Robust Privacy-Preserving Recommendation Systems Driven by Multimodal Federated Learning

Chenyuan Feng, Daquan Feng, Guanxin Huang, Zuozhu Liu, *Member, IEEE*, Zhenzhong Wang, and Xiang-Gen Xia, *Fellow, IEEE*

***Abstract*— Recommendation system (RS) is an important information filtering tool in nowadays digital era. With the growing concern on privacy, deploying RSs in a federated learning (FL) manner emerges as a promising solution, which can train a high-quality model on the premise that the server does not directly access sensitive user data. Nevertheless, some malicious clients can deduce user data by analyzing the uploaded model parameters. Even worse, some Byzantine clients can also send contaminated data to the server, causing blockage or failure of model convergence. In addition, most existing researches on federated recommendation algorithms only focus on unimodality learning, ignoring the assistance of multiple modality data to promote recommendation accuracy. Therefore, this article designs an FL-based privacy-preserving multimodal RS framework. To distinguish various modality data, an attention mechanism is introduced, wherein different weight ratios are assigned to various modal features. To further strengthen the privacy, local differential privacy (LDP) and personalized FL strategies are designed to identify malicious clients and bolster the resilience against Byzantine attacks. Finally, two multimodal datasets are established to verify the effectiveness of the proposed algorithm. The superiority of our proposed techniques is confirmed by the simulation results.**

***Index Terms*— Byzantine attack, federated learning (FL), local differential privacy (LDP), multimodal learning, recommendation system (RS).**

Nomenclature

U and I	User set and item set, respectively.
N and M	Number of users and that of items, respectively.
D and D_u	Global dataset and local dataset of client u, respectively.
$(\mathbf{x}_u^i, y_u^i)$	Training data $\mathbf{x}_u^i$ and their corresponding label y_u^i.
$\boldsymbol{x}_u^i$	Input multimodal data in ith training sample of user u.
y_u^i and $\hat{y}_u^i$	True label and the estimate label for ith training sample of user u.
$\mathbf{X}_{\text{txt}}$, $\mathbf{X}_{\text{img}}$, $\mathbf{X}_{\text{ID}}$, and $\mathbf{X}_{\text{beh}}$	Input text sequence, image data, identification (ID) data, and historical viewing content sequences, respectively.
$\mathbf{h}_{\text{txt}}^{(l)}$	Input of the lth Transformer block in text feature extraction module.
$\mathbf{Q}_{\text{txt},i}^{(l)}$, $\mathbf{K}_{\text{txt},j}^{(l)}$, and $\mathbf{V}_{\text{txt},j}^{(l)}$	Query, Key, and Value vectors of word i in text feature extraction module's lth transformer.
$d_{\mathbf{Q}_{\text{txt},i}^{(l)}}$ and $d_{\mathbf{Q}_i^{\text{beh}}}$	Row number in $\mathbf{Q}_{\text{txt},i}^{(l)}$ and $\mathbf{Q}_i^{\text{beh}}$, respectively.
$\mathbf{W}_{\text{txt},q}^{(l)}$, $\mathbf{W}_{\text{txt},k}^{(l)}$, and $\mathbf{W}_{\text{txt},v}^{(l)}$	Trainable weight matrices in text feature extraction module's lth attention block.
$\mathbf{W}_{\text{txt},1}^{(l)}$, $\mathbf{W}_{\text{txt},2}^{(l)}$, $\mathbf{b}_{\text{txt},1}^{(l)}$, and $\mathbf{b}_{\text{txt},2}^{(l)}$	Training weight matrices and bias vectors in text feature extraction module's lth Transformer block.
$\mathbf{f}^{\text{img}}$, $\mathbf{f}^{\text{txt}}$, $\mathbf{f}^{\text{ID}}$, and $\mathbf{f}^{\text{beh}}$	Image, text, ID, and behavior feature, respectively.
$\mathbf{F}_{\text{ID}}$ and K_o	Dense ID feature matrix of user or content and the dimension of ID feature vector, respectively.
$\mathbf{W}_{\text{Em}}$ and K_i	Learnable embedding matrix and the dimension of input ID sequence, respectively.

Manuscript received 12 September 2023; revised 29 January 2024 and 15 April 2024; accepted 2 June 2024. Date of publication 19 June 2024; date of current version 5 May 2025. The work of Chenyuan Feng was supported by the National Natural Science Foundation of China under Grant 62301328. The work of Daquan Feng and Guanxin Huang was supported in part by the Key Research and Development Plan of the Ministry of Science and Technology under Grant 2023YFF0612403, in part by Guangdong Provincial Key Laboratory under Grant 2023B1212060076, and in part by Shenzhen University High-Level Human and Social Sciences Team Project for Leading Scholar Innovation. Part of this work was presented at the 2023 IEEE Wireless Communications and Networking Conference (WCNC) [DOI: 10.1109/WCNC55385.2023.10118906]. *(Corresponding author: Daquan Feng.)*

Chenyuan Feng and Daquan Feng are with Guangdong Provincial Key Laboratory of Intelligent Information Processing, Shenzhen Key Laboratory of Digital Creative Technology, Shenzhen University, Shenzhen 518060, China (e-mail: fengchenyuan@szu.edu.cn; fdquan@szu.edu.cn).

Guanxin Huang is with the Research and Development Department, BYD Company Ltd., Shenzhen 518118, China (e-mail: huang.guanxin@byd.com).

Zuozhu Liu is with the Zhejiang University-University of Illinois at Urbana Champaign Institute, Zhejiang University, Haining, Zhejiang 314400, China (e-mail: zuozhuliu@intl.zju.edu.cn).

Zhenzhong Wang is with the Technical Management Center, China Media Group, Beijing 100859, China (e-mail: wangzhenzhong@cctv.com).

Xiang-Gen Xia is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: xianggen@udel.edu).

Digital Object Identifier 10.1109/TNNLS.2024.3411402

2162-237X © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

$\mathbf{W}^{\text{beh}}_{Q,i}$, $\mathbf{W}^{\text{beh}}_{K,i}$, and $\mathbf{W}^{\text{beh}}_{V,i}$	Training weight matrices to linearly project the input content sequence in the ith subspace.
$\mathbf{Q}^{\text{beh}}_{i}$, $\mathbf{K}^{\text{beh}}_{i}$, and $\mathbf{V}^{\text{beh}}_{i}$	Query, Key, and Value vectors in the ith subspace of behavior feature extraction module.
K_h	Total number of subspaces in the multihead attention mechanism.
$\mathbf{h}^{\text{beh}}_{i}$	Attention head vector in the ith subspace in the multihead attention mechanism.
$\mathbf{W}^{O}$	Training weight matrix in the multihead attention mechanism.
$\alpha^{\mathbf{f}_i}$	Attention weights assigned for the ith feature for feature fusion.
$\mathbf{W}_i$ and $\mathbf{b}_i$	Weight matrix and bias vector at the ith FC layer in the prediction network, respectively.
$\mathbf{y}_L$ and $\widehat{\mathbf{y}}_L$	Output of the Lth FC layer in the prediction network and the prediction vector.
$\theta^t_{u,l}$ and $\tilde{\theta}^{t+1}_{u,l}$	Local model of user u at tth communication round and its noisy version with LDP, respectively.
$\theta^t_{u,p}$	Local personalized model of user u at tth communication round.
λ	Proportion coefficient of the penalty term.
$\boldsymbol{w}^t$	Global model at tth communication round.
$\boldsymbol{u}^t$	Received local model from user u at tth communication round.
$\Theta^t_{u,p}$	Personalized reference model for user u.
$l_u(\theta^t_{u,l}; \mathbf{z}^i_u)$	Error between the prediction and true labels.
$g_{u,l}(\theta^t_{u,l}, D_{u,l})$ and $\hat{g}_{u,l}(\theta^t_{u,l}, D_{u,l})$	Local model gradient of user u and its clipped version.
$g_{u,l}(\theta^t_{u,l}, D_{u,l})$	Personalized model gradient of user u.
C, ϵ, and σ^2	Clipping threshold, privacy budget, and the additive white Gaussian noise (AWGN) power used in LDP technique.
$m^{t+1}_{u,l}$ and $v^{t+1}_{u,l}$	First and the second momentum of user local model of client u after tth round, respectively.
$m^{t+1}_{u,p}$ and $v^{t+1}_{u,p}$	First and the second momentum of user personalized model of client u after tth round, respectively.
η_l and η_p	Learning rate of user local update and user personalized update, respectively.
β_1 and β_2	Learning rate, hyperparameters related to the first and the second momentum tuning, respectively.
α^t_u	Attention weight coefficient of user u's local model in the tth global model aggregation.
$b^t_{u,v}$	Weight coefficient of uploaded parameters from user v on user u at the tth personalized reference model generation.

I. Introduction

RECOMMENDATION systems (RSs) that are based on user preferences are a useful tool for implementing personalized services and are essential for information filtering in order to tackle the issue of information overload. Nevertheless, conventional recommendation algorithms are typically implemented on cloud servers in order to acquire adequate resources for training and data storage. Recently, large-scale centralized data collecting has grown more challenging due to the possibility of unintentional disclosure or misuse of user data kept on the server. As a result, data security and privacy protection become crucial considerations in RSs [1], [2], [3], [4], [5], [6].

A. Distributed Learning-Based RS

The federated recommendation algorithm is a novel and highly regarded method since it secures the original data without the need for centralized learning (CL) and collecting of users' sensitive data. Federated RSs employ a distributed training framework to anticipate users' personal preferences by cooperatively training a shared RS model between a server and numerous client devices, eliminating the need to transmit users' sensitive data, in contrast to centralized RSs that have privacy problems [7], [8]. During this process, clients optimize their local models with their own collected data and transmit the local model parameters to the server, and the server will collect and aggregate the uploaded user model parameters to construct a global model and distribute it to all clients [9]. The decentralized data collection and storage of federated recommendation algorithms avoid potential attacks during the transmission of sensitive data and effectively enhance the user privacy [10]. In most federated learning (FL) scenarios, the interaction data uploaded from clients to the server are considered secure and can be shared with others [11], [12], [13]. However, recent research has shown that attackers can partially leak sensitive information by analyzing uploaded parameter information, such as model parameters or gradient information [14].

B. Privacy-Preserving RS

To further ensure the data security, encryption-based or obscureness-based FL algorithms are proposed to protect the interaction data. As for encryption-based federated RS algorithms, the protection of uploaded parameters is realized by using homomorphic encryption or blockchain techniques [7], [15], [16]. As for obscureness-based FL algorithms, the protection of interaction data is realized by blurring raw data, such as adding Gaussian noises or Laplace noises to the uploaded parameters [2]. Federated RS based on homomorphic encryption is proposed, which requires clients to perform homomorphic encryption before uploading model parameters to the server, in [17] and [18]. The encrypted model parameters are then aggregated on the server to generate a global encrypted model. Wei et al. [19] reveal that parameter interactions between clients and servers may suffer from model inference attacks, where malicious users analyze global model parameters and infer other users' original data. To solve this problem, local differential privacy (LDP) is proposed, which adds noises to model parameters. Compared with homomorphic encryption-based FL algorithms, LDP-based FL algorithms have lower computational complexity and better flexibility.

C. Data Sparsity in RS

In addition to concerns about data privacy, recommendation algorithms also have to deal with data sparsity [20], which

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

significantly impairs recommendation performance. Recent studies show that multiple modality data, such as user evaluation, commodity description, and pictures, can effectively alleviate the data sparsity issue [21], [22]. It shows that the correlations between users can be constructed by analyzing user attribute features, such as age and occupation. In addition, valuable visual information as well as textual information, such as item images, comments, and introductions, can be explored as important supplementary sources of information for recommending similar products to users [23], [24]. In [25], a multimodal video RS is proposed based on emotional analysis, which analyzes physiological parameters, such as facial expressions, arm movements, and human instantaneous reactions to analyze users' preferences. In [26], a multimodal RS is proposed to learn artist information from semantic information and analyze the music popularity through audio signals. In [27], a multimodal and socially-aware movie RS is proposed to mine the similarities among users and movies based on user social relationships, user rating data, poster images, and textual descriptions of movies. In [21], a multimodal news RS is proposed, which uses target detection algorithms to extract regions of interest (ROIs) from news images and predict the user preference by analyzing correlations of news text and image ROI data with co-attention transformers models. In [28], a multimodal knowledge graph attention network is proposed to solve the cold start problems in the RSs, where all modality information is treated as entity nodes in the knowledge graph and information among nodes are aggregated through graph attention networks.

The abovementioned multimodal RS algorithms all focus on a CL manner. As far as we know, there are only a few studies on the deployment of multimodal RS algorithms in an FL scenario. Due to the differences in acquisition equipment and user habits, some users may have only one or a few modality of data. Therefore, how to solve the modality missing issue and maximize the utilization efficiency of various modality data is a promising research direction of federated recommendation algorithm in the future.

Based on in-depth literature research, there are still four main kinds of challenges in the implementation of robust federated RSs.

1) *Privacy Data Security Issue:* The attackers can deduce users' original sensitive data by analyzing the parameters uploaded by clients, leading to information leakage.
2) *Data Imbalance Issue:* Users' collected data vary in quantity and quality, which introduces statistic bias in model aggregation and consequently degrades learning performance.
3) *Byzantine Attack Problem:* Since FL is an open-distributed training manner, competitors can create Byzantine clients to upload pointless or misleading data to the server, hindering the convergence of the global recommendation model.
4) *Importance Distinctiveness of Multimodal Data and Missing Modality Issue:* Not all users have multiple modality data, and various modality data possessed by different users have varying contribution values to their respective recommendation models.

Motivated by these issues, this article studies the design of a privacy-preserving RS based on multimodal FL. The main innovations of this article are summarized as follows.

1) To address the heterogeneity of users' multimodal data, an adaptive multimodal federated recommendation system (AMMFRS) is designed for the local training model. Furthermore, an attention module is proposed to effectively employ various modality features and discern the significance of various modality information. A feature completion technique is introduced to deal with the missing modality issue.
2) To protect the user privacy, this article proposes a robust personalized privacy-preserving federated learning (RP^{3}FL) manner as for model parameter interaction. This study designs a personalized federated training technique to mitigate the performance loss brought about by the introduction of LDP and data imbalance. Thus, an enhanced privacy federated recommendation algorithm that maintains high recommendation performance is achieved.
3) To avoid the convergence failure caused by Byzantine attacks and enhance the robustness of our algorithm, this article proposes a secure model aggregation strategy to adaptively allocate aggregation weights to each users.
4) As far as we know, there is few open-source multimodal dataset for RS training; therefore, we construct our own movie and joker training dataset based on crawling information from public websites. This work conducts comprehensive trials in many scenarios and experimental settings, confirming the superiority and efficacy of the suggested algorithm in terms of prediction accuracy.

The rest of this article is organized as follows. Section II introduces an FL paradigm for multimodal RSs. In Section III, AMMFRS algorithm is formulated for local training model design. In Section IV, the RP^{3}FL manner is illustrated in detail. Numerical simulations and experimental results are shown in Section V, and finally, Section VI concludes this article. The main notations in this article are presented in the Nomenclature.

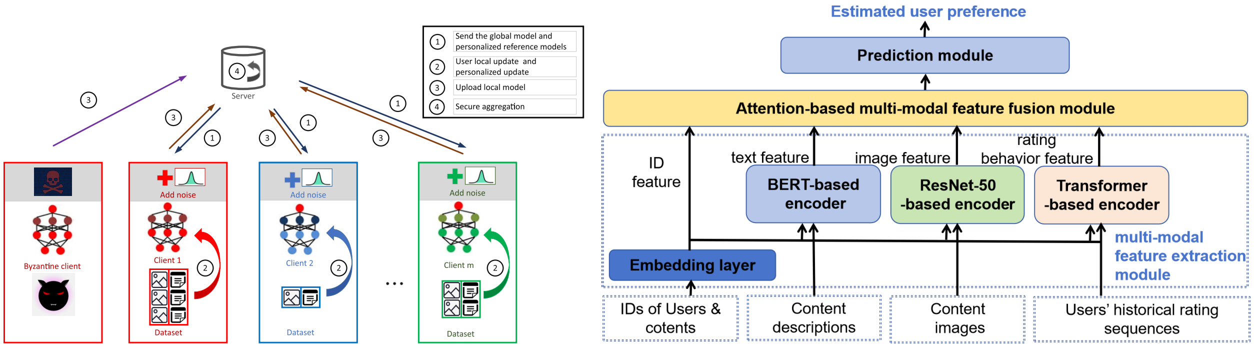
II. System Model

In this work, assuming that a set $U = \{u_1, u_2, \dots, u_N\}$ consists of N clients and a set $I = \{i_1, i_2, \dots, i_M\}$ consists of M items, a set of multimodal data, D, is distributed on N clients, where $D = \bigcup_{u=1}^{N} D_u$, where D_u represents the multimodal data of the uth client. The amount of data between local datasets of different clients are unbalanced, i.e., $\|D_u\| \neq \|D_v\|$, $u \neq v \in N$. Each training data for the prediction problem is denoted as $(\mathbf{x}_u^i, y_u^i)$, where $\mathbf{x}_u^i$ consists of K_m modality information and the value of label $y_u^i \in Y$ is client u's level of interest in item i, $Y = \{y_u^i \mid u \in U, i \in I\}$. In this work, we consider image, text, identification (ID), and rating data, which means $K_m = 4$.

As shown in Fig. 1(a), the federated training process can be described as follows.

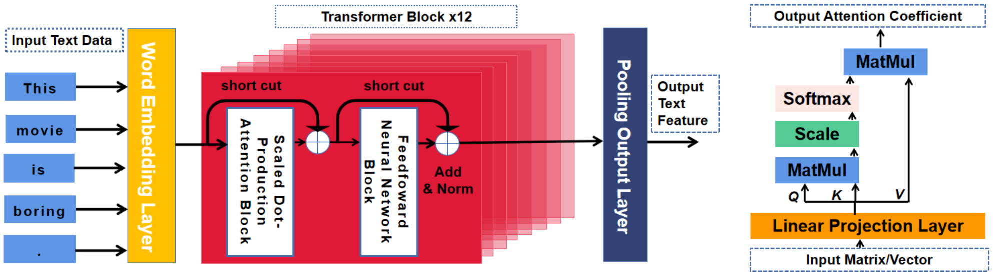
1) At the beginning of a new round of training, the server random sends the latest global model and personalized reference model parameters to each client.
2) Each client updates its local model and personalized model based on received models according to the AMMFRS algorithm and RP^{3}FL manner.
3) After local training, the selected honest clients will add Gaussian noise to the local models based on the LDP scheme before uploading them to the server. While malicious clients will upload random parameters or the opposite of their local model parameters to the server to launch Byzantine attacks and prevent the updating of the global model.

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 1. Framework of our proposed AMMFRS algorithm trained in an RP^{3}FL manner. (a) Illustration of RP^{3}FL manner. (b) Illustration of AMMFRS local model. The input data consist of content descriptions (e.g., movie synopsis posters and joker text), content images (e.g., movie posters and joker illustrations), the IDs of all users and contents, and users rating sequence. The multimodal feature extraction module, multimodal attention layer, and prediction layer comprise the AMMFRS-based local model. The attention layer weights and concatenates each feature information to build a new feature representation. Ultimately, these feature vectors will be input into the prediction layer, where two fully connected (FC) layers will compute an estimate of each user's preference probability for all contents.



(a) (b)

Fig. 2. Architecture of text feature extraction module. (a) Architecture of pretrained BERT-based model. (b) Architecture of the scaled dot-production attention block in the transformer block. The original input data are the description of each content. Similar to BERT [29], the output of the word embedding layer is the sum of the token embedding, the segmentation embedding, and the position embedding. The output is the feature representation of the input text. Each transformer block consists of one scaled dot-production attention block and one FNN block.

4) Once received the parameters uploaded by selected users, the server dynamically aggregates the local model parameters based on secure aggregation algorithms and updates the global model.
5) Repeat steps 1)–4) continuously until the model converges. In the FL manner, the procedure of steps 1)–4) is usually called as one communication round. The details of the AMMFRS local training model, RP^{3}FL interaction manner among the server, and users based on LDP techniques will be illustrated in the following.

III. AMMFRS-Based Local Model Design

As shown in Fig. 1(b), the AMMFRS-based local training model comprises three primary modules, which are the multimodal feature extraction module, multimodal attention module, and prediction module. The input data include content illustration figures, content description, users' and contents' IDs, and users' historical rating on contents. The four extraction submodules are located in the multimodal feature extraction module. To create a new feature vector representation, the attention layer concatenates and weights each piece of feature data. Ultimately, the prediction layer will receive these feature vectors and utilize them to compute an estimate of user preference for all contents.

A. Multimodal Feature Extraction Module

The multimodal feature extraction module consists of four submodules, each of which has a missing modality completion mechanism and is in charge of processing various modality data to produce a unique feature representation.

1) Text Features: Semantic text can help mine the similarity between different contents as well as users, which is helpful to predict user preferences. In this work, we design a text feature extraction module based on pretrained bidirectional encoder representations from transformers (BERTs)-based submodule [29]. The input text data are first converted into high-dimensional vectors for subsequent training. As shown in Fig. 2, our BERT-based feature extraction submodule consists of $L = 12$ Transformer blocks, where one feedforward neural network (FNN) block and one scaled dot-production attention block make up each Transformer block. Given an input text sequence $\mathbf{X}_{\text{txt}} = (\mathbf{x}_{\text{txt},1}, \mathbf{x}_{\text{txt},2}, \dots, \mathbf{x}_{\text{txt},n})$ with n words, where $\mathbf{x}_{\text{txt},i}$ represents the ith word identifier. In this article, the maximum length of the input sequence is considered to be 512, namely, $n = 512$. The learning procedure can be expressed as follows:

$$\begin{aligned}\mathbf{h}_{\text{txt}}^{(0)} &= \mathbf{X}_{\text{txt}}\\ \mathbf{h}_{\text{txt}}^{(l+1)} &= \text{FNN}^{(l)}\left(\text{ATT}^{(l)}\left(\mathbf{h}_{\text{txt}}^{(l)}\right)\right), \quad l = 0, \dots, L-1\\ \mathbf{f}^{\text{txt}} &= \text{MaxPool}\left(\mathbf{h}_{\text{txt}}^{(L)}\right)\end{aligned} \tag{1}$$

where $\mathbf{h}_{\text{txt}}^{(l)}$ denotes the input of lth hidden Transformer block, and $\text{ATT}^{(l)}(\cdot)$ and $\text{FNN}^{(l)}(\cdot)$ denote the outputs of the attention block and the FNN block in Transformer block l, respectively. The following Transformer block will get the output of the lth

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

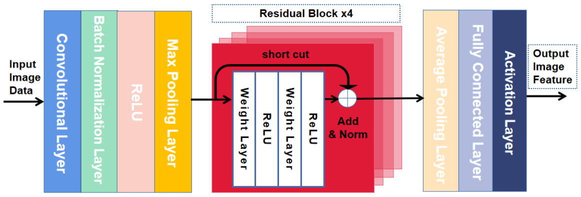


Fig. 3. Architecture of image feature extraction module.

Transformer block, and the last Transformer block's output will be utilized to construct the text features for the classification task, namely, $\mathbf{f}^{\text{txt},i}$.

Feeding $\mathbf{h}_{\text{txt}}^{(l)}$ into the attention block allows the model to capture the relationships among different words in a sentence to better understand the contextual semantics of each word. For $i \neq j = 1, \ldots, N, l = 0, \ldots, L-1$, the ith element of $\text{ATT}^{(l)}(\mathbf{h}_{\text{txt}}^{(l)})$ denoted by $\text{ATT}_i^{(l)}(\mathbf{h}_{\text{txt}}^{(l)})$ can be expressed as

$$\text{ATT}_i^{(l)}\left(\mathbf{h}_{\text{txt}}^{(l)}\right) = \sum_j \left(\text{softmax}\left(\frac{\mathbf{Q}_{\text{txt},i}^{(l)}\left(\mathbf{K}_{\text{txt},j}^{(l)}\right)^{\text{T}}}{\sqrt{d_{\mathbf{Q}_{\text{txt},i}^{(l)}}}}\right)\mathbf{V}_{\text{txt},j}^{(l)}\right)$$
$$\mathbf{Q}_{\text{txt},i}^{(l)} = \mathbf{W}_{\text{txt},q}^{(l)}\mathbf{h}_{\text{txt},i}^{(l)}, \quad \mathbf{K}_{\text{txt},j}^{(l)} = \mathbf{W}_{\text{txt},k}^{(l)}\mathbf{h}_{\text{txt},j}^{(l)}$$
$$\mathbf{V}_{\text{txt},j}^{(l)} = \mathbf{W}_{\text{txt},v}^{(l)}\mathbf{h}_{\text{txt},j}^{(l)} \tag{2}$$

where $\mathbf{Q}_{\text{txt},i}^{(l)}$, $\mathbf{K}_{\text{txt},j}^{(l)}$, and $\mathbf{V}_{\text{txt},j}^{(l)}$ denote the queried word i's Query vector, Key vector, and Value vector of another word j in transformer block l, respectively, $(\mathbf{K}_{\text{txt},j}^{(l)})^{\text{T}}$ is the transpose of matrix $\mathbf{K}_{\text{txt},j}^{(l)}$, $d_{\mathbf{Q}_{\text{txt},i}^{(l)}}$ denotes the row number of $\mathbf{Q}_{\text{txt},i}^{(l)}$, and $\mathbf{W}_{\text{txt},q}^{(l)}$, $\mathbf{W}_{\text{txt},k}^{(l)}$, and $\mathbf{W}_{\text{txt},v}^{(l)}$ are the linear-projection weight matrices in attention block l. For $l = 0, \ldots, L-1$, the output of FNN can be expressed as

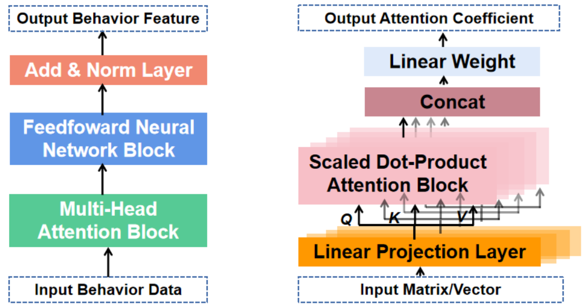
$$\text{FNN}^{(l)}\left(\text{ATT}^{(l)}\left(\mathbf{h}_{\text{txt}}^{(l)}\right)\right)$$
$$= \mathbf{W}_{\text{txt},2}^{(l)}\text{ReLU}\left(\mathbf{W}_{\text{txt},1}^{(l)}\text{ATT}^{(l)}(\mathbf{h}_{\text{txt}}^{(l)}) + \mathbf{b}_{\text{txt},1}\right)^{(l)} + \mathbf{b}_{\text{txt},2}^{(l)} \tag{3}$$

where $\mathbf{W}_{\text{txt},1}^{(l)}$, $\mathbf{W}_{\text{txt},2}^{(l)}$, $\mathbf{b}_{\text{txt},1}^{(l)}$, and $\mathbf{b}_{\text{txt},2}^{(l)}$ denote the training weight matrices and bias vectors of the corresponding training models in the lth Transformer block, respectively.

2) Image Features: The visual characteristics of the contents indicate how similar they are to one another, which might give useful information for forecasting user preferences. In this article, we adopt an image feature extraction module based on ResNet50 architecture [30], [31]. As shown in Fig. 3, this model is mostly made up of several residual blocks, each of which has a shortcut link and several convolutional layers. It can successfully address the gradient disappearance issue brought on by too many neural network (NN) layers by enabling the network to learn the residual data of the input image rather than the image itself. The convolutional layer's output is

$$\mathbf{h}_{\text{img}} = \text{ReLU}(\text{Conv2d}(\mathbf{x}_{\text{img}}, \mathbf{w}) + \mathbf{b}) \tag{4}$$

where $\mathbf{x}_{\text{img}}$ with a size of 3 × 224 × 224 denotes the input picture, $\text{Conv2d}(\mathbf{x}, \mathbf{w})$ is the convolution operation, $\mathbf{w}$ and $\mathbf{b}$ are the learnable weights and bias, respectively, and rectified linear unit (ReLU) denotes an activation function. The output of the convolutional layer $\mathbf{h}_{\text{img}}$ will be transferred to four successive residual blocks, one pooling layer, and ultimately an FC layer, to generate the image feature $\mathbf{f}^{\text{img}}$ with a dimension of 1000.



(a)

(b)

Fig. 4. Architecture of user behavior feature extraction module. (a) Overall architecture. (b) Multihead attention block.

3) ID Features: Considering that one-hot encoders typically embed the ID information and produce highly sparse and high-dimension feature vectors, we create an embedding module to transform user and content IDs into dense low-dimension representations. Given the user/item ID sequence $\mathbf{X}_{\text{ID}} = [\mathbf{x}_{\text{ID},1}, \ldots, \mathbf{x}_{\text{ID},i}, \ldots, \mathbf{x}_{\text{ID},N}]^{\text{T}}$, the output of the embedding module is: $\mathbf{F}_{\text{ID}} = \mathbf{X}_{\text{ID}}\mathbf{W}_{\text{Em}}$, where $\mathbf{F}_{\text{ID}} \in \mathbb{R}^{N \times K_o}$ is a dense user/item ID feature matrix, each row $\mathbf{f}^{\text{ID}}$ in $\mathbf{F}_{\text{ID}} \in \mathbb{R}^{K_o}$ is one ID feature, $\mathbf{W}_{\text{Em}} \in \mathbb{R}^{K_i \times K_o}$ is the embedding matrix, K_i denotes the length of input one-hot ID vector, and artificially set parameter $K_o \ll K_i$ denotes the dimension of output ID feature. The value of K_i depends on the characteristic of training data, and K_o is set as 64 in this work.

4) Rating Behavior Features: The most crucial factor in predicting users' future interests is their prior rating sequences. To process the relevance of user historical rating data, we construct a rating behavior feature extraction module based on Transformer encoder [33], [34]. The Transformer encoder is made up of one multihead attention block and one FNN block, as shown in Fig. 4. To capture the dependencies between various contents, the multihead attention mechanism processes in parallel distinct portions of the input representation of the user's historical rated contents. In order to acquire the behavior features, the feature representation is finally transferred to the FNN block, which consists of residual connections and layer normalization.

The multihead attention mechanism uses several linear transformations to project linear projections on the input data, resulting in several distinct representations [32]. The attention mechanism is applied to different subspaces, allowing the model to focus on various areas of the input data at the same time. The final output is created by joining the attention-weighted representations from each subspace and passing them to the final linear layer. Let $\mathbf{X}_{\text{beh}} = [\mathbf{x}_{\text{beh},1}, \ldots, \mathbf{x}_{\text{beh},i}, \ldots, \mathbf{x}_{\text{beh},N}]^{\text{T}}$ denote an input user-content rating sequence, multihead attention mechanism adopts three different weight matrices, $\mathbf{W}_{Q,i}^{\text{beh}}$, $\mathbf{W}_{K,i}^{\text{beh}}$, and $\mathbf{W}_{V,i}^{\text{beh}}$, to linearly project the input rating sequence $\mathbf{X}_{\text{beh}}$ into different subspaces. For subspace i, Query, Key, and Value vectors can be defined as

$$\mathbf{Q}_i^{\text{beh}} = \mathbf{X}_{\text{beh}}\mathbf{W}_{Q,i}^{\text{beh}}, \quad \mathbf{K}_i^{\text{beh}} = \mathbf{X}_{\text{beh}}\mathbf{W}_{K,i}^{\text{beh}}, \quad \mathbf{V}_i^{\text{beh}} = \mathbf{X}_{\text{beh}}\mathbf{W}_{V,i}^{\text{beh}} \tag{5}$$

where K_h is the total number of subspaces. In subspace i, the attention head is

$$\mathbf{h}_i^{\text{beh}} = \text{softmax}\left(\frac{\mathbf{Q}_i^{\text{beh}}\left(\mathbf{K}_i^{\text{beh}}\right)^{\text{T}}}{\sqrt{d_{\mathbf{Q}_i^{\text{beh}}}}}\right)\mathbf{V}_i^{\text{beh}} \tag{6}$$

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

where $(\mathbf{K}_i^{\text{beh}})^{\text{T}}$ is the transpose of matrix $\mathbf{K}_i^{\text{beh}}$, and $d_{\mathbf{Q}_i^{\text{beh}}}$ denotes the row number of $\mathbf{Q}_i^{\text{beh}}$. The attention-weighted representations from all subspaces are concatenated and fed into the multihead attention mechanism, which may be described as

$$\text{MultiHead}(\mathbf{X}_{\text{beh}}, K_h) = \text{Concat}\left(\mathbf{h}_1^{\text{beh}}, \dots, \mathbf{h}_{K_h}^{\text{beh}}\right)\mathbf{W}^O \tag{7}$$

where $\mathbf{W}^O$ is the trainable weight matrix. In this work, K_h is set as 8, which is the same as the classical multihead attention mechanism [32]. The FNN block consists of a two-layer NN and ReLU activation function. Finally, the output of the final Transformer encoder is created utilizing residual connections and layer normalization, which acts as a behavior feature for user-item ratings, $\mathbf{f}^{\text{beh}}$.

5) Missing Modality Data Completion Mechanism: To deal with the probable missing modality issue, this article employs a mean imputation module. The main approach is to replace the missing modality feature with the expected value of existing analogous features.

B. Multimodal Attention Fusion

One simple way to fuse various modalities is to concatenate their characteristics directly. However, this amounts to assigning a set weight to each modality and ignoring the variations in the ways that different modality features influence the model update [35]. Moreover, within the FL framework, different modality aspects' contributions to the recommendation models differ in terms of efficacy. In order to get the ideal allocation strategy, this article proposes a multimodal feature fusion method based on an attention mechanism that may be adjusted as needed. Let $\mathbf{f} = \{\mathbf{f}^{\text{txt}}, \mathbf{f}^{\text{img}}, \mathbf{f}^{\text{ID}}, \mathbf{f}^{\text{beh}}\}$ denote the multimodal feature. In order to create feature vectors of the same dimension, the input feature is projected using two nonlinear projection layers into a k_d-dimensional vector $\mathbf{f}_{k_d}^i$, which is expressed as

$$\mathbf{f}_{k_d}^i = \text{ReLU}(\mathbf{W}_2(\text{ReLU}(\mathbf{W}_1\mathbf{f}^i + \mathbf{b}_1) + \mathbf{b}_2)) \tag{8}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ denote the weight matrices and bias vectors of the first and second projection layers, respectively. After nonlinearly projecting each modality feature into its own attention module, we can obtain the attention weights $\alpha^{\mathbf{f}_i} = \tanh(\mathbf{W}\mathbf{f}_{k_d}^i + \mathbf{b})$, where $\mathbf{W}$ and $\mathbf{b}$ denote the weight matrix and bias vector, respectively, and $\alpha^{\mathbf{f}_i}$ is the attention weight for modality i. The sum of the elementwise multiplication between the k_d-dimensional vectors of each modality and its corresponding attention weight represents the multimodal features' final output, which is $\mathbf{F} = \sum_{i \in \{\text{txt, img, ID, beh}\}} \alpha^{\mathbf{f}_i} \odot \mathbf{f}_{k_d}^i$, where $\odot$ is the elementwise multiplication, and k_d is set as 256 in this work.

C. Prediction Network

The Sigmoid function can be used to determine the predicted user preference by feeding the multimodal attention fusion module's output into an FC layer. The prediction network is made up of L FC layers, and given the output $\mathbf{F} \in \mathbb{R}^d$ of the multimodal attention fusion module, the FC layer's output is represented as follows:

$$\begin{aligned} \mathbf{y}_0 &= f(\mathbf{W}_0\mathbf{F} + \mathbf{b}_0) \\ \mathbf{y}_{l+1} &= f(\mathbf{W}_l\mathbf{y}_l + \mathbf{b}_l), \quad l = 1, \dots, L-1 \\ \widehat{\mathbf{y}}_L &= \text{Sigmoid}(\mathbf{y}_L) \end{aligned} \tag{9}$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ are the weight matrix and bias vector at the ith FC layer in the prediction network, respectively, $f(\cdot)$ is the activation function, $\mathbf{y}_L$ is the output of the Lth FC layer in the network, and $\widehat{\mathbf{y}}_L$ is the prediction vector.

IV. RP^{3}FL-Based Training Manner

The RP^{3}FL manner is proposed to enhance the privacy and robustness in this section. LDP algorithms are used to add noise to model parameters uploaded from each client to the server, in order to enhance the privacy. Since adding noise can have a certain degree of damage to model accuracy, a personalized training strategy is designed to construct a globally generalized recommendation model while retaining local personalized recommendation models. In addition, an adaptive adjustment of the local update learning rate strategy and adding penalty terms to the local optimization objective are used to prevent the local personalized model from detaching from the global model update after long-term training. Furthermore, an adaptive weighting aggregation strategy is used to adjust the client's aggregation weight for secure aggregation to avoid damage to the global model update caused by attacking clients. To alleviate the performance loss caused by the two aforementioned problems, this article proposes a personalized FL strategy, which mainly consists of two stages: the local model update stage and the personalized model update stage.

A. User Local Update

In the local training stage, each client performs further updates based on the global model issued by the server. To enhance the model flexibility and mitigate the negative effects of imbalanced data, this article proposes a target function adaptive update method in the local training stage. The target function adaptive update introduces a penalty term to prevent the local model from deviating too far from the global model update direction due to significant differences between local data samples and those of other users. Thus, given a local training dataset sampled from user local dataset, namely, $D_{u,l} \subset D_u$, the modified objective function of local model updating of user u at tth communication round is

$$\mathcal{L}_{u,l}\left(\theta_{u,l}^t, D_{u,l}\right) = \mathbb{E}_{\mathbf{z}_u^i \in D_u}\left[l_u\left(\theta_{u,l}^t; \mathbf{z}_u^i\right)\right] + \lambda\left\|\theta_{u,l}^t - \boldsymbol{w}^t\right\|_2 \tag{10}$$

where $\theta_{u,l}^t \in \mathbb{R}^d$ and $\boldsymbol{w}^t \in \mathbb{R}^d$ denote the local model of user u and the global model parameters at tth communication round, λ is the proportion coefficient of the penalty term and is a hyperparameter that controls the adjustment amplitude, $\|\theta_{u,l}^t - \boldsymbol{w}^t\|_2$ represents the penalty term that constrains the local model update, and $l_u(\theta_{u,l}^t; \mathbf{z}_u^i)$ represents the error between the prediction and true labels on local training data sample $\mathbf{z}_u^i = (x_u^i, y_u^i)$ with local model $\theta_{u,l}^t$ of user u at tth round, which is defined as

$$\begin{aligned} l_u\left(\theta_{u,l}^t; \mathbf{z}_u^i\right) &= l_u\left(\theta_{u,l}^t; \left(\boldsymbol{x}_u^i, y_u^i\right)\right) \\ &= y_u^i \log\left(\hat{y_u^i}\right) + \left(1 - y_u^i\right)\log\left(1 - \hat{y_u^i}\right) \end{aligned} \tag{11}$$

where $\boldsymbol{x}_u^i$ denotes the input multimodal data in ith training sample of user u, and y_u^i and $\hat{y_u^i}$ denote the true label and the estimate label for ith training sample of user u, trained by AMMFRS algorithm. To satisfy the requirements of LDP, the gradient of the objective function needs to be clipped, and the gradient of local training loss function can be defined as

$$\begin{aligned} g_{u,l}\left(\theta_{u,l}^t, D_{u,l}\right) &= \nabla\mathcal{L}_u\left(\theta_{u,l}^t, D_{u,l}\right) \\ \hat{g}_{u,l}\left(\theta_{u,l}^t, D_{u,l}\right) &= \frac{g_{u,l}\left(\theta_{u,l}^t, D_{u,l}\right)}{\max\left(1, \frac{\|g_{u,l}(\theta_{u,l}^t, D_{u,l})\|_2}{C}\right)} \end{aligned} \tag{12}$$

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

where $g_{u,l}(\theta^t_{u,l}, D_{u,l})$ and $\hat{g}_{u,l}(\theta^t_{u,l}, D_{u,l})$ represent the local model gradient and its clipped version under the clipping threshold of C. The local model is updated using the Adam optimizer in the local training stage, and the noisy model is blurred with the LDP technique, which can be described as follows:

$$\begin{aligned} m^{t+1}_{u,l} &= \beta_1 m^t_{u,l} + (1-\beta_1)\hat{g}_{u,l}\big(\theta^t_{u,l}, D_{u,l}\big) \\ v^{t+1}_{u,l} &= \beta_2 v^t_{u,l} + (1-\beta_2)\hat{g}_{u,l}\big(\theta^t_{u,l}, D_{u,l}\big) \odot \hat{g}_{u,l}\big(\theta^t_{u,l}, D_{u,l}\big) \\ u^{t+1}_{u,l} &= \frac{m^{t+1}_{u,l}}{\sqrt{v^{t+1}_{u,l}} + \epsilon} \\ \theta^{t+1}_{u,l} &= \boldsymbol{w}^t - \eta_l \frac{\sqrt{1-\beta_2}}{1-\beta_1} u^{t+1}_{u,l} \\ \tilde{\theta}^{t+1}_{u,l} &= \theta^{t+1}_{u,l} + N(0, \sigma^2 I) \end{aligned} \tag{13}$$

where $\tilde{\theta}^{t+1}_{u,l}$ denotes the noisy local model of user u for uploading to the server after tth round of user local model update, η_l denotes learning rate of user local model training, $m^{t+1}_{u,l}$, $v^{t+1}_{u,l}$, and $g^{t+1}_{u,l}$ represent the first momentum, the second momentum, and the gradient of local model parameters of client u after tth round of user local update, respectively, and β_1 and β_2 denote the hyperparameters related to the first- and second-momentum tuning, respectively. In this stage, the unperturbed local model $\theta^{t+1}_{u,l}$ is saved locally for the subsequent personalized training stage, and the noisy local model $\tilde{\theta}^{t+1}_{u,l}$ is uploaded to the server.

B. User Personalized Update

As a popular privacy protection technology, LDP techniques can perturb interactive data before user model parameters are uploaded to the server, aiming to protect the privacy of user personal data while allowing mathematical analysis of model parameters to ensure the feasibility of model training. However, during the implementation process, the additional noise introduced by the LDP technique will degrade the performance of the recommendation algorithm model. Additionally, the FL training process is susceptible to data imbalance, which refers to the difference in the size and distribution of datasets among different clients.

1) LDP Schemes: The basic idea of the LDP technique is adding noise to blur the original data, and the noise is added in a controllable and quantifiable way. Let D and D' denote two adjacent datasets that differ only by one entity. Let $\epsilon > 0$ denote the boundary value used to differentiate all outputs on the adjacent datasets, and $\delta \in [0, 1)$ denote a probability value that represents the ratio of the probability of an event that cannot be limited by e^ϵ between the two adjacent datasets after adding the privacy mechanism. Given any δ, a larger ϵ can make the distinguishability of adjacent datasets clearer, resulting in a higher risk of privacy leakage. A random algorithm $\mathbf{M}$ satisfies (ϵ, δ)-LDP, if the following condition is satisfied:

$$\Pr[\mathbf{M}(D)] \le e^\epsilon \Pr\big[\mathbf{M}\big(D'\big)\big] + \delta. \tag{14}$$

This work applies the Gaussian mechanism with ℓ_2-norm sensitivity to the FL framework to ensure (ϵ, δ)-LDP. The specific implementation of the mechanism is to add Gaussian noise with mean 0 and variance $\sigma^2 I$ to the local model parameter. The ℓ_2-norm sensitivity can be defined as follows:

$$\triangledown \ell_2 = \max_{D_i, D_i'} \big\| g(D_i) - g\big(D_i'\big)\big\|_2 \tag{15}$$

where $g(D_i) = \triangledown \ell(D_i, \theta_i)$ denotes the gradient of local loss function of the ith user, and $\ell(D_i, \theta_i)$ denotes the ith user's local loss function calculated with the local dataset D_i and the model parameters θ_i. The detail of the local loss function will be introduced in Section IV-B2.

To measure the upper bound of noise perturbation in (ϵ, δ)-LDP, this article uses the method of using norm clipping to clip the model gradient parameters, so that the maximum value of the gradient is equal to the norm clipping threshold. Assuming a norm clipping threshold C is given, the sensitivity can be bounded as: $\triangledown \ell_2 \le (2\eta C/|D_i|)$, where η represents the local model update speed. In order to measure the level of privacy protection, this article uses (ϵ, δ)-LDP moments account method to calculate the privacy budget ϵ under the FL framework, which can be defined as

$$\epsilon = \frac{\triangledown \ell_2 \sqrt{2T \ln \frac{1}{\delta}}}{\sigma} \tag{16}$$

where T represents the number of total communication rounds, and σ denotes the standard deviation of the additive Gaussian noise. Therefore, the privacy budget ϵ can effectively reflect the level of privacy protection for the current interaction data. The smaller the ϵ is, the higher the level of privacy protection is, and vice versa.

2) User Personalized Model Update: In the personalized training stage, each client further optimizes the unperturbed model from the local training stage based on the characteristics of their own data distribution and trains a personalized local model to improve the accuracy of preference prediction. Based on the received reference model from the server, the gradient of loss function of personalized training is defined as

$$g_{u,p}(\theta^t_{u,p}, D_{u,p}) = \nabla \mathcal{L}_u\big(\theta^t_{u,p}, D_{u,p}\big) \tag{17}$$

where $\theta^t_{u,p}$ and $D_{u,p} \subset D_u$ denote the personalized user model and the personalized training dataset of user u at tth round, respectively. The personalized user model at the tth round of the personalized training stage based on the Adam optimizer can be expressed as follows:

$$\begin{aligned} m^{t+1}_{u,p} &= \beta_1 m^t_{u,p} + (1-\beta_1) g^t_{u,p}\big(\theta^t_{u,p}, D_{u,p}\big) \\ v^{t+1}_{u} &= \beta_2 v^t_{u,p} + (1-\beta_2) g^t_{u,p}\big(\theta^t_{u,p}, D_{u,p}\big) \odot g^t_{u,p}\big(\theta^t_{u,p}, D_{u,p}\big) \\ u^{t+1}_{u,p} &= \frac{m^{t+1}_{u,p}}{\sqrt{v^{t+1}_{u,p}} + \epsilon} \\ \theta^{t+1}_{u,p} &= \gamma \theta^t_{u,p} + (1-\gamma)\big(c\theta^{t+1}_{u,l} + \Theta^t_{u,p}\big) - \eta_p \frac{\sqrt{1-\beta_2}}{1-\beta_1} u^{t+1}_{u,p} \end{aligned} \tag{18}$$

where $\theta^{t+1}_{u,l}$ is defined in (13), $\Theta^t_{u,p}$ denotes the personalized reference model for user u received from the server at the tth round and will be specified later, η_p is the initial learning rate of user personalized model training, $m^{t+1}_{u,p}$, $v^{t+1}_{u,p}$, and $g^{t+1}_{u,p}$ represent the first momentum, the second momentum, and the gradient of personalized model parameters of client u after the tth personalized update round, respectively, and c and γ are the coefficient weights of the tth user personalized model and the $(t+1)$th user local model, respectively.

In conclusion, the LDP-based user local model training is to solve the information leakage issue in the traditional FL manner, and the introduction of a personalized user model is to solve the problem of data imbalance and the model performance degradation caused by LDP techniques.

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

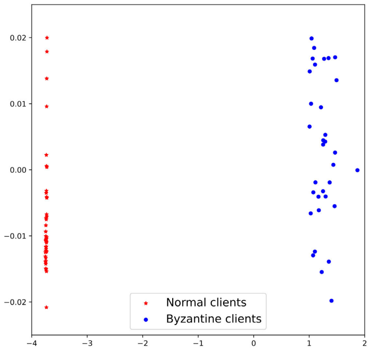


Fig. 5. Visualization of 2-D data of model parameters for both Byzantine and normal clients.

C. Server Secure Aggregation

As described in [36], the traditional FL is vulnerable to the attacks from Byzantine clients. These clients can influence the aggregation process by uploading arbitrary data to the server, which may cause slow convergence rate or even convergence failure of the global model. Considering the coexistence of both normal and Byzantine clients, the data received by the server from client u can be represented as

$$\boldsymbol{w_u}^{t+1} = \begin{cases} *, & \text{if client } u \text{ is Byzantine attacker} \\ \tilde{\theta}_{u,l}^{t+1}, & \text{otherwise} \end{cases} \tag{19}$$

where $*$ represents an arbitrary value, and $\boldsymbol{w_u}^{t+1}$ represents the received model parameters from client u after tth round of user local update.

Furthermore, this work applies PCA data dimensionality reduction to the model parameters of Byzantine clients and normal clients during the training process. The high-dimensional model parameter data are mapped to 2-D data for visualization, as shown in Fig. 5. As can be seen from Fig. 5, there is a significant difference between the model parameters of Byzantine clients and normal clients. By comparing and analyzing the model parameters of the two, Byzantine clients can be effectively detected and identified.

To resist the influence of Byzantine clients, this work proposes to introduce a secure aggregation mechanism in the model aggregation process, which dynamically assigns different weights to each client. In this way, Byzantine clients will be assigned lower aggregation weights, while normal clients will be assigned higher aggregation weights, greatly reducing the impact of Byzantine clients on global model updates. The secure aggregation mechanism uses cosine similarity to calculate the difference between the local model received from the clients and the stored global model at the server before aggregation, which can be expressed as

$$s_{u,l}^{t+1} = \frac{\boldsymbol{w}^{t} \cdot \boldsymbol{w_u}^{t+1}}{\left\|\boldsymbol{w}^{t}\right\| \left\|\boldsymbol{w_u}^{t+1}\right\|}. \tag{20}$$

Geometric median, as a widely-used robust estimator, is adopted in this work. Specifically, parameters that are closer to the median should be assigned greater attention values. Therefore, cosine similarity combined with geometric median can be used to measure the aggregation weights of all clients involved in the aggregation. The attention weight coefficient can be defined as

$$\alpha_u^{t+1} = \text{Softmax}\left(1 - \left\| \frac{s_{u,l}^{t+1} - \text{Median}\left(\mathbf{s}_l^{t+1}\right)}{\max\left(\mathbf{s}_l^{t+1}\right) - \min\left(\mathbf{s}_l^{t+1}\right)} \right\|\right) \tag{21}$$

where α_u^{t+1} denotes the attention weight coefficient of user u's local model in global model aggregation, and $\mathbf{s}_l^{t+1} = \{s_{1,l}^{t+1}, \ldots, s_{N,l}^{t+1}\}$ is the list of differences between the global model and all local models of N users at tth round. It indicates that when the cosine similarity distance between local model parameters and global model parameters is closer to the median value, the aggregation weight assigned to them is higher, and vice versa. And the global model is aggregated as follows:

$$\boldsymbol{w}^{t+1} = \sum_{u} \alpha_u^{t+1} \boldsymbol{w}_u^{t+1}. \tag{22}$$

To further prevent honest users from malicious users that inject biased data into the global model, attention-based aggregation mechanism is used for personalized reference model generation. Similarly, the personalized reference model for user u is calculated as follows:

$$\Theta_{u,p}^{t+1} = \sum_{v=1, v\neq u} b_{u,v}^{t+1} \boldsymbol{w}_v^{t+1}. \tag{23}$$

where $b_{u,v}^{t+1}$ denotes the weight coefficient of uploaded parameters from user v on user u's personalized reference model generation. The weights of personalized reference models can be defined as

$$b_{u,v}^{t+1} = \text{Softmax}\left(1 - \left\| \frac{p_{u,v}^{t+1} - \min\left(\mathbf{p}_u^{t+1}\right)}{\max\left(\mathbf{p}_u^{t+1}\right) - \min\left(\mathbf{p}_u^{t+1}\right)} \right\|\right) \tag{24}$$

where $\mathbf{p}_u^{t+1} = \{p_{u,v}^{t+1}\}_{v\neq u}^{N-1}$ is the list of cosine similarity between the uploaded parameters from the target user u and the other user v, which can be defined as

$$p_{u,v}^{t+1} = \frac{\boldsymbol{w}_u^{t+1} \cdot \boldsymbol{w}_v^{t+1}}{\left\|\boldsymbol{w}_u^{t+1}\right\| \left\|\boldsymbol{w}_v^{t+1}\right\|}, \quad u \neq v. \tag{25}$$

The detail of RP^{3}FL interaction procedure is listed in Algorithm 1.

V. Experimental Results

This section will introduce our constructed datasets for simulation, experimental environment settings, comparison methods, evaluation metrics, and performance analysis.

A. Dataset Construction

In this work, we construct two unbalanced multimodal datasets based on published movie and joke datasets since as far as we know, there is few open-source standard multimodal datasets for RS training. The MovieLens-1M movie (MMMovie) dataset[1] consists of 6040 users' rating information on over 3900 movies. It contains a total of 1 000 209 rating records, the value of which is an integer from 0 to 5. The user feature information comprises the user ID, age, gender, and occupation, while the movie feature information consists of links to the IMDb webpage[2] and 18 different movie categories. This work crawls the movie poster and introduction from the IMDb website as the image data and text data of multimodal data in order to retrieve the image and text data corresponding to the film. To construct positive and negative sample sets, data with a score value greater than or equal to four points is

[1] https://grouplens.org/datasets/movielens/1m/

[2] https://www.imdb.com/

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Robust Privacy-Preserving Personalized Federated Learning (RP^{3}FL)

```
Input: Initial global model $\boldsymbol{w}^0 = \mathbf{0}$, initial personalized reference model $\{\Theta^0_{u,p}\}^N_u = \mathbf{0}^N_u$, maximum number of communication round $T$, local training datasets $D_u = \{\mathbf{X}_{txt}, \mathbf{X}_{img}, \mathbf{X}_{ID}, \mathbf{X}_{beh}\}$, learning rate for local user model update $\eta_l$, learning rate for personalized user model update $\eta_p$, noise power $\sigma^2$, privacy budget $\epsilon$, learning hyperparameters $\beta_1$, $\beta_2$, $C$.
Output: Final global model $\boldsymbol{w}^T$ and final personalized user models $\{\theta^T_{u,p}\}^N_u$
for round $t = 0$ to $T - 1$ do
    Server sends $\boldsymbol{w^t}$ and $\Theta^t_{u,p}$ to users
    for all client $u \in U$ do
        Make recommendation prediction based on AMM-FRS algorithm
        Update user local model and user noisy model $\theta^{t+1}_{u,l}, \tilde{\theta}^{t+1}_{u,l} \leftarrow$ UserLocalUpdate($\boldsymbol{w^t}, \theta^{t+1}_{u,l}$) based on (10) - (13)
        Update personalized user model $\theta^{t+1}_{u,p} \leftarrow$ UserPersonalizedUpdate($\Theta^t_{u,p}, \theta^t_{u,p}, \theta^{t+1}_{u,l}$) based on (17) - (18)
        Upload noisy model $\tilde{\theta}^{t+1}_{u,l}$
    end for
    Collect noisy user local models $\{\boldsymbol{w^{t+1}_u}\}^N_u \leftarrow$ UserUpload$\{*, \tilde{\theta}^{t+1}_{u,l}\}^N_u$ according to (19)
    Update global model $\boldsymbol{w^{t+1}} \leftarrow$ ServerSecureAggregation($\boldsymbol{w^t}, \{\boldsymbol{w^{t+1}_u}\}^N_u$) based on (20) - (22)
    Update personalized reference model $\Theta^{t+1}_{u,p} \leftarrow$ ServerSecureAggregation($\boldsymbol{w^{t+1}_u}, \{\boldsymbol{w^{t+1}_v}\}^N_{v \neq u}$) based on (23) - (25)
end for
```

taken as positive sample data, that is, content that users are interested in; data with a score value less than four points is taken as negative sample data, that is, content that users are not interested in. In addition, in order to construct user historical request sequence data, this article sorted each user score data in time order on the training set, verification set, and test set, constructed the past four positive sample data of the same user into a sequence as the user historical request sequence of the positive samples, and constructed the user request sequence of the negative samples in the same way. So far, we have finished compiling the multimodal MMMovie dataset.

The Jester joke dataset[3] provides rating data from over 82 000 users on 150 jokes collected from November 2006 to March 2015. It contains a total of over 2 300 000 evaluation records, the value of which is an integer from −10 to 10. The input information includes the user IDs, joke IDs, and the specific contents of the joke. Since no image data are provided by this dataset, we only use its text information as multimodal data. To construct positive and negative sample sets, we consider rating values between 2 and 10 as positive samples and rating values between −10 and 1 as negative samples. Similar to the construction of MMMovie dataset, the Multimodal Jester (MMJoke) dataset is constructed. Table I shows the statistical information of the movie and joke datasets.

TABLE I
DATASETS STATISTICAL PROPERTIES

Datasets	Interaction	Items	Users	Sparsity
MovieLens-1M	1000209	3706	6040	95.53%
Jester	2300000	150	82366	81.38%

Considering the heterogeneous data generated by distributed users in real life, the amount of data allocated to each client is set as different. This article proposes a term of user balance index (UBI) to measure the degree of difference in the number of local dataset samples between different users, which can be defined as UBI $= (\min(\mathcal{D})/\max(\mathcal{D}))$, where $\mathcal{D} = \{|D_1|, |D_2|, \ldots, |D_N|\}$ with $|D_i|$ representing the number of samples in user i's training dataset. The more closer the value of UBI is to 1, the more balanced the sample quantity among different users is. This article will conduct experiments under two different degrees of data imbalance. The amount of data allocated to each client in Unbalanced Multimodal MovieLens-1 (UMMMovie) dataset with different values of UBI settings are shown in Fig. 6. So far, the UMMMovie and Unbalanced Multimodal Jester (UMMJoke) datasets are finally constructed. In this experiment, 70% of the dataset is randomly selected as training data, 20% as validation data, and 10% as test data.

B. Performance Metrics

This article uses the following performance metrics: F1-score, Accuracy, and area under the receiver operating characteristic (ROC) curve (AUC). Among these, the AUC is a commonly-used classifier performance indicator. The true positive rate (TPR) is the vertical coordinate and the false positive rate (FPR) is the horizontal coordinate of the ROC curve, which illustrates how well classifiers perform at various thresholds. AUC values range from 0.5 to 1. The classifier's performance improves with increasing value. The recommendation model's ability to accurately forecast a user's level of interest in a particular piece of material is indicated by the recommendation algorithm.

Accuracy is an indicator of the proportion of samples correctly classified by a classifier to the total number of samples, whose value ranges from 0 to 1. Accuracy can be defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{26}$$

where TP represents the number of true positive cases, TN represents the number of true negative cases, FN represents the number of false negative cases, and FP represents the number of false positive cases.

F1-score is a metric that takes into account the Precision and Recall of the model, whose value ranges from 0 to 1. The higher the values of Accuracy and F1-score are, the better the prediction performances of the recommended models are. F1-score can be defined as

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{27}$$

where Recall is a metric that measures the proportion of true positive samples correctly predicted by the classifier to the total number of true positive samples, and Precision is a metric

[3] https://eigentaste.berkeley.edu/dataset/

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

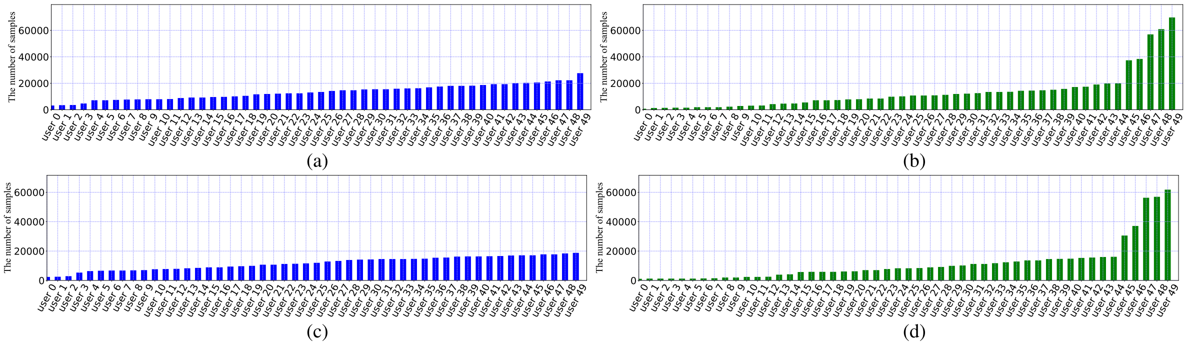


Fig. 6. Allocation of data samples to each client with different datasets and different settings of UBI value. (a) UMMMovie dataset, high UBI. (b) UMMMovie dataset, low UBI. (c) UMMJoke dataset, high UBI. (d) UMMJoke dataset, low UBI.

TABLE II
PERFORMANCE COMPARISON OF VARIOUS MODELS ON THE MMMOVIE DATASETS

		Item2Vec [37]	**NeualCF** [38]	**AotuInt** [39]	**DCN-V2** [40]	**AMMFRS (ours)**
CL	ACC	0.7094	0.7345	0.7221	0.7211	**0.7628**
	AUC	0.7547	0.7822	0.7595	0.7609	**0.8257**
	F1	0.7215	0.7360	0.7283	0.7281	**0.7539**
FL	ACC	0.6401	0.7004	0.6499	0.6595	**0.7105**
	AUC	0.6977	0.7591	0.6887	0.6968	**0.7899**
	F1	0.6601	0.7081	0.6882	0.6851	**0.7153**

that measures the proportion of true positive samples correctly predicted by the classifier to the total number of samples predicted as positive.

C. Baselines

To assess the efficacy of our proposed multimodal learning RS model, we use the following RS model for a thorough comparison.

1) *Item2Vec [37]:* The primary objective is content similarity analysis. Using the popular Word2Vec approach, it discovers the latent representation of the contents.
2) *NeuralCF [38]:* This algorithm blends deep learning with conventional collaborative filtering techniques. It uses NNs to learn latent representations of all users and content.
3) *AotuInt [39]:* The algorithm utilizes a multihead self-attention NN with residual connections to extract the features of ratings and semantic information from user ratings and content descriptions as input.
4) *DCN-V2 [40]:* This algorithm automatically and efficiently learns explicit and implicit cross features via a deep cross-network.

To evaluate the effectiveness of our proposed attention mechanism, we also compare our model with the following federated RS models to evaluate the effectiveness of our proposed attention mechanism.

1) *FedAvg [41]:* This method is a classic FL method proposed by Google. Each client device trains a local model based on the stochastic gradient descent method, and the server will generate a global model by taking an average of uploaded model parameters.
2) *Per-FedAvg [42]:* This method is a personalized variation of the FedAvg algorithm. A meta-learning method is proposed to train multiple personalized models and one global shared model at the same time.
3) *APPLE [43]:* This method proposes a personalized FL algorithm that can adaptively learn the benefit that each client can obtain from other clients' models.

D. Performance Analysis

1) Impact of Our Proposed AMMFRS Algorithm: The experiment contrasts the proposed algorithm with five state-of-the-art (SOTA) deep learning recommendation algorithms, namely, Item2Vec, NeuralCF, AutoInt, and DCN-V2. All these algorithms were proposed under a CL manner. To make a fair comparison, we conduct the experiments with both CL and FL manners. As for the distributed training manner, each user will construct their own local modals based on Item2Vec, NeuralCF, AutoInt, or DCN-V2 algorithm, and the user-server cooperation is akin to the FedAvg algorithm, a classical FL architecture [41]. The experimental results of all models are shown in Table II.

The findings demonstrate that, for both centralized and distributed training methods, our proposed algorithm outperforms the other four SOTA algorithms in terms of recommendation accuracy. This suggests that the multimodal learning and attention-based model aggregation mechanism of our AMMFRS approach can yield better recommendation performance in both CL and FL scenarios. Out of all of them, the performance of Item2Vec is the worst, highlighting the limitations of assuming content-relatedness as the only factor in predicting user preference. Moreover, FL's learning performance would decline relative to that of CL due to the statistic bias caused by unbalanced and heterogeneous data. It is noteworthy to notice that our proposed algorithm has the least performance degradation when compared to the other FL-based approaches. This may be attributed to our attention mechanism and adaptive model aggregation approach.

As shown in Table III, it is a similar story with MMJoke datasets. From the experimental results, it can be seen that the proposed algorithm has the best performance indicators in both centralized and distributed scenarios. The proposed method can better understand user preferences by incorporating text, image, and user request sequence data into user preference learning, modeling the inherent correlations between them, and dynamically allocating weights for various modal features. All

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PERFORMANCE COMPARISON OF VARIOUS MODELS ON MMJOKE DATASETS

		Item2Vec [37]	**NeuralCF** [38]	**AutoInt** [39]	**DCN-V2** [40]	**AMMFRS (ours)**
CL	ACC	0.5800	0.7927	0.7877	0.7879	**0.8385**
	AUC	0.5706	0.8507	0.8410	0.8422	**0.9053**
	F1 score	0.5985	0.6694	0.6676	0.6648	**0.7659**
FL	ACC	0.5288	0.7703	0.6858	0.6800	**0.7743**
	AUC	0.5749	0.8221	0.7081	0.7010	**0.8354**
	F1 score	0.3775	0.6274	0.4351	0.4879	**0.6336**

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT (ϵ, σ)-LDP SETTINGS ON THE UMMMOVIE DATASET

		Centralized	**FedAvg** [41]	**Per-FedAvg** [42]	**APPLE** [43]	**RP^{3}FL (ours)**
ACC	σ=0	**0.7628**	0.7105	0.7009	0.7004	0.7332
	σ=0.1		0.7030	0.6873	0.6604	**0.7285**
	σ=0.2		0.6783	0.6039	0.5245	**0.7105**
AUC	σ=0	**0.8257**	0.7899	0.7869	0.7815	0.7918
	σ=0.1		0.7748	0.7565	0.7203	**0.7896**
	σ=0.2		0.7423	0.6716	0.5311	**0.7583**
F1 score	σ=0	**0.7539**	0.7153	0.7199	0.7091	0.7254
	σ=0.1		0.7142	0.6941	0.6097	**0.7446**
	σ=0.2		0.6455	0.6732	0.5618	**0.7174**
ϵ	σ=0.1		0.00024	0.00024	0.00024	0.00024
	σ=0.2		0.00012	0.00012	0.00012	0.00012

TABLE V
PERFORMANCE COMPARISON UNDER DIFFERENT (ϵ, σ)-LDP SETTINGS IN UMMJOKE DATASET

		Centralized	**FedAvg** [41]	**Per-FedAvg** [42]	**APPLE** [43]	**RP^{3}FL(ours)**
ACC	σ=0	**0.8385**	0.7743	0.7625	0.7723	0.8345
	σ=0.1		0.7086	0.6892	0.6916	**0.8084**
	σ=0.2		0.6828	0.6696	0.5926	**0.7835**
AUC	σ=0	**0.9053**	0.8354	0.8225	0.8377	0.9001
	σ=0.1		0.7633	0.7457	0.7600	**0.8720**
	σ=0.2		0.7253	0.7089	0.5903	**0.8458**
F1 score	σ=0	**0.7659**	0.6336	0.6320	0.6300	0.7530
	σ=0.1		0.5754	0.5696	0.5245	**0.7264**
	σ=0.2		0.5344	0.5006	0.5204	**0.6990**
ϵ	σ=0.1		0.00024	0.00024	0.00024	0.00024
	σ=0.2		0.00012	0.00012	0.00012	0.00012

of these features can greatly enhance the model prediction performance.

2) Impact of Our Proposed RP^{3}FL Manner: In this experimental setup, all federated recommendation algorithms use LDP techniques to resist privacy inference attacks. σ represents different noise levels, i.e., different degrees of privacy protection. The higher the noise level σ, the smaller the privacy budget ϵ, and the higher the degree of privacy protection, but the recommendation performance of the algorithm will be worse. The performance comparison of various algorithms under different noise levels is shown in Tables IV and V. From the experimental results on the UMMMovie and UMMJoke datasets, it can be seen that our proposed algorithm outperforms FedAvg, Per-FedAvg, and APPLE algorithms in all three performance metrics. It is worth mentioning that the performance degradation of our proposed algorithm is the smallest when σ increases. Since a larger σ represents a higher noise level and stronger privacy protection, the proposed algorithm can maintain relatively satisfactory prediction accuracy with strengthened privacy protection.

E. Discussion

1) Robustness to Data Imbalance: Fig. 6 displays the unique data split configuration for each user with varying UBI levels in UMMMovie and UMMJoke datasets. Our proposed algorithm outperforms the other algorithms, as shown in Tables VI and VII and Figs. 7 and 8. In particular, the FedAvg, Per-FedAvg, and APPLE algorithms perform worse than the proposed algorithm in low UBI situations by 1.5%, 1.7%, and 1.0%, respectively, whereas the AUC performance by using our algorithm fell by 0.5% in high UBI circumstances. The proposed algorithm provides the least amount of performance deterioration and the most stability when there is data imbalance, as can be seen from the abovementioned tables and figures. This can be attributed to the superiority of our attention-based mechanism.

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

TABLE VI

PERFORMANCE COMPARISON UNDER DIFFERENT UBI SETTINGS IN UMMMOVIE DATASET

		FedAvg [41]	**Per-FedAvg** [42]	**APPLE** [43]	**RP^{3}FL (ours)**
ACC	low UBI (UBI=0.0074)	0.7058	0.6937	0.6965	**0.7311**
	high UBI (UBI=0.1075)	0.7105	0.7009	0.7004	**0.7332**
AUC	low UBI (UBI=0.0074)	0.7777	0.7739	0.7735	**0.7877**
	high UBI (UBI=0.1075)	0.7899	0.7869	0.7815	**0.7918**
F1 score	low UBI (UBI=0.0074)	0.7132	0.7112	0.7053	**0.7277**
	high UBI (UBI=0.1075)	0.7153	0.7199	0.7091	**0.7254**

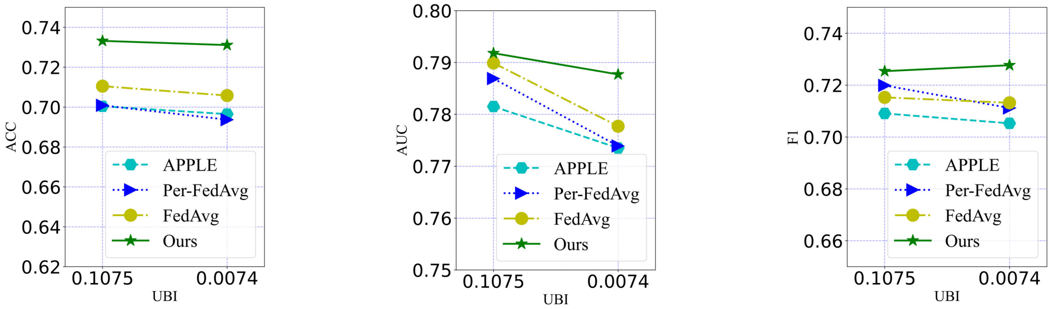


Fig. 7. Performance comparison under different UBI settings in UMMMovie dataset.

TABLE VII

PERFORMANCE COMPARISON UNDER DIFFERENT UBI SETTINGS IN UMMJOKE DATASET

		FedAvg [41]	**Per-FedAvg** [42]	**APPLE** [43]	**RP^{3}FL (ours)**
ACC	low UBI (UBI=0.0146)	0.7370	0.6892	0.7564	**0.8344**
	high UBI (UBI=0.1172)	0.7743	0.7625	0.7723	**0.8345**
AUC	low UBI (UBI=0.0146)	0.7868	0.7813	0.8301	**0.9000**
	high UBI (UBI=0.1172)	0.8354	0.8225	0.8377	**0.9001**
F1 score	low UBI (UBI=0.0146)	0.5777	0.5765	0.6256	**0.7488**
	high UBI (UBI=0.1172)	0.6336	0.6320	0.6300	**0.7530**

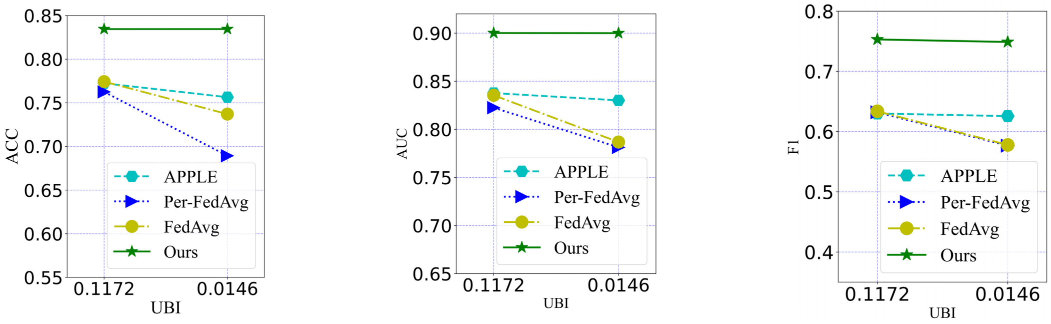


Fig. 8. Performance comparison under different UBI settings in the UMMJoke dataset.

2) Robustness to Byzantine Attacks: In order to commit malicious assaults, the Byzantine clients in the experiment will upload the global model parameters that are the reverse of what is received to the server. Tables VIII and IX present a performance comparison of different methods in various datasets with varying number of Byzantine clients. First, in terms of all three performance criteria, the proposed method beats the FedAvg, Per-FedAvg, and APPLE algorithms, according to the experimental results on UMMMovie and UMMJoke datasets. Second, the forecast accuracy of all algorithms often diminishes as the percentage of Byzantine clients in the whole user group rises. While our designed algorithm exhibits the least amount of performance loss among these four techniques as the number of Byzantine clients rises. In particular, for the UMMMovie dataset, the performance of the suggested approach only drops by 5.8% when the percentage of Byzantine clients rises from 0% to 40%, whereas the AUC values of the FedAvg, Per-FedAvg, and APPLE algorithms decline by 35.4%, 35.4%, and 10.5%, respectively. The experimental results show that the proposed algorithm can still maintain high prediction performance even in the face of varying degrees of Byzantine client attacks, which reflects the robustness and superior performance of the proposed algorithm.

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII

PERFORMANCE COMPARISON UNDER DIFFERENT BYZANTINE CLIENT PROPORTIONS IN UMMMOVIE DATASET

		Centralized	**FedAvg** [41]	**Pre-FedAvg** [42]	**APPLE** [43]	**RP^{3}FL (ours)**
ACC	Byzantine Clients 0%	**0.7614**	0.7105	0.7009	0.7004	0.7332
	Byzantine Clients 30%		0.5428	0.5402	0.6467	**0.6898**
	Byzantine Clients 40%		0.5396	0.5402	0.6385	**0.6801**
AUC	Byzantine Clients 0%	**0.8257**	0.7899	0.7869	0.7815	0.7918
	Byzantine Clients 30%		0.7061	0.7280	0.7263	**0.7513**
	Byzantine Clients 40%		0.5104	0.5086	0.6814	**0.7462**
F1 score	Byzantine Clients 0%	**0.7498**	0.7153	0.7199	0.7091	0.7254
	Byzantine Clients 30%		0.6865	0.6860	0.6843	**0.6909**
	Byzantine Clients 40%		0.6773	0.6858	0.6691	**0.6896**

TABLE IX

PERFORMANCE COMPARISON UNDER DIFFERENT BYZANTINE CLIENT PROPORTIONS IN UMMJOKE DATASET

		Centralized	**FedAvg** [41]	**Pre-FedAvg** [42]	**APPLE** [43]	**RP^{3}FL (ours)**
ACC	Byzantine Clients 0%	**0.8385**	0.7743	0.7625	0.7723	0.8345
	Byzantine Clients 30%		0.6296	0.6220	0.6525	**0.7988**
	Byzantine Clients 40%		0.6266	0.6296	0.6321	**0.7747**
AUC	Byzantine Clients 0%	**0.9053**	0.8354	0.8225	0.8377	0.9001
	Byzantine Clients 30%		0.5865	0.7365	0.7192	**0.8806**
	Byzantine Clients 40%		0.5046	0.5150	0.6155	**0.8643**
F1 score	Byzantine Clients 0%	**0.7659**	0.6336	0.6320	0.6300	0.7530
	Byzantine Clients 30%		0.5312	0.5312	0.5386	**0.7231**
	Byzantine Clients 40%		0.5303	0.5308	0.5364	**0.7099**

TABLE X

ABLATION STUDY TO EVALUATE THE EFFECTIVENESS OF EACH COMPONENT ON UMMMOVIE DATASETS

ID	Text	Image	Rating Behavior	ACC	AUC	F1 score
✓	✓			0.6604	0.6994	0.6814
✓		✓		0.6994	0.6994	0.6814
✓			✓	0.7142	0.7817	0.7280
✓	✓	✓		0.7219	0.7609	0.7247
✓	✓		✓	0.7604	0.8247	0.7479
✓		✓	✓	0.7609	0.8253	0.7498
✓	✓	✓	✓	**0.7614**	**0.8257**	**0.7498**

TABLE XI

COMPUTATION COST COMPARISON OF DIFFERENT LOCAL MODELS AND TRAINING MANNERS

Local Models	Computational Complexity (FLOPS)	Parameter Quantity	Running Time (s)
Item2Vec	51200	790600	5509
NeuralCF	731136	642401	373
AutoInt	31049216	841993	12504
DCN-V2	18229248	791910	6271
AMMFRS(ours)	1268273664	4249829	6283
Training Manners	Computational Complexity (FLOPS)	Parameter Quantity	Running Time (s)
Centralized	1268273664	424982900	628300
FedAvg	63413683200	212491450	65889
Per-FedAvg	63413683200	212491450	55893
APPLE	63413683200	212491450	65955
RP^{3}FL(ours)	63413683200	212491450	70111

3) Ablation Study: The AMMFRS is composed of four basic components to realize multimodal feature extraction, including text feature, image feature, ID feature, and the behavior feature. Among them, the ID feature extraction module is used to differentiate between various contents and users and therefore is nonremovable. We offer a performance comparison of various combinations of the aforementioned modules to assess each one's effectiveness using UMMMovie datasets in terms of average recommendation accuracy. First, as presented in Table X, all of our proposed modules could contribute to enhancing the performance. Second, for unimodal recommendation algorithms, the performance gain brought by adding rating behavior data is the most important, while the gains of text and image information on recommendation performance are similar, which indicates that there is redundancy in these two modalities. It is a similar story when it comes to bimodal recommendation algorithms. It can be concluded that, given limited computing capacity, a pretty high recommendation performance could be obtained by using only the rating information and either text or image data of contents.

4) Computational Overhead and Time Efficiency: To evaluate the computational overhead and time efficiency, we also provide the performance comparison in terms of computational complexity, parameter quantity, and running time. We utilize floating-point operations per second (FLOPS), which is frequently employed in machine learning applications where floating-point operations are necessary, as a measure of computer performance when assessing computational complexity. Table XI illustrates how our algorithm's computational complexity and parameter quantity are larger than those of existing local models based on unimodal recommendation algorithms. This is reasonable given that our local training model considers information from four modalities. It is worth mentioning that because our multimodal feature extraction is carried out in parallel, our execution time is comparable to that of the two unimodal methods, namely, Item2vec and AutoInt. Additionally, all FL-based training architectures achieve reduced computational overhead and faster execution times when compared to CL training methods. This is because the multipoint distributed

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

computing framework significantly alleviates the pressure associated with single-point centralized computing tasks. Our training method can offer better privacy protection and more accurate recommendation outcomes at a minor execution time penalty as compared to previous FL-based algorithms.

VI. Conclusion

This article studies the design of a robust multimodal federated recommendation algorithm with strengthened privacy protection and a strong resistance to Byzantine attacks and inference attacks. To address the data heterogeneity issue, this article proposes the multimodal attention module based on an attention mechanism to adaptively alter the aggregation weights. Furthermore, LDP and personalized FL techniques are used to enhance the privacy of the federated recommendation model. Specifically, an adaptive modification of the learning rate strategy and penalty terms are used to prevent the local personalized model from detaching from the global model update. Furthermore, to counter potential Byzantine attacks, an adaptive weighting aggregation strategy is used to adjust the client's aggregation weight for secure aggregation. Finally, the effectiveness of the various strategy improvements is verified through comparative experiments with current mainstream algorithms. The experimental results show that the proposed algorithm ensures enhanced data privacy and model robustness while maintaining high recommendation accuracy for both movie and joker recommendation.

References

[1] D. Feng, G. Huang, C. Feng, B. Cao, Z. Wang, and X.-G. Xia, "EAPS: Edge-assisted privacy-preserving federated prediction systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Glasgow, U.K., Mar. 2023, pp. 1–6, doi: 10.1109/WCNC55385.2023.10118906.

[2] T. Li, L. Song, and C. Fragouli, "Federated recommendation system via differential privacy," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 2592–2597, doi: 10.1109/ISIT44484.2020.9174297.

[3] X. Jiang, B. Liu, J. Qin, Y. Zhang, and J. Qian, "FedNCF: Federated neural collaborative filtering for privacy-preserving recommender system," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Padua, Italy, Jul. 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9892909.

[4] C. Feng, H. H. Yang, D. Hu, Z. Zhao, T. Q. S. Quek, and G. Min, "Mobility-aware cluster federated learning in hierarchical wireless networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8441–8458, Oct. 2022, doi: 10.1109/TWC.2022.3166386.

[5] G. Lin, F. Liang, W. Pan, and Z. Ming, "FedRec: Federated recommendation with explicit feedback," *IEEE Intell. Syst.*, vol. 36, no. 5, pp. 21–30, Sep. 2021, doi: 10.1109/MIS.2020.3017205.

[6] C. Feng, H. H. Yang, S. Wang, Z. Zhao, and T. Q. S. Quek, "Hybrid learning: When centralized learning meets federated learning in the mobile edge computing systems," *IEEE Trans. Commun.*, vol. 71, no. 12, pp. 7008–7022, Dec. 2023, doi: 10.1109/TCOMM.2023.3310529.

[7] D. Chai, L. Wang, K. Chen, and Q. Yang, "Secure federated matrix factorization," *IEEE Intell. Syst.*, vol. 36, no. 5, pp. 11–20, Sep. 2021, doi: 10.1109/MIS.2020.3014880.

[8] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018, doi: 10.1109/TIFS.2017.2787987.

[9] K. Muhammad et al., "FedFast: Going beyond average for faster training of federated recommender systems," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Long Beach, CA, USA, Jun. 2020, pp. 1234–1242, doi: 10.1145/3394486.3403176.

[10] K. Wei et al., "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3388–3401, Sep. 2022, doi: 10.1109/TMC.2021.3056991.

[11] J. Xu, W. Du, Y. Jin, W. He, and R. Cheng, "Ternary compression for communication-efficient federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1162–1176, Mar. 2022, doi: 10.1109/TNNLS.2020.3041185.

[12] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 9587–9603, Dec. 2023, doi: 10.1109/TNNLS.2022.3160699.

[13] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "FedAUX: Leveraging unlabeled auxiliary data in federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 5531–5543, Sep. 2023, doi: 10.1109/TNNLS.2021.3129371.

[14] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 14774–14784, doi: 10.5555/3454287.3455610.

[15] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6103–6115, Nov. 2022, doi: 10.1109/TNNLS.2021.3072238.

[16] M. Cao, L. Zhang, and B. Cao, "Toward on-device federated learning: A direct acyclic graph-based blockchain approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2028–2042, Apr. 2023, doi: 10.1109/TNNLS.2021.3105810.

[17] M. Gong et al., "A multi-modal vertical federated learning framework based on homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1826–1839, 2024, doi: 10.1109/TIFS.2023.3340994.

[18] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 70–82, Jul. 2020, doi: 10.1109/MIS.2020.2988525.

[19] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020, doi: 10.1109/TIFS.2020.2988575.

[20] Y. Qiang, "Federated recommendation systems," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Los Angeles, CA, USA, Dec. 2019, p. 1, doi: 10.1109/BigData47090.2019.9005952.

[21] C. Wu, F. Wu, T. Qi, and Y. Huang, "MM-Rec: Visiolinguistic model empowered multimodal news recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Jul. 2022, pp. 2560–2564, doi: 10.1145/3477495.3531896.

[22] F. Wu, C. Lyu, and Y. Liu, "A personalized recommendation system for multi-modal transportation systems," *Multimodal Transp.*, vol. 1, no. 2, Jun. 2022, Art. no. 100016, doi: 10.1016/j.multra.2022.100016.

[23] L. Jin, Z. Li, and J. Tang, "Deep semantic multimodal hashing network for scalable image-text and video-text retrievals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 1838–1851, Apr. 2023, doi: 10.1109/TNNLS.2020.2997020.

[24] T. Hoang, T.-T. Do, T. V. Nguyen, and N.-M. Cheung, "Multimodal mutual information maximization: A novel approach for unsupervised deep cross-modal hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6289–6302, Sep. 2023, doi: 10.1109/TNNLS.2021.3135420.

[25] S. Pingali, P. Mondal, D. Chakder, S. Saha, and A. Ghosh, "Towards developing a multi-modal video recommendation system," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Padua, Italy, Jul. 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9892382.

[26] S. Oramas, O. Nieto, M. Sordo, and X. Serra, "A deep multimodal approach for cold-start music recommendation," in *Proc. 2nd Workshop Deep Learn. Recommender Syst.*, Como, Italy, Aug. 2017, pp. 32–37, doi: 10.1145/3125486.3125492.

[27] Z. Zhao et al., "Social-aware movie recommendation via multimodal network learning," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 430–440, Feb. 2018, doi: 10.1109/TMM.2017.2740022.

[28] H. Liu, C. Li, and L. Tian, "Multi-modal graph attention network for video recommendation," in *Proc. IEEE 5th Int. Conf. Comput. Commun. Eng. Technol. (CCET)*, Beijing, China, Aug. 2022, pp. 94–99, doi: 10.1109/CCET55412.2022.9906399.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL)*, vol. 1, Jun. 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[30] B. Koonce, "ResNet 50," in *Convolutional Neural Networks With Swift for Tensorflow*, Berkeley, CA, USA: Apress, 2021, doi: 10.1007/97814842616826.

[31] J. Liu and Y. Jin, "A comprehensive survey of robust deep learning in computer vision," *J. Autom. Intell.*, vol. 2, no. 4, pp. 175–195, Nov. 2023, doi: 10.1016/j.jai.2023.10.002.

[32] V. Ashish et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008, doi: 10.5555/3295222.3295349.

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.

[33] S. Liu, H. Zhang, and Y. Jin, "A survey on computationally efficient neural architecture search," *J. Autom. Intell.*, vol. 1, no. 1, Dec. 2022, Art. no. 100002, doi: 10.1016/j.jai.2022.100002.

[34] J. Yang, Y. Xu, H. Cao, H. Zou, and L. Xie, "Deep learning and transfer learning for device-free human activity recognition: A survey," *J. Automat. Intell.*, vol. 1, no. 1, Dec. 2022, Art. no. 100007, doi: 10.1016/j.jai.2022.100007.

[35] Q. Cui and Y. Song, "Tracking control of unknown and constrained nonlinear systems via neural networks with implicit weight and activation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5427–5434, Dec. 2021, doi: 10.1109/TNNLS.2021.3085371.

[36] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 1–11, doi: 10.5555/3294771.3294783.

[37] O. Barkan and N. Koenigstein, "ITEM2VEC: Neural item embedding for collaborative filtering," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Vietrisul Mare, Italy, Sep. 2016, pp. 1–6, doi: 10.1109/MLSP.2016.7738886.

[38] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, Perth, WA, Australia, 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[39] W. Song et al., "AutoInt: Automatic feature interaction learning via self-attentive neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1161–1170, doi: 10.1145/3357384.3357925.

[40] R. Wang et al., "DCN V2: Improved deep & cross network and practical lessons for web-scale learning to rank systems," in *Proc. Int. World Wide Web Conf. (WWW)*, Ljubljana, Slovenia, Apr. 2021, pp. 1785–1797, doi: 10.1145/3442381.3450078

[41] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Ft. Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[42] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 3557–3568, doi: 10.5555/3495724.3496024.

[43] J. Luo and S. Wu, "Adapt to adaptation: Learning personalization for cross-silo federated learning," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2022, pp. 2166–2173, doi: 10.24963/ijcai.2022/301.

Chenyuan Feng received the B.E. degree in electrical and electronics engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2016, and the Ph.D. degree in information system technology and design from Singapore University of Technology and Design (SUTD), Singapore, in 2021.

She holds three Chinese invention patents and has one edited book related to federated learning in intelligent systems. Her research interests include edge intelligence, multimedia intelligence, and federated learning for future-generation communication.

Dr. Feng was a recipient of Marie Skłodowska-Curie Global Fellowship (EU Talent Program) and 2021 IEEE ComComAp Best Paper Award. She serves as a TPC Member for 2023 IEEE the 23rd International Conference on Communication Technology, a Track Chair for 2024 IEEE the 24th International Conference on Communication Technology, and also a Committee Member for Special Committee on Metaverse in Shenzhen City Computer Federation.

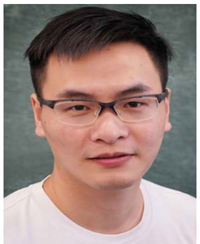
Daquan Feng received the Ph.D. degree in information engineering from the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu, China, in 2015.

He was a Visiting Student with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, from 2011 to 2014. After graduation, he was a Research Staff with the State Radio Monitoring Center, Beijing, China, and then, a Post-Doctoral Research Fellow with Singapore University of Technology and Design, Singapore. He is currently an Associate Professor with Shenzhen Key Laboratory of Digital Creative Technology, Guangdong Province Engineering Laboratory for Digital Creative Technology, and Guangdong–Hong Kong Joint Laboratory for Big Data Imaging and Communication, College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. His research interests include ultra-reliable and low latency communications (URLLC), mobile edge computing (MEC), and massive Internet of Things (IoT) networks.

Dr. Feng is an Associate Editor of IEEE COMMUNICATIONS LETTERS, *ICT Express*, and *Digital Communications and Networks*.

Guanxin Huang received the M.S. degree in information and communication engineering from Shenzhen University, Shenzhen, China, in 2023.

He is currently working with the Research and Development Department, BYD Company Ltd., Shenzhen.



Zuozhu Liu (Member, IEEE) received the B.Eng. degree from Zhejiang University, Hangzhou, China, in 2015, and the Ph.D. degree from Singapore University of Technology and Design, Singapore, in 2019.

He was a Post-Doctoral Research Fellow with the Department of Statistics and Applied Probability, National University of Singapore, Singapore. Since September 2020, he has been an Assistant Professor with Zhejiang University. His research works are published in top journals and conferences such as IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, IEEE TRANSACTIONS ON COMMUNICATIONS, *Patterns*, NeurIPS, ICLR, and ACL. His research interests include machine learning, deep learning, and artificial intelligence (AI) for healthcare.

Zhenzhong Wang received the Ph.D. degree in electromagnetic field and microwave technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2010.

Since 2010, he has been with the Technology and Management Center of China Central Television, which was renamed as China Media Group, Beijing, in 2018, where he is currently a Professorate Senior Engineer. His research interests include 4K/8K ultra high definition (UHD) production, media content distribution, and 5G transmission.

Xiang-Gen Xia (Fellow, IEEE) is currently the Charles Black Evans Professor with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, USA. He has authored the book *Modulated Coding for Intersymbol Interference Channels* (Marcel Dekker, New York, 2000) and coauthored the book *Array Beamforming Enabled Wireless Communications* (CRC Press, New York, 2023). His current research interests include space–time coding, multi-in multi-out (MIMO) and orthogonal frequency division multiplexing (OFDM) systems, digital signal processing, and synthetic aperture radar (SAR) and inverse synthetic aperture radar (ISAR) imaging.

Dr. Xia received the National Science Foundation (NSF) Faculty Early Career Development (CAREER) Program Award in 1997, the Office of Naval Research (ONR) Young Investigator Award in 1998, and the 2019 Information Theory Outstanding Overseas Chinese Scientist Award, The Information Theory Society of Chinese Institute of Electronics. He has served as an Associate Editor for numerous international journals including IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is a Technical Program Chair of the Signal Processing Symposium, GLOBECOM 2007 in Washington, DC, USA, and a General Co-Chair of ICASSP 2005 in Philadelphia, PA, USA.

Authorized licensed use limited to: SHANDONG UNIVERSITY. Downloaded on October 09,2025 at 08:59:04 UTC from IEEE Xplore. Restrictions apply.