Cluster Computing (2022) 25:4075–4096
https://doi.org/10.1007/s10586-022-03644-w

Federated recommenders: methods, challenges and future

Zareen Alamgir[1] · Farwa K. Khan[1] · Saira Karim[1]

Received: 22 August 2021 / Revised: 28 April 2022 / Accepted: 29 May 2022 / Published online: 25 June 2022

Abstract
Abstract Web users are flooded with information on the internet, and they feel overwhelmed by the different choices they have to make online daily. Recommender systems come to their rescue by suggesting products best aligned with their interests. To achieve this, traditional recommenders transfer users' personal data from the client to the server and dig for information about the user's interests and tastes. Moving data to the cloud violates the user confidentiality requirement and poses severe threats to user privacy and security. Moreover, with the tremendous increase in data size, it is no longer possible to collect and process massive data in the cloud. With the emergence of federated learning, numerous innovative recommender models are devised to solve these issues. In these models, the user data never leaves the client-side, and only the inferred results are sent back to the server for aggregating and updating the master model. Hence, the federated recommenders preserve user privacy and save the hassle of transferring enormous data to the cloud. This paper meticulously studies the recently proposed federated recommenders and classifies them based on the enhancements introduced in the prediction model, security scheme, or optimization technique. We identify the challenges faced by current federated recommenders and observe that most issues are inherently due to various aspects of federated learning, such as heterogeneous and non-IID data, malicious users, distributed framework, and non-reliable edge devices. While some emerge due to the coupling of the recommendation process in the federated paradigm. This research summarizes the current limitations, highlights the areas that need improvements, and presents future paths. In short, it paves the way for the development of robust federated recommenders that can handle the challenges of federated learning and, at the same time, generate high-quality recommendations.

Keywords Federated learning · Recommendation systems · Federated recommenders · Matrix factorization · Meta learning

1 Introduction

Over the last few decades, recommender systems have gained significant importance and completely revolutionized the online world. Companies continuously strive to enhance their recommender systems to thrive and cope with the dynamics of today's fast-changing world. The tech giants like Amazon, Netflix, and Youtube have used the recommenders to their advantage and captured a vast market share by providing personalized recommendations to their users.

The designs and techniques used for recommendations have evolved over the years and continue to do so. Numerous innovative algorithms and ideas are proposed to handle challenges faced by recommenders in different domains [1, 2]. Traditional recommendation algorithms are sequential and work on small datasets. With the massive increase in the size of data, there came an age of distributed and cloud-based recommendation algorithms. However, now even these systems are unable to handle the enormous distributed data and its challenges. Meanwhile, the data continue to grow at a tremendous rate due to a huge upsurge in the number of online users and mobile data traffic [3]. Cloud-based recommenders face various

✉ Zareen Alamgir
zareen.alamgir@nu.edu.pk

Farwa K. Khan
l191881@lhr.nu.edu.pk

Saira Karim
saira.karim@nu.edu.pk

[1] Computer Science Department, National University of Computer and Emerging Sciences, Lahore, Pakistan

problems in dealing with massive data scattered around the world. In addition to this, they have many data privacy and security loopholes. To address these issues, Google introduced the concept of federated learning [4]. Federated learning (FL) is a new emerging computing paradigm that aims to train machine learning models on data distributed across multiple devices while preserving data privacy and security [5]. It does not require data transfer from the user's device to a central storage(server); instead, it focuses on updating the global model at the server by aggregating updates from local models trained by clients. In general, FL allows multiple organizations to collaborate and develop ML models without exchanging data [6].

FL architectures are now gaining momentum motivated by their abilities to provide collaborative data analytics while maintaining privacy [7]. Recommendation systems were deployed over centralized data repositories for a long time, yet modern requirements demand federated recommender methods. Recently researchers have started working in this direction, and many innovative federated recommenders(FR) have been proposed [8, 9]. Current FRs have to deal with the challenges of the federated paradigm along with the recommendation issues. A reasonable amount of work has been done to improve security in FR [10, 11], and make it more personalized [12, 13].

Federated recommenders have gained significant importance in the current era of data science and edge computing [14]. The existing FRs are still in their early stages. There is a dire need to thoroughly analyze the existing techniques, identify the shortcomings and point out the future paths leading to robust and scalable models. Currently, there is no survey or study on this topic. To the best of our knowledge this is the first survey on the emerging paradigm of FRs. The federated recommendation is challenging and significantly different from the traditional one due to its distributed nature, data transfer limitations, and privacy concerns. Numerous surveys have been conducted on FL and recommendation systems, but none touched the area of FRs [6, 5]. To fill the gap, this survey dives deep into the current literature of federated recommenders and provides a baseline for researchers.

Contributions of this survey The contributions of this survey are multi-fold: (1) It provides a brief overview of traditional recommenders and FL. (2) It presents comprehensive details on state-of-the-art federated recommenders. (3) It discusses significant challenges faced by federated recommenders and also highlights the open issues and future directions. (4) It presents the applications that can benefit from federated recommenders. (5) This survey also summarizes datasets, platforms, and frameworks used for federated recommenders.

The list of abbreviations used in the paper is given in Table 1. The rest of the paper is organized as follows. Section 2 briefly describes the traditional recommender systems and FL models. Section 3 presents the detailed literature review on federated recommenders. In sect. 4, we described major applications, datasets, tools, and platforms for FL. Section 5 highlights the challenges and limitations in this area. Section 6 summarizes the future work , and finally, Sect. 7 concludes the paper.

2 Background

This section briefly overviews and presents the current state of the art on the traditional recommendation systems and FL to equip the readers with the essential techniques and background.

2.1 Recommender systems

Recommenders have gained immense importance in the past few decades. New innovative algorithms are being developed by multidisciplinary efforts using techniques from Deep Learning, Big data, Machine Learning, and Data mining to improve recommendation accuracy. Figure 1 shows the major recommendation algorithms currently in use. Almost all recommender systems attempt to identify links between users and items based on the previous history and predict future connections. They rely on explicit or implicit feedback to understand user traits and interests. Explicit feedback mainly refers to the rating given by a user to an item, whereas implicit feedback does not involve any direct signal given by users. Instead, it uses a view-history of a user to understand what kind of items he may like. In most cases, implicit feedback is considered as valuable as explicit feedback.

Content based recommenders suggest items to a user similar to the ones adored by him in the past [15]. They generate user profiles based on the user's data and require additional information (features) about items to make accurate future predictions, as shown in Fig. 2. Interestingly, content-based models can recommend new items with given features and no previous ratings. However, a problem occurs when a new item arrives with entirely different or unseen features [16]. Besides this, it is challenging to generate appropriate and reliable features for items. Another drawback of content-based filtering is that it recommends similar types of items and does not give creative out-of-the-box recommendations [2].

Collaborative filtering based recommenders are the most popular ones, and they rely only on user-item interactions and do not need any additional information. Collaborative filtering(CF) consists of three main steps: first pair-wise similarities between users are computed, next missing ratings of items for each user are predicted, and

Table 1 List of abbreviations

Acronym	Explanation	Acronym	Explanation
ALS	Alternating Least Square	HR	Hybrid based Recommenders
CF	Collaborative Filtering	MF	Matrix Factorization
CNN	Convolution Neural Networks	ML	Machine Learning
DL	Deep Learning	MLP	Multi-layer Perceptron
DP	Differential Privacy	PCA	Principle Component Analysis
FedAvg	Federated Average	RNN	Recurrent Neural Networks
FL	Federated Learning	SGD	Stochastic Gradient Descent
FR	Federated Recommenders	TL	Transfer Learning
HE	Homomorphic Encryption	VFL	Vertical Federated Fearning
HFL	Horizontal Federated Learning		

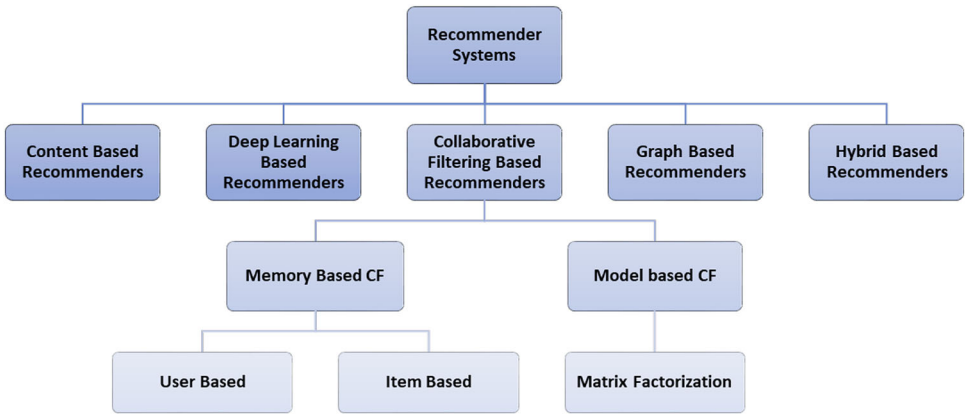


Fig. 1 Recommender systems

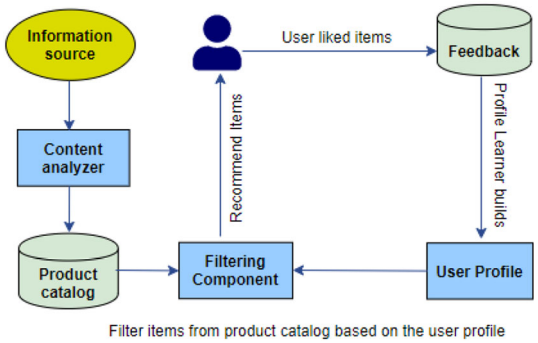


Fig. 2 Content based recommender

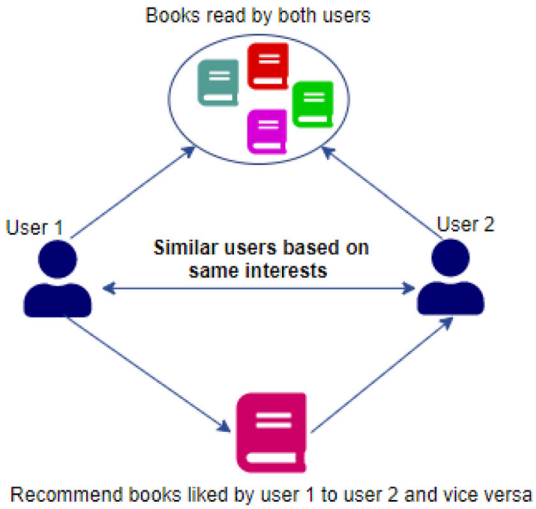


Fig. 3 Collaborative based filtering

finally, the top K most suitable items are selected for recommendation. The above approach is categorized as memory-based CF as it works by extracting essential information from a user-item rating matrix, the utility matrix. Figure 3 shows the details of the CF technique.

Matrix factorization is a state-of-the-art model-based CF approach that learns user-item interactions and generates new recommendations. It decomposes a large sparse user-item matrix into a low-dimensional dense matrix that captures the essence of the data and gives precise representation. The users and items are represented as low dimensional latent vectors, and interaction between an item and a user is obtained by calculating the dot product of their relevant vectors, as shown in Fig. 4.

Let interaction matrix M of size $n * m$ contains ratings. Matrix factorization goal is to factorize M such that:

$$M = X \cdot Y^T$$

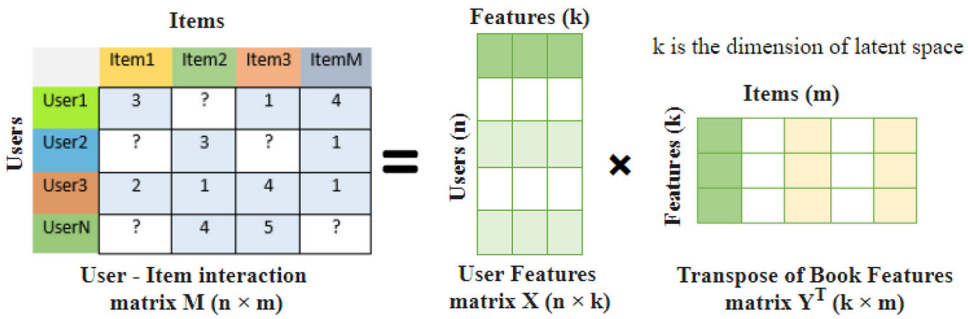


Fig. 4 Matrix factorization

here X represents the user matrix $(n * k)$, Y represents the item matrix $(m * k)$, and k is the dimension of latent space where user-item interactions are modeled as dot product $m_{u_i} = x_u \cdot y_i^T$, which captures the interaction between user u and item i. To learn latent vectors (x_u and y_i) regularized squared error on set of known ratings should be minimized as in [17]. Commonly used algorithms for matrix factorization are Alternating Least Squares, Singular value decomposition, Principle component analysis, and Stochastic gradient descent [17]. Among these, ALS and SGD are the most popular and widely used for recommendation. Matrix factorization models differ mainly in terms of the constraints enforced on user and item vectors or by the nature of the objective function. In [18], a detailed survey of recommender systems using matrix factorization is presented.

Hybrid recommenders combine different recommendation algorithms with various techniques from data mining and machine learning to overcome the issues faced by traditional ones. They are designed to increase accuracy and handle problems like cold start, data sparsity, scalability, diversity, and user privacy. The most popular and simple hybrid recommenders blend content-based filtering with CF algorithms [19, 20]. These recommenders usually attain high accuracy and low recommendation error. Some researchers have mixed user-user CF with item-item CF to improve the overall performance and accuracy. Now, researchers are developing complex hybrid systems consisting of various components that use data mining techniques to extract useful information from the data[21]. Complex HR mix collaborative filtering with clustering [22], sequential pattern analysis [23], and demographic filtering [24] to improve the accuracy and reduce the error. Cano et al. conducted a comprehensive survey of the hybrid recommenders proposed in the last decade [25].

Deep learning based recommenders DL is a sub-field of ML that relishes extreme popularity as it can learn multiple levels of data representation. Researchers are incorporating DL models in recommenders to enhance recommendation quality. A recent survey [26] shows that different DL architectures like Multilayer perceptron(MLP), Auto-encoder, Convolution neural network (CNN), Attentional models, Recurrent neural network(RNN) are used in recommenders. MLP is the most widely used DL model in recommenders [27]. In [28], authors proposed a news recommender system based on RNN to learn user representations. For Google play app recommendations, Cheng et al. [29] deployed a generalized linear model and a feed-forward neural network to improve generalization and memorization. For rating prediction and a better learning framework, a model based on a combination of MLP and RNN is proposed in [30]. CNN is also explored for the recommendation potential [31].

Graph-based recommenders construct a graph to capture user-item interactions, identify connections, and make valuable predictions. In [32], a graph CNN based recommender is proposed for specially designed user-item CF. In [33], authors developed a graph-based collaborative ranking framework called GRank to capture user preferences from implicit feedbacks using the Tripartite Preference Graph (TPG). In [34], a feature selection technique is proposed that can increase the performance of graph-based recommenders and tackle baseline problems in CF. Graph matching is a key enabling technique in recommenders; mining and matching relationships from interaction contexts in a social networks is very useful [35]. Knowledge-based graph recommenders [36] are also quite popular and they serve as side information in recommender systems.

The traditional recommender suffers from several unresolved issues and problems that have severe effects on the performance. Researchers are trying to come up with new innovative ideas and hybrid systems to resolve these issues. The most crucial challenge is to protect user privacy effectively. Currently, the user's data is shifted to the cloud for model training and generating recommendations, and this leads to various data security and privacy problems. Furthermore, it is not easy to collect data in one place as data is scattered around the world in different organizations and countries. At the same time, organizations and users are concerned about their data privacy and security.

2.2 Federated learning (FL)

FL is a new paradigm developed by Google to train the machine learning models in a distributed environment without compromising data security and privacy. It employs edge devices to learn a shared global model without moving the user data to a central server. Each edge device shares summarized model updates with the server, and the server combines these updates to improve the global model. FL offers multiple benefits in addition to preserving data privacy. It employs advance optimization techniques [37, 38] to build intelligent models that reduce data traffic, latency, and power consumption.

FL can be categorized based on infrastructure, data distribution, and privacy mechanisms. This section briefly explores the different characterization of FL and the motivation behind it. So the readers have an idea about the primary grounds available for building secure and robust FRs.

2.2.1 FL infrastructure

FL can work in a centralized as well as decentralized manner.

Centralized federated learning employs a hub and a spoke topology, where the hub is the central server and spokes represent clients. The central server is responsible for carrying out the model training process and resource provisioning [39, 40]. It gathers model updates from the clients participating during the learning process and builds a global model. The server is a single point of failure and may become a bottleneck when there are many clients. The simple centralized FL model, shown in Fig. 5, works as follows: the server selects clients based on some eligibility criteria for the training process and sends current model updates and the training program to the selected clients. Next, each client updates the model using local data and sends the locally updated model to the server. Then the server updates the master model based on the results received from clients in the current round.

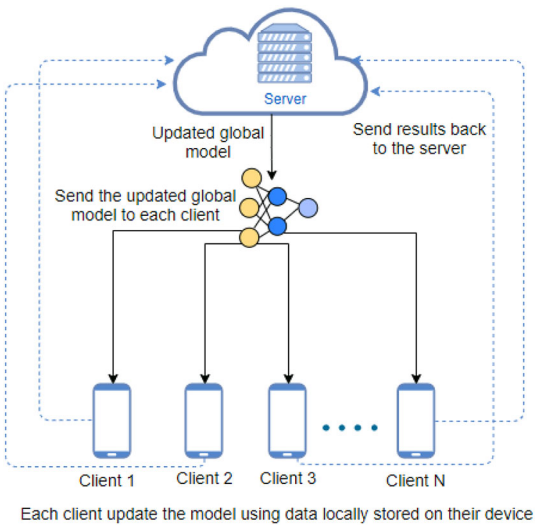


Fig. 5 Centralized FL

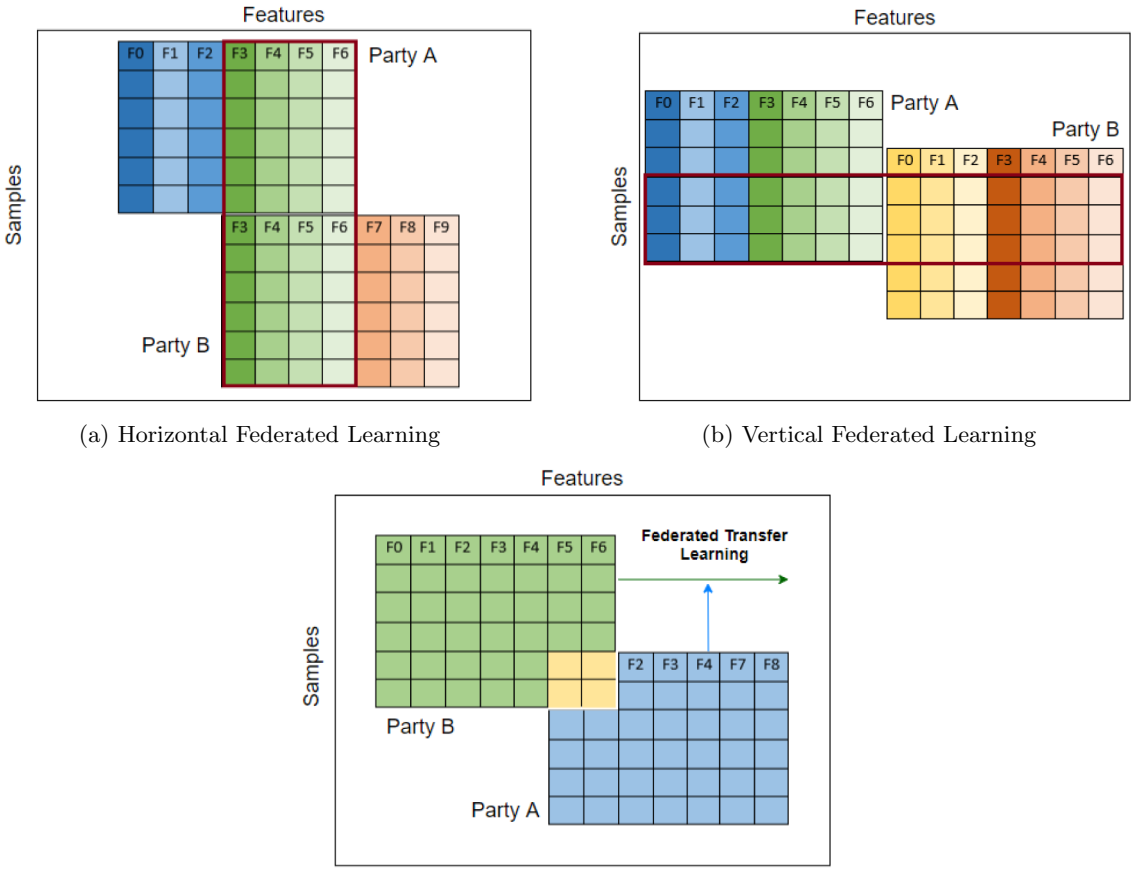
Decentralized FL uses a peer-to-peer communication mechanism, where clients coordinate to build a robust global model instead of communicating with the central server for model updates. A central authority is only responsible for establishing the learning process. Usually, a sparse graph network is used as the communication topology for clients to send/receive messages from their peers.

2.2.2 FL data distribution

FL is divided into three different categories based on the distribution characteristics of the data.

Horizontal FL, also known as sample-based FL, is employed when different training datasets have the same set of features but different sample space, as shown in Fig. 6a. For example, two bank branches located in different regions have distinct users but similar features. HFL can work with a client-server architecture based on centralized FL and also with a peer-to-peer architecture based on decentralized FL. In a client-server architecture, clients jointly train the model with the server's help, as shown in Fig. 7. The users compute training gradients of items and send the masked results to the server. The server performs secure aggregation of the item gradients and sends the updated model to each client. The users locally train user profiles. These steps repeat until convergence. In HFL, a server is assumed to be honest but curious, and participants are considered honest. This indicates that only a server can compromise data security and privacy. Peer-to-peer HFL does not involve a central server and all participants act as trainers or workers. They jointly train the model without server and exchange model weights with each other using a secure channel.

Vertical FL, also known as feature-based FL, is used when different training datasets have the same training samples but different feature space, as shown in Fig. 6b. For example, different organizations in the same region most likely have the same set of users but different features. VFL can extract valuable insight from such datasets. It builds a model by aggregating different features from distinct datasets and computes the training loss while preserving privacy. VFL is a business-to-business paradigm where most participants are organizations, and they jointly improve business performance. In VFL security, participants are honest but curious. For secure computation between participants, an independent semi-honest third



(a) Horizontal Federated Learning

(b) Vertical Federated Learning

(c) Transfer Federated Learning

Fig. 6 Federated learning categories

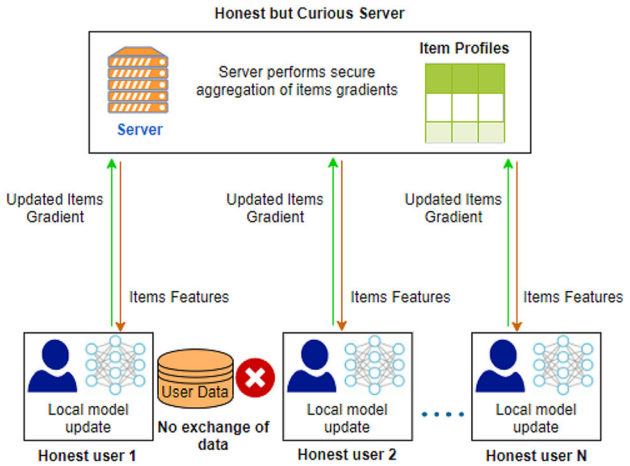


Fig. 7 Horizontal federated learning client-server architecture

party helps the participants train the ML model. The third-party does not collude with both parties and only collects intermediate results to compute the gradient and share the relevant model parameters with each participant.

Federated transfer learning, a combination of FL and transfer learning, is useful when datasets differ in both sample and feature space, as shown in Fig. 6c. For example, consider two different organizations (a bank in the United States and an e-commerce company in China). They both have distinct features and a small intersection of users due to geographical restrictions. HFL requires the same feature space and VFL requires the same sample space, but this is not the case in most scenarios. FTL addresses data-related issues and builds an accurate ML model when data have small overlapping features and samples.

2.2.3 FL privacy mechanisms

Different privacy techniques are employed in FL for data protection and security. *Differential privacy* [41] is a common privacy technique used to add noise to the input, intermediate results, or output data so that reverse-engineering the user's data become hard. It adds randomness to the user collected data to preserve privacy, but this compromises accuracy. *Secure multiparty computation* [42] preserve privacy without compromising accuracy. In MPC, multiple parties jointly compute common functions without showing their private data to each other. It is considered

secure if parties only learn results rather than other information. Some studies have proposed a hybrid approach to combine differential privacy and secure MPC to provide privacy and accuracy [43]. Another approach *Homomorphic encryption* [44] protects user privacy by exchanging parameters after the encryption mechanism. It allows calculations on encrypted data and provides the same results as if a computation is performed on decrypted data. Unlike differential privacy, it doesn't transmit the data and the model.

3 Federated recommenders state-of-the-art

Federated recommender systems have gained a lot of attention in recent times due to their potential in preserving user privacy and computational power. This section presents the latest algorithms and frameworks proposed in the domain of FRs. The researchers have devised innovative FR models that exploit different techniques to handle the various challenges of the federated paradigm and recommendation. All proposed FR models work on one or more of the following areas: prediction model design efficacy, model security, and model/resource optimization. We have characterized the current FR systems from an algorithmic perspective based on the area they primarily work on; the characterization is presented in Table 2.

Most of the research conducted on FR is mainly based on collaborative filtering and matrix factorization. Some researchers have also explored the realm of meta-learning, deep-learning, and reinforcement learning for constructing robust recommenders. Collaborative filtering is a simple and effective technique that generates valuable recommendations, and hence, it is one of the first that is tailored and adopted in FRs. The majority of the federated collaborating filtering models are based on matrix factorization. Figure 8 shows the basic architecture of federated matrix factorization that adopts horizontal FL and uses encryption

Table 2 Characterization of federated recommenders

Category	Technique
Prediction model	Matrix Factorization [8, 9, 45–48]
	Meta learning [12, 49–51]
	Reinforcement learning [52, 53]
Model security	Encryption [54, 47, 55, 56]
	Differential privacy [57–59]
	Selective data sharing [45, 60, 46]
Model optimization	Early convergence [48]
	Resource optimization [50, 61]

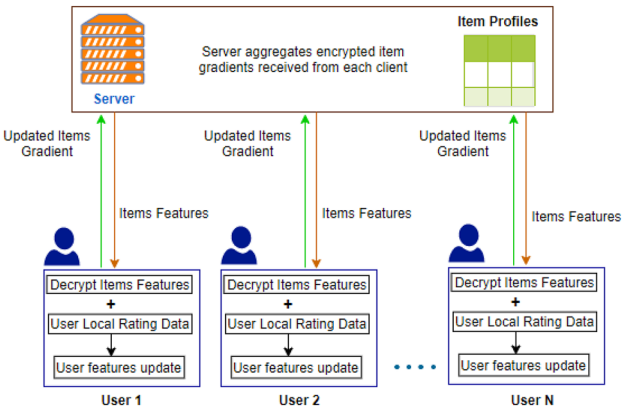


Fig. 8 Federated matrix factorization

to ensure user privacy. The federated recommender employs the concept of FL to train a shared predictive model while preserving user privacy, and unlike traditional recommenders, it does not move the user's private data to a central server.

Ammad-ud-din et al. [8] proposed the earliest implementation of a federated CF recommender system that gives personalized recommendations based on users' implicit feedback. The system uses the underlying idea of ALS algorithm to learn user and items vectors by alternating between updates to user and item vectors. However, unlike ALS, Federate CF employs a stochastic gradient technique to learn item vectors at each client and aggregates the gradients to update the master model. This federated CF model performs three steps: (1) Update all item vectors on the server and share them with other clients. (2) Update user vectors locally on the client using the local data and item vector transmitted by the server. (3) Update the gradients of an item on each client and sent it back to the server; the server aggregates these gradients to update the item. Empirical results showed that the performance and recommendation accuracy of the federated CF system is comparable to the standard CF technique. Flanagan et al. [9] presented a federated multi-view matrix factorization (MVMF) model for personalized recommendations. It is the first federated model designed to handle heterogeneous data from multiple sources and integrates various data views to capture insights. For example, a multi-view movie recommender can incorporate user personal information and movie-related details with the historical user-movie watch data to produce accurate recommendations. MVMF is an extension of standard CF and applies SGD to learn user vectors, item vectors, and user-item interactions. Latter, uses joint factorization to capture the relationships between these features. It outperforms FCF and gives a significant improvement in the performance of highly sparse matrices. One of the limitations of this approach is a higher payload than simple FCF because of the side

information included in a model. Computation time increased from 24 to 52%. The model shows promising results for cold start problems, especially for new users, but predictions need further improvements for new items. Table 3 provides a brief overview of emerging federated recommendation models.

The privacy and security of user data are of utmost concern in the recommender system. In FL, the clients share the model updates with the server. These updates may contain sufficient information to reveal model features and client data, possibly leaking some user information. To overcome this potential hazard, various techniques are proposed by researchers [54, 62–64]. Chai et al. [54] proposed a secure framework (FedMF) to handle user privacy issues in FRs. FedMF, a federated matrix factorization system, deploys horizontal FL client-server architecture and uses additive homomorphic encryption for protecting user's privacy. FedMF employs SGD for matrix factorization. Authors showed that gradient information sent to the server could harm the user's privacy, so they used Paillier encryption [65] for the prototype of FedMF, implemented with two different settings (FullText and PartText). In FullText implementation, a user sends gradients of all items, while in PartText, users only send gradients of items they have rated. Hence, PartText is computationally efficient, but it exposes the items. Yongjie et al. [62] also propose a privacy-preserving recommender based on federated MF that takes care of the model-privacy, existence-privacy, and value-privacy. The authors formulate a two-stage randomized response method to balance privacy and computing cost.

Another recommender, FedeRank [45] handles privacy concerns by giving control to users to decide the amount of sensitive data that can be shared. It is one of the initial attempts to develop a federated pair-wise recommender system. The central server contains the latent representation of items, and each client builds a local training set for the pair-wise relationship among the user and a pair of items. In each round, the server distributes the global model to the set of selected clients. Each client then selects some samples from its local dataset and updates its local model. The client is given a choice to send a fraction of its positive item interaction updates to the central server, whereas the negative feedback, which is assumed to be insensitive data, is always sent back to the server. In another work, Qin et al. [60] also used the fact that almost every user has some sensitive data and some public data to develop a novel privacy-preserving federated

Table 3 Summary of federated recommenders (prediction model)

Approach	Dataset	Discussion
Federated CF using matrix factorization [8]	MovieLens, in-house data	Pioneer method and produces results comparable to standard CF but susceptible to model attacks and cold-start problems
Federated multi-view matrix factorization using side information [9]	MovieLens, bookcrossing	Outperforms Federated CF and handles cold-start problem but at the cost of higher network communication and payload
FedeRank, a pair-wise FR with user-controlled feedback [45]	Amazon music, Library thing, MovieLens	Allows a user to decide how to share private sensitive preferences. Achieves higher accuracy than baseline centralized methods and federated CF
Federated pair-wise learning using MF and Bayesian Personalized Ranking, [46]	Foursquare	Allow user to control the amount of data sharing with server to overcome vulnerability, performance is comparable with baseline methods, but cold-start problem exists
Momentum-based FR using factored item similarity model [47]	Last.fm, MovieLens, Citeulike	Use gradients (instead of model parameters) to detect Byzantine clients and avoid the model poisoning attack. Can be extended to handle targeted poisoning attacks
Federated meta-learning (FedMeta) shares parameterized algorithm (meta-learner) [12]	FEMNIST, Shakespeare, Sent140,	Handles the statistical and systematic challenges in FL. Converges fast, has low communication cost and high accuracy than FedAvg but requires large local training data
Meta-learning and Neural network [49]	MovieLens	Improved personalized recommendations but higher error rate as compared to centralized algorithm. Does not ensure model security and user privacy
Fed4Rec - FL with model agnostic meta-learning [51]	Globo	Online Page recommendation with joint-learning from public and private users data. Achieve higher accuracy than baseline methods
Distributed asynchronous deep reinforcement learning framework [52]	Outbrain	Combines ideas from advantage actor-critic model (A3C) and FL. Enhances user privacy and performs comparably with baseline methods
Contextual recommender for big data [53]	YFCC100M	Preserve privacy using differential privacy, handles scarcity, no performance loss but slow convergence rate

recommender. The server is given access only to users' public data and the item catalog. Anelli et al. [46] proposed federated pair-wise learning that uses matrix factorization and Bayesian Personalized Ranking. The proposed method allows users to control the amount of sensitive data shared with the server to overcome vulnerability.

Gao et al. [66] demonstrated the potential privacy threats in all three types of federated recommender systems that include horizontal, vertical, and transfer federated matrix factorization. Whereas Chen et al. [47] proposed a robust federated recommendation system that can detect byzantine clients. For real-time personalizations and data sparsity, they used factored item similarity model [67] and momentum-based Adam optimizer [68]. Adam optimizer has fast convergence properties and by using model parameters in Adam, byzantine clients can easily attack. So, authors [47] utilized the gradients of the model rather than model parameters for detecting byzantine clients. Their proposed strategy can also be adapted to other optimizers: AdaGrad, SGD with momentum, and RMSProp. A comprehensive summary of important research articles that focused on model security and data privacy is given in Table 4.

The federated recommenders suffer from a colossal communication cost and generate a lot of network traffic. This is due to the fact that the accuracy of standard FRs converges after processing and aggregating many rounds of clients' local updates. Muhammad et al. [48] devised a system, FedFast, that focuses on enhancing the convergence speed of the model to achieve better accuracy. They extended the federated averaging algorithm (FedAvg) [69] in the context of the neural recommendation model (General Matrix Factorization, GMF). FedAvg is a synchronous algorithm that allows clients to perform updates on local models multiple times before averaging the resultant models. It helps in reducing the number of communication rounds. Figure 9 shows the basic architecture of FedFast. The two main contributions of FedFast includes: a better selection of clients and intelligent aggregation of local training models for faster convergence. Unlike traditional FL systems that select clients at random, FedFast clusters clients with similar characteristics into groups. Users in a group benefit from training experience from their peers, thus accelerating the learning process. A comparison of FedFast with FedAvg and GMF in terms of communication speed and quality of recommendation shows that FedFast outperforms FedAvg in most cases and GMF in some. FedFast converges more quickly than FedAvg without compromising recommendation accuracy. There are a few limitations in FedFast. It is susceptible to cold start problems and needs to be retrained through transfer learning to support new users and items. Furthermore, it is vulnerable to model inversion attacks during training. This vulnerability can be avoided using privacy techniques. Table 5 summarizes the recent ideas on resource efficient federated recommender systems.

Ribero et al. [57] proposed a federated recommender that significantly reduces communication costs while preserving user confidentiality. They introduced differential privacy to FRs; this is not an easy task as the basic idea and mechanism of both techniques contradict one another. The federated system is iterative in nature, where the global model is trained iteratively by incorporating the clients' local updates. On the other hand, differential privacy does not work well with the iterative models as it inserts noise in the data to ensure privacy. Adding noise in each iteration is not feasible; in this case, the amount of noise will explode. Ribero et al. built differentially private prototypes via matrix factorization to learn the global model without transmitting users' data, statistics, or preferences to the central server. The proposed model eliminates iteration and requires only two global steps. Thus, it effectively

Table 4 Summary of federated recommenders (model security)

Approach	Dataset	Discussion
Matrix factorization and homomorphic encryption [54]	MovieLens, Filmtrust	Resilient to model inversion attacks but not scalable and has high computation cost
Cross-domain POI, MLP, homomorphic encryption [56]	Foursquare, MovieLens	Privacy preserving and handles data sparsity and cold-start problem but high algorithm complexity
Matrix factorization and differential private prototypes [57]	Synthetic, eICU, MovieLens	Privacy preserving, low communication cost but assumes that individuals are grouped into entities
FedRec—News Recommender based on Deep Learning and differential privacy [58]	News	Privacy preserving and handles the cold-start problem. Low performance than Federated CF but comparable to SOTA news RS
Blockchain CF and differential privacy [59]	News	Ensure privacy and reduces computing and bandwidth requirement. Improves news push accuracy while recommendation accuracy is low

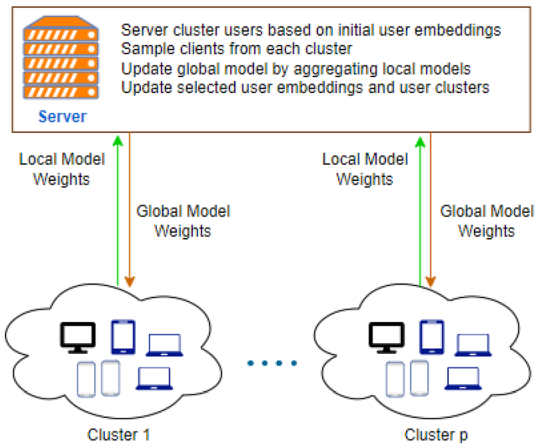


Fig. 9 Key components of FedFast architecture

incorporated differential privacy in federated recommender systems. This approach leads to a significant reduction in communication costs and privacy risks.

Few researchers introduced the idea of meta-learning and deep learning to FR. Chen et al. [12] developed a Federated meta-learning framework (FedMeta) in the context of a robust, content-based recommendation model. Figure 10 shows the basic architecture of FedMeta. It is the first method that combines the best of two promising domains; meta-learning and FL. FedMeta extends meta-learning [70] in a federated setting to handle statistical and systematic challenges. Statistical challenges include non-IID and heterogeneous data. Many devices and their constraints in terms of storage, computation, and communication capacities pose systematic challenges. FedMeta shares a parameterized algorithm instead of a global model. Each client receives parameters from the server, trains the model using the locally stored training dataset, computes the loss function on the test dataset, and sends it to the server. The server updates the parameters with the help of loss functions collected from clients. This work implements a model agnostic meta-learning algorithm (MAML), first-order MAML (FOMAML), meta-SGD to handle non-convex problems. The authors mold the recommendation problem into a classification problem to use FedMeta for recommendations.

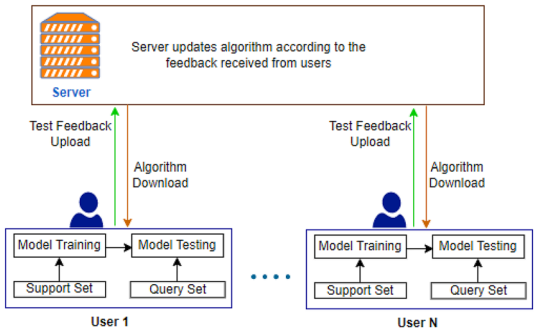


Fig. 10 Federated meta learning framework

Jalalirad and Scavuzzo [49] incorporated the concepts of REPTILE meta-learning algorithm in FRs [71]. However, unlike REPTILE meta-learning, edge devices are not required to communicate with each other. Each device trains model in parallel and communicate with the server independently. The algorithm runs for multiple cycles, and each cycle consists of two phases: a global training and a local training phase. In global training, each client trains the model locally for a few steps and sends its parameter vectors to the server. The server aggregates the parameter vectors and transmits the resultant vector to the clients. Then clients update their model with the received parameter vector, train it for a few steps, and send updated vectors to the server. The global phase runs for a predefined number of rounds, then local training begins. In local training, the clients train the model locally and do not interact with the server. This phase facilitates tweaking the model according to its local data at each client. In experiments, the model used was a three-layer neural network consisting of two fully connected layers and one embedding layer. The results reveal that a NN with the learned item and user embeddings can give a robust FR.

In another research, Lin et al. [50] introduced meta matrix factorization (MetaMF) for rating predictions.

Table 5 Summary of federated recommenders (model optimization)

Approach	Dataset	Discussion
Matrix factorization using Deep learning [48]	MovieLens, TripAdvisor, Yelp	Fast convergence, low communication cost, high accuracy but susceptible to the cold-start problem and model attacks
Meta-matrix factorization and Multi-layer perceptron [50]	Douban, Hetrec-ML, Movielens1M, Ciao	High performance on rating prediction model, handles systematic constraints on RAM, storage and computation. Susceptible to cold-start user problem, model inversion attacks and requires huge personalized data
Collaborative filtering and Multi arm bandit, RL [61]	MovieLens, Last-FM, MIND	Reduced 90% of model payload, lower communication cost but theoretical convergence not guaranteed

Figure 11 shows the basic architecture of MetaMF. It is specifically designed to work in a mobile environment with limited resources and constraints (RAM, storage, communication bandwidth). MetaMF consists of three modules: collaborative memory module, meta recommender module deployed on the server, and private prediction module installed on the user devices. The collaborative module is responsible for learning the user vector. The meta recommender component creates private item embeddings for each user using a rise-dimensional generation strategy to tackle the issue of high-dimension user embeddings. MetaMF builds a private rating prediction model using a multi-layer perceptron. The prediction module then estimates ratings using item embedding and rating prediction models received from previous modules. This approach does not address the cold-start user problem and privacy issues.

Significant efforts have been made to apply reinforcement learning techniques and methodologies to improve federated recommendation models. Shi et al. [52] proposed a federated recommender that uses reinforcement learning and the asynchronous advantage actor-critic (A3C) model. The goal is to learn a deep neural network that adapts using the input from the local models of client devices. The local model uses private information of the client, items interactions, and any available contextual information to train the model locally using A3C. The updated global model is sent back to the client asynchronously. In Federated CF [8, 9] and similar recommender systems, the model payload increases linearly with the increase in the number of items. Farwa et al. [61] proposed the first method to tackle the item-dependent payload optimization challenge for FRs using reinforcement learning. Multi-armed bandit solution, a classical approach to reinforcement learning, is used to intelligently select a part of the global model to transmit to all participating users in each FL iteration. The proposed method achieves up to 90% reduction in model payload with only 4–8% loss in recommendation performance on highly sparse datasets.

In another approach, Zhou et al. [53] used reinforcement learning to devise an online federated contextual distributed recommender system (DRS) that effectively handles big data without posing any threat to user privacy. The system is based on a contextual multi-armed bandit (CMAB) [72] and uses Trusted Third Party (TTP) to prevent vulnerable behavior in DRS. The framework consists of agents, items, users, and TTP. An agent (individual/company) maintains an enormous repository of items. When a user arrives, it recommends items to a user based on context. An item-tree cluster is created to handle massive data efficiently. The proposed DRS is adaptive in nature and claims to have low power consumption and latency.

Recently, federated recommenders have been introduced to various real-world applications and fields like media, social networks, transportation, and others to improve the quality of recommendations. Tao Qi et al. [58] developed a news recommendation system (FedRec) using the click and skip behavior of users on the news website. The model is composed of two sub-models: news and user model. The news model has different layers to perform tasks like word embeddings, learning word representations using a convolution neural network (CNN). The user model learns users embeddings by combining the long-term and short-term interests. FedRec model is privacy-preserving and applies local differential privacy. It embeds Laplacian noise to data for safeguarding user privacy and handles recommendations challenges like cold-user and cold-item. Yichuan et al. [59] also developed a privacy-preserving news recommendation system. The proposed system uses blockchains with FL and protects data privacy by adding Laplacian noise to the training model through differential privacy. In another work, Zhao et al. [51] presented a page recommendation framework Fed4Rec that provides

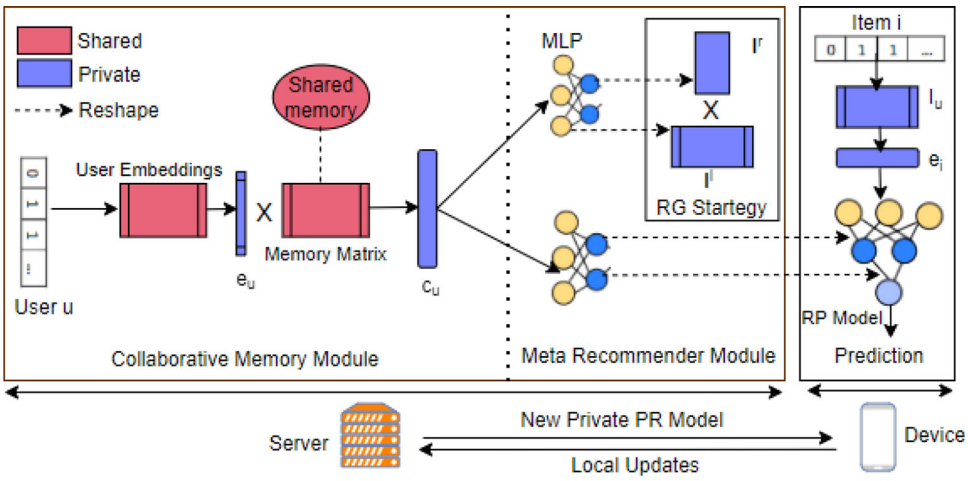


Fig. 11 Meta matrix factorization

recommendations to both the public and private users using FL and model agnostic meta-learning. Fed4Rec comprises two layers: one is RNN, and the second is the attention layer. In each round, the central server trains this model using the sequence of page visits of the public users available at the server and shares the global model with selected clients. Each client only updates the attention layer of the model using its private data and sends it back to the server.

Jayant et al. [73] introduced FRs to driver assistance system. Based on historical driving data, they devised a mechanism to analyze the driver's stress level, driving behavior, and fuel efficiency based on historical driving data to recommend a suitable driver for the requested trip. To predict stress level, Long Short-Term Memory for fully CNN is used, and to identify driver behavior Hemming Markov Model (HMM) model is adopted. The relationship between stress and behavior is established using correlation analysis. Authors in [55] adopted vertical federated factorization and homomorphic encryption for secure electric vehicle charging point recommendation.

Federated recommendation is a promising and progressive research area that can have many practical applications. Many innovative and creative ideas are being introduced to the realm of FRs. Recently, Li-e Wang et al. [56] proposed a novel federated cross-domain POI (Point-of-Interest) recommendation framework that used homomorphic encryption to protect user privacy. The system utilizes auxiliary domain data to address data sparsity and cold-start problems.

4 Applications, datasets and platforms

FL has a unique modeling mechanism that facilitates training models with data from multiple parties without compromising data privacy. One of the promising applications of federated recommenders is e-commerce or e-shopping. Smart retail provides personalized services to customers using machine learning techniques. User interests, user purchasing power, and item features are the main characteristics of innovative retail business, but these three datasets are likely to be scattered across different departments. Data is heterogeneous, and it is difficult to break the data barriers. Federated learning recommenders can address these problems while preserving privacy.

A computerized healthcare system can benefit immensely from FRs [74]. Medical datasets are sensitive, private, and usually scattered across different hospitals; for example, a patient may go to one hospital for a blood test and another for a pathology test. It is hard to accumulate data from various medical centers. Previous machine learning approaches have performance issues due to data insufficiency and lack of data labels. FRs can play an essential role in recommending treatment for a patient based on medical history without compromising the patient's privacy. A federated recommender with transfer learning is the most desirable solution for tackling data issues and improving healthcare.

The entertainment industry is another important application of FRs. Entertainment includes movies, videos, TV programs, and music. This industry is growing fast and needs customer retention to get better ratings and views. FRs can help understand users' interests and suggest items according to their taste without moving the enormous data in one place. Similarly, FRs can help generate personalized recommendations for books, newspapers, blogs, and documents, thus improving users' experience and saving their time from tiresome searches.

Services recommendations involve traveling, restaurants, experts for consultation. Tourism is increasing globally, and recommendation applications can play an important role in recommending places and trip advisors to users according to their interests. On the other hand, food recommendations are also prevalent. Restaurants can increase their sales and revenue by providing users with appropriate food suggestions. Many apps are developed for ordering online food from favorite restaurants. These apps can increase their rating and orders by providing a recommendation facility based on the user likes\dislikes. A FL recommender helps users by providing relevant suggestions based on the data on their devices.

Datasets Data plays a significant role in analyzing and testing new algorithms. To examine the efficiency, robustness, scalability, and reliability of a federated algorithm, one needs datasets that can capture the real-world essence. Table 6 shows datasets that are commonly used by researchers in evaluating FRs.

Platforms Some open-source platforms are developed for simulations of FL algorithms. Few production-oriented platforms are also available. Table 7 provides detail of these platforms.

5 Challenges and limitations

The current FRs face various issues while employing the constraints and requirements of the recommendation engine in the federated paradigm. Most of the challenges arise due to different aspects of FL, such as heterogeneous and non-IID data, distributed environment, malicious users, and edge devices. FRs must preserve user privacy, perform optimal client selection and find a good mix between communication cost and model accuracy. This section summarizes the current limitations, highlights the areas

Table 6 Federated recommender datasets

Dataset name	Description	Refs.
LEAF	Contains datasets of Twitter, Shakespeare, FEMNIST and many more	[75]
MovieLens	It has four datasets of MovieLens 100K, 1M, 10M, and 20M	[76]
YFCC100M	Dataset by Yahoo Flicker with 100M media objects	[77]
BookCrossings	Dataset scrapped from book rating website	[78]
eICU	It's a collaborative research database for critical care research	[79]
Yelp	It contain ratings, reviews provided by users to different businesses	[80]
TripAdvisor	TripAdvisor hotel dataset can be scrapped manually	[81]
News	News dataset scrapped from commercial news websites	[82]
Hetrec-movielens	It is an extension of MovieLens10M dataset	[83]
Last.fm	This dataset contains music listening information from 2K users	[83]
CiteULike	This data contains information of users and their articles with/without tag information	[84]
FilmTrust	FilmTrust is a dataset crawled from the entire FilmTrust website	[62]
Foursquare	This data considered as a reference for evaluating PoI recommendation models	[46]
UAH-DriveSet	This dataset contain features of the vehicle, road, and traffic	[73]

Table 7 Federated learning frameworks

Framework	Description	Refs.
Simulation based frameworks		
TensorFlow Federated	Designed for FL development environment. It combines tensorflow and communication	[85]
LEAF	It provides an evaluation framework and open-source datasets for FL	[75]
PySyft	It is a python library that supports deep learning, FL, and multi-party computation	[86]
Flower	New FL framework that supports system-related challenges and algorithmic research	[87]
IBM FL	It's a python framework for FL that supports Deep Neural Networks, Decision ID3, Linear classification with SGD, K-mean, and Naïve Bayes. It's an ongoing effort	[88]
Production-oriented platforms		
FATE	It's an open source project that supports federated AI ecosystem by providing secure framework	[89]
Clara training framework	Based on centralized FL with data privacy protection	[90]
PaddleFL	It provides strategies for FL and training along with demonstration	[91]

that need improvements, and suggests possible directions for future work that can yield fruitful results.

5.1 Trade off between communication cost and accuracy

FL requires several communication rounds between clients and the server to train a high-quality central model. Network communication is quite expensive and has adverse effects on system speed. Different attempts are made to reduce the communication cost in FL [12]. However, little work is done in the area of FRs. We can combine quantization techniques with federated averaging to reduce cost with minimal compromise on accuracy. It is important to note that there is a trade-off between communication cost and accuracy. Accuracy usually improves with the increase in data, but more data leads to an upsurge in the communication cost. It is challenging to find an optimal balance between accuracy and communication cost [8]. We need to develop computationally efficient federated recommendation models that converge fast and have lower communication costs with improved accuracy. Different researchers tried to achieve improved accuracy with lower communication cost [12, 48, 57]. In [61], an effort is made to reduce the model size by sharing a partial global model using the multi-armed bandit approach. However, theoretical bounds

regarding the convergence of the model and regret bounds for the proposed reward function demand a thorough investigation. Another aspect to consider is the dependence of model size on the number of items to recommend, making large federated recommender systems infeasible. One of the key challenges is to break this direct dependence.

Furthermore, we can reduce communication costs by deploying a scheme that minimizes the number of clients in each FL round without compromising recommendation quality. We can utilize different clustering algorithms to cluster similar users and select a few representative clients from each cluster. This area can be a focus for future research on FRs. In [48], it is highlighted that reduction in the total number of users in each training round can make FedFast more efficient. In most studies, it is assumed that each user/client is equally important irrespective of its contribution in terms of the amount of data. Treating users based on their contribution can be another exciting aspect of federated recommender systems.

5.2 Model security and user privacy concerns

It is one of the major issues that need to be addressed in FRs. Although FL guarantees that data never leaves the client, there is still a chance of information leakage that can violate user privacy. One way to handle privacy issues in FRs is to use DP [54, 62, 57, 53, 58] or HE [54, 62]. DP adds randomness to the user data to protect privacy while compromising accuracy, and the situation worsens if convergence requires more FL rounds. On the other hand, HE is limited to very few operations and is computationally expensive. The need is to develop a privacy-preserving federated mechanism that is efficient in model training, protects the user data, and does not impact accuracy. Few papers have studied the privacy threats for different types of recommenders [60]. In FedeRank [45], authors propose a unique idea to share partial model updates with the central server to avoid model inversion attacks. A recent approach [64] blends differential privacy with locality sensitive hashing to design scalable federated recommender systems. A promising research area is to identify potential threats for various types of FRs using different optimization algorithms.

Moreover, the server can compromise data security and user privacy if it cannot detect malicious users. In FL, the server is exposed to model-poisoning attacks as the global model is built by aggregating client updates and has no insight into how these updates are generated. Mischievous clients can train the local model on backdoor data and send the malicious updates to the server; this could affect the global model after federated averaging. In [92], a model replacement methodology is proposed that exposes these risks of FL. In addition to model poisoning, fake users can make shilling attacks in recommenders to degrade the recommendation quality. In traditional recommenders, different efforts are made to address this issue, but they are not yet incorporated in FRs [93, 94]. It is critically important to make the server more secure to detect attacks and sustain its reliability. The current FRs are vulnerable to malicious attacks [8, 48, 49]. Some thoughtful efforts are required in this direction to make FL models protected against such attacks.

5.3 Handling huge heterogeneous and non-IID data

Massive data is an invaluable resource for model training and generating high-quality recommendations. However, the network communication cost increases enormously with the increase in data and model size. We need an optimal mechanism to effectively utilize big data in a federated environment. Zhuo et al. [53] proposed an online contextual federated recommender that can handle big data and data scarcity but has a lower convergence rate. In [9], it is highlighted that model size dependencies on the number of items to recommend should be removed. Moreover, the data in FL is usually heterogeneous and non-IID (Independent and Identical). The statistical heterogeneity and non-Identical data distribution make federate recommendations challenging and complicated. The real-world federated datasets suffer from various issues like feature distribution skewness and label distribution skewness. The uneven distribution of the data is mainly because different clients hold different amounts of data. Better characterization of datasets is an open question, and it requires the development of optimal strategies to deal with heterogeneous and unbalanced data in federated recommender systems. Meta-learning can come to the rescue; it is an effective approach for learning non-identical and personalized data. It is also popular for model adaptation and predicting the performance of ML algorithms. In [12], the recommendation is performed by applying federated meta-learning to overcome the issues in data. Granular computing [95] can also be used for dealing with the imbalanced datasets. It an important technique for identifying the optimal granularity under an imbalanced dataset.

5.4 Selecting best architecture for FRS-scalable and reliable

Current work in FRs is mainly related to centralized FL. The issue with centralized FR is that the central server is a single point of failure, as it is responsible for model updates, communications between clients and resource provisioning [40]. The decentralized approach relies on

peer-to-peer communication and eliminates the need for a central server. In a fully decentralized FR, all the clients hold their private information and a copy of a global model. Clients compute gradients of the local and global models using their private data and communicate updates to their neighboring clients in a connected graph topological network. The neighboring node merges its global model with the received model and then updates the local and resultant merged global models.

The decentralized recommenders face some significant challenges due to clients' limited availability and lack of trusted authority to manage model training and resource provisioning [96]. For the decentralized network, an optimization algorithm and a decentralized variant of SGD have been considered [97–99]. Heged et al. [100] presented an empirical comparison of centralized FL and decentralized gossip learning for recommender systems. Gossip learning does not require a server; all clients exchange models directly and are equally important in forming a P2P network. The results showed that decentralized gossip learning is comparable to centralized FL. Thus, an efficient, cheap, and fully trusted decentralized network can replace a centralized approach in FRs.

5.5 Selecting best mode of communication

Synchronous communication requires clients to exchange information immediately in real-time. In contrast, asynchronous communication does not compel all clients to respond immediately to exchange data. Both approaches have pros and cons; rigorous experimental comparisons are required on real-world datasets to determine which works best for FRs. In federated training, the server selects limited numbers of devices in each epoch. Asynchronous communications do not rely on all the selected devices to respond in each epoch; instead, it immediately updates the global model whenever the client receives the updates. It introduces the staleness problem, i.e., model updates occur when some users are computing their gradients, resulting in gradients computation using outdated parameters. In [8], it is highlighted that it could be an online learning environment where updates from clients arrive in a continuously asynchronous fashion for real-world scenarios; this scenario needs to be tested for FRs. Synchronous communication requires all the selected devices to respond before the aggregation step at each epoch's end. It is not guaranteed that every device would be available at the end of the epoch due to different battery times, computational complexity, and lousy networking. Waiting for slow devices or stragglers makes synchronous communication drop the current epoch on server timeout. Synchronous models resolve the staleness problem, but it is slow and costly. More work is required to make it fault-tolerant to an arbitrary number of slow devices. K Muhammad et al. [48] used extended synchronous FedAvg algorithm [69] and show that the proposed solution has fast convergence and low communication cost, but it can further be improved by reducing the total number of users in each training round.

5.6 Incorporating recommendation constraints

The federated recommender attempts to improve recommendation accuracy while overcoming various recommendation challenges and constraints. Most of the recently proposed FRs are based on the idea of CF and use MF, DL models, and rating data to generate recommendations, but they usually do not include side information. We can incorporate side information such as user and item features to improve recommendation quality [9]. The inclusion of additional features not only adds more value to the recommender systems but can also help address recommendation challenges like the cold-start problem. Cold-start problem is common in recommenders, and much work has been done to cope with this issue. However, few FRs have attempted to handle this problem yet [9, 58]. In the field of FRs, the cold-start is still an open issue, and a proper mechanism is required to deal with it. Furthermore, different approaches like collaborative filtering and content-based can be combined to improve FRs' efficiency and accuracy. Several traditional hybrid recommenders show promising results [21, 101], but in the domain of FRs, a hybrid approach still serves as future work.

We need FRs for specialized real-world applications that incorporate domain-specific constraints and requirements while maintaining user privacy. For example, in the case of event recommendations, we have to consider various aspects in addition to the user-item interactions like event lifetime, geographical location, reviews, and others [102, 103]. Similarly, we need advanced federated job recommenders, tag recommenders, medical recommenders, hotel recommenders [104]. Few FRs are developed for specific domains. Recently proposed Fed4Rec [51] recommender works well for page recommendation. Nevertheless, Fed4Rec is a general framework and can be extended to various domains such as video recommendation or product recommendation. Similarly, the proposed privacy-related federated recommender like PPRSF [60] can be deployed to the real world information services that deal with sensitive user information, such as medical and financial services.

Furthermore, to improve recommendation precision we need to integrate emerging recommendation concepts in a federated environment, like utilizing helpfulness rating [105], user reviews, and short-term user interactions. Recommenders consider the entire user-item interaction history to model user preferences while ignoring short-term

transactions, and thus they ignore the change in users' preferences over time. This leads to poor quality and erratic recommendations. Session-based recommender systems tend to learn changes in user preferences over time by considering user-item interaction history divided into sessions. A session can multiple purchased items in one shopping event, web pages visited by a user in one internet surf, or songs listened for a particular duration (a day or a month). Designing session-based recommendation systems using FL can be an exciting area to explore.

Lastly, current FRs are mostly simulated, which means their performance is examined in a controlled virtual environment. It is unknown how they might perform in the real world, which could be significantly different due to various factors and unforeseen circumstances. The need is to deploy the current FRs in the real world to assess their performance and recommendation quality [12, 48].

6 Discussions and future direction

This study reviews the latest developments in federated recommenders and highlights the shortcomings in current approaches along with the opportunities and future work. The crux of this study is that the future is federated recommendation systems. Recommenders are deployed over centralized data repositories for a long time, yet modern requirements demand federated recommender methods. It is tricky and cumbersome to incorporate the best recommendation techniques in a federated paradigm that has multiple issues of its own. Recently, researchers have started working in this area, it is new, and there is ample room for improvement and research. This section presents research directions that are exciting to investigate in the future. The Fig. 12 point out possible future directions, classified based on the high-level challenges identified in the previous section.

One of the critical focus areas in FRs is to enhance the system's accuracy without increasing the communication cost(payload). Researchers have explored various directions in this line of research; however, each has its shortcomings and limitations. A thorough investigation should be conducted in this avenue to achieve the best trade-off between accuracy and network cost. Future work can study the effects of using additional local training to update the local model and analyze the impact on the global model. This can improve model accuracy and provide user-tailored recommendations without increasing the communication payload. In addition, analysis of the communication payloads and system efficiency could help evaluate the other practical aspects of such systems. Another possible direction to reduce communication payload is to decrease the number of clients in each FL round without affecting recommendation quality. We can utilize advanced clustering algorithms to cluster similar users and select a few representative clients from each cluster.

Furthermore, studies have shown that incorporating side-information such as user and item feature vectors enhances recommendation quality for datasets inherently sparse. The production and real-world datasets are sparse; thus, they can greatly benefit from this. The proposed FED-MVMF approach [9] takes advantage of side-information in a federated paradigm to yield better recommendations without moving the user's personal data and features to a central server, thus maintaining user privacy. Moreover, federated MVMF can overcome the cold-start problem. However, all these benefits are achieved at the cost of increased payload. Additional studies are needed in this direction to develop multi-view techniques that do not significantly increase payload. The current FED-MVMF approach [9] does not perform well for cold-start item scenarios mainly due to the lower quality of the item side-information source. Detailed experiments should be carried out to explore the reasons and rectify them. One possible solution direction to deal with the cold start problem without increasing payload is to reduce the data requirements of MetaMF [50] using few-shot or zero-shot learning. The emerging optimization techniques such as elephant herding optimization,butterfly and earthworm optimization, can be utilized to effectively reduce communication and handle systems constraints [106, 107]. The federated optimization algorithms allow for low participation and local updating at clients [108].

Another critical key area for future work is security in FRs. The new emerging techniques for security need to be rigorously tested for their ability to handle attacks and threats. Providing privacy to the users is of utmost concern in the federated paradigm. The current techniques deployed to ensure user data privacy are faulty and have many performance issues. Differential privacy, a promising technique for ensuring user privacy, does not work well with FRs as they need multiple iterations to converge the model. Ribero et al. [57] introduce the idea of federated prototyping to overcome this issue. However, the idea is in the early stage and works only in a specific scenario where each user's data is linked with different entities in the dataset. As future work, this approach can be extended to real-world scenarios (from commerce or content sites) where each entity represents a single individual. Moreover, rigorous investigations should be carried out to find error bounds for the reconstructed matrix in the proposed federated prototyping technique [57]. Another approach to secure user data is to encrypt the user gradients before sending them to the server using techniques like homomorphic encryption, which has shown promising results in FR. However, incorporating homomorphic encryption in

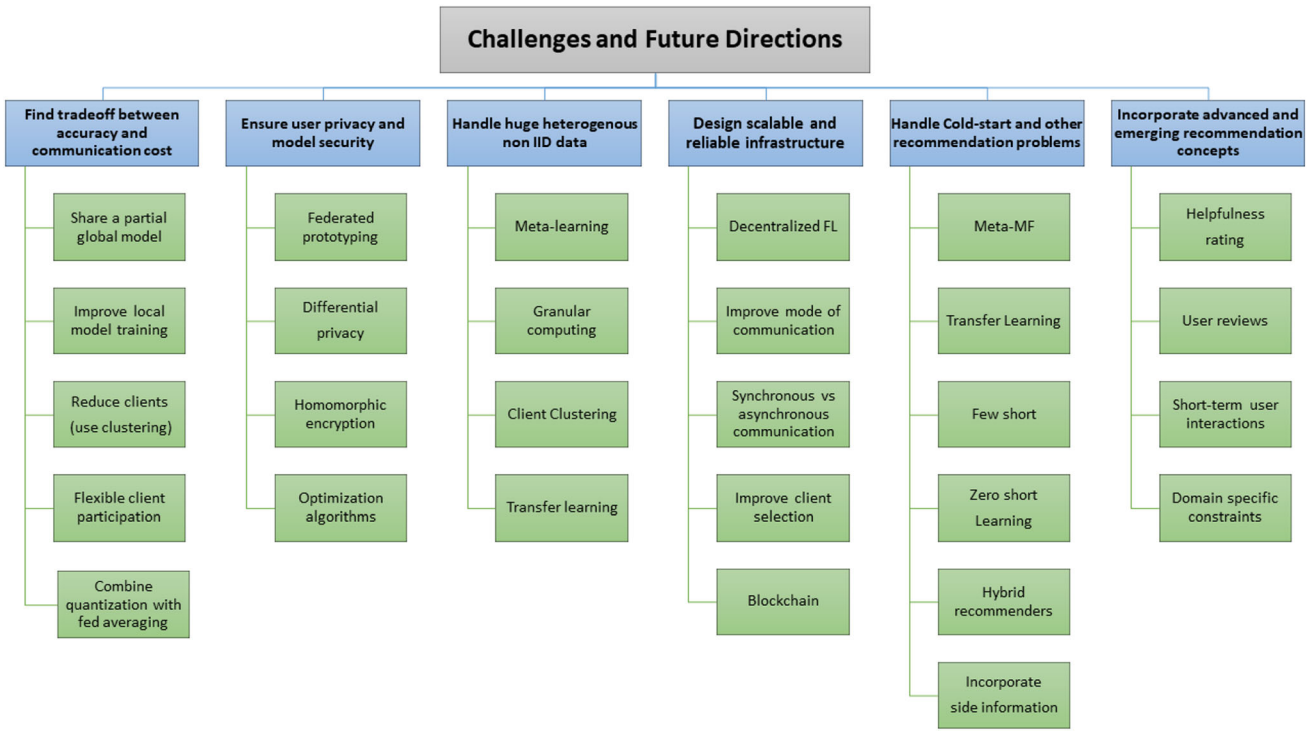


Fig. 12 High-level classification of challenges and future directions (challenges are in blue and future directions are in green) (Color figure online)

the model to ensure privacy directly affects the computing power and time, increasing the cost of a recommenders model training by many folds [62]. One possible research direction is to improve the homomorphic encryption's efficiency when performing operations on ciphertext to make these techniques practical and feasible for big data.

Gao et al. [66] study the potential privacy threats in all three types of federated recommender systems and observed that it is possible to infer private user information from the model parameters in the existing FRS. Different solutions are devised to overcome this issue. Few approaches propose to use meta-learning. The federated meta-learning framework (FedMeta) shares a parameterized algorithm (or metalearner) instead of a global model to preserve user privacy. At the same time, some researchers use gradients (instead of model parameters) to solve the problem. The gradients also help filter out Byzantine clients in a momentum-based FRS. These ideas are in the initial stages and need rigorous evaluation for real-world scenarios. A detailed, comprehensive study should be conducted to measure the threats against emerging techniques; FedMeta, ALS-based MF, and other recommenders. Furthermore, as future work, we need to explore the effects of different privacy attacks and the extent to which private data can be inferred. Another promising future direction is extending the current works to defend against targeted poisoning attacks and model aversion attacks.

One important future work is to test the existing FRs by simulating the real-world scenario, where client updates arrive asynchronously (online learning), to analyze the potential of existing recommenders [8]. Furthermore, most of the current FR systems are built on centralized FL. It is important to note that fully decentralized FL has significant benefits over centralized one as there is no single point of failure. Furthermore, in such systems, clients have a low cost of entry independent of the network size; hence they are scalable. FL is compared with the decentralized Gossip Learning framework for recommendation potential in [100],?. The results show that the performance of Gossip learning is comparable to FL in terms of convergence time and communication cost.

The area of decentralized recommenders is promising and needs to be explored further to build robust and scalable recommenders. We can achieve decentralization in FR by adopting blockchain technology. The central server can be replaced by the peer-to-peer blockchain mechanism, where blockchain nodes manage the task of combining the model updates, thus, overcoming the issue of unreliability in FR due to a single point of failure. Additionally, blockchain can offer reliable verification mechanisms to identify malicious local model updates. Moreover,

blockchain can also reward FR clients to encourage their participation and honest behavior [109, 110].

However, decentralized FR is not without issues and has its demons to handle. In decentralized FR systems, users tend to incur most of the training cost, and this causes performance degradation and an increase in communication payload, which can, in turn, frustrate users who are primarily using edge devices with limited resources. Eventually, this can reduce the acceptance of the federated recommender system, especially if users have to wait quite long to enjoy high-quality personalized recommendations. One possible solution is to develop techniques that converge fast and minimize the number of clients required to train the model, reducing the load on communication-related battery consumption. The need of the hour is to build a system that combines the best of both worlds (centralized and decentralized) to generate accurate recommendations without overloading the user devices.

Federated Recommendation is an ongoing field of research and has various promising future directions. Detailed future work is required to theoretically analyze the convergence of the different recommender models under various constraints and the optimal use of early stopping to get the maximum model performance [52]. Most FRS treat all users equally, paying no regard to the amount of data contributed by each user. It is unfair to users who provide more data. Some mechanism should be devised to assign weightage to users according to the amount of data they contribute [62]. Finally, the ranking prediction task [27] is vital in the area of recommendation system, and we need to evaluate the performance of MetaMF [50] and other FRs on the ranking prediction.

7 Conclusion

Federated learning is a promising solution for recommender systems in the era of Big data and cloud computing. Traditional recommenders are designed to run on a single processor, so they fail to thrive in today's world as they cannot deal with gigantic and distributed data. In the upcoming systems, federated recommender systems are the way to go. This paper provides a brief overview of traditional recommenders and presents new emerging federated recommenders. We highlighted the main challenges faced by the existing FRs and pointed out the future work. This area has enormous potential, and much research is needed to improve the recommender's performance. We also present the applications, datasets, and tools for FRs to facilitate future work. However, paper did not discussed the latest recommenders proposed for specific applications as it is beyond the scope. This paper provides a comprehensive review of the latest research on federated recommender systems and shows the future paths.

Author contributions It is a review article author ZA propose the idea for the paper. The literature review is conducted by ZA, FKK and SK. ZA and SK identified the challenges and future work. FKK constructed the figures and gathered information on tools, frameworks and datasets.

Funding None.

Data availability It is a review article and it summarizes the various datasets used in emerging federated recommenders. The details are mentioned in the data section of the paper.

Declarations

Conflict of interest All the authors declared that they have no conflict of interest to disclose.

Ethical approval The manuscript "Federated Recommenders: Methods, Challenges and Future" meets all the ethical requirements led by the Journal. This submission is guaranteed with the confirmation that the above-mentioned manuscript has not been published, accepted for publication elsewhere, or under editorial review for publication elsewhere. Furthermore, the work is not plagiarized and includes proper references wherever required.

References

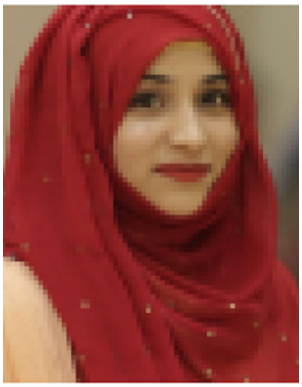
1. Wu, L., Xiangnan, H., Wang, X., Zhang, K., Wang, M.: A survey on neural recommendation: from collaborative filtering to information-rich recommendation. http://arxiv.org/abs/2104.13030v22104.13030v2 (2021)
2. Javed, U., Shaukat, K., Hameed, I.A., Iqbal, F., Alam, T.M., Luo, S.: A review of content-based and context-based recommendation systems. Int. J. Emerg. Technol. Learn. **16**, 274–306 (2021)
3. Awaysheh, F., Alazab, M., Garg, S., Niyato, D., Verikoukis, C.: Big data resource management and networks: taxonomy, survey, and future directions. IEEE Commun. Surv. Tutor. **23**, 2098–2130 (2021)
4. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency. http://arxiv.org/abs/1610.05492 (2016)
5. Wahab, O., Mourad, A., Otrok, H., Taleb, T.: Federated machine learning: survey, multi-level classification, desirable criteria and future directions in communication and networking systems. IEEE Commun. Surv. Tutor. **23**, 1342–1397 (2021)
6. Wang, X., Han, Y., Leung, V., Niyato, D., Yan, X., Chen, X.: Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun. Surv. Tutor. **22**, 869–904 (2020)
7. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. **10**, 1–19 (2019)
8. Ammad-Ud-Din, M., Ivannikova, E., Khan, S., Oyomno, W., Fu, Q., Tan, K., Flanagan, A. Federated collaborative filtering for privacy-preserving personalized recommendation system. http://arxiv.org/abs/1901.09888 (2019)

9. Flanagan, A., Oyomno, W., Grigorievskiy, A., Tan, K., Khan, S., Ammad-Ud-Din, M. Federated multi-view matrix factorization for personalized recommendations. http://arxiv.org/abs/2004.04256 (2020)
10. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191. Association for Computing Machinery, New York, NY, USA (2017)
11. Geyer, R.C., Klein, T., Nabi, M.: Differentially private federated learning: a client level perspective. http://arxiv.org/abs/1712.07557 (2017)
12. Chen, F., Luo, M., Dong, Z., Li, Z., He, X.: Federated meta-learning with fast convergence and efficient communication. http://arxiv.org/abs/1802.07876 (2018)
13. Smith, V., Chiang, C., Sanjabi, M., Talwalkar, A.: Federated multi-task learning. http://arxiv.org/abs/1705.10467 (2017)
14. Abreha, H., Hayajneh, M., Serhani, M.: Federated learning in edge computing: a systematic survey. Sensors **22**, 450 (2022)
15. Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., Venkatesh, B.: Content-based movie recommendation system using genre correlation. In: Satapathy, S.C., Bhateja, V., Das, S. (eds.) Smart Intelligent Computing And Applications, pp. 391–397. Springer, Singapore (2019)
16. Son, J., Kim, S.: Content-based filtering for recommendation systems using multiattribute networks. Expert Syst. Appl. **89**, 404–412 (2017)
17. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
18. Ramlatchan, A., Yang, M., Liu, Q., Li, M., Wang, J., Li, Y.: A survey of matrix completion methods for recommendation systems. Big Data Min. Anal. **1**, 308–323 (2018)
19. Lin, C., Wang, L., Tsai, K.: Hybrid real-time matrix factorization for implicit feedback recommendation systems. IEEE Access **6**, 21369–21380 (2018)
20. Alfarhood, M., Cheng, J.: DeepHCF: a deep learning based hybrid collaborative filtering approach for recommendation systems. In: Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 89-96 (2018)
21. Patro, S., Mishra, B., Panda, S., Kumar, R., Long, H., Taniar, D., Priyadarshini, I.: A hybrid action-related K-nearest neighbour (HAR-KNN) approach for recommendation systems. IEEE Access **8**, 90978–90991 (2020)
22. Zarzour, H., Al-Sharif, Z., Al-Ayyoub, M., Jararweh, Y. A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques. In: Proceedings of the 2018 9th International Conference On Information And Communication Systems (ICICS). pp. 102-106 (2018)
23. Sánchez, P., Bellogín, A.: Building user profiles based on sequences for content and collaborative filtering. Inf. Process. Manag. **56**, 192–211 (2019)
24. Pereira, N., Varma, S.: Financial planning recommendation system using content-based collaborative and demographic filtering. In: Tiwari, S., Trivedi, M.C., Mishra, K.K., Misra, A.K., Kumar, K.K. (eds.) Smart Innovations in Communication and Computational Sciences, pp. 141–151. Springer, Singapore (2019)
25. Çano, E., Morisio, M.: Hybrid recommender systems: a systematic literature review. Intell. Data Anal. **21**(6), 1487–1524 (2017)
26. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: a survey and new perspectives. ACM Comput. Surv. (CSUR) **52**(1), 1–38 (2019)
27. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 191–198 (2016)
28. Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1933–1942 (2017)
29. Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10. Association for Computing Machinery, New York, NY, USA (2016)
30. Li, P., Wang, Z., Ren, Z., Bing, L., Lam, W.: Neural rating regression with abstractive tips generation for recommendation. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 345–354 (2017)
31. Nguyen, H.T., Wistuba, M., Grabocka, J., Drumond, L.R., Schmidt-Thieme, L.: Personalized deep learning for tag recommendation. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 186–197. Springer (2017). https://doi.org/10.1007/978-3-319-57454-7_15
32. Chen, L., Wu, L., Hong, R., Zhang, K., Wang, M.: Revisiting graph based collaborative filtering: a linear residual graph convolutional network approach. Proc. AAAI Conf. Artif. Intell. **34**, 27–34 (2020)
33. Shams, B., Haratizadeh, S.: Graph-based collaborative ranking. Expert Syst. Appl. **67**, 59–70 (2017). https://doi.org/10.1016/j.eswa.2016.09.013
34. Musto, C., Basile, P., Lops, P., de Gemmis, M., Semeraro, G.: Introducing linked open data in graph-based recommender systems. Inf. Process. Manag. **53**(2), 405–435 (2017). https://doi.org/10.1016/j.ipm.2016.12.003
35. Leng, J., Jiang, P.: Mining and matching relationships from interaction contexts in a social manufacturing paradigm. IEEE Trans. Syst. Man Cybern. **47**, 276–288 (2017)
36. Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., He, Q. A survey on knowledge graph-based recommender systems. In: Proceedings of the IEEE Transactions on Knowledge and Data Engineering, pp. 1–1 (2020)
37. Li, G., Wang, G., Dong, J., Yeh, W., Li, K.: DLEA: a dynamic learning evolution algorithm for many-objective optimization. Inf. Sci. **574**, 567–589 (2021)
38. Li, W., Gai, W., Gandomi, A.: A survey of learning-based intelligent optimization algorithms. Arch. Comput. Methods Eng. **28**, 1–19 (2021)
39. Etemadi, M., Ghobaei-Arani, M., Shahidinejad, A.: Resource provisioning for IoT services in the fog computing environment: an autonomic approach. Comput. Commun. **161**, 109–131 (2020)
40. Shahidinejad, A., Ghobaei-Arani, M., Masdari, M.: Resource provisioning using workload clustering in cloud computing environment: a hybrid approach. Clust. Comput. **24**, 319–342 (2021)
41. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), vol. 7, pp. 94–103. IEEE (2007)
42. Mugunthan, V., Polychroniadou, A., Byrd, D., Balch, T.H.: Smpai: secure multi-party computation for federated learning. Neural Information Processing. https://www.jpmorgan.com/jpmpdf/1320748217124.pdf (2019)

43. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., Zhou, Y.: A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pp. 1–11 (2019)
44. Giacomelli, I., Jha, S., Joye, M., Page, C.D., Yoon, K.: Privacy-preserving ridge regression with only linearly-homomorphic encryption. Cryptology ePrint Archive, Report 2017/979. https://eprint.iacr.org/2017/979 (2017)
45. Anelli, V., Deldjoo, Y., Di Noia, T., Ferrara, A., Narducci, F. FedeRank: user controlled feedback with federated recommender systems. http://arxiv.org/abs/2012.11328 (2020)
46. Anelli, V.W., Deldjoo, Y., Di Noia, T., Ferrara, A., Narducci, F.: How to put users in control of their data in federated top-n recommendation with learning to rank. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing. SAC '21, pp. 1359–1362. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3412841.3442010
47. Chen, C., Zhang, J., Tung, A.K.H., Kankanhalli, M., Chen, G.: Robust federated recommendation system. http://arxiv.org/abs/2006.08259 (2020)
48. Muhammad, K., Wang, Q., O'Reilly-Morgan, D., Tragos, E., Smyth, B., Hurley, N., Geraci, J., Lawlor, A.: Fedfast: going beyond average for faster training of federated recommender systems. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1234–1242. Association for Computing Machinery, New York, NY, USA (2020)
49. Jalalirad, A., Scavuzzo, M., Capota, C., Sprague, M.: A simple and efficient federated recommender system. In: Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, pp. 53–58 (2019)
50. Lin, Y., Ren, P., Chen, Z., Ren, Z., Yu, D., Ma, J., Rijke, M.D., Cheng, X.: Meta matrix factorization for federated rating predictions. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 981–990. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3397271.3401081
51. Zhao, S., Bharati, R., Borcea, C., Chen, Y. Privacy-aware federated learning for page recommendation. In: Proceedings of the 2020 IEEE International Conference On Big Data (Big Data), pp. 1071–1080 (2020)
52. Shi, B., Tragos, E.Z., Ozsoy, M.G., Dong, R., Hurley, N., Smyth, B., Lawlor, A.: Dares: an asynchronous distributed recommender system using deep reinforcement learning. IEEE Access **9**, 83340–83354 (2021). https://doi.org/10.1109/ACCESS.2021.3087406
53. Zhou, P., Wang, K., Guo, L., Gong, S., Zheng, B.: A privacy-preserving distributed contextual federated online learning framework with big data support in social recommender systems. In: Proceedings of the IEEE Transactions on Knowledge and Data Engineering (2019)
54. Chai, D., Wang, L., Chen, K., Yang, Q.: Secure federated matrix factorization. http://arxiv.org/abs/1906.05108 (2019)
55. Wang, X., Zheng, X., Liang, X.: Charging station recommendation for electric vehicle based on federated learning. J. Phys. **1792**, 012055 (2021)
56. Wang, L.-E., Wang, Y., Bai, Y., Liu, P., Li, X.: POI recommendation with federated learning and privacy preserving in cross domain recommendation. In: Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6. IEEE (2021)
57. Ribero, M., Henderson, J., Williamson, S., Vikalo, H. Federating recommendations using differentially private prototypes. http://arxiv.org/abs/2003.00602 (2020)
58. Qi, T., Wu, F., Wu, C., Huang, Y., Xie, X.: FedRec: privacy-preserving news recommendation with federated learning. http://arxiv.org/abs/2003.09592 (2020)
59. Wang, Y., Tian, Y., Yin, X., Hei, X.: A trusted recommendation scheme for privacy protection based on federated learning. CCF Trans. Netw. **3**(3), 218–228 (2020)
60. Qin, J., Liu, B., Qian, J.: A novel privacy-preserved recommender system framework based on federated learning. In: Proceedings of the 2021 The 4th International Conference On Software Engineering and Information Management. pp. 82-88 (2021)
61. Khan, F.K., Flanagan, A., Tan, K.E., Alamgir, Z., Ammad-Ud-Din, M.: A payload optimization method for federated recommender systems. In: Proceedings of the Fifteenth ACM Conference on Recommender Systems. RecSys '21. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3460231.3474257
62. Du, Y., Zhou, D., Xie, Y., Shi, J., Gong, M.: Federated matrix factorization for privacy-preserving recommender systems. Appl. Soft Comput. **111**, 107700 (2021)
63. Ali, W., Kumar, R., Deng, Z., Wang, Y., Shao, J.: A federated learning approach for privacy protection in context-aware recommender systems. Comput. J. **64**, 1016–1027 (2021)
64. Hu, H., Dobbie, G., Salcic, Z., Liu, M., Zhang, J., Lyu, L., Zhang, X. Differentially private locality sensitive hashing based federated recommender system. Concurrency and Computation: Practice And Experience. pp. e6233 (2021)
65. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, pp. 223–238. Springer (1999)
66. Gao, D., Tan, B., Ju, C., Zheng, V., Yang, Q. Privacy threats against federated matrix factorization. http://arxiv.org/abs/2007.01587 (2020)
67. Kabbur, S., Ning, X., Karypis, G.: Fism: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 659–667 (2013)
68. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. http://arxiv.org/abs/1412.6980 (2014)
69. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
70. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, Vol. 70, pp. 1126–1135. JMLR.org (2017)
71. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. http://arxiv.org/abs/1803.02999 (2018)
72. Lu, T., Pál, D., Pál, M.: Contextual multi-armed bandits. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 485–492 (2010)
73. Vyas, J., Das, D., Das, S.K.: Vehicular edge computing based driver recommendation system using federated learning. In: Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 675–683. IEEE (2020)
74. Antunes, R., Costa, C., Küderle, A., Yari, I., Eskofier, B.: Federated learning for healthcare: systematic review and architecture proposal. ACM Trans. Intell. Syst. Technol. (TIST) (2022)
75. Caldas, S., Duddu, S.M.K., Wu, P., Li, T., Konečnỳ, J., McMahan, H.B., Smith, V., Talwalkar, A.: LEAF: a benchmark for federated settings. http://arxiv.org/abs/1812.01097 (2019)

76. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. **5**(4), 1–19 (2015)
77. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.-J.: Yfcc100m: the new data in multimedia research. Commun. ACM **59**(2), 64–73 (2016)
78. Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web, pp. 22–32. ACM, (2005)
79. Pollard, T.J., Johnson, A.E., Raffa, J.D., Celi, L.A., Mark, R.G., Badawi, O.: The eicu collaborative research database, a freely available multi-center database for critical care research. Sci. Data **5**, 1–13 (2018)
80. Yelp: yelp open dataset. https://www.yelp.com/dataset
81. TripAdvisor: tripadvisor dataset. http://times.cs.uiuc.edu/wang296/Data/
82. Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., et al.: Mind: a large-scale dataset for news recommendation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3597–3606 (2020)
83. Cantador, I., Brusilovsky, P., Kuflik, T.: Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In: RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems, pp. 387–388 (2011). https://doi.org/10.1145/2043932.2044016
84. Wang, H., Chen, B., Li, W.-J.: Collaborative topic regression with social regularization for tag recommendation. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 2719–2725. AAAI Press (2013)
85. Authors, T.T.: TensorFlow federated. https://www.tensorflow.org/federated (2019)
86. Ryffel, T., Trask, A., Dahl, M., Wagner, B., Mancuso, J., Rueckert, D., Passerat-Palmbach, J.: A generic framework for privacy preserving deep learning (2018)
87. Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., Lane, N.D.: Flower: a friendly federated learning research framework (2020)
88. IBM: IBM federated learning. https://github.com/IBM/federated-learning-lib.git (2020)
89. Authors, T.F.: Federated AI technology enabler. https://www.fedai.org/ (2019)
90. Authors, T.C.T.F.: NVIDIA Clara. https://developer.nvidia.com/clara (2019)
91. Authors, T.P.: PaddleFL. https://github.com/PaddlePaddle/PaddleFL (2019)
92. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: Chiappa, S., Calandra, R. (eds.) International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 108, pp. 2938–2948. PMLR (2020)
93. Chirita, P.-A., Nejdl, W., Zamfir, C.: Preventing shilling attacks in online recommender systems. In: Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, pp. 67–74. Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1097047.1097061
94. Zhou, W., Wen, J., Qu, Q., Zeng, J., Cheng, T.: Shilling attack detection for recommender systems based on credibility of group users and rating time series. PLoS ONE **13**, e0196533 (2018)
95. Leng, J., Chen, Q., Mao, N., Jiang, P.: Combining granular computing technique with deep learning for service planning under social manufacturing contexts. Knowl.-Based Syst. **143**, 295–306 (2018)
96. Khorsand, R., Ghobaei-Arani, M., Ramezanpour, M.: A self-learning fuzzy approach for proactive resource provisioning in cloud environment. Software **49**, 1618–1642 (2019)
97. Colin, I., Bellet, A., Salmon, J., Clémençon, S.: Gossip dual averaging for decentralized optimization of pairwise functions. http://arxiv.org/abs/1606.02421 (2016)
98. Bellet, A., Guerraoui, R., Taziki, M., Tommasi, M.: Personalized and private peer-to-peer machine learning. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 473–481 (2018)
99. Elgabli, A., Park, J., Bedi, A.S., Bennis, M., Aggarwal, V.: Gadmm: Fast and communication efficient framework for distributed machine learning. http://arxiv.org/abs/1909.00047 (2019)
100. Hegedűs, I., Danner, G., Jelasity, M.: Decentralized recommendation based on matrix factorization: a comparison of gossip and federated learning. In: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 317–332. Springer (2019)
101. Dhruv, A., Kamath, A., Powar, A., Gaikwad, K.: Artist recommendation system using hybrid method: a novel approach. In: Shetty, N.R., Prasad, N.H., Nalini, N. (eds.) Emerging Research in Computing, Information, Communication and Applications, pp. 527–542. Springer, Singapore (2019)
102. Ma, Y., Chen, G., Wei, Q.: Finding users preferences from large-scale online reviews for personalized recommendation. Electron. Commer. Res. **17**(1), 3–29 (2017)
103. Chen, J., Zhang, C., Niu, Z.: Identifying helpful online reviews with word embedding features. In: Proceedings of the International Conference on Knowledge Science, Engineering and Management, pp. 123–133. Springer (2016)
104. Lee, P.-J., Hu, Y.-H., Lu, K.-T.: Assessing the helpfulness of online hotel reviews: a classification-based approach. Telemat. Inform. **35**(2), 436–445 (2018)
105. Cheng, Z., Ding, Y., Zhu, L., Kankanhalli, M.: Aspect-aware latent factor model: rating prediction with ratings and reviews. http://arxiv.org/abs/1802.07938 (2018)
106. Li, W., Wang, G., Alavi, A.: Learning-based elephant herding optimization algorithm for solving numerical optimization problems. Knowl.-Based Syst. **195**, 105675 (2020)
107. Li, W., Wang, G. Elephant herding optimization using dynamic topology and biogeography-based optimization based on learning for numerical optimization. Engineering With Computers. pp. 1-29 (2021)
108. Li, J., Li, Y., Tian, S., Xia, J.: An improved cuckoo search algorithm with self-adaptive knowledge learning. Neural Comput. Appl. **32**, 1–31 (2019)
109. Leng, J., Zhou, M., Zhao, J., Huang, Y., Bian, Y. Blockchain security: a survey of techniques and research directions. In Proceedings of the IEEE Transactions on Services Computing. pp. 1–1 (2020)
110. Li, C., Yuan, Y., Wang, F. Blockchain-enabled federated learning: a survey. In: Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), pp. 286–289 (2021)

Zareen Alamgir She is an associate professor at the Department of Computer Science, NUCES Lahore. Her research interests include Data Science, Federated Learning, Big Data, and Recommendation Systems. She has written many articles in the field of data analysis and algorithms.



Farwa K. Khan She is currently pursuing a Master in Computer science from NUCES, Lahore. Her research interest includes Federated Learning, Data Science, and Data Analysis.

Saira Karim She is an assistant professor at the Department of Computer Science, NUCES Lahore. Her research interests include Social Network Analysis, Federated Learning, Algorithms, and Graph neural networks.