPU-CPU acceleration scheme for this architecture. The experimental results show its anti-interference ability and GPU-CPU acceleration ability during wireless transmission, which is 10 times the error control capability and 100 times faster than existing WFL schemes.

INDEX TERMS Internet of Things, wireless federated learning, LDPC, GPU-CPU.

I. INTRODUCTION

RECENTLY, Japan and the European Union reached consensus on strengthening cooperation on the new generation of communication standard “Beyond 5G”. With rise of the concept beyond 5G, the Internet of Things (IoT) has developed rapidly [1]. It is widely used in scenarios that require low-power computing and remote access control, such as health detection [2], logistics [3], and automated home [4]. As a result, many IoT devices have been spawned,

which are designed to achieve automatic data transmission and provide ubiquitous wireless connections to billions of devices with sensing [5], communication [6], computing and control capabilities, such as smartphones and sensors. In addition, the huge amount of data generated by the devices of the IoT can be used to extract useful information through machine learning, thus achieving a variety of intelligent IoT services. Emerging applications of the intelligent IoT include automobile driving [7], [8], unmanned aerial vehicles

(UAV) [9], [10], robots [11], health care [12], supply chain finance, and so on. However, in the 6G era, the paradigm shift from ‘interconnected things’ to ‘interconnected intelligence’ through modern machine learning technology is facing three major challenges. First, data transfer with private information to a cloud server is vulnerable to eavesdropping and data tampering attacks. Second, due to the limited wireless channel transmission resources, the machine learning training model which aggregates a large amount of distributed data through the wireless network channel may lead to network congestion, which causes network delay to be too long and affects the efficiency of machine learning.

In order to solve the above problems, researchers have studied and designed federated machine learning (FL) [13]. FL is a novel distributed learning system, which is also considered as a promising privacy-sensitive and low-latency intelligent IoT application solution with the ability to utilize distributed computing resources. We consider a scenario where there is such a network structure with a server and several clients in a wireless network environment. In this wireless network environment, if all data of the client is simply transmitted to the server, it will be a heavy burden for the wireless network with limited resources, and the server cannot obtain all the wireless data quickly, which will affect the performance of FL model. Compared to traditional machine learning, FL has more prospects for data privacy. FL allows the training data stay on the local, and only requires each IoT device to upload its locally updated model to an edge aggregation server over wireless transmission during model training. This enhances devices privacy and data security by preventing data collected on IoT devices from being leaked to other devices and aggregation servers. Among them, WFL [14], [15], [16], [17], [18] is the most recent research hotspot, and many new studies have also been generated around WFL. Such as communication problems [19], [20], user selection problems [21], [22], [23], [24], [25], security problems [26], [27], and automatic modulation classification problems [28], [29]. WFL relies on both cloud servers and edge devices. A complete federated learning process includes multiple rounds. Each participant uses local data to perform local training to obtain a local model, and sends the weights parameter to the cloud server through wireless channel coding which can do correction and error detection of wireless transmitted digital signals and enhance the ability of data to withstand various interference. The cloud server aggregates the weights parameter of each participant to obtain a global model. Then, the weights parameter of the global model are sent back to the edge devices for the next round of training until the model converges. WFL can share federated learning models to complete learning tasks without sharing training data. Because the data does not leave the local during the training process, so the data security of the participants is guaranteed, and a lot of communication overhead is saved.

Although WFL has the above characteristics, its performance is still affected by the following factors [30]. First, earning model parameters will be disturbed by bit

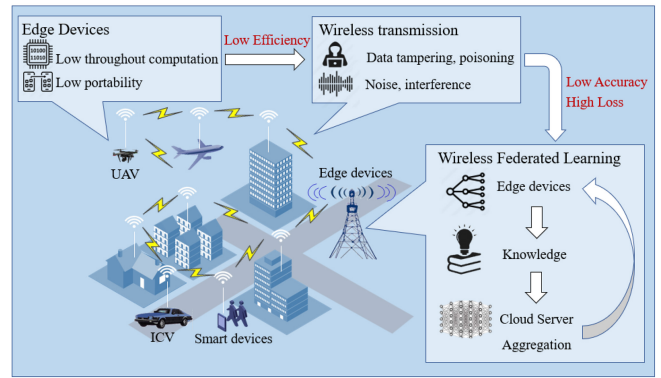


FIGURE 1. Knowledge exchange by wireless transmission in intelligent IoT network.

errors because of interference and noise during wireless transmission. In the wireless transmission medium, unstable uplink and downlink transmission will cause the parameter weights of the model to be incorrectly transmitted. Second, traditional edge devices with CPU acceleration are inefficient due to the low throughput computation, especially in accelerating the encoding and decoding process during wireless transmission. Third, current hardware-level GPU acceleration methods cannot optimize complex operations, for instance, complex matrix multiplication in wireless channel coding in the WFL environment.

To solve the above challenges, first, we designed a new architecture that optimize WFL with wireless channel coding. Specifically, we embed the encoding and decoding process into the local learning process and the server parameter aggregation process of WFL. Through wireless channel coding, the model parameters obtained by learning and aggregation can be checked and corrected while transmitting in the wireless channel, so the transmission process of model parameters of WFL is more reliable. Second, we also designed a software defined CPU-GPU hybrid architecture to speed up the encoding and decoding process of model parameters by wireless channel coding. Compared with traditional hardware acceleration, software-defined GPU-CPU acceleration can accelerate the wireless channel coding process of specific model parameters through software programming. Third, The software-defined acceleration method we designed makes the acceleration not limited to hardware conditions. It can optimize the transmission of different parameters of different edge devices through software programming, and has strong portability. Finally, it is worth mentioning that our work is the first we know to consider combining the parameter transmission process with the wireless channel coding work, using communication optimization learning. Simulation experiment shows that the WFL architecture with wireless channel coding embedded is more anti-interference compared with the WFL without wireless channel coding. In addition, we also prove that the GPU-CPU hybrid acceleration architecture can greatly improve the wireless channel coding efficiency, compared with pure CPU acceleration method and pure GPU acceleration method.

The remainder of this article is organized as follows. In Section II we will introduce recent related works about WFL optimization and GPU acceleration methods. Section III we will introduce the system model of anti-interference WFL architecture embedded with wireless channel coding. Section IV shows the algorithm we designed to implement the anti-interference WFL architecture. In Section V we designed experiment to simulate a real wireless environment to test the anti-interference and acceleration ability of the anti-interference WFL architecture. Finally, Section VI concludes this paper and points out future work.

II. RELATED WORKS

In recent years, research on WFL has received a lot of attention. Researchers have tried to optimize WFL from different aspects. Under the condition of limited wireless network resources [31] and client energy resources [23], [32] participating in WFL local training, Zhou proposed a bandwidth allocation algorithm with low energy consumption [33], which enables clients to engage in learning more sustainably. Xu and Wang proposed to intelligently select clients participating in WFL local learning based on energy consumption from the long-term perspective of learning as a whole [34], not limited to learning rounds. By optimizing the joint client selection and bandwidth allocation under long-term client energy constraints, the long-term performance of wireless federated learning is guaranteed in complex network environments. Based on the expected convergence speed of the WFL algorithm, Chen et al. quantifies the influence of wireless factors on WFL, and under the given user selection and uplink resource block (RB) allocation scheme [35], derives the optimal transmit power for each user, thereby optimizing user selection and uplink RB allocation to minimize the WFL loss function. The above studies all aim to optimize the resource allocation of clients participating in WFL local training, and do not consider the influence of the instability of the wireless channel on the transmission of model parameters between the client and the server in complex wireless environment.

In the wireless network environment, wireless channel coding is the guarantee of fast and correct data transmission. Afshin Abdi compresses the stochastic gradient (SG) transmitted in WFL based on Random Linear Coding (RLC) to reduce communication overhead and accelerate convergence [36]. Amiri and Gündüz proposes a new simulation scheme called A-Distributed Stochastic Gradient Descent (A-DSGD) by exploiting the additional properties of wireless MACs for over-the-air gradient computation [37]. In A-DSGD, these devices first thin out their gradient estimates and then project them into a low-dimensional space imposed by the available channel bandwidth. These predictions are sent directly through the MAC without using any digital codes. Because A-DSGD utilizes the limited bandwidth more efficiently and the natural alignment of gradient estimates across channels, it converges faster. Although the above researches optimize the communication process of model

parameter transmission between client and server in wireless environment, they do not consider the influence of wireless transmission error on model parameters. We simulate the communication process of model parameter transmission between client and server in WFL, and embedded wireless channel coding inside, under different bit error rates, the accuracy of model parameter transmission is ensured, so as to ensure the accuracy and reduce the loss of WFL.

With the development of hardware technology, the computer has more and more CPU calculation and operation units, which makes the program run faster and faster. However, in some scenarios, compared to the CPU, the number of computing units of the GPU is much higher, even if the computing power of the GPU is not as good as CPU, it can perform thousands of calculations at the same time, which is more efficient. In the communication process of WFL, GPU acceleration can greatly accelerate the process of encoding and decoding model parameters, which improves the efficiency of learning. Ling and Cautereels studied GPU as a digital signal processing accelerator for cloud RAN [38], which improves the throughput of data and accelerates the decoding process of LDPC. Chance Tarver works by changing the parallelization strategy of mapping GPU cores to blocks, using many GPU cores to quickly compute a codeword for low latency, or using cores to process multiple codewords simultaneously for targeted high-throughput applications [39]. The above researches are based on GPU hardware optimization to speed up the encoding and decoding process. By analyzing the operation characteristics of wireless channel coding, we propose a GPU-CPU hybrid architecture at the software level, which accelerates the encoding and decoding process of model parameter communication between the WFL client and server.

III. SYSTEM MODEL

In order to reduce the influence of wireless channel instability on the communication process of WFL and improve the anti-interference ability of WFL, we designed an WFL architecture with wireless communication coding embedding in, and designed a software-defined GPU-CPU hybrid acceleration architecture to accelerate the encoding and decoding process between the client and the aggregation server. Next we will introduce our designed anti-interference WFL architecture embedded with wireless communication coding, and a software-defined GPU-CPU hybrid architecture for accelerating wireless channel coding.

The proposed framework of anti-interference WFL architecture embedded with wireless channel coding is shown in Fig. 2, which consists of three planes as listed below.

A. ANTI-INTERFERENCE WIRELESS FEDERATED LEARNING PLANE

This plane is formed by connecting an aggregation server with many edge devices with limited wireless communication resources, such as smartphones, tablets, unmanned autonomous, drones and industrial equipment, etc. network

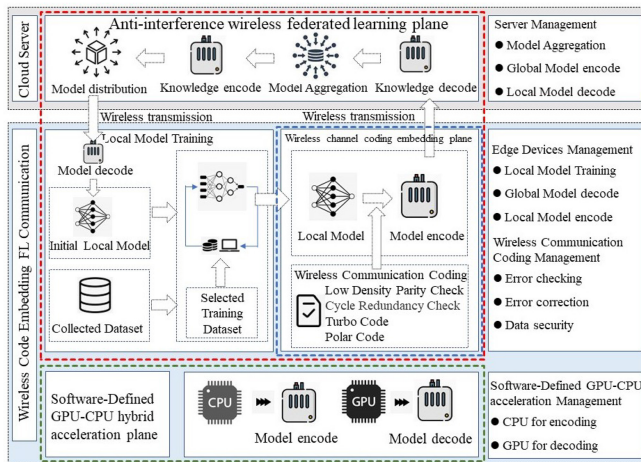


FIGURE 2. Anti-interference wireless federated learning framework.

composition. In order to handle the high-frequency communication between aggregation servers and edge devices, it is necessary to ensure fast and stable wireless communication. However, the instability of the wireless channel limits the communication process between the aggregation server and the edge device. If there is no error correction method, when error happens in the wireless transmission, or when it is attacked by byzantine attacks, data poisoning, and model inference, the model trained by WFL will be unreliable, especially in an open wireless network environment.

B. WIRELESS CHANNEL CODING EMBEDDING PLANE

The wireless channel coding embedding plane is composed of the aggregation server and the edge client. Each end is further divided into a wireless channel coding module and a wireless channel decoding module. The wireless channel coding module uses the aggregation model parameters of the aggregation server and the local model parameters of the edge client as information symbols, and encodes the block model parameters, so that the wireless transmitted data has error detection and even error correction capabilities. The wireless channel decoding module code decodes the local model parameters received by the aggregation server and the global model parameters received by the edge client. Through wireless channel coding, even if it is subjected to unstable wireless transmission, after error detection and error correction of the parity bit, the finally decoded model parameters are close to or even the same as those before coding, so the performance of WFL can be guaranteed.

Commonly used wireless channel codes mainly include Low Density Parity Code (LDPC), Cycle Redundancy Check (CRC), Turbo Code and Polar Code.

C. CPU-GPU HYBRID ACCELERATION ARCHITECTURE PLANE

Today, many studies have realized hardware-accelerated wireless channel coding, such as using a graphics processing unit (GPU) as an alternative to FPGA and ASIC decoders

to accelerate the decoding process of LDPC. According to the characteristics of wireless channel coding and decoding in mathematical calculation, we defined a CPU-GPU hybrid acceleration architecture from the software aspect, and dynamically accelerate the encoding and decoding process of model parameters, which is simpler and more portable than hardware acceleration.

The wireless channel coding embedded anti-interference wireless federated learning framework we designed, while retaining the characteristics of FL data not leaving the local, through wireless channel coding, further guarantees the communication security between the aggregation server and the edge device. First, the aggregation server encodes the initial model parameters and sends them to all edge devices participating in the learning through wireless transmission. The edge device decodes the received model parameters, initializes the local model, and performs training according to the local data. After the local training round ends, each edge device encodes its model parameters and returns them to the aggregation server through wireless transmission. The aggregation server decodes all model parameters, and aggregates all model parameters into parameters of a global model through an algorithm. Repeating this process until the global model sent to the edge devices reaches convergence, and the entire WFL task is completed. In this process, the GPU-CPU hybrid acceleration architecture we designed dynamically optimizes the communication process between the aggregation server and the edge clients, and reasonably allocates CPU acceleration and GPU acceleration to the encoding and decoding processes. On the premise of reducing communication overhead and saving computing resources, the accuracy of model parameter transmission is guaranteed.

In order to realize this anti-interference WFL architecture, next we will show the specific implementation algorithm of our designed wireless channel coding embedded in wireless federated learning and software-defined GPU-CPU hybrid acceleration architecture.

IV. ALGORITHM

The specific implementation algorithm of wireless channel coding embedded in WFL and the software-defined GPU-CPU hybrid acceleration architecture is proposed in this section.

A. WIRELESS CHANNEL CODING EMBEDDING IN WIRELESS FEDERATED LEARNING

In order to realize the anti-interference WFL architecture, we embedded wireless channel coding in both the edge client side and aggregation server side of WFL. For the edge client, decoding is performed when receiving global model parameters from the aggregation server, and encoding is performed when sending local model parameters to the aggregation server. For the aggregation server, encoding is performed every time the aggregated global model parameters are sent

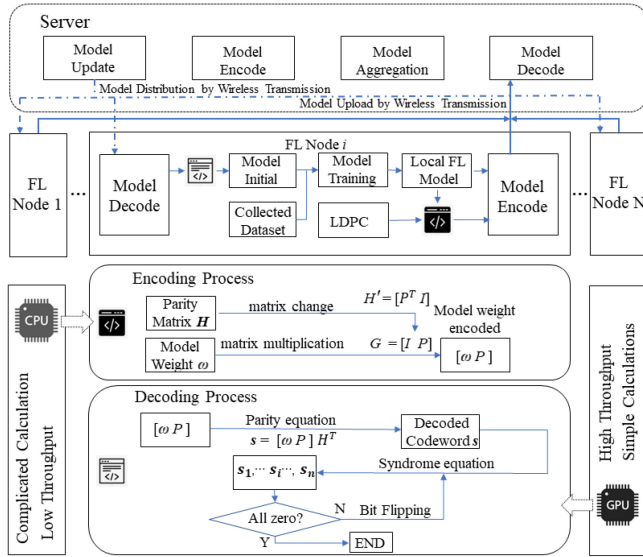


FIGURE 3. Wireless coding embedded wireless federated learning algorithm.

to the edge client, and decoding is performed when receiving an update of the local model parameters uploaded by the edge client.

As shown in Figure 3, this framework consists of one cloud server and N edge clients. Each user k has a local dataset D_k . For each local dataset D_k , $D_k = \{x_{ki}, y_{ki}\}_{i=1}^{D_k}$, where $\{x_{ki}\} \in \mathbb{R}^d$ is the input vector of user k , and y_{ki} is its corresponding output (we currently only consider single-output federated learning algorithm). We define the vector ω_i as the local model parameter trained by x_{ki} and vector g as the global model parameter aggregated by ω_i . For each client i , their task is to train the optimal parameter ω_i to minimize the loss function of the local model. And the entire FL process translates into solving the following optimization equation:

$$F_k(\omega_i, x_{k1}, y_{k1}, \dots, x_{kD_k}, y_{kD_k}) = \frac{1}{D_k} \sum_{i=1}^{D_k} f(\omega_i, x_{ki}, y_{ki}) \quad (1)$$

$$g = \sum_{k=1}^K \frac{n_k}{n} \omega_k \quad (2)$$

where $f(\omega_i, x_{ki}, y_{ki})$ is the loss function describing the performance of the FL model obtained by the input vector x_{ki} and output vector y_{ki} . Among them, g is the weight parameter of the aggregation model obtained based on the model parameter ω_i uploaded by the edge client to the aggregation server according to the federated average algorithm. As with any machine learning, the most important factor affecting learning performance is the weight parameters of the machine learning model. For the research in this paper, it is the weight parameter g of the aggregation model. Whether g can be transmitted quickly and accurately is directly related to the performance of WFL. Therefore,

we embedded wireless channel coding in the WFL model, to design an anti-interference WFL architecture with error checking and error correction, and perform wireless channel coding on the weight parameter g of the aggregation model of wireless transmission so that g can be transmitted accurately, ensuring performance of WFL.

B. WIRELESS ENCODING

Wireless channel coding is a process of matrix transformation and matrix operation. First, we need to divide the global model parameters g into information symbols for wireless transmission, and transform it into a binary matrix as the basis of operation. The global model parameter g contains the weight parameter of each node of the model whose type is a floating-point number. We need to convert the weight parameter g of the floating-point number into a binary matrix for encoding operation. Dividing the floating-point number weight parameter g into an integer part F and a fractional part f , and the subscript represents the number of digits. We convert the integer and fractional parts to binary and store them in the matrices $A = [a_1, \dots, a_i, \dots, a_n]$ and $B = [b_1, \dots, b_i, \dots, b_n]$, and then merge the matrices A and B separately:

$$[a_i] = \begin{cases} [0]_b, (F_n \setminus 2 = 0) \\ [1]_b, (F_n \setminus 2 = 1) \end{cases} \quad i \in [1, n] \quad (3)$$

$$[b_i] = \begin{cases} [0]_b, (f_n \times 2 < 0) \\ [1]_b, (f_n \times 2 \geq 1) \end{cases} \quad i \in [1, n] \quad (4)$$

where F_n is the integer part of F_{n-1} divided by 2, written as $F_n = \lfloor F_{n-1}/2 \rfloor$, and f_n is the fractional part of f_{n-1} multiplied by 2, written as $f_n = f_{n-1} \times 2 - \lfloor f_{n-1} \times 2 \rfloor$.

Because the storage method of floating-point numbers in the computer follows the IEEE 754 floating-point number counting standard, it can be expressed as:

$$V = (-1)^S \times M \times 2^E \quad (5)$$

Therefore, we also need to convert the matrix after the matrix A and B are combined into a representation form composed of sign bits S , exponent codes E , and mantissa bits M . According to the positive or negative of the weight, the value of the sign bit S is taken, and the positive number is 0, and the negative number is 1. After moving the decimal point, the value of the exponent E is equal to the number of digits moved after the representation is reproduced in scientific notation. Take the mantissa M according to the precision to get the value of the mantissa bits. Finally, a binary matrix of weight parameters $X = [SME]$ is obtained.

Secondly, we need to construct the parity matrix H applicable to the global model information symbol binary matrix. Parity matrix H is a sparse matrix composed of only 0 and 1. According to the shape of the segmented global model information symbol binary matrix, the shape of the parity matrix H is determined. As follows, on the basis of confirming the shape of the parity matrix H , we define the $m \times n$

mother matrix $M(H)$ of the parity matrix H , and use the 0 and 1 in $M(H)$ to be all 0 of $L \times L$. The parity matrix H can be obtained by replacing the sub-matrix with the cyclic sub-matrix $P^{a_{ij}}$ of $L \times L$. We define cyclic shift sub-matrix P and parent matrix $M(H)$ as:

$$P = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \quad M(H) = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ 0 & 1 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \dots & 0 & 1 \\ 1 & 0 & \dots & 1 & 0 \end{bmatrix} \quad (6)$$

Replacing each 1 in the above formula with an $L \times L$ cyclic shift sub-matrix $P^{a_{ij}}$, and replace each 0 with an $L \times L$ all-zero matrix to obtain a $mL \times nL$ H matrix:

$$M(H) = \begin{bmatrix} P^{a_{11}} & 0 & \dots & 0 & P^{a_{1n}} \\ 0 & P^{a_{21}} & \dots & P^{a_{2(n-1)}} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & P^{a_{(m-1)2}} & \dots & 0 & P^{a_{(m-1)n}} \\ P^{a_{m1}} & 0 & \dots & P^{a_{m(n-1)}} & 0 \end{bmatrix} \quad (7)$$

where a_{ij} is the shift term. The unit matrix is cyclically shifted to the right by a to obtain the cyclic sub-matrix $P^{a_{ij}}$. When storing the H matrix, we only need to store the value of each a_{ij} in the above formula, instead of storing the position of each 1.

As long as $M(H)$ is expanded, we get the constructed parity matrix H . In order to perform matrix operations with the global model information symbol binary matrix, we need to change the parity matrix H into a generator matrix G through matrix transformation. First, we perform row-column transformation on the H matrix, and change the right half of the matrix into an identity matrix to obtain the intermediate matrix $H' = [P^T, I]$. Convert the left half of the middle matrix H' to rank and exchange it with the identity matrix of the right half to obtain the generator matrix $G = [I, P]$. The specific process is as follows:

$$H' = \begin{bmatrix} P^{a_{11}} & \dots & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & \dots & P^{a_{21}} & 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & P^{a_{(m-1)2}} & 0 & 0 & \dots & 1 & 0 \\ P^{a_{m1}} & \dots & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad (8)$$

$$G = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & P^{a_{11}} & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & P^{a_{n2}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & P^{a_{(m-1)n}} \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & P^{a_{mn}} \end{bmatrix} \quad (9)$$

After obtaining the generator matrix G , we do $r = X \times G$ to get the encoded information r .

$$[SME] \times \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & P^{a_{11}} & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & P^{a_{2n}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & P^{a_{(m-1)n}} \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & P^{a_{mn}} \end{bmatrix} \quad (10)$$

Obviously, because the left side of the generator matrix G is an identity matrix, the left side of the encoded one-dimensional matrix r we get is the same as the global model information symbol binary matrix X , and the check bit is behind it. This is why G is called a generator matrix. On the basis of retaining the original information, it generates a check digit for verification, which provides a basis for error detection and correction in subsequent decoding.

C. WIRELESS DECODING

In wireless signal transmission, noise, pulses caused by alternating current or lightning, transmission equipment failure and other factors will cause bit errors, causing the transmitted binary signal to send bit flips (from 0 to 1 or from 1 to 0). Therefore, we add wireless channel coding and add check bits on the basis of the original information symbols. If a bit-flip error occurs in the wireless transmission of the global model parameter g , it can check the position where the bit flip occurs and correct it. We embed a bit-flip decoding method in the client, and the client performs decoding after receiving the encoded global model parameter g from the server.

We set when clients received the original information symbols with parity bits like $r = [r_1 \dots r_i \dots r_n]$.

Now the parity matrix H will do its job. We do matrix multiplication $r \times H^T$, then we can get multiple syndromes s .

$$s_i = \sum_{k=1}^n p_{ik} \cdot r_k \quad (11)$$

When we compute the syndrome s , we use modular two addition. When the results of the syndrome s_i are all 0, it means that the wireless transmission is correct. When the result of the syndrome s_i is 1, it means that the coded information matrix r involved in the calculation of the syndrome s_i may have bit errors during wireless transmission. At this time, in the syndrome s_i whose calculation result is 1, the coded information matrix r_i that participated in the most calculation times has the greatest possibility of error in wireless transmission, so we perform bit flipping on the r_i that participated in the most calculation times, and then re-calculate the syndrome s until the result of the syndrome s is all 0, and we have obtained the correct binary information matrix r of the global model parameters. After re-converting the binary information matrix r into floating-point numbers and importing them into each node of the local model, the local model update of the clients are completed, and a new round of local training can be started.

Algorithm 1: LDPC Wireless Channel Coding Algorithm

Input: WFL model parameter i , code length N , code rate r , parity matrix H , generator matrix G .

Output: global model parameters g .

```

1 for convergence == 0 do
    // Local model parameters obtained
    // by local training.
2 local_model[] = local_training(i, local_data);
    // Turning local model parameter
    // into bit matrix called
    // information matrix.
3 info_matrix[] = Binary(local_model, N, r);
    // Encoding
4 Encoded[] = LDPC_Encoder(info_matrix, G);
    // Wireless transmission
    // Decoding
5 local_model = LDPC_Decoder(Encoded);
    // Cloud server Aggregation
6 global_model_parameters g =
    Aggregation(local_model);
    // Judge whether convergence
7 if convergence == 1 then
8     Output g;
9     Break; // Convergence
10 else if convergence == 0 then
11     i = g; // Next round
12 End
    
```

In the next section, we will design experiments to test our designed software-defined GPU-CPU empowered WFL with Embedding LDPC Communication Coding in terms of anti-interference and acceleration ability.

V. EXPERIMENT

A. EXPERIMENT SETUP

Our experiments are deployed on the hardware with 2.90-GHz Intel Xeon Gold 6326 CPU, 256-G RAM, and 8-T disk. The operating system is Linux Ubuntu 20.04 LTS and the simulations are conducted on Python 3.8. The parameters of the our experiments are presented as follows.

We set the number of clients $m = 10$, the global iteration round $G = 100$ (that is, the number of communication iterations between the server and the client), the local iteration round $L = 5$, the number of clients participating in training in each round $k = 5$, the number of samples for each round of local training is $s = 32$, and the learning rate $lr = 0.01$. The frame of encoder code length $N = 648$ and the code rate $r = 1/2$.

Obviously, different from the ideal conditions, in the actual environment, the wireless transmission will be affected by various factors. Errors occur during the transmission of model parameters, which affects the convergence of

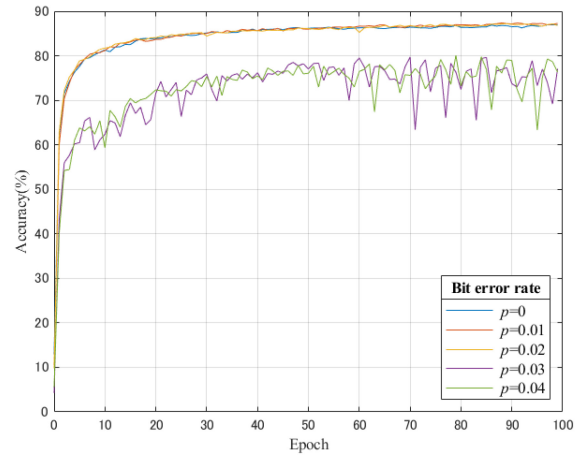


FIGURE 4. Model accuracy curves with wireless coding under different bit error rates.

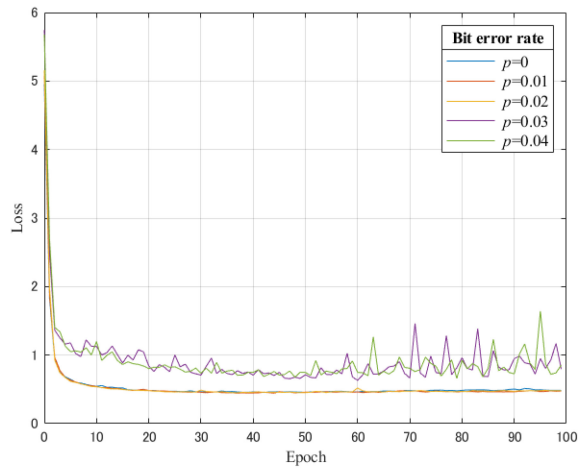


FIGURE 5. Model loss curves with wireless coding under different bit error rates.

the model and affects the accuracy and loss of learning. Therefore, we set the parameter p of the bit error rate to indicate the degree of bit errors that occur during wireless transmission. It represents the probability that a bit flip may occur in the binary information during transmission.

Next, we will conduct experiments on the anti-interference performance and encoding-decoding acceleration capabilities of our proposed framework.

B. EXPERIMENT RESULT

1) Anti-interference experiment: The convergence of our proposed wireless channel coding embedded anti-interference WFL architecture over $p = 0.01, 0.02, 0.03, 0.04$ and traditional federated learning without code embedded are shown as Fig. 4 and Fig. 5.

From Fig. 4, We can see that along the number of epoch increases, the Accuracy of our proposed Code-embedded anti-interference WFL architecture gradually increases and converges to a high stable value. Specially, We can see that when the bit error rate is low, the WFL architecture with wireless channel coding embedding has similar performance

TABLE 1. Performance comparison between coded and non-coded wireless federated learning.

Bit error rate p	0.005	0.006	0.007	0.008	0.009	0.01	0.02	0.03	0.04
Accuracy with wireless coding%	86.64	86.62	86.58	86.51	86.27	87.27	86.81	77.08	76.31
Loss with wireless coding	0.493	0.492	0.508	0.512	0.492	0.474	0.478	0.796	0.854
Accuracy without wireless coding%	86.70	76.43	75.73	73.29	75.42	76.26	num	num	num
Loss without wireless coding	0.487	0.824	0.817	0.786	0.813	0.782	num	num	num

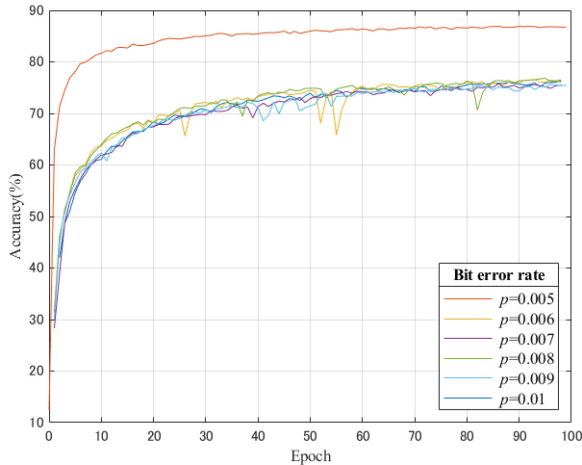


FIGURE 6. Model accuracy curves without wireless coding under different bit error rates.

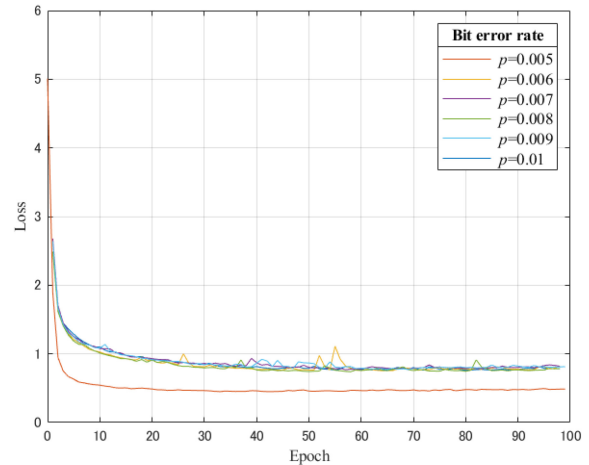


FIGURE 7. Model loss curves without wireless coding under different bit error rates.

to traditional FL. Therefore, the impact of embedding wireless channel coding on FL performance is negligible. As the bit error rate increases, the time it takes for the wireless channel coding embedded WFL architecture to reach convergence increases and fluctuates slightly. The same effect can be seen from Fig. 5. Fig. 5 is the change curve of the loss and epoch rounds of the anti-interference WFL architecture and traditional WFL under the same conditions. When the bit error rate is low, the loss of anti-interference WFL is almost the same as that of traditional WFL under ideal conditions. The loss for anti-interference WFL increases slightly as the bit error rate increases. Therefore, from the two aspects of model accuracy and loss, the anti-interference WFL architecture will not have a negative effect on the convergence of the model. Therefore, we designed an experiment to simulate the WFL embedded with wireless channel coding and ordinary WFL without wireless channel coding in the actual environment, and compared the performance of the two architecture under the influence of bit errors. As shown in Fig. 6 and Fig. 7, the change curve of model accuracy and loss with epoch rounds.

We plot the accuracy, loss and bit error rate p of the environment where the anti-interference WFL embedded with wireless channel coding embedded and ordinary WFL without wireless channel coding embedded is plotted into the following table.

From Table 1, we can find that before the bit error rate p reaches 0.3, the performance of wireless federated learning embedded in anti-interference wireless channel coding is stable at the level of wireless federated learning under

ideal conditions. When the bit error rate p reaches 0.3, There is a slight decrease in accuracy and loss. In contrast, the common federated learning without code embedding, only when the bit error rate p is between 0.001 and 0.005, the learning performance is close to the level of wireless federated learning under ideal conditions. When the bit error rate p reaches 0.01, the learning performance drops greatly, and the learning performance is not even as good as the performance of the anti-interference federated learning framework in the environment of 10 times bit error rate (i.e., $p = 0.01$). When the bit error rate p exceeds 0.01, because there are too many bit errors in the transmitted model parameters, the output of model accuracy and loss are num, and the learning cannot proceed normally. Therefore, the embedding of wireless channel coding greatly improves the anti-interference ability of the wireless federated learning framework, enabling the learning to receive accurate model parameters in a complex environment for model update and training, ensuring the wireless federated learning performance.

2) GPU-CPU hybrid acceleration experiment: To reduce the impact of encoding embedding on learning efficiency, we designed a software-defined GPU-CPU hybrid acceleration architecture.

According to the characteristics of wireless channel coding, we divide the acceleration into two parts. The first part is the calculation of converting the model parameters of the floating point type into binary matrices, and the calculation of generating the parity matrix H through matrix transformation. The second part is the multiplication of binary matrix and generator matrix and the syndrome calculation of bit flip

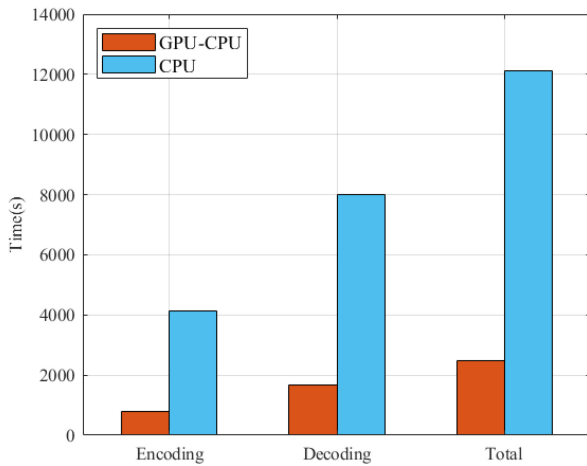


FIGURE 8. Encoding and decoding time comparison of different acceleration methods.

decoding. The first part is mainly characterized by small calculations but high computational complexity, which is suitable for CPU acceleration. The second part is a large amount of calculation, high throughput, repetitive and simple calculation content, suitable for GPU acceleration. Under the condition of the same bit error rate p , we accelerate the coding process with pure CPU acceleration and GPU-CPU hybrid acceleration respectively to obtain a comparison chart of the time spent by each acceleration method when the model reaches convergence.

Obviously, compared with the GPU-CPU hybrid acceleration method, CPU acceleration method spent more time to reach convergence. The convergence speed of the GPU-CPU hybrid acceleration method is 5 times that of the pure CPU acceleration method.

VI. CONCLUSION

In this article, we proposed a software-defined GPU-CPU empowered efficient wireless federated learning architecture with LDPC communication coding embedded, which integrates FL and wireless coding to enhance the performance of wireless federated learning while guaranteeing information privacy between the clients who participate in FL. In this proposed framework, we first discuss the influence of the framework itself on learning performance from two aspects of model accuracy and loss. Next, we designed a simulation experiment to simulate the accuracy and loss of our proposed framework and the common framework under different bit error rates p . In addition, we compared pure CPU acceleration and GPU-CPU hybrid acceleration to improve the encoding speed. Finally, the experimental results show that the anti-interference wireless federated learning framework we proposed can ensure the accurate transmission of model parameters in complex environments. Moreover, the GPU-CPU acceleration module defined at the software level improves the speed of encoding calculation in the framework, improves the learning efficiency of the framework, and solves the problem of learning efficiency decline caused

by encoding embedding. In future work, we will study the impact of parity matrix on communication optimization when using LDPC codes to optimize WFL communication under different WFL tasks. In addition, we will compare different wireless channel codes to optimize the communication process of WFL based on the rules and characteristics of model parameters.

ACKNOWLEDGMENT

Therefore, the authors gratefully acknowledge technical and financial support from the Ministry of Education and King Abdulaziz University, Jeddah, Saudi Arabia.

REFERENCES

- [1] W. Bao, C. Wu, S. Guleng, J. Zhang, K.-L. A. Yau, and Y. Ji, "Edge computing-based joint client selection and networking scheme for federated learning in vehicular IoT," *China Commun.*, vol. 18, no. 6, pp. 39–52, Jun. 2021, doi: [10.23919/JCC.2021.06.004](https://doi.org/10.23919/JCC.2021.06.004).
- [2] M. Karunakaran, J. Balusamy, and K. Selvaraj, "Machine learning models based mental health detection," in *Proc. 3rd Int. Conf. Intell. Comput. Instrum. Control Technol. (ICICICT)*, 2022, pp. 835–842, doi: [10.1109/ICICICT54557.2022.9917622](https://doi.org/10.1109/ICICICT54557.2022.9917622).
- [3] L. Feng, "Intelligent logistics and distribution system based on Internet of Things," in *Proc. IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. (IMCEC)*, 2016, pp. 228–231, doi: [10.1109/IMCEC.2016.7867206](https://doi.org/10.1109/IMCEC.2016.7867206).
- [4] M. D'souza, N. Wilfred, R. Pereira, T. Rayen, and A. Telgote, "Home automation using Internet of Things," in *Proc. Int. Conf. Energy Commun. Data Anal. Soft Comput. (ICECDS)*, 2017, pp. 559–561, doi: [10.1109/ICECDS.2017.8389498](https://doi.org/10.1109/ICECDS.2017.8389498).
- [5] X. Lu and L.-L. Cheng, "Target tracking in Internet of Things based on sensing subtraction and compressed sensing," in *Proc. 11th Int. Comput. Conf. Wavelet Active Media Technol. Inf. Process. (ICCWAMTIP)*, 2014, pp. 192–195, doi: [10.1109/ICCWAMTIP.2014.7073388](https://doi.org/10.1109/ICCWAMTIP.2014.7073388).
- [6] H. Hui, C. Zhou, S. Xu, and F. Lin, "A novel secure data transmission scheme in industrial Internet of Things," *China Commun.*, vol. 17, no. 1, pp. 73–88, Jan. 2020, doi: [10.23919/JCC.2020.01.006](https://doi.org/10.23919/JCC.2020.01.006).
- [7] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016, doi: [10.1109/JIOT.2016.2612119](https://doi.org/10.1109/JIOT.2016.2612119).
- [8] D. P. Moya Osorio et al., "Towards 6G-enabled Internet of Vehicles: Security and privacy," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 82–105, 2022, doi: [10.1109/OJCOMS.2022.3143098](https://doi.org/10.1109/OJCOMS.2022.3143098).
- [9] N. Nomikos, P. K. Gkonis, P. S. Bithas, and P. Trakadas, "A survey on UAV-aided maritime communications: Deployment considerations, applications, and future challenges," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 56–78, 2023, doi: [10.1109/OJCOMS.2022.3225590](https://doi.org/10.1109/OJCOMS.2022.3225590).
- [10] Y. Huang et al., "Connectivity guarantee within UAV cluster: A graph coalition formation game approach," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 79–90, 2023, doi: [10.1109/OJCOMS.2022.3232600](https://doi.org/10.1109/OJCOMS.2022.3232600).
- [11] M. S. Uddin, M. Gianni, and A. Lab, "Long range robot teleoperation system based on Internet of Things," in *Proc. 2nd Int. Conf. Comput. Commun. Syst. (ICCCS)*, 2017, pp. 163–167, doi: [10.1109/CCOMS.2017.8075288](https://doi.org/10.1109/CCOMS.2017.8075288).
- [12] F. Hu, D. Xie, and S. Shen, "On the application of the Internet of Things in the field of medical and health care," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber Phys. Social Comput.*, 2013, pp. 2053–2058, doi: [10.1109/GreenCom-iThings-CPSCom.2013.384](https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.384).
- [13] L. Li, Y. Fan, and K.-Y. Lin, "A Survey on federated learning," in *Proc. IEEE 16th Int. Conf. Control Autom. (ICCA)*, 2020, pp. 791–796, doi: [10.1109/ICCA51439.2020.9264412](https://doi.org/10.1109/ICCA51439.2020.9264412).
- [14] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021, doi: [10.1109/TWC.2020.3024629](https://doi.org/10.1109/TWC.2020.3024629).

[15] X. Lin, J. Wu, J. Li, X. Zheng, and G. Li, "Friend-as-learner: Socially-driven trustworthy and efficient wireless federated edge learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 269–283, Jan. 2023, doi: [10.1109/TMC.2021.3074816](https://doi.org/10.1109/TMC.2021.3074816).

[16] G. Shi, S. Guo, J. Ye, N. Saeed, and S. Dang, "Multiple parallel federated learning via over-the-air computation," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1252–1264, 2022, doi: [10.1109/OJCOMS.2022.3194821](https://doi.org/10.1109/OJCOMS.2022.3194821).

[17] I. Donevski, N. Babu, J. J. Nielsen, P. Popovski, and W. Saad, "Federated learning with a drone orchestrator: Path planning for minimized staleness," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1000–1014, 2021, doi: [10.1109/OJCOMS.2021.3072003](https://doi.org/10.1109/OJCOMS.2021.3072003).

[18] G. Ma, C. Zhang, S. Dang, and B. Shihada, "Paving the way for distributed artificial intelligence over the air," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1103–1118, 2022, doi: [10.1109/OJCOMS.2022.3188051](https://doi.org/10.1109/OJCOMS.2022.3188051).

[19] H. Zhang, H. Tian, M. Dong, K. Ota, and J. Jia, "FedPCC: Parallelism of communication and computation for federated learning in wireless networks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 6, pp. 1368–1377, Dec. 2022, doi: [10.1109/TETCI.2022.3170471](https://doi.org/10.1109/TETCI.2022.3170471).

[20] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated learning over wireless IoT networks with optimized communication and resources," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16592–16605, Sep. 2022, doi: [10.1109/JIOT.2022.3151193](https://doi.org/10.1109/JIOT.2022.3151193).

[21] Y. Fu and X. Wang, "Traffic prediction-enabled energy-efficient dynamic computing resource allocation in cran based on deep learning," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 159–175, 2022, doi: [10.1109/OJCOMS.2022.3146886](https://doi.org/10.1109/OJCOMS.2022.3146886).

[22] C. Sun, S. Wu, and T. Cui, "User selection for federated learning in a wireless environment: A process to minimize the negative effect of training data correlation and improve performance," *IEEE Veh. Technol. Mag.*, vol. 17, no. 3, pp. 26–33, Sep. 2022, doi: [10.1109/MVT.2022.3153274](https://doi.org/10.1109/MVT.2022.3153274).

[23] T. Zhou, X. Li, C. Pan, M. Zhou, and Y. Yao, "Multi-server federated edge learning for low power consumption wireless resource allocation based on user QoE," *J. Commun. Netw.*, vol. 23, no. 6, pp. 463–472, Dec. 2021, doi: [10.23919/JCN.2021.000040](https://doi.org/10.23919/JCN.2021.000040).

[24] H.-S. Lee, "Device selection and resource allocation for layerwise federated learning in wireless networks," *IEEE Syst. J.*, vol. 16, no. 4, pp. 6441–6444, Dec. 2022, doi: [10.1109/JSYST.2022.3169461](https://doi.org/10.1109/JSYST.2022.3169461).

[25] S. Liu, G. Yu, X. Chen, and M. Bennis, "Joint user association and resource allocation for wireless hierarchical federated learning with IID and non-IID data," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7852–7866, Oct. 2022, doi: [10.1109/TWC.2022.3162595](https://doi.org/10.1109/TWC.2022.3162595).

[26] Q. Pan, J. Wu, X. Zheng, W. Yang, and J. Li, "Differential privacy and IRS empowered intelligent energy harvesting for 6G Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22109–22122, Nov. 2022.

[27] H. Zhang, C. Yang, and B. Dai, "When wireless federated learning meets physical layer security: The fundamental limits," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPs)*, 2022, pp. 1–6, doi: [10.1109/INFOCOMWKSHPs54753.2022.9797923](https://doi.org/10.1109/INFOCOMWKSHPs54753.2022.9797923).

[28] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020, doi: [10.1109/TCOMM.2019.2944169](https://doi.org/10.1109/TCOMM.2019.2944169).

[29] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021, doi: [10.1109/TWC.2020.3025446](https://doi.org/10.1109/TWC.2020.3025446).

[30] G. Li, J. Wu, S. Li, W. Yang, and C. Li, "Multitentacle federated learning over software-defined Industrial Internet of Things against adaptive poisoning attacks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1260–1269, Feb. 2023, doi: [10.1109/TII.2022.3173996](https://doi.org/10.1109/TII.2022.3173996).

[31] Q. Pan, J. Wu, A. K. Bashir, J. Li, W. Yang, and Y. D. Al-Otaibi, "Joint protection of energy security and information privacy for energy harvesting: An incentive federated learning approach," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3473–3483, May 2022, doi: [10.1109/TII.2021.3105492](https://doi.org/10.1109/TII.2021.3105492).

[32] Y. Sun, H. Esaki, and H. Ochiai, "Adaptive intrusion detection in the networking of large-scale LANs with segmented federated learning," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 102–112, 2021, doi: [10.1109/OJCOMS.2020.3044323](https://doi.org/10.1109/OJCOMS.2020.3044323).

[33] T. T. Vu, H. Q. Ngo, M. N. Dao, D. T. Ngo, E. G. Larsson, and T. Le-Ngoc, "Energy-efficient massive MIMO for federated learning: Transmission designs and resource allocations," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 2329–2346, 2022, doi: [10.1109/OJCOMS.2022.3222749](https://doi.org/10.1109/OJCOMS.2022.3222749).

[34] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2021, doi: [10.1109/TWC.2020.3031503](https://doi.org/10.1109/TWC.2020.3031503).

[35] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "Performance optimization of federated learning over wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6, doi: [10.1109/GLOBECOM38437.2019.9013160](https://doi.org/10.1109/GLOBECOM38437.2019.9013160).

[36] A. Abdi, Y. M. Saidutta, and F. Fekri, "Analog compression and communication for federated learning over wireless MAC," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2020, pp. 1–5, doi: [10.1109/SPAWC48557.2020.9154309](https://doi.org/10.1109/SPAWC48557.2020.9154309).

[37] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020, doi: [10.1109/TWC.2020.2974748](https://doi.org/10.1109/TWC.2020.2974748).

[38] J. Ling and P. Cautereels, "Fast LDPC GPU decoder for cloud RAN," *IEEE Embedded Syst. Lett.*, vol. 13, no. 4, pp. 170–173, Dec. 2021, doi: [10.1109/LES.2021.3052714](https://doi.org/10.1109/LES.2021.3052714).

[39] C. Tarver, M. Tonnemacher, H. Chen, J. Zhang, and J. R. Cavallaro, "GPU-based, LDPC decoding for 5G and beyond," *IEEE Open J. Circuits Syst.*, vol. 2, pp. 278–290, Jan. 2021, doi: [10.1109/OJCS.2020.3042448](https://doi.org/10.1109/OJCS.2020.3042448).



ZIHONG LI (Graduate Student Member, IEEE) received the B.S. degree from the School of Automation, Beijing Institute of Technology, Beijing, China, in 2021. He is currently pursuing the M.S. degree with the Graduate School of Information, Production and System, Waseda University, Fukuoka, Japan. His research interests are focusing on wireless federated learning and wireless channel coding.



YANG HONG (Graduate Student Member, IEEE) received the B.S. degree from the School of Cyber Science and Engineering, Sichuan University, Chendu, China, in 2022. He is currently pursuing the M.S. degree with the Graduate School of Information, Production and System, Waseda University, Fukuoka, Japan. His research interests are focusing on digital twin and automated vehicles.



ALI KASHIF BASHIR (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Korea University, South Korea. He is the Leader with the Department of Computing and Mathematics, Manchester Metropolitan University, U.K., and also an Adjunct Professor with the Woxsen School of Business, Woxsen University, India. He is a member of IEEE Industrial Electronic Society and ACM and the Distinguished Speaker of ACM. He has authored over 200 research articles; and received over three million USD funding as PI and Co-PI from research bodies of South Korea, Japan, EU, U.K., and Middle East. His research interests include Internet of Things, wireless networks, distributed systems, network/cyber security, network function virtualization, and machine learning. He is serving as the Editor-in-Chief of the IEEE FUTURE DIRECTIONS NEWSLETTER. He is also serving as an Area Editor of KSII Transactions on Internet and Information Systems; an Associate Editor of *IEEE Internet of Things Magazine*, *IEEE ACCESS*, *Peer J Computer Science*, *IET Quantum Computing*, and *Journal of Plant Disease and Protection*. He is leading many conferences as a chair (program, publicity, and track) and had organized workshops in flagship conferences like IEEE Infocom, IEEE Globecom, and IEEE Mobicom.



YASSER D. AL-OTAIBI (Senior Member, IEEE) received the B.Sc. degree in information technology, the M.Sc. degree in information systems, the Graduate Diploma degree in research studies, and the Ph.D. degree in information systems from Griffith University, Brisbane, Australia, in 2010, 2012, 2013, and 2018, respectively. He is an Associate Professor with the Department of Information Systems, Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include IT adoption and acceptance, wireless sensor networks, computer security, and the Internet of Things. He is the recipient of the 2012 and 2013 Griffith Awards for academic excellence during his graduate studies at Griffith University.



JUN WU (Senior Member, IEEE) received the Ph.D. degree in information and telecommunication studies from Waseda University, Japan, in 2011, where he is currently a Professor with the Graduate School of Information, Production and Systems. He is the Chair of IEEE P21451-1-5 Standard Working Group for Internet of Things. He is the author or the coauthor of more than 200 peer-reviewed journal/conference papers within the above-mentioned topics. His research interests include the intelligence and security techniques of Internet of Things, edge computing, big data, and 5G/6G. His publications have received a few distinctions, which includes the Best Paper Award of IEEE Transactions on Emerging Topics in Computing, in 2020, the Best Paper Award of International Conference on Telecommunications and Signal Process in 2019, the Best Conference Paper Award of the IEEE ComSoc Technical Committee on Communications Systems Integration and Modeling in 2018. He has served as the Track Chair for VTC 2019 and VTC 2020 and a TPC Member of more than ten international conferences, including ICC and GLOBECOM. He serves as an Associate Editor for the IEEE SYSTEMS JOURNAL and IEEE NETWORKING LETTERS. He has served as the Guest Editor for the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION, IEEE SENSORS JOURNAL, *Sensors*, and *Frontiers of Information Technology & Electronic Engineering*.