A Federated Deep Learning Framework for Privacy-Preserving Consumer Electronics Recommendations

Jintao Wu, Jingyi Zhang, Muhammad Bilal *Senior Member IEEE*, Feng Han, Nancy Victor, and Xiaolong Xu

***Abstract*—Recommender systems (RSs) have proven to be highly effective in guiding consumers towards well-informed purchase decisions for electronics. These systems can provide personalised recommendations that consider individual preferences, past purchases and current market trends by collecting and analysing massive amounts of consumer data. However, RSs have traditionally employed centralised storage of users' consumption records and item interactions, which may potentially lead to privacy concerns. In particular, centralised data storage may prove unworkable in the future with the advent of regulations such as the General Data Protection Regulation. In turn, this can lead to an urgent need for decentralised recommendation frameworks for consumer electronics. In this study, we propose a federated learning recommender system (FRS) for the recommendation task in the consumer electronics industry. However, this is rather challenging due to its privacy protection, model scalability and personalisation requirements. First, the federated recommender system for consumer electronics (FRS-CE) adopts an outer product and two proposed feature fusion operations to construct an interaction map between users and items. Second, the FRS-CE uses a lightweight convolution operation to extract high-order features from the interaction map. Finally, the proposed model employs an adaptive aggregation mechanism to update the global model, which enhances the scalability of the system. Extensive experiments conducted on two real-world datasets have demonstrated the effectiveness of the FRS-CE in generating consumer electronics recommendations with privacy protection.**

***Index Terms*—Data privacy, federated learning, convolutional neural network, consumer electronics recommendation**

I. INTRODUCTION

Over the past few years, consumer electronics have become an essential part of people's daily lives, transforming the way people interact with the world. Smartphones, tablets, laptops, wearable devices, digital cameras and other products have played increasingly important roles in meeting our daily requirements for entertainment, communication, work, study and other aspects. However, with the continuous expansion of the consumer electronics market and the advent of the era of big data, users are faced with an increasing number of options and information resources when purchasing these products. Unfortunately, such a wide array of information to choose from often leads to decision-making difficulties and unsatisfactory outcomes in terms of quality. Recommender systems (RSs) have proven to be a practical technology [1] to efficiently extract insightful information from enormous volumes of data [2]. The RS development strategy involves the accurate prediction of user preferences and item ratings to assist them in discovering items that may be of interest to them [3][4]. Currently, the applications of RSs are widespread and include several industries, namely, entertainment [5], e-commerce [6], news [7], e-learning [8], healthcare [9] and many others. Thus, it has become a crucial component of many large companies, such as Google, Facebook, Amazon and Netflix [10].

Traditional RSs typically make recommendations by analysing past user behaviours or identifying similarities between users/items, which are mainly categorised into collaborative filtering (CF), content-based filtering and hybrid RSs [11]. CF is a popular approach used in RSs to recommend items to users by identifying other users who have similar preferences and tastes [12]. Content-based filtering analyses the characteristics of items preferred by users and recommends similar items based on those features [13]. Hybrid RSs leverage the strengths of different recommendation techniques by combining two or more algorithms, resulting in more effective and precise recommendations for users [14]. Although traditional RSs have achieved a certain level of success, their applications in recommending consumer electronics remain limited due to some drawbacks.

1) Data Privacy [15]: Traditional RSs usually operate on a

Manuscript received 27 April 2023; revised 6 July 2023 and 26 September 2023; accepted 12 October 2023. This research was supported by the Natural Science Foundation of Jiangsu Province of China (No. BK20211284), the Financial and Science Technology Plan Project of Xinjiang Production and Construction Corps (No. 2020DB005), the Natural Science Foundation of China (No. 62177014), the General Program of Basic Science Research in Higher Education Institutions of Jiangsu Province (Natural Sciences) (No. 23KJB520020), and the Construction and Application Demonstration of city-oriented Epidemic Prevention and Control Decision Support Platform (No. 2021AB034).

(Corresponding author: Muhammad Bilal)

Jintao Wu, Jingyi Zhang and Xiaolong Xu are with the Department of School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: wjt@nuist.edu.cn; jy_zhang0114@163.com; xlxu@ieee.org).

Muhammad Bilal is with the School of Computing and Communications, Lancaster University, Bailrigg, Lancaster LA1 4WA, United Kingdom (e-mail: m.bilal@ieee.org).

Feng Han is with the Department of the College of Information Science and Technology, Shihezi University, Shihezi 832003, China (e-mail: hf_inf@shzu.edu.cn)

Nancy Victor is with the School of Information Technology and Engineering, Vellore Institute of Technology, India (e-mail: nancyvictor@vit.ac.in)
.

Digital Object Identifier XXXX

centralised cloud computing architecture that requires centralised storage and analysis of massive amounts of item information and user consumption data. However, this architecture is faced with a common issue: it is challenging to share data among different consumer electronics platforms while maintaining commercial competitiveness without risking the privacy of users and commercial secrets.

2) Model Scalability [16]: As the number of users and the sales of consumer electronics continue to grow, traditional RSs become burdened with processing increasing amounts of user behaviour data and item information. This overload results in a significant increase in computational overhead, which makes it challenging to maintain the performance and efficiency of the recommendations.

3) Cold Start [17]: Traditional RSs struggle to provide effective recommendations in the absence of adequate user or item historical data. This issue is a common problem, especially in consumer electronics markets with fast replacement rates, in which the cold-start problem of recommendations is more noticeable. Thus, applying traditional recommendation algorithms directly to recommend consumer electronics becomes difficult due to the aforementioned challenges. Therefore, new algorithms and techniques must be developed to address the unique characteristics and requirements of consumer electronics RSs. Specifically, these algorithms and techniques must have the capability to extract useful information accurately and effectively from heterogeneous or sparse data sources, enhance the scalability of the RSs to handle large-scale and real-time scenarios and mitigate the cold-start problem caused by limited historical data while protecting the privacy of merchants and users.

Federated learning (FL) has emerged as a highly promising approach for training machine learning (ML) models while ensuring the privacy and security of user data [18]. FL coordinates multiple clients to execute model training without sharing their raw data, thereby achieving effective global model training. In FL, each device trains an ML model locally using its own dataset, after which the central server aggregates the learned parameters retrieved from multiple devices [19]. In recent years, the continuous maturation of mobile edge computing (MEC) technology has provided technical support for reducing the training latency of FL. In particular, MEC leverages distributed edge networks to perform computational tasks or applications near edge users, thus removing the requirement of cloud computing [20]. Edge computing provides advantages for FL development by enabling local computing on devices, thereby reducing the risk of data transmission and privacy breaches [21]. Moreover, edge computing can enhance the efficiency of model training and inference, thus leading to reduced bandwidth overhead and latency. At present, FL has become one of the key technologies used in solving privacy protection for cloud-edge intelligent collaborative computing [22]. Thus, recommending consumer electronics based on edge computing technology and an FL strategy presents a promising approach.

In the present study, we propose a novel federated RS for consumer electronics called FRS-CE. Its core principle is based on FL and cloud-edge collaboration, both of which are widely employed in RSs. Specifically, we use a lightweight CNN (MobileNet) at the edge to capture useful nonlinear relationships between users and items. In the proposed method, the convolutional layer is actually applied on the interaction map describing the relationship between users and items to obtain high-order features. In this paper, we attempt to incorporate the features of users and items into the interaction map and enhance its expressive power by proposing two different feature fusion mechanisms. Instead of uploading all models as in traditional FL, we suggest implementing an adaptive aggregation mechanism during the model update phase, allowing for the selective uploading of edge-trained models. The proposed method effectively alleviates the cold-start problem in consumer electronics recommendations through the fusion mechanism. At the same time, the method can reduce the energy consumption of federated model training and the computing overhead of the edge through the adaptive aggregation mechanism. The main contributions of our work are summarised as follows:

- The proposed FRS-CE can leverage distributed data across multiple edge devices for prediction and modelling without leaking user/item privacy. In addition, our designed adaptive aggregation mechanism (AAM) effectively reduces the computational cost during the training process, thus achieving model scalability.
- We propose two feature fusion techniques, namely, horizontal fusion (HF) and vertical fusion (VF), which integrate high-dimensional user and item features into the user–item interaction map obtained by outer product operation, thus mitigating the cold-start phenomenon. Furthermore, the proposed method applies a lightweight CNN to the obtained interaction map to efficiently capture useful nonlinear relationships between users and items.
- We conducted extensive experiments on two real-world datasets, and the systematic experimental results demonstrate that our FRS-CE is an effective approach for providing accurate recommendations for consumer electronics while protecting user privacy.

The remainder of this paper is organised as follows. To facilitate the explanation that follows, we will first discuss some related works on RSs in Section II. In Section III, we describe the algorithm and implementation of the proposed FRS-CE in detail. Subsequently, in Section IV, we perform various experiments presented to verify the superiority of our proposed FRS-CE. Finally, we present the conclusion in Section V.

II. RELATED WORK

A. Recommender System

RSs are software tools and algorithms that assist users in finding and selecting items that align with their preferences and interests [10]. By analysing user feedback data on various items, RSs predict user preferences or ratings, with the ultimate goal of providing each user with targeted recommendations to improve their overall experience. Traditional recommendation methods can be classified as CF, content-based filtering and hybrid recommendation techniques.

First proposed by Goldberg et al. [23] in 1992, CF utilises the connections between users or items, such as through sharing, favouriting and comments, to calculate the similarities and recommend the most relevant items to users. When Amazon applied this method based on the improved CF model proposed by Linden et al. [24], it quickly led to increased sales for the company. However, as mentioned earlier, CF also has limitations, such as data sparsity, scalability and 'grey sheep' [25]. To address the limitations, researchers have proposed content-based filtering techniques that involve analysing the content of users' favourite items and recommending similar items based on that analysis [26]. One application of content-based CF can be seen in the RS proposed by Wang et al. [27], which helps authors decide where to submit their manuscripts. In particular, their system recommends a conference or journal for a researcher based on the similarities between other published papers and the user's manuscript. However, this approach relies solely on a user's own characteristics and can suffer from the cold-start problem when dealing with new users who have yet to provide sufficient data for analysis. By integrating multiple recommendation techniques, a hybrid RS can compensate for the limitations and biases of each individual approach. For instance, if one technique struggles to generate accurate predictions for certain users or items, other techniques can fill in the gaps and produce more precise recommendations, resulting in a more robust RS [28]. Despite their advantages, hybrid RSs may face challenges when attempting to effectively incorporate certain types of content information, such as textual descriptions or visual images associated with users or items [16]. Consequently, it may be necessary to use other specialised recommendation approaches to accurately capture and leverage these types of information.

B. Deep Learning for RSs

There has been a surge of interest in deep learning-based RS models in recent years [29]. These models leverage powerful neural network architectures to analyse and learn from large-scale datasets, enabling them to determine and exploit complex relationships among users, items and other relevant features. Consequently, deep learning techniques have shown great promise in enhancing the accuracy, scalability and robustness of RSs across various domains and applications. In fact, empirical studies have consistently shown the superior performance of deep learning-based RSs compared with traditional approaches [30].

Recently, many CF-based techniques have been enhanced by incorporating deep learning, thus giving rise to different models, such as NCF [31], DMF [32], SocialCDL [33], CoDAE [34] and ECAE [35]. These models leverage the rating matrix to learn latent features that capture user–item interactions and use these to predict ratings for unseen user–item pairs. For example, NCF is an approach that combines neural networks and matrix factorisation to learn CF models. NCF uses the embedding vectors generated by matrix factorisation as inputs for a deep neural network instead of representing users and items as separate feature vectors. This network utilises a multilayer perceptron to model the latent factors of user–item interactions in the high-dimensional space. These methods can better capture nonlinear and complex patterns that exist in user–item data by combining CF with deep learning, thus achieving improved recommendation accuracy and performance.

The deep learning-based RSs discussed earlier typically adopt multilayer perceptrons to process user and item features separately without considering their interactions [36]. In the past few years, there have been several efforts to create RSs that can capture direct interactions between users and items using CNN architectures, such as ConvNCF [37] and CFM [38]. These systems have shown promising results in terms of improving the accuracy and effectiveness of recommendations. For instance, to represent user–item interactions in ConvNCF, an interaction map can be created by using the outer product matrix between the embedding vectors of users and items. This map is then fed into a convolutional layer where a generic filter is applied to learn higher-order features. Furthermore, thanks to the powerful high-order connection modelling capability of graph neural networks (GNNs) [39], graph-based RSs, such as NGCF [40] and CVGA [41], have attracted widespread attention and produced several research results that are far superior to traditional neural network-based CF models.

However, these methods typically store user/item information centrally in the cloud, resulting in the leakage of user privacy data. Additionally, according to strict privacy protection under the General Data Protection Regulation, the use of centralised data storage may not be feasible in the future, urging a decentralised framework of recommendation.

C. FL for Recommender System

The concept of FL [42] was first introduced by Google in 2016. FL is a method of distributed ML in which the model learning process is distributed across multiple client devices. Through the combination of multiple local models, a global model can be created without exposing the user's private data to the server, thus improving user privacy [43]. In recent years, FL-based models have been studied in the field of RSs, in which federated CF (FCF) [19] and FedMF [44] are two pioneering works. Later, researchers developed a federated RS called A-FRS [45], which is capable of resisting poisoning attacks from clients. One of its key techniques is the use of an item similarity model for learning user/item embeddings. For social recommendation tasks, Liu et al. [43] proposed a new FL framework called FeSoG, which comprehensively integrates clients' local user privacy data. This approach effectively tackles the three challenges of data heterogeneity, communication privacy protection and demand for individualised modelling. FGC [15] is the most recent work that combines GCN [46] with a federated RS, which collects parameters from participating clients and returns aggregated results to them for knowledge sharing and training acceleration. The central server ensures data privacy by keeping the privacy-sensitive scenario embedding local. This is achieved by only collecting the model weights and service embedding. Moreover, this approach significantly improves accuracy compared with local-only approaches.

In recent years, the integration of FL and RSs in edge computing environments has brought numerous benefits. These benefits include reduced latency and increased responsiveness due to localised processing [47], enhanced privacy protection by keeping sensitive data on edge devices [48], improved scalability and resource efficiency through distributed computing at the edge [49] and customised recommendations enabled by leveraging contextual information from nearby edge devices [50, 51, 18]. Overall, combining FL and RSs in edge environments offers numerous advantages, such as faster response times, personalised experiences and enhanced privacy, scalability and efficiency.

III. PROPOSED METHODS

A. Problem Formulation

A typical recommendation scenario includes M users and N items. A user–item interaction matrix $R \in \mathbb{R}^{M \times N}$ is obtained from users' implicit feedback:

$$R_{ui} = \begin{cases} 1, & \text{if interaction (user } u, \text{item } i\text{) is observed;} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

If the value of R_{ui} is 1, then there is an interaction (e.g. purchase history, click history or rating data) between user u and item i, but this does not necessarily mean that user u likes item i, and vice versa. Obtaining implicit feedback is less difficult than obtaining explicit feedback; however, it makes recommendation more challenging due to the uncertainty in interpreting the user's behaviour. The main purpose of the RS is to estimate users' preferences for items by analysing their historical behaviour information and providing personalised recommendations for different users. In fact, the recommendation problem can be formulated as a problem of estimating the scores of unobserved items in R, which are used to rank the items. As previously mentioned, the precision of score prediction in RSs is not exclusively affected by implicit feedback but also significantly depends on the inherent characteristics of users/items. Thus, incorporating user/item features (content-based recommendation), apart from conventional CF approaches that consider implicit/explicit feedback, has the potential to augment the efficacy of recommending consumer electronics. For example, varying age cohorts manifest diverse preferences and brand recognition towards consumer electronics. Consequently, the current research proposes a hybrid RS that combines implicit feedback with user–item features for consumer electronics recommendations. The task addressed in this paper is defined as follows:

Input: Observed user vector v_u (u-th row of R), observed item vector v_i (i-th column of R), user attribute a_u and item attribute a_i.

Output: The predicted rating value pr_{ui} of user u for item i.

B. General Framework

The framework of the proposed FRS-CE method primarily consists of three modules *(Feature Fusion, Convolution Operation and Model Update), as* illustrated in Fig. 1. The *Feature Fusion* module utilises user and item information collected by local base stations to explicitly model the pairwise correlations between dimensions in the embedding space and fuse user and item features. The *Convolution Operation* module uses a lightweight convolution operation to extract high-dimensional semantic features from the constructed interaction map. In doing so, the module can contribute to the generation of a local lightweight model. The *Model Update* module is responsible for sending the local model trained on the edge to the cloud. Its function is to perform aggregation and to send the updated model to the participating edge to continue iterative training.

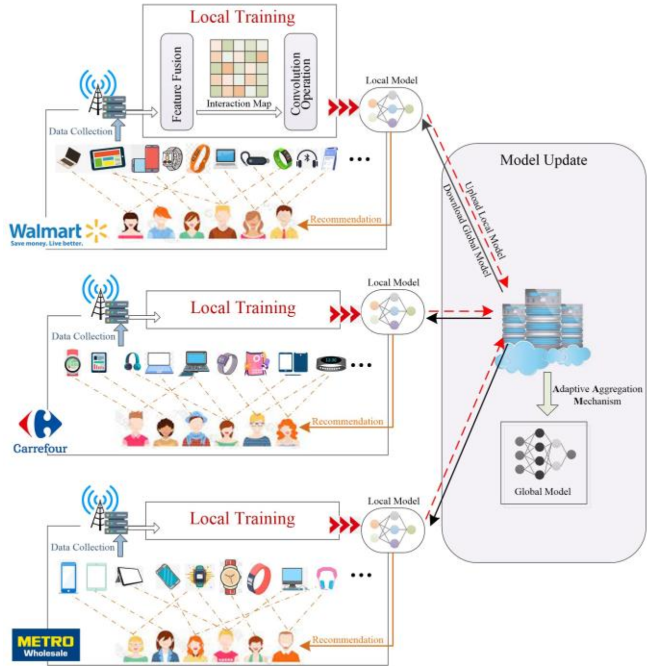


Fig. 1. Architecture of the FRS-CE.

C. Interaction Map Generation

Given a user u and item i, we use an observed user vector $v_u = \{R_{u1}, R_{u2}, \ldots, R_{uN}\}$ (u-th row of R) to represent u and an observed item vector $v_i = \{R_{1i}, R_{2i}, \ldots, R_{Mi}\}$ (i-th column of R) to represent i. Next, we improve the feature expression ability of high-dimensional binary sparse vectors v_u and v_i by projecting them to dense vectors through the following fully connected layer:

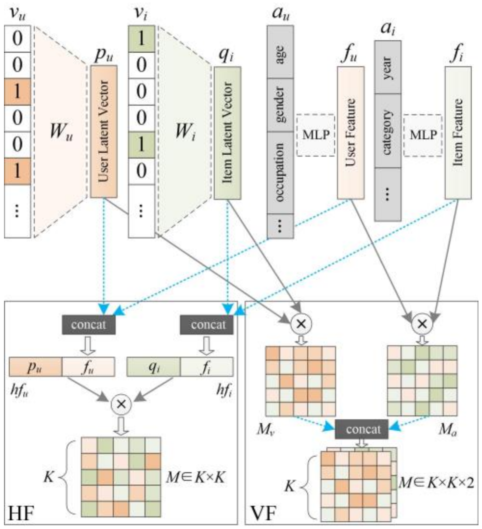
$$p_u = F_u(v_u \mid W_u) \quad (2)$$

$$q_i = F_i(v_i \mid W_i) \quad (3)$$

where $W_u \in \mathbb{R}^{M \times K}$ ($W_i \in \mathbb{R}^{N \times K}$), denoting the model parameters of the single-layer neural network F_u (F_i) for user u (item i). Therefore, the obtained user (item) embedding p_u (q_i) can be regarded as a K-dimensional user (item) latent vector.

Similar to ONCF [37], in the current paper, we still use the outer product of p_u and q_i to obtain an interaction map $M \in \mathbb{R}^{K \times K}$. The purpose of this step is to consider the correlations between the embedding dimensions, thus enriching the information contained in the features. Furthermore, this $K \times K$ matrix is more convenient for the subsequent convolution operation [37].

However, the above method does not consider the cold-start problem. When a batch of new users (new items) joins, their v_u (v_i) are all zero vectors (i.e. they do not contain any interaction information). This can lead to the same interaction maps generated by different types of new users (items) and directly affect users' ratings for items. To solve this problem, the current study combines the attributes of users (e.g. age, gender and occupation) and items (e.g. year and category) into the interaction map. According to different concatenating directions, we propose two feature fusion methods listed below.



(a) Horizontal Fusion (b) Vertical Fusion

Fig. 2. Process of feature fusion.

HF. In this method, a single-layer neural network is used to extract user feature $f_u = \{f_{u1}, f_{u2}, \ldots, f_{uK/2}\}$ and item feature $f_i = \{f_{i1}, f_{i2}, \ldots, f_{iK/2}\}$ from user attribute a_u and item attribute a_i, respectively. Then, we horizontally concatenate f_u (f_i) to p_u (q_i) to obtain horizontal user features $hf_u = \{ f_{u1}, \ldots, f_{uK/2}, p_{u1}, \ldots, p_{uK/2}\}$ ($hf_i = \{f_{i1}, \ldots, f_{iK/2}, q_{i1}, \ldots, q_{iK/2}\}$). Finally, as shown in Fig. 2(a), an interaction map with a scale of $K \times K$ is obtained by performing the outer product operation on hf_u and hf_i.

VF. This method also uses a single-layer neural network to extract the user feature f_u and item feature f_i, with a doubling of their respective scales. Then, a feature interaction map M_f with a scale of $K \times K$ is obtained by directly performing outer product operations on f_u and f_i. Finally, M_f and the outer product of p_u and q_i are stacked vertically to form an interaction map with a scale of $K \times K \times 2$, as shown in Fig. 2(b).

Next, we train user (item) embedding p_u (q_i) and features f_u (f_i) separately, without sharing their respective neural network layers, to provide more flexibility to the interaction map. We denote the result of feature fusion as M_{ui} for the convenience of description.

D. Convolutional Operation

In this section, we use hidden layers to extract high-level semantic information and predict ratings from an interaction map M_{ui} that describes the relationship between user u and item i. The hidden layer discussed here can be abstracted as $r_{ui} = F_\Theta(M_{ui})$, where F_Θ represents the hidden layer model with parameter Θ, and r_{ui} represents the final prediction score of user u for item i. Technically speaking, F_Θ can be designed as any function that takes a matrix as input and outputs a value between 0 and 1. A straightforward approach is to use a multilayer perceptron (MLP), such as the NCF framework [31]. However, a large amount of training data is required for learning a good model using MLP, and the edge may not be able to provide sufficient data due to the consideration of user privacy and the limitation of edge storage resources. At the same time, MLP has the disadvantages of many parameters and large models. For instance, when the output scale is 32 × 32, and the input is a 64 × 64 matrix, MLP needs more than 4 million parameters (here, the matrix needs to be flatted into a vector), which can affect the upload (or download) speed of the local model (during federated training). To some extent, the emergence of the convolution-based NCF framework (ONCF [37]) alleviates the above problems because CNN uses much fewer parameters than MLP, owing to the fact that it can stack layers in a locally connected manner.

The structure of hidden layers F_Θ is crucial to the construction of RSs, especially when performing distributed federated training. This is because complex hidden layers can lead to slow model aggregation and high transmission costs, whereas lightweight hidden layers often lead to underfitting of training and affect rating prediction.

MobileNet [52] is a lightweight convolutional neural network whose main goal is to reduce the size and computational complexity of the model as much as possible while maintaining its accuracy. The design idea of MobileNet is to replace the traditional convolutional layer with a depthwise separable convolution layer—a feature that can greatly reduce the amount of computation and model size. Furthermore, its lightweight network structure enables MobileNet to be easily deployed on mobile devices with limited computing resources, thus providing technical support for the development of edge intelligence [52].

As shown in Fig. 3, we use the MobileNet as the backbone network to extract higher-order information from the aforementioned interaction map and achieve rating prediction, thereby reducing the computational complexity at the edge. However, there are many improved versions of MobileNet (e.g. MobileNet V4); the classic version is used in the present study to demonstrate the efficiency of the proposed method. In each layer, our model applies a single filter (size: 2 × 2) to each input channel using a depthwise convolution (the input of the first layer is an interaction map), after which a linear combination of the output of the depthwise layer is created by using 32 simple 1 × 1 convolutions (pointwise). After each convolution, batch normalisation and rectified linear unit (ReLU) are used. In this work, we normalise the results of the last layer to a probability between 0 and 1 by using the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$, which is employed to represent the user's preference for the item.

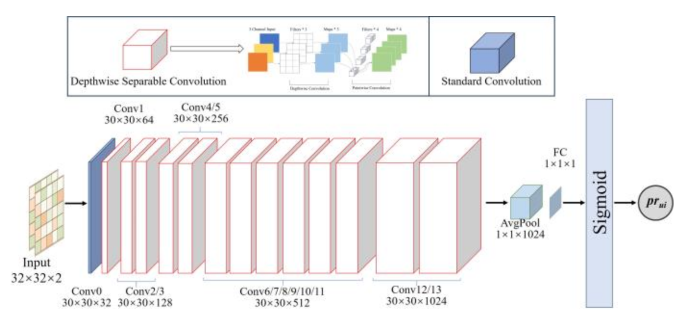


Fig. 3 Lightweight recommendation model deployed at the edge (using vertical fusion as input).

Compared with ONCF, our model reduces the computation and the number of parameters of $\frac{1}{N} + \frac{1}{k^2}$ in each layer, where

k^2 represents the size of the depthwise convolution kernel, and N represents the number of channels of the output. In our experiment, we set the output channel of each layer to 32 and the kernel size to 2 × 2. Therefore, the computational complexity and the number of parameters per layer have been reduced by approximately 28% compared with ONCF. This means that the scalability of the recommendation model for consumer electronics products has been enhanced to a certain extent.

E. Model Update Based on AAM

Based on the high privacy requirements of each edge device, our approach follows the basic FL mechanism. The core idea is to conduct distributed model training on multiple edge devices with local data without exchanging any of the local data. In this case, by only exchanging local model parameters or intermediate results, we are able to construct a global recommendation model based on virtual fused data. Specifically, in our approach, each edge device uses the private dataset it holds to train a local model, after which it sends the local model parameters to the cloud server for aggregation and proceeds to update the global model. Next, the cloud server sends the updated global model as a new round of shared models to participating edge devices for new iterative training, and the training ends when the global model converges or reaches a certain recommendation accuracy.

In our research, we find that the communication between the edge device and the cloud server is the main factor affecting the efficiency of FL. The distance between the mobile edge device and the cloud server is usually far. Moreover, FL requires multiple rounds of training, thus requiring more communication time and energy consumption. Here, the communication time and energy consumption mainly involve the two modules of model transmission and cloud parameter aggregation. Let T and E represent the transmission time and energy consumption after each round of training. This can be formalised as (4) and (5):

$$T = 2 \cdot \max\{e_t^1, e_t^2, ..., e_t^N\} + C_c \tag{4}$$

$$E = \sum_{i=1}^{N} 2 \cdot e_t^i \cdot (ep_t^i + cp_t) + C_c \cdot cp_c \tag{5}$$

where the first term of T represents the model upload and download delay, e_t^i is the transmission parameter delay of the i-th edge, and C_c is the cloud parameter aggregation delay. In addition, ep_t^i is the transmission power of the i-th edge, cp_t is the transmission power of the cloud, cp_c is the computing power of the cloud, and N is the number of edge devices participating in the federated training.

During the traditional FL process, model aggregation entails the server gathering local models from all participating devices and subsequently aggregating their model parameters. This approach involves considering all local models contributed by each device. However, this process is vulnerable to interference from noisy or low-quality models. Yet, despite failing to meet verification standards, these subpar models must be uploaded to the cloud server and included in the construction of the global model. Unfortunately, this inclusion can undermine the performance of other high-quality models, hinder the convergence of the global model and result in raised energy consumption for both transmission and computing tasks. Thus, we tackle these challenges by embedding a verification module at the edge and using local data to verify the local model. Given that only verified local models can be uploaded to the cloud, this process helps filter out noisy models. As a result, the model aggregation process within the cloud environment exclusively involves verified local models, which are referred to as the AAM.

The delay of edge transmission and cloud parameter aggregation with an embedded verification module can be formalised as (6) and (7):

$$e_t^i = \frac{(F_\theta^i + MLP_\theta^i) \cdot s_{min}}{\min\{e_v^i, c_v\}} \cdot val_e^i(d^i) \tag{6}$$

$$C_c = \sum_{i=1}^{N} \frac{F_\theta^i + MLP_\theta^i}{t_{min}} \cdot val_e^i(d^i) \tag{7}$$

where F_θ^i and MLP_θ^i respectively represent the parameter size of the lightweight convolutional neural network and multilayer perceptron involved in the proposed method; s_{min} represents the minimum time required to transmit one unit of data; e_v^i and c_v denote the data transmission rates at the edge and cloud, respectively; t_{min} denotes the minimum time for a single calculation; and val_e^i and d^i represent the verification function and verification data on the i-th edge, respectively. In fact, the result of val_e^i is either 1 or 0, representing the option of whether to upload the local model or not. We formalise this as (8), where r represents the current training round. This approach increases the validation threshold with increasing training rounds, which means that the conditions for model uploading are becoming more stringent. Given that high-quality model parameters are extensively explored in the early stages of training and then fine-tuned for accelerated convergence in the later stages, this approach is in line with actual practice.

$$val_e^i(d^i) = \begin{cases} 1 & \text{if MAE} < k / r^2 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The cloud server will perform model parameter aggregation on all collected local models. The proposed method obtains the next round of the shared global model by employing the conventional aggregation operation of the average algorithm FedAvg [32]. Notably, this aggregation specifically involves the validated local models rather than all the local models contributed by the participants. Finally, the cloud server sends the obtained global model to all participating edge devices for updating local models and starting new training.

F. Implementation Details

To optimise the cross-entropy objective function, we employ the FRS-CE algorithm using a batch size of 32 and a learning rate of 0.001, after which we perform training using the Adam [53] optimiser. The training epoch is set to 50. Initially, we generate latent embedding vectors $p_u(q_i)$ of dimensions N × K (for items, M × K) for each participating user. Subsequently, we used a one-layer MLP to extract features from users and items, thus mapping the feature vectors (f_u and f_i) to the same dimensions as the embedding vectors (p_u and q_i). The embedding vector dimensions are set to either 16 or 32, depending on whether we choose HF or VF in subsequent steps, as elaborated in Part III of this paper. Next, we use HF or VF operations to obtain interaction maps of dimensions K × K × 1 (K × K × 2). These maps are then fed into a neural network based on MobileNet to obtain rating predictions, after which the model parameters are updated through backpropagation. Each edge node is equipped with the same randomly initialised FRS-CE model at the start of training. Then, we shuffle all observed interactions at each edge node and sequentially obtain a minibatch of data in each epoch. The model is trained on the minibatch data of the edge node during each epoch. The model is uploaded to the cloud centre for model aggregation if the training result satisfies the constraints in Equation (8). The aggregated model is then transmitted back to the edge nodes before the start of the next epoch to update their model parameters. If no edge node uploads model parameters to the cloud centre for more than three epochs, then we consider the model to have reached the fitting state. At this point, the model training should be terminated.

IV. EXPERIMENTS

A. Datasets and Experiment Setup

TABLE I
DATASET INFORMATION

Dataset	Interaction	User	Item	Sparsity
MovieLens-1M	1000209	6040	3706	95.53%
Amazon-Electronics	972961	15876	52471	99.88%

We employed two benchmark datasets for recommendation in our experiments: MovieLens-1M and Amazon-Electronics. The characteristics of the two datasets are summarised in TABLE I. MovieLens-1M is a widely used classical benchmark for evaluating recommendation models, consisting of approximately 1 million explicit ratings from 6040 users for 3706 movies. For user and movie information, we selected the user's age, gender and occupation, as well as the year and category of the movie. Then, we mapped the information into numerical data to extract features. Amazon-Electronics is a subset of Amazon-Review and does not include feature data. Hence, to validate the effectiveness of our proposed algorithm, only rating data were employed, and users who had less than 20 interactions were excluded from the analysis. The ratings of the two datasets ranged from 1 to 5 as an integer value. To ensure consistency with the range of the model output, we scaled down the score by a factor of 5. To simulate different clients within the federated framework, we randomly divided the dataset into three parts, each representing the edge nodes' local data. Moreover, we split the data on each edge node: 10% were used as test data, 10% of the other 90% were used as validation data, and the remaining were used as training data.

To ensure fair comparisons, we followed the original authors' recommendations to set the embedding dimension of NCF to 32 and the embedding dimension of ConvNCF to 64, as the models all have different structures. Next, we set the embedding dimensions of the two models to 16 and 32, respectively, to ensure that the dimensions of the interaction map input to the MobileNet layers of the two models(FRS-CE(HF) and FRS-CE(VF)) were 32 × 32. In addition, to generate a single-channel interaction map to validate the effectiveness of the model, we opted to use 32-dimensional embeddings due to the absence of feature data in the Amazon-Electronics dataset. We utilised the Adam optimiser with a 0.001 learning rate and initialised parameters using Kaiming initialisation [54]. All of the models used ReLU and Sigmoid activation functions in the hidden and output layers, respectively. We also implemented our FRS-CE in PyTorch. All experiments were conducted on a machine running Ubuntu 18.04 LTS with Intel Core i7-9700K CPU 3.60 GHz, 32 GB of RAM.

B. Compared Approaches and Metrics

We implemented the following approaches in the same experimental setting to demonstrate the effectiveness of our proposed approach FRS-CE:

- **NCF** [31]**:** This approach feeds to the standard MLP for learning the interaction function by only concatenating user embedding and item embedding.
- **ConvNCF** [37]**:** This approach utilises convolutional layers and pooling layers to extract features of users and items and then fuses them into a fully connected layer for prediction.
- **FedNCF:** This approach integrates the NCF model with FL, thus enabling local training of model parameters at edge nodes. These are subsequently uploaded to the cloud for aggregation and model updating.
- **FedConvNCF:** This approach integrates the ConvNCF model with FL, thus enabling local training of model parameters at edge nodes. These are subsequently uploaded to the cloud for aggregation and model updating.
- **FRS-CE(HF):** This approach, first introduced in Section 4, involves horizontally concatenating f_u (f_i) and p_u (q_i), followed by creating an interaction map via the outer product.
- **FRS-CE(VF):** This approach is similar to FRS-CE(HF) but first forms an interaction map by taking the outer product of f_u and p_u, which is then vertically concatenated to create the final interaction map.

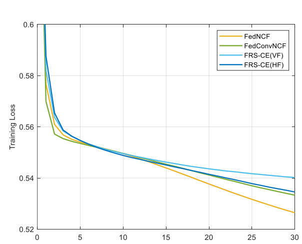
In our experiments, we employed MAE and RMSE as evaluation metrics, which are widely utilised in regression prediction. Each experiment was repeated at least 10 times to ensure the reliability of our results, and the final result was obtained by taking the average value. In the nonfederated scenario, the NCF and ConvNCF models were trained independently on each edge node, and their test results were simply averaged. In comparison, in the federated scenario, both FedNCF and FedConvNCF models were trained and tested using the same methods as our proposed model.

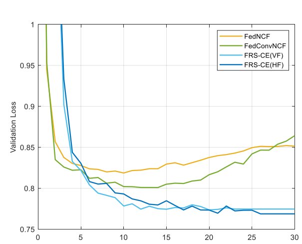
C. Results

TABLE II
EXPERIMENTAL RESULTS

Dataset	Model	RMSE	MAE
MovieLens-1M	NCF	0.9184	0.7323
	ConvNCF	0.9051	0.7195
	FedNCF	0.8952	0.7118
	FedConvNCF	0.8836	0.7002
	FRS-CE(HF)	**0.8747**	**0.6918**
	FRS-CE(VF)	**0.8691**	**0.6884**
Amazon-Electronics	NCF	1.0437	0.7583
	ConvNCF	1.0083	0.6747
	FedNCF	1.0069	0.6973
	FedConvNCF	0.9574	0.6622
	FRS-CE	**0.9467**	**0.6528**

The results of the experiments are summarised in TABLE II, which shows that the proposed FRS-CE method is highly competitive. Compared with other methods, the proposed method achieved lower RMSE and MAE, including the method with HF and VF. According to the experimental results, non-federated learning recommendation methods such as NCF and ConvNCF had the worst performance in different datasets. This is because their local model parameters cannot be shared and aggregated, resulting in the underfitting of the local model due to insufficient data. In comparison, FedNCF and FedConvNCF performed slightly better because they were directly applied to the federated framework. This result allowed the sharing of model parameters across different edge devices and improved model accuracy. However, because we considered user and item features and used CNN to extract more valuable high-order information, our two methods maintained the best recommendation performance. Notably, using AAM for model aggregation instead of traditional global aggregation improved the scalability of the model. These results provide two meaningful conclusions: (1) the interaction between users and items, as well as the features of users and items, are crucial for predicting user–item ratings, and (2) convolutional operations effectively extract useful features from the interaction map between users and items.





(a) Training data (b) Validation data

Fig. 4. Behaviour of loss with respect to change for each model's training data and validation data.

From the perspective of RMSE and MAE, our method slightly outperformed the other two FL methods in terms of recommendation performance. To comprehensively explore the convergence performance of these four FL methods, we summarised the behaviour of the MSE loss function relative to the epoch for each model with the best hyperparameters in Fig. 4. The x-axis represents the epoch, and the y-axis represents the MAE loss of the ratings. In particular, Figs. 4(a) and 4(b) demonstrate how the loss changes during training and validation on the MovieLens dataset, respectively. Although the MAE values of our two methods were higher than those of FedNCF and FedConvNCF during training, our MAE values were smaller than theirs during validation. Furthermore, the loss of FedNCF and FedConvNCF also slightly increased during validation as the number of epochs increased, indicating that these two methods had overfitting issues. Such issues pose a risk of recommending anomalies for consumer products. In contrast, our two methods showed more stable performances during training. Therefore, they are expected to provide a better user shopping experience.

TABLE III
EXPERIMENTAL RESULTS OF NEW USERS (10%)

Dataset	Model	RMSE	MAE
MovieLens-1M	NCF	1.0635	0.8639
	ConvNCF	1.0442	0.8547
	FedNCF	1.0438	0.8545
	FedConvNCF	1.0242	0.8434
	FRS-CE(HF)	**1.0114**	**0.8380**
	FRS-CE(VF)	**1.0095**	**0.8241**
Amazon-Electronics	NCF	1.3617	1.0088
	ConvNCF	1.3254	1.0865
	FedNCF	1.2413	0.9869
	FedConvNCF	1.2092	1.0304
	FRS-CE	1.2337	1.0639

Table III shows the experimental results regarding the cold-start problem after we sampled 10% of user (item) information from two datasets as new users. The results indicated that the traditional method still achieved the worst performance, followed by FedNCF. In comparison, FedConvNCF obtained the best prediction. Our two methods showed closer performance to the FedConvNCF method. There are three possible reasons for this phenomenon:

1) Insufficient data: We only sampled a small portion of the data to train the model due to the limitation of computing power, making it difficult for the model to find the correct features and solve the cold-start problem.

2) Insufficient feature extraction: Lightweight convolution models may not be able to capture the deep correlation between users and items.

3) Incomplete parameter aggregation: Our method used AAM for selective model uploading, which may miss excellent parameters. Nevertheless, our method still performed the

closest to the optimal method in terms of dealing with cold-start problems. This result indicates that the proposed method can alleviate this problem to some extent.

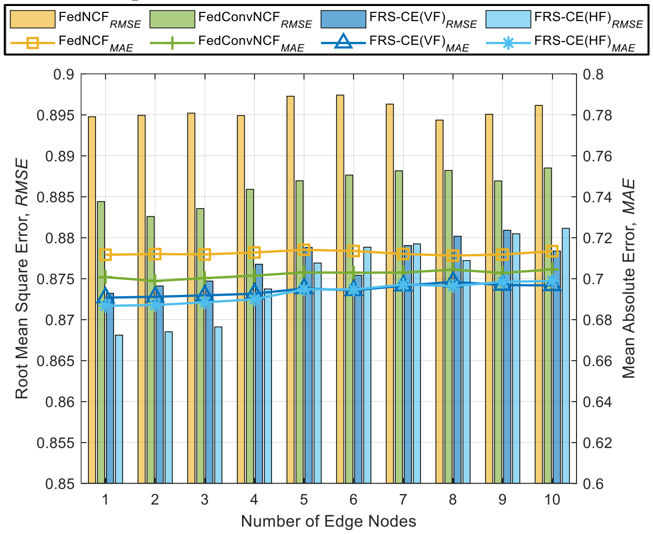


Fig. 5. Performance comparison of recommendation models based on FL across different numbers of edge nodes.

We conducted a series of experiments across different numbers of edge nodes to investigate the effectiveness of recommending consumer electronics products. During the experiments, we compared four methods: FedNCF, FedConvNCF, FRS-CE(HF) and FRS-CE(VF). The MovieLens-1M dataset was used for evaluation. The results, depicted in Fig. 5., were plotted with the number of edge nodes on the x-axis and RMSE and MAE for recommendation on the left and right y-axes, respectively. The results showed that the FedNCF had the poorest performance, followed by FedConvNCF. In contrast, our FRS-CE(HF) and FRS-CE(VF) methods consistently achieved superior recommendation effectiveness across different numbers of edge nodes. Notably, we observed that the quantity of edge nodes had no direct impact on the recommendation performance in accordance with the characteristics of FL.

These results underscore the superiority of our FRS-CE(HF) and FRS-CE(VF) approaches over traditional methods, such as FedNCF and FedConvNCF. In addition, the independence of recommendation performance from the number of edge nodes further supports the suitability of FL in addressing challenges in recommending consumer electronics products.

This study contributes to the field of consumer electronics RSs and highlights the potential of FL techniques in achieving accurate and privacy-preserving recommendations in distributed environments.

V. CONCLUSIONS AND FUTURE WORK

Especially with rising concerns about data privacy, the traditional recommendation method of using centralised data storage to provide reliable recommendations is no longer feasible in light of consumer electronics with extensive application scenarios and massive amounts of user data. In this work, we proposed a CNN-based federated RS (i.e. FRS-CE) that collects parameters from participating edges and returns aggregated results to them for knowledge sharing and training acceleration. Using distributed data across multiple edge devices, the proposed FRS-CE method enables modelling and prediction without compromising user/project privacy. We also alleviate the cold-start problem to a certain extent by designing two feature fusion techniques (i.e. HF and VF) to fuse user and item features into the user–item interaction map. Our AAM can also effectively reduce computation costs during the training process, resulting in higher model scalability. Furthermore, the effectiveness of the proposed FRS-CE and its variations against the four competing baseline methods are validated via extensive experiments performed on two real-world datasets.

In our investigation, we encountered an efficiency challenge related to the use of lightweight convolutional models in our proposed system. This limitation stems from the inherent trade-off between model complexity and computational efficiency, which affects the overall training efficiency. Thus, we will continue to improve our federated recommendation method in several areas, including exploring cold-start scenarios, handling recommendations for larger-scale consumer electronics, enhancing system stability and robustness, improving the local CNN model structure without compromising data privacy, and balancing costs, among others.

REFERENCES

[1] B.T. Betru, C.A. Onana, and B. Batchakui, "Deep learning methods on recommender system: A survey of state-of-the-art," *Int. J. Comput. Appl.*, vol. 162, no. 2, pp. 17–22, Dec. 2017.

[2] L. Wang, X. Zhang, and R. Wang R, "Diversified service recommendation with high accuracy and efficiency," *Knowl-Based Syst.*, vol. 204, pp. 106196, Sep. 2020.

[3] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, "Recommender Systems: An Introduction," New York, USA, 2010.

[4] F. Ricci, R. Lior, and S. Bracha, "Introduction to recommender systems handbook," Recommender systems handbook. Boston, MA: springer US, 2010, pp. 1–35.

[5] M. Perano, G. L. Casali, Y. Liu, and T. Abbate, "Professional reviews as service: A mix method approach to assess the value of recommender systems in the entertainment industry," *Technol. Forecast. Soc. Change*, vol. 169, pp. 120800, Aug. 2021.

[6] Z. Zhu, S. Wang, F. Wang, and Z. Tu, "Recommendation networks of homogeneous products on an E-commerce platform: Measurement and competition effects," *Expert Syst. Appl.*, vol. 201, pp. 117128, Sep. 2022.

[7] C. Wu, F. Wu, Y. Huang, and X. Xie, "Personalized News Recommendation: Methods and Challenges," *ACM Trans. Inf. Syst.*, vol. 41, no. 1, pp. 1–50, Jan. 2023.

[8] T. Liu, Q. Wu, L. Chang, and T. Gu, "A review of deep learning-based recommender system in e-learning environments," *Artif. Intell. Rev.*, vol. 55, no. 8, pp. 5953–5980, May. 2022.

[9] W. Yue, Z. Wang, J. Zhang and X. Liu, "An Overview of Recommendation Techniques and Their Applications in Healthcare," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 701–717, Apr. 2021.

[10] M. M. Afsar, T. Crump, and B. Far, "Reinforcement learning based recommender systems: A survey," *ACM Comput. Surv*., vol. 55, no. 7, pp. 1–38, Jul. 2022.

[11] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, Jun. 2015.

[12] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. Fourteenth Conf. UAI*, San Francisco, CA, USA, Jul. 1998, pp. 43–52.

[13] J. Son, and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," *Expert Syst. Appl.*, vol. 89, pp. 404–412, Dec. 2017.

[14] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[15]Y. Yin, Y. Li, H. Gao, T. Liang and Q. Pan, "FGC: GCN-Based Federated Learning Approach for Trust Industrial Service Recommendation," *IEEE Trans. Industr. Inform.*, vol. 19, no. 3, pp. 3240–3250, Mar. 2023.

[16]M. M. Afsar, T. Crump, and B. Far, "Reinforcement Learning based Recommender Systems: A Survey," *ACM Comput. Surv.*, vol. 55, no. 7, pp. 1–38, Dec. 2022.

[17]W. Zhang and J. Wang, "A Collective Bayesian Poisson Factorization Model for Cold-start Local Event Recommendation," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl.*, New York, NY, USA, Aug. 2015, pp. 1455–1464.

[18]K. Wang, Q. He, F. F. Chen, H. Jin and Y. Yang, "FedEdge: Accelerating Edge-Assisted Federated Learning," in *Proc. WWW '23*, New York, NY, USA, Apr. 2023, pp. 2895–2904.

[19]M. Ammad-Ud-Din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," 2019. *arXiv preprint arXiv:1901.09888*.

[20]Y. C. Hu, M. Patel, D. Sabella, N. Sprecher and V. Young, "Mobile edge computing—A key technology towards 5G," *ETSI white paper*, vol.11, no. 11, pp. 1–16, Sep. 2015.

[21]S. Ko, K. Lee, H. Cho, Y. Hwang, H. S. Jang, "Asynchronous federated learning with directed acyclic graph-based blockchain in edge computing: Overview, design, and challenges," *Expert Syst. Appl.*, vol. 223, pp. 119896, Aug. 2023. doi: https://doi.org/10.1016/j.eswa.2023.119896

[22]J. Yang, J. Zheng, Z. Zhang, Q. I. Chen, D. S. Wong, and Y. Li, "Security of federated learning for cloud-edge intelligence collaborative computing," *Int. J. Intell. Syst.*, vol. 37, no. 11, pp. 9290–9308, Aug. 2022.

[23]D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Commun. ACM, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[24]G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Itemto-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[25]X. Su, and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, Aug. 2009.

[26]F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," *Recommender systems handbook*, pp. 1–34, 2015.

[27]D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowl.-Based Syst.*, vol. 157, pp. 1–9, Oct. 2018.

[28]P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Eighteenth national conference on Artificial intelligence*, Menlo Park, CA, USA, 2002.

[29]S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–35, Jan. 2019.

[30]A. Singhal, P. Sinha, and R. Pant, "Use of deep learning in modern recommendation system: A summary of recent works," *Int. J. Comput. Appl.*, vol. 180, no. 7, pp. 17–22, Dec. 2017.

[31]X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *Proc. WWW'17*, Perth, Australia, Apr. 2017, pp. 173–182.

[32]H. J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *IJCAI*, Melbourne, Australia, Aug. 2017, pp. 3203–3209.

[33]Y. Pan, F. He, and H. Yu, "Learning social representations with deep autoencoder for recommender system," *WWW*, vol. 23, pp. 2259–2279, Mar. 2020.

[34]Y. Pan, F. He, and H. Yu, "A correlative denoising autoencoder to model social influence for top-N recommender system," *Front. Comput. Sci.*, vol. 14, pp. 1–13, Dec. 2019.

[35]Y. Pan, F. He, and H. Yu, "A novel enhanced collaborative autoencoder with knowledge distillation for top-N recommender systems," *Neurocomputing*, vol. 332, pp. 137–148, Mar. 2019.

[36]S. Lee, and D. Kim, "Deep learning based recommender system using cross convolutional filters," *Inf. Sci.*, vol. 592, pp. 112–122, May. 2022.

[37]X. He, X. Du, X. Wang, F. Tian, J. Tang, and T. S. Chua, "Outer product-based neural collaborative filtering," in *IJCAI-18*, Stockholm, Sweden, Jul. 2018, pp. 2227–2233.

[38]X. Xin, B. Chen, X. He, D. Wang, Y. Ding and J. Jose, "CFM: Convolutional Factorization Machines for Context-Aware Recommendation," in *IJCAI*, Macao, China, Aug. 2019, vol. 19, pp. 3926–3932.

[39]X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in *Proc. ACM SIGIR '20*, New York, NY, USA, Jul. 2020, pp. 639–648.

[40]X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural Graph Collaborative Filtering," in *Proc. ACM SIGIR'19*, New York, NY, USA, Jul. 2019, pp. 165–174.

[41]Y. Zhang, Y. Zhang, D. Yan, S. Deng, and Y. Yang, "Revisiting Graph-based Recommender Systems from the Perspective of Variational Auto-Encoder," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, pp. 1–28, Jul. 2023.

[42]B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *in Proc. AISTATS, PMLR,* Ft. Lauderdale, FL, USA, Apr. 2017, vol. 54, pp. 1273–1282.

[43]Z. Liu, L. Yang, Z. Fan, H. Peng, and P. S. Yu. 2022, "Federated Social Recommendation with Graph Neural Network," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–24, Aug. 2022.

[44]D. Chai, L. Wang, K. Chen, and Q. Yang, "Secure federated matrix factorization," *IEEE Intell. Syst.*, vol. 36, no. 5, pp. 11–20, Sept.-Oct. 2020.

[45]C. Chen, J. Zhang, A. K. H. Tung, M. Kankanhalli, and G. Chen, "Robust federated recommendation system," 2020. arXiv:2006.08259.

[46]M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *NIPS*, vol. 29, 2016.

[47]X. Xu, H. Li, Z. Li and X. Zhou, "Safe: Synergic Data Filtering for Federated Learning in Cloud-Edge Computing, " *IEEE T. IND. INFORM.*, vol. 19, no. 2, pp. 1655–1665, Feb. 2023.

[48]Y. W. Zhang, J. Pan, L. Y. Qi, Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *FGCS*, vol. 114, pp. 336–348, Jan. 2021.

[49]H. Dai, J. S. Wu, Y. Wang, C. Z. Xu, "Towards scalable and efficient Deep-RL in edge computing: A game-based partition approach," *J. PARALLEL DISTR. COM.*, vol. 168, pp. 108–119, Oct. 2022.

[50]X. Su, G. Sperlì, V. Moscato, A. Picariello, C. Esposito and C. Choi, "An Edge Intelligence Empowered Recommender System Enabling Cultural Heritage Applications," *IEEE T. IND. INFORM.*, vol. 15, no. 7, pp. 4266–4275, Jul. 2019.

[51]G Liang, C Sun, J Zhou, F Luo, J Wen and X Li, "A general matrix factorization framework for recommender systems in multi-access edge computing network," *MOBILE NETW. APPL.*, vol. 27, no. 4, pp. 1629–1641, Feb. 2022.

[52]A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, ... and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. arXiv preprint arXiv:1704.04861.

[53]D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization", 2014. arXiv preprint arXiv:1412.6980.

[54]K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1026-1034.

Jintao Wu received the Ph.D. degree in computer science and technology from Anhui University, China, in 2020. He is currently a Lecturer with the School of Software, Nanjing University of Information Science and Technology. His research interests include service computing, edge computing, and computer vision.

Jingyi Zhang is a postgraduate student in the School of Software, Nanjing University of Information Science and Technology, China. His research focuses on recommendation system and big data.

Muhammad Bilal (Senior Member, IEEE) received the Ph.D. degree in information and communication network engineering from the School of Electronics and Telecommunications Research Institute (ETRI), Korea University of Science and Technology, in 2017. From 2017 to 2018, he was with Korea University, where he was a Postdoctoral Research Fellow with the Smart Quantum Communication Center. In 2018, he joined Hankuk University of Foreign Studies, South Korea, where he was an Associate Professor with the Division of Computer and Electronic Systems Engineering. In 2023, Dr. Bilal became part of Lancaster University, where he now serves as a Senior Lecturer (Associate Professor) within the School of Computing and Communications. Throughout his career, Dr. Bilal has been actively involved in various research projects, including projects funded by the Korean Government Institute for Information & Communications Technology Promotion (IITP), the Ministry of Trade, Industry & (MOTIE) Korea, the BK21 program, National Research Foundation of Korea (NRF), and National Science Foundation China.

Dr. Bilal is a prolific author, known for his wide-ranging contributions to numerous articles published in internationally renowned, top-tier journals. His pioneering work has also led to the successful acquisition of multiple US and Korean patents. His research interests encompass Network Optimization, Cyber Security, the Internet of Things, Vehicular Networks, Information-Centric Networking, Digital Twin, Artificial Intelligence, and Cloud/Fog Computing.

Dr. Bilal is an esteemed member of the editorial boards for several prominent journals, including IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE INTERNET OF THINGS JOURNAL, IEEE Future Directions in Technology, Policy, and Ethics Newsletter, Alexandria Engineering Journal (Elsevier), and Physical Communication (Elsevier). He also serves as the Co-Editor-in-Chief of the International Journal of Smart Vehicles and Smart Transportation. In addition, he has actively contributed as a Technical Program Committee Member for leading international conferences such as the IEEE Vehicular Technology Conference (VTC), the IEEE International Conference on Communications (ICC), ACM SigCom, and the IEEE Consumer Communications & Networking Conference (CCNC).

Feng Han obtained a master's degree in computer technology from Xi'an Jiaotong University. He is currently an associate professor and master 's tutor in the School of Information Science and Technology of Shihezi University. Research interests include digital image processing and applications, and multi-sensor remote sensing applications based on UAV platforms.

Nancy Victor (Member, IEEE) received the B.Tech. degree in information technology from Kerala University, in 2010, the M. Tech. degree in computer and information technology (CIT) from Manonmaniam Sundaranar University, in 2012, and the Ph.D. degree from the Vellore Institute of Technology, Tamil Nadu, India, in 2021. She is currently an Assistant Professor (Senior) with the School of Information Technology and Engineering, Vellore Institute of Technology, where she has been a Faculty Member, since 2013. Prior to this, she has worked as an Assistant Professor at MCET, Kerala, India. She has a decade of experience in the teaching field. She has published her research works in various international conferences and journals. Her research interests include deep learning, natural language processing, data privacy, and big data. She is a member of LITD'27, Bureau of Indian Standards. She is also a member of the Indian Science Congress and a Life Member of the Indian Society for Technical Education. She has received several recognitions including the Gold Medal from the Governor Dr. K. Rosaiah, for securing university first rank in M.Tech. (CIT) degree. She has also been selected for the "Advance Training for Professionals," fully funded by the Atal Bihari Vajpayee Indian Institute of Information Technology and Management, Gwalior, to meet the immediate trained human resource requirements of IT and ITES industry.

Xiaolong Xu (Member, IEEE) received the Ph.D. degree in computer science and technology from Nanjing University, China, in 2016. He was a Research Scholar with Michigan State University, USA, from April 2017 to May 2018. He is currently a Full Professor with the School of Software, Nanjing University of Information Science and Technology. He has published more than 100 peer-review articles in international journals and conferences, including the IEEE TSC, IEEE TFS, IEEE TKDE, IEEE TPDS, IEEE T-ITS, IEEE TII, ACM TOSN, ACM TOMM, ACM TIST, ACM SIGIR, IEEE ICWS, ICSOC, etc. He was selected as the Highly Cited Researcher of Clarivate 2021. He received the best paper awards from IEEE CBD 2016, IEEE CPSCom 2020, SPDE 2020, IEEE CyberSciTech2021, CENET 2021, and EAI Cloudcomp 2021, and the distinguish paper award from IEEE Smart City 2021. His research interests include edge computing, the Internet of Things (IoT), cloud computing, and big data.