DOI. No. 10.13052/2794-7254.008

Federated Learning Enhancement Through Transfer and Continual Learning Integration: Analyzing Effects of Different Levels of Dirichlet Distribution

Boyuan Zhang and Mohammad Reza Shikh-Bahaei*

Abstract: Machine learning plays a pivotal role in modern technology, driving advancements across various domains such as healthcare, finance, and autonomous systems. Federated Learning (FL) offers a significant advantage over traditional machine learning by enabling decentralized model training without requiring data to be centralized, thereby enhancing privacy and security. With the advent of 6G networks, which promise ultra-reliable low-latency communications (URLLC) and massive machine-type communications (mMTC), FL can be significantly enhanced. 6G's improved bandwidth and latency characteristics will enable more efficient data exchange and model updates, further enhancing the adoption of FL. However, the performance of FL can be significantly affected by data distribution, particularly in non-IID (non-Independent and Identically Distributed) scenarios, where FL tends to perform poorly. This paper proposes a novel approach to enhance FL by integrating Transfer Learning (TL) and Continual Learning (CL), named Integrated Federated Transfer and Continual Learning (IFTCL). TL can extract features from client training samples to benefit subsequent clients, while CL mitigates catastrophic forgetting caused by heterogeneous data across clients. This integration improves FL performance under varying degrees of heterogeneous data distributions simulated by Dirichlet distribution, enhancing accuracy, convergence speed, and reducing communication overhead. The proposed method's feasibility is validated using a publicly available radar recognition dataset.

Keywords: federated learning, transfer learning, continual learning, Dirichlet distribution.

1. Introduction

With the advent of sixth-generation (6G) technology, the significant increase in data volume has brought considerable attention to machine learning, which is expected to play a crucial role in the development of 6G wireless networks. These networks, offering ultra-reliable low-latency communication (URLLC) [1] and extensive machine-type communication (mMTC) [2], are poised to revolutionize technology and convenience. Machine learning influences a wide range of applications that are closely tied to daily activities such as: healthcare [3], finance [4], and transportation [5]. Its ubiquity in everyday applications underscores its significance in the contemporary digital landscape. However, it also gets access to vast amounts of personal data that require protection from unauthorized access and misuse.

These advancements facilitate more efficient data exchange, seamless real-time applications, and improved performance of various digital services. However, as 6G bandwidth and connectivity improve, the accompanying surge in data volume and increased network complexity have made privacy issues more severe [6]. The potential for data breaches and unauthorized access is greater than ever, making privacy protection a top priority. Traditional centralized machine learning methods, which aggregate data from multiple sources into a central repository, pose significant privacy risks.

Federated Learning (FL) addresses these concerns by enabling decentralized model training. In FL, the data remains on local devices, and only model updates are shared with a central server, thus ensuring that personal data is not exposed or transmitted [7]. This method not only preserves privacy but also complies with stringent data protection regulations, making FL a compelling solution for privacy-conscious applications. However, the performance of FL is highly dependent on the distribution of the dataset. When the samples in each client are uniformly distributed across each training client, the training results are generally excellent [8, 9]. However, in real-world scenarios, data is often non-IID (non-Independent and Identically Distributed), meaning that data distributions can vary significantly between clients [10, 11]. This heterogeneity can lead to substantial challenges in model convergence and accuracy [12]. For instance, certain clients may have data that is biased or skewed towards specific classes or features, causing the global model to perform poorly when aggregated from these disparate local models. This imbalance can slow down the convergence rate, reduce overall model accuracy, and increase the communication burden due to the need for more frequent synchronization and updates to achieve acceptable performance.

To address these challenges, this paper proposes a novel approach that combines FL with Transfer Learning (TL) and Continual Learning (CL) to enhance FL under varying degrees of Dirichlet distribution. TL facilitates the extraction of useful features from a set of clients [13], which can then be utilized by subsequent set of clients to form a complete feature extractor, thereby promoting knowledge transfer to remaining clients and improving overall learning efficiency by freezing the feature extractor layers of FL network to reduce communication

Department of Engineering, King's College London, London, United Kingdom
E-mail: boyuan.zhang@kcl.ac.uk; m.sbahaei@kcl.ac.uk
*Corresponding Author
Manuscript received 12 September 2024, accepted 24 September 2024, and ready for publication 21 December 2024.

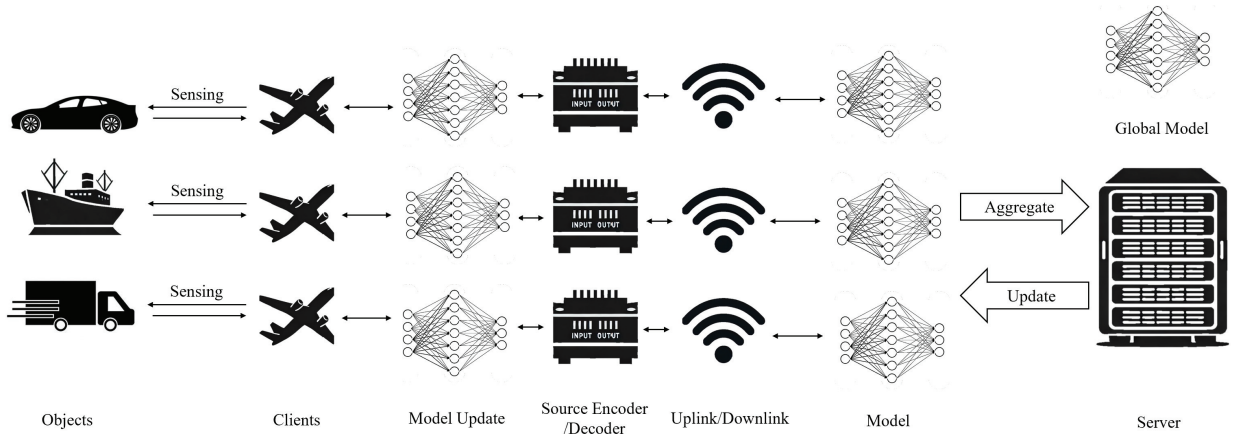

Figure 1.
System model demonstration.

burden [14]. Meanwhile, CL helps to mitigate the issue of catastrophic forgetting [15], which occurs when a model trained on new data overwrites previously learned information, especially reduce the impact of heterogeneity between different sets of clients.

By leveraging these two techniques, our approach aims to enhance the training performance of FL in non-IID environments, modelled by Dirichlet distribution [16], improving accuracy, convergence speed, and reducing communication overhead.

The contributions of this paper are as follows:

(1) *Investigation of FL and FTL performance*: We explore the performance of federated learning and federated transfer learning under different levels of non-IID data distributions, providing insights into its weakness under non-IID condition.
(2) *Introduction of IFTCL*: IFTCL approach by integrating TL and CL with FL is introduced, which demonstrate its ability to enhance training accuracy, accelerate convergence, and reduce communication overhead in non-IID environments.
(3) *Empirical validation*: We validate the practicality and effectiveness of FL, FTL and IFTCL algorithms on a publicly available radar recognition dataset, highlighting its potential for real-world applications.

2. System Model

In this section, the basic system models are presented, representing the context to carry out the data collection, communication and federated training process. A simple demonstration of the system model is shown in Figure 1. The clients are responsible for collecting radar images from different geographical locations and various objects. After the data is collected, the collected raw data is stored in each client. Communication between the clients and the server is conducted via wireless communication. Through multiple rounds of aggregation and updates between clients and the server, federated learning is eventually completed.

In this experiment, in order to simulate different levels of non-IID, each FL client is assigned a portion of the whole dataset with varying quantities and distributions of data according to a Dirichlet distribution, simulating a non-IID environment for FL. Every client utilizes a Convolutional Neural Network (CNN) with the same architecture to train for object recognition tasks.

2.1. Data Distribution

The non-IID nature of data is a challenge in FL. The Dirichlet distribution can effectively model such data distributions, which is a multivariate probability distribution used to describe the distribution of probability vectors. The probability distribution function for this distribution is:

$$P(p \mid \alpha) = \frac{1}{B(\alpha)} \prod_{m=1}^{M} p_m^{\alpha_m - 1} \tag{1}$$

where p is an M-dimensional vector representing the probability of each object's tag m for each client, and α_m is a positive parameter that determines the concentration of the generated distribution.

From this, we can conclude that the smaller the value of α, the more uneven the distribution of different tags across different clients which is caused by bigger variation of different objects in each client. Especially, when α is particularly small, it is highly possible that not all tags will be included on each client.

2.2. Basic FL Model

Federated learning involves a set of K clients, each with its local dataset D_k, and a central server. The goal is to train the global model w in the server by aggregating locally computed parameters without sharing the actual data. The optimization problem can be formulated as the minimization of loss function:

$$\min F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{2}$$

where: $F_k(w)$ is the local loss function for client k, n_k is the number of data points in client k, n is the total number of data points.

Each client k performs local updates by minimizing its local objective function using gradient descent:

$$w_k^{t+1} = w_k^t - \eta \nabla F_k \left(w_k^t \right) \tag{3}$$

where η is the learning rate.

After a certain number of local updates, clients send their local models to the server, which aggregates them to update the global model:

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1} \tag{4}$$

The algorithm above is called the Federated Averaging algorithm (FedAvg).

2.3. Wireless Communication Model

The communication process includes the formation of the local parameter in each training iteration in each client, the uplink of the trained model parameters to the server, the aggregation of the encoded parameters, decoding of the parameters, and the downlink of the parameters back to each client.

Hence that the communication system need source encoding and some kinds of encryption. The overall process of encoding can be represented as:

$$\theta_k^t = \phi(w_k^t) \tag{5}$$

Therefore, after the procedure of encoding, each client will uplink its encoded parameters. The transmission process can be denoted as the following algorithm:

$$\mu_k^t = \beta(\theta_k^t) \tag{6}$$

After the transmission, the server will begin to aggregate the coefficient by the conditional distribution of samples in each client. Finally, the fusion of the coefficient will be decoded, and gain the aggregated weight.

3. IFTCL Algorithm

This section delves into the structure of CNN, transfer learning and continual learning, providing an understanding of their definitions and mechanisms. It discusses why traditional Federated Transfer Learning (FTL) alone struggles to achieve optimal training outcomes under highly non-IID conditions. Finally, we introduce the Integrated Federated Transfer and Continual Learning (IFTCL) method, and explains its operational principles.

3.1. Composition of CNN

When it comes to tasks such as object recognition, speech recognition, image segmentation, and natural language processing, we often use Convolutional Neural Networks (CNNs). A CNN is a deep learning model composed of multiple layers, each responsible for different functions. A CNN can be roughly divided into two parts:

(1) Feature Extraction Part: Comprising multiple convolutional and pooling layers, this part is responsible for extracting feature information of the object, including shapes, contours, textures, colors, and spatial relationships. These layers gradually extract increasingly abstract features, transforming the input data into high-dimensional feature representations. It can be expressed as: w^f.

(2) Classification Part: Mainly composed of fully connected layers, this part inputs the high-dimensional features generated by the feature extraction part into a classifier to make the final classification decisions. The output layer of this part is usually a softmax layer, which produces a probability distribution over the various classes. It can be expressed as: w^c.

3.2. Transfer Learning

Transfer Learning (TL) is a machine learning technique where a model developed for a particular task is reused as the starting point for a model on a second task. By leveraging the knowledge gained from the first task, TL can significantly improve the performance and efficiency of the model on the new task.

Based on the information above, in FL, if clients can be divided into two subsets, we can pre-train one subset first and then use transfer learning to transfer the pre-trained biases gradients to the remaining subsets for further FL. At this point, if the pre-trained model parameters can fully represent the object's features, we only need to freeze the feature extraction parts of the CNN on these remaining clients and directly train the classification parts to obtain the specific object labels. However, this method, known as Federated Transfer Learning (FTL), is less effective in scenarios where the data is highly non-IID.

3.3. Continual Learning

Continual Learning (CL), also known as lifelong learning, is a significant concept in machine learning aimed at enabling models to retain and accumulate knowledge over a continuous learning process without forgetting previously learned information. Unlike traditional machine learning models that are trained on a fixed dataset, continual learning models are designed to adapt and learn from data that arrives incrementally.

There are many methods of CL, such as: (1) replaying a small portion of stored old data along with new data; (2) incorporating additional terms into the loss function to ensure that learning new tasks does not interfere significantly with previously learned tasks; (3) isolating or partitioning the model parameters to prevent interference between tasks. Among all three methods, the first method that using replaying mechanism is the easiest to carry out, and it can retain the same network structure as FL and FTL for later comparision. Therefore, we utilize the replaying strategy in our new approach.

3.4. Integrated Federated Transfer and Continual Learning

Since traditional FL and FTL has its limitations, we propose an Integrated Federated Transfer Continual Learning (IFTCL)

approach. This method combines the advantages of federated learning (transferring parameters instead of the entire model), transfer learning (leveraging learned knowledge), and continual learning (avoiding catastrophic forgetting).

The procedure of IFTCL can be described as follows: First, we partition N clients into $M+1$ subsets $S_1, S_2, \ldots S_M, S_R$ rather than only two sets. Initially, federated learning is used to train on the data of first set S_1. At this time, a rough feature extractor $w^f_{S_1}$ can be trained.

Then, using transfer learning, the feature extraction parts are transferred to the second subset through the server, where federated learning continues. At this point, because the model parameters are not frozen, due to the nature of transfer learning, the second set start learning based on the first rough feature extractor. The second client set will form a relatively complete feature extractor.

However, due to the non-IID nature in each client, the composition of data in each client is highly different. The knowledge learned by the previous subset will gradually be forgotten by the next subset during training. To address this, after training on S_2, CL will enable experience replay strategy by transfering parameters to previous trained set S_1 for continual learning for a certain round.

The following scenario is quite similar to the one described above. After CL, its feature extraction parts will undergo TL on S_3. Once TL is completed, it will go through a certain number of CL rounds on S_1 and S_2. This process will continue in the same manner until reaching S_M. In order to balance the training iterations in each client and compare the performance of FL and FTL later, we make the iterations in each client are the same in total.

Compared to the FTL method mentioned earlier, this approach aims to train the feature extractor more effectively. In traditional FTL, clients are divided into two groups, S and S_R, with the clients in S participating in federated training independently to obtain the feature extractor. However, in our IFTCL method, the set S in FTL is further subdivided into M smaller subsets, which first learn individually and then leverage transfer learning to pass knowledge to the next client.

The advantage of this approach lies in the fact that the ultimate goal is to minimize the global loss function through the aggregation phase of federated learning by minimizing the loss function of each client during local training. In scenarios where data among clients is highly heterogeneous, each client contributes differently to the global model during aggregation when minimizing its local loss function. This contrasts with IID data scenarios where each client's gradient descent direction is generally consistent. The multiple rounds of transfer learning in IFTCL reduce the aggregation process among clients, significantly mitigating this issue. Additionally, due to the substantial heterogeneity of client data, continual learning is employed to ensure that the previous training results are not forgotten during subsequent federated learning stages, necessitating the use of replay.

After all these procedures, a well optimized feature extractor is trained, we can continue the procedure in FTL, transfer the feature extractor to S_R and freeze it for further classification.

4. Simulations and Results

4.1. Experiment Setup

In this experiment, the MSTAR dataset was utilized, which is widely recognized in Synthetic Aperture Radar (SAR) imagery. For the purposes of this study, radar images of eight distinct objects were selected from the MSTAR dataset. The data distribution among clients was modeled using a Dirichlet distribution with parameter α set to 0.3 and 1. The α parameter controls the degree of non-IID distribution, allowing the evaluation of the performance of federated learning methods under various non-IID conditions.

The experimental configuration involved $K = 6$ clients, each equipped with a convolutional neural network consisting of two convolutional layers. These two layers are selected to be the feature extraction part for pre-training in FTL and IFTCL. In the process of transfer learning, three clients are selected for pre-training and the remaining three clients are chosen for classification. Meanwhile, in IFTCL, the selected three clients are designed to be three subsets, which means that local training is carried out in each client, which greatly decreases the communication burden. In this experiment, the total number of training rounds was set to 100, with 3,318 out of 4,459 samples being utilized for training.

4.2. Data Distribution

According to the Dirichlet distribution, data is allocated among six clients, as illustrated in Figure 2. Notably, different clients

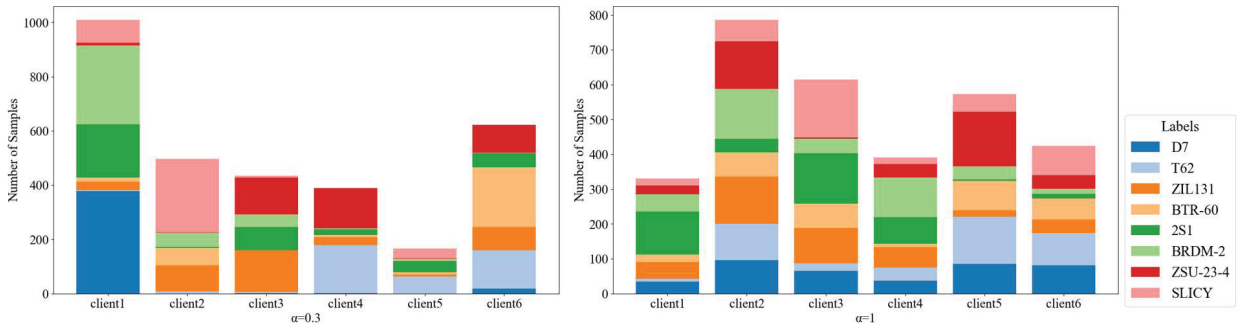

Figure 2.

Visual presentation of different parameters in non-IID data for each client.

receive varying quantities of data, with smaller values of α resulting in more imbalanced distributions. When $\alpha = 1$, we can clearly tell the distribution is uneven and unbalanced, but there are still eight kinds of objects in each client. In cases where the non-IID coefficient α is particularly small, as $\alpha = 0.3$, certain clients may receive very few samples for specific categories, or even none at all.

4.3. Training Performance

The evaluation of these scenarios is based on four critical metrics: communication overhead, convergence speed, accuracy.

4.3.1. *Training Accuracy*

As shown in Figures 3 and 4, these graphs illustrate the training performance of FL, FTL and IFTCL under different Dirichlet parameters α. From a broader perspective, comparing the two graphs, it is evident that all three methods demonstrate that as the Dirichlet parameter α increases, the training process converges faster, and the final training results become more accurate. Specifically, when $\alpha = 1$, the data is most evenly distributed compared to $\alpha = 0.3$, leading to a higher overall accuracy, with the fastest convergence rate.

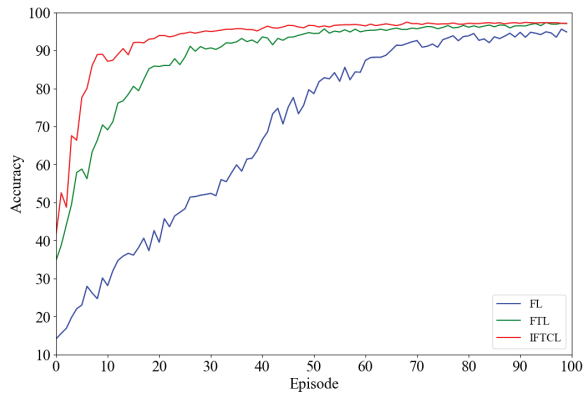

Figure 3.
Three algorithms training results comparison when $\alpha = 1$.

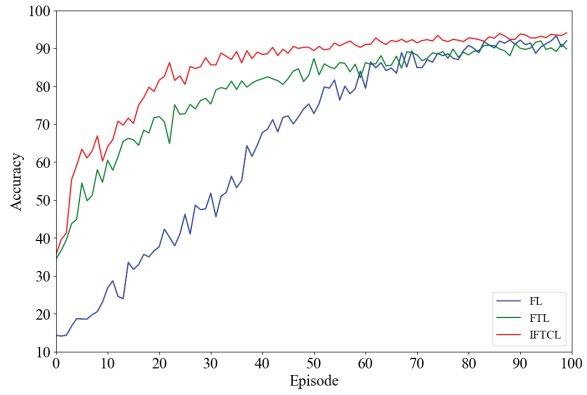

Figure 4.
Training performance when $\alpha = 0.3$.

Table 1.

The overall communication overhead comparison	
	Communication Overhead
FL	315.64MB
FTL	263.55MB
IFTCL	36.92MB

We can observe from the graph that regardless of the value of α, the initial accuracy is highest for IFTCL, followed by FTL, and then FL. This is because the pre-training step in transfer learning inherently boosts accuracy. However, the final accuracy may not always follow this pattern. When $\alpha = 1$, the results align with this trend, but when α is smaller, the significant differences in data across clients can cause issues. Specifically, in FTL, the pre-trained feature extractor may not sufficiently capture all the features, and the frozen feature layer could negatively impact subsequent classification. From the graph, it can be seen that when $\alpha = 0.3$, the accuracy of FTL slightly lags behind that of traditional federated learning. In contrast, IFTCL, having developed a more comprehensive feature extractor through training, consistently maintains higher accuracy than both federated learning and federated transfer learning.

4.3.2. *Convergence Speed*

In terms of convergence speed, when α is relatively large, such as $\alpha = 1$, it is obvious that the standard FL has the slowest convergence, only approaching convergence after 75 iterations. In contrast, FTL converges after 27 iterations, while IFTCL achieves convergence even faster, in just 18 iterations. On the other hand, when $\alpha = 0.3$, FL converges after about 82 iterations, FTL after around 50 iterations, and IFTCL after 25 iterations.

4.3.3. *Computation of Communication Overhead*

The communication overhead is defined as the volume of uplink and downlink data required for the communication process until the training reaches convergence. The results are shown in Table 1, We can see that traditional FL has the most communication overhead. FTL mitigates the burden greatly, because it needs fewer rounds than FL to achieve convergence, IFTCL results in a even lower communication overhead of 36.92 MB, which is much smaller than that of FL.

5. Conclusion

The Dirichlet distribution effectively models varying levels of non-IID conditions by adjusting its parameter, α. When α is small, some clients may lack certain labels from the dataset. In studying the impact of federated learning under different degrees of non-IID conditions, it has been observed that as distribution variation increases, the training accuracy and convergence speed of federated learning decreases. Federated transfer learning, known for reducing communication overhead and speed up convergence, can outperform traditional federated learning when the degree of non-IID is high (e.g., $\alpha = 1$). However, when α is small (e.g., $\alpha = 0.3$), the high heterogeneity causes FTL to underperform compared to FL due to an inadequately trained feature extractor in the pre-training stage.

To address this issue, the proposed integration of federated transfer and continual learning trains and transfers feature extractors multiple times, using replay mechanisms to avoid forgetting caused by highly non-IID data. This reduces aggregation challenges during the pre-training stage. It consistently outperforms both FL and FTL across varying α values, achieving higher training accuracy, faster convergence, and lower communication overhead. These results have been validated using the MSTAR dataset.

References

[1] P. Popovski et al., "Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC)," in *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5783–5801, Aug. 2019.

[2] H. Zhao et al., "QUIC-Enabled Data Aggregation for Short Packet Communication in mMTC," *IEEE INFOCOM 2022 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, New York, NY, USA, 2022, pp. 1–2.

[3] K. Shailaja, B. Seetharamulu and M. A. Jabbar, "Machine Learning in Healthcare: A Review," *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2018, pp. 910–914.

[4] Y. W. Bhowte, A. Roy, K. B. Raj, M. Sharma, K. Devi and P. LathaSoundarraj, "Advanced Fraud Detection Using Machine Learning Techniques in Accounting and Finance Sector," *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Chennai, India, 2024, pp. 1–6.

[5] G. Meena, D. Sharma and M. Mahrishi, "Traffic Prediction for Intelligent Transportation System using Machine Learning," *2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)*, Jaipur, India, 2020, pp. 145–148.

[6] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6G be?," *Nature Electronics*, vol. 3, no. 1, pp. 20–29, Jan. 2020.

[7] Bonawitz, K., et al. (2017). Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*.

[8] A. Elgabli, J. Park, S. Ahmed, and M. Bennis, "L-FGADMM: Layer-Wise Federated Group ADMM for Communication Efficient Decentralized Deep Learning," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, Seoul, Korea (South), 2020, pp. 1–6.

[9] J. T. Raj, "Building Decentralized Image Classifiers with Federated Learning," in *2020 IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, 2020, pp. 489–494.

[10] Y. Deng and X. Yan, "Federated learning on heterogeneous opportunistic networks," in *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, Nanjing, China, 2024, pp. 447–451.

[11] M. Baughman, N. Hudson, I. Foster, and K. Chard, "Balancing federated learning trade-offs for heterogeneous environments," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Atlanta, GA, USA, 2023, pp. 404–407.

[12] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, Kuala Lumpur, Malaysia, 2022, pp. 965–978.

[13] Y. Ma, S. Chen, S. Ermon, and D. B. Lobell, "Transfer learning in environmental remote sensing," *Remote Sensing of Environment*, vol. 301, p. 113924, Feb. 2024.

[14] J. Chen, J. Li, R. Huang, K. Yue, Z. Chen, and W. Li, "Federated transfer learning for bearing fault diagnosis with discrepancy-based weighted federated averaging," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[15] L. Wang, X. Zhang, H. Su and J. Zhu, "A Comprehensive Survey of Continual Learning: Theory, Method and Application," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5362–5383, Aug. 2024.

[16] J. Yao, Y. Zhang, Z. Xu, J. Sun, J. Zhou, and X. Gu, "Joint Latent Dirichlet Allocation for non-iid social tags," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, Turin, Italy, 2015, pp. 1–6.

Biographies

Boyuan Zhang received the B.Eng. degree in Information Engineering from Jinan University, Guangdong, China. He is currently pursuing M.Sc. degree in Mobile & Personal Communications at King's College London. His research interests include wireless communication and machine learning.

Mohammad Reza Shikh-Bahaei received the B.Sc. degree from the University of Tehran, Tehran, Iran, in 1992, the M.Sc. degree from the Sharif University of Technology, Tehran, in 1994, and the Ph.D. degree from King's College London, U.K., in 2000. He has worked for two start-up companies and for National Semiconductor Corporation, Santa Clara, CA, USA (now part of Texas Instruments Incorporated), on the design of third-generation (3G) mobile handsets, for which he has been awarded three U.S. patents as an inventor and a co-inventor. In 2002, he joined King's College London, as a Lecturer, where he is currently a full Professor with the Department of Engineering. He has authored or coauthored numerous journal articles and conference papers. He has been engaged in research of wireless communications and signal processing since 1995 both in academia and industrial organizations. His research interests include learning-based integrated sensing and communication, full duplex communication, RIS-assisted networks, and secure and energy-efficient communication over wireless networks. He was the Founder and the Chair of the Wireless Advanced (formerly SPWC) Annual International Conference from 2003 to 2012. He was a recipient of the overall King's College London Excellence in Supervisory Award in 2014.