Privacy preserving Federated Learning based Recommendation Engine

Navya Manjari Uppaluri[1],

[1]Faculty of Computer Science, Dalhousie University, Halifax, Canada
nv574116@dal.ca,

Abstract

In today's data-driven environment, safeguarding user privacy is a top priority, particularly in machine learning applications. Our study introduces an innovative approach that combines the privacy-preserving attributes of federated learning with the advanced capabilities of transformer-based models, specifically tailored for recommendation systems. Federated learning emerges as a decentralized alternative to traditional machine learning, enhancing both user privacy and data security. The models performance is analyzed using the movielens-1m datasets. The empirical results are compelling: the federated model achieves a notable 0.865 MAE in the global model. This research not only highlights the effectiveness of federated learning in boosting model accuracy but also emphasizes its crucial role in preserving user privacy. Our findings illustrate that integrating federated learning can lead to enhanced performance in recommendation systems without sacrificing data privacy. Consequently, this research marks a significant step forward in developing more effective, privacy-conscious machine learning solutions, contributing to the broader field of ethical and responsible AI.

Introduction

In the evolving digital ecosystem, recommendation systems have become integral in shaping user experiences across diverse platforms, from e-commerce to social networking. These systems are instrumental in tailoring user interactions, as observed by Kim and Chen (Kim and Chen 2015). They underline the substantial growth in this field, particularly noting advancements in methodologies like collaborative filtering, which have adapted to modern digital trends including social media. Yet, the effectiveness of these recommendation systems is often contingent upon the analysis of substantial user data. This reliance raises significant privacy and data security concerns (Mazeh and Shmueli 2020). Traditional models, primarily based on centralized data collection, pose notable privacy risks despite their efficiency. Such concerns are increasingly relevant amidst evolving data protection regulations and growing public demand for enhanced data privacy. In response, federated learning presents a viable solution. As a distributed machine learning approach, it facilitates learning from decentralized data sources without necessitating direct data sharing, thereby aligning with the surging need for privacy-preserving techniques in machine learning (Ammad-Ud-Din et al. 2019). Applying federated learning to recommendation systems could effectively mitigate the privacy challenges inherent in traditional methodologies.

Problem of Study

Recommendation systems have evolved significantly since their inception, transitioning from traditional methods to incorporating advanced techniques like deep learning. This evolution is crucial for understanding the context in which transformer-based federated learning models operate within recommendation systems.

Traditional recommendation systems have been classified mainly into collaborative filtering, content-based filtering, and hybrid approaches. Collaborative filtering relies on the assumption that users who agreed in the past tend to agree again in the future. In contrast, content-based filtering recommends items similar to those a user liked in the past, based on item features. Hybrid systems combine these approaches to leverage their respective strengths while mitigating their weaknesses (Dong et al. 2022).

Several studies have refined these traditional methods. For instance, user-based and item-based collaborative filtering have been extensively optimized to improve scalability and accuracy. Despite their success, traditional methods struggle with privacy issues and limited ability to capture complex user behavior patterns.

The advent of deep learning has brought significant advancements in recommendation systems. Neural networks, with their ability to model complex non-linear relationships, have been increasingly applied to enhance the performance of these systems. This shift is evident in the growing body of literature exploring various deep learning architectures for recommendation purposes (Li et al. 2023).

Li et al. (Li et al. 2023) provide a comprehensive survey of the recent developments in recommender systems, highlighting the integration of deep learning methods, especially in personalized and group-based systems. Their survey categorizes personalized recommendation systems into collaborative filtering, content-based, knowledge-based, and hybrid systems, noting the increased popularity of deep learning-enhanced methods.

Furthermore, Dong et al. (Dong et al. 2022) reviews the history of web recommender systems, emphasizing the transformation brought about by the integration of deep learning techniques. Their work traces the progression from early recommendation models to the current state, where deep learning plays a pivotal role. Challenges remain in computational complexity, the need for large datasets, and integrating deep learning with user privacy considerations.

Context-aware recommender systems (CARS) represent a significant step forward, incorporating contextual information to refine recommendations. Casillo et al. (Occhioni et al. 2023) discusses the introduction of context analysis techniques in recommender systems, especially highlighting their application in the cultural heritage field. This approach aligns with the current trend towards more personalized and situation-specific recommendation systems, a domain where deep learning, including transformer-based models, shows great promise. Recent studies have successfully integrated contextual data to improve the relevance of recommendations in various domains.Issues such as the complexity of context integration, scalability, and maintaining user privacy remain.

Related Work

The concept of Federated Learning (FL) has emerged as a significant advancement in the realm of machine learning, particularly in the context of decentralized data and privacy preservation.

McMahan et al. (McMahan et al. 2017) introduced the concept of Federated Learning as a method for learning deep networks from decentralized data. They proposed an approach termed FedAvg, which involves aggregating locally-computed updates to learn a shared model while keeping the training data on the mobile devices. This method addresses the privacy concerns and communication efficiency in decentralized settings, laying the groundwork for subsequent research in FL.

Li et al. (Li et al. 2020) conducted a comprehensive review of applications in Federated Learning, highlighting the evolution of FL in addressing challenges such as data silos and sensitivity. They explored various optimization paths and identified significant applications of FL in industrial engineering and computer science. Their work provides a foundational understanding of FL's role in diverse fields, including its potential in recommendation systems.

Javeed et al. (Javeed et al. 2023) discussed the integration of FL in Personalized Recommendation Systems (PRS), particularly focusing on security and privacy challenges in next-generation Consumer Electronics (CE). They emphasized how FL enhances data privacy and security in PRS by sharing local recommender parameters while keeping the training data localized.

Z. Jie et al. (Jie et al. 2023) proposed a federated recommendation system based on historical parameter clustering to address the challenges of non-independent and identically distributed data in FL. Their approach involves a weighted average of historical and global parameters, enhancing the recommendation system's accuracy.

Muhammad et al. (Muhammad et al. 2020) introduced FedFast, an innovative technique for accelerating the training of federated recommender systems. By employing a novel sampling technique and active aggregation strategy, FedFast improves the convergence speed and accuracy of federated recommendation models, demonstrating its effectiveness across various benchmark datasets.

Studies have shown FL's effectiveness in collaborative filtering, matrix factorization, and personalized recommendation systems while addressing data privacy. Handling non-IID data, communication overhead, and ensuring model convergence are ongoing challenges.

Methodology

This paper introduces two recommendation systems that leverage FL. The proposed models are Transformers and Feed-Forward Networks, to enhance the efficiency and effectiveness of the recommendation process. The Movie Recommendation system uses two types of model architectures, the BST and Feed-Forward models, as its global and local models. In order to train our global model, FL is used to aggregate small client models that are trained on non-independent and non-identically distributed (non-iid) data. The weights of these models are then transferred to the server for averaging the weights. For product recommendation, we use the BERT and Feed-Forward Neural Network models, leveraging their strength in text classification.

The Federated Averaging Adam Algorithm in both of our applications, which has been proven to be robust in various studies (Nilsson et al. 2018), (Karimireddy et al. 2021). This ensures optimal performance and precision for the recommendation systems. To develop a Movie Recommendation System using Federated Learning (FL), the methodology illustrated in Figure 1 is adopted. This methodology involves three primary entities: Users, Ratings, and Movies. The dataset is organized by partitioning the movie IDs into sequences. To control the number of sequences generated for each user, sequence length and step size parameters are utilized.

The dataset consists of three key tables: User, Ratings, and Movies. The Users table encompasses user-id, gender, age group, and occupation, while the Movies table includes movie-id, title, and genre. The Ratings table contains user-id, movie-id, ratings, and timestamps. To prepare the data for modeling, sequences of movie IDs and their corresponding ratings are generated. The output is then reorganized to ensure that each sequence is represented as a separate record in the DataFrame, linking user characteristics with the rating data to capture the nuances of user preferences. Non-iid datasets are created for each client, enabling local training. The Federated Averaging Algorithm is employed to aggregate insights from these client models into the global model, optimizing the Movie Recommendation System for diverse user preferences and characteristics.

Dataset Preparation

Movielens 1m dataset (Harper and Konstan 2023) is used for Movie Recommendation system and Amazon review dataset

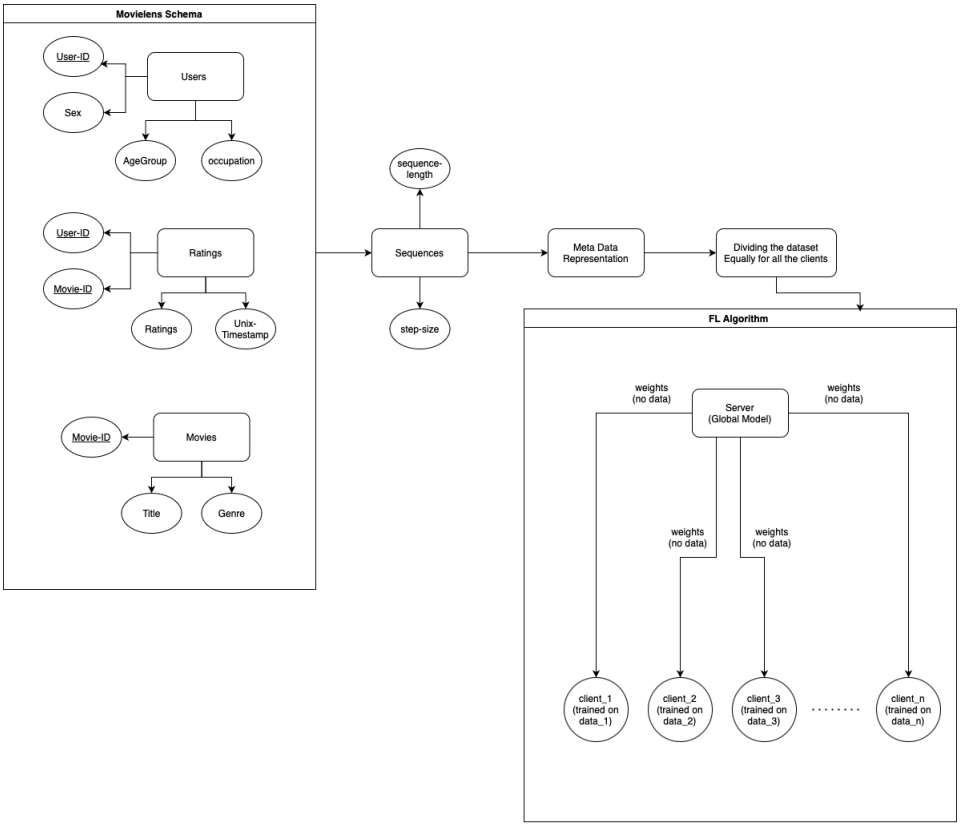

Figure 1: Proposed Methodology of Movie Recommendation Model by using movielens dataset

user_id	sequence_movie_ids	sequence_ratings	sex	age_group	occupation
user_1	movie_3186,movie_1721,movie_1270,movie_1022	4.0,4.0,5.0,5.0	F	group_1	occupation_10
user_1	movie_1270,movie_1022,movie_2340,movie_1836	5.0,5.0,3.0,5.0	F	group_1	occupation_10
user_1	movie_2340,movie_1836,movie_3408,movie_1207	3.0,5.0,4.0,4.0	F	group_1	occupation_10
user_1	movie_3408,movie_1207,movie_2804,movie_260	4.0,4.0,5.0,4.0	F	group_1	occupation_10
user_1	movie_2804,movie_260,movie_720,movie_1193	5.0,4.0,3.0,5.0	F	group_1	occupation_10
...	...	...	...	...	...

Table 1: Sample table from the Movielens dataset, which has undergone sequencing and metadata creation.

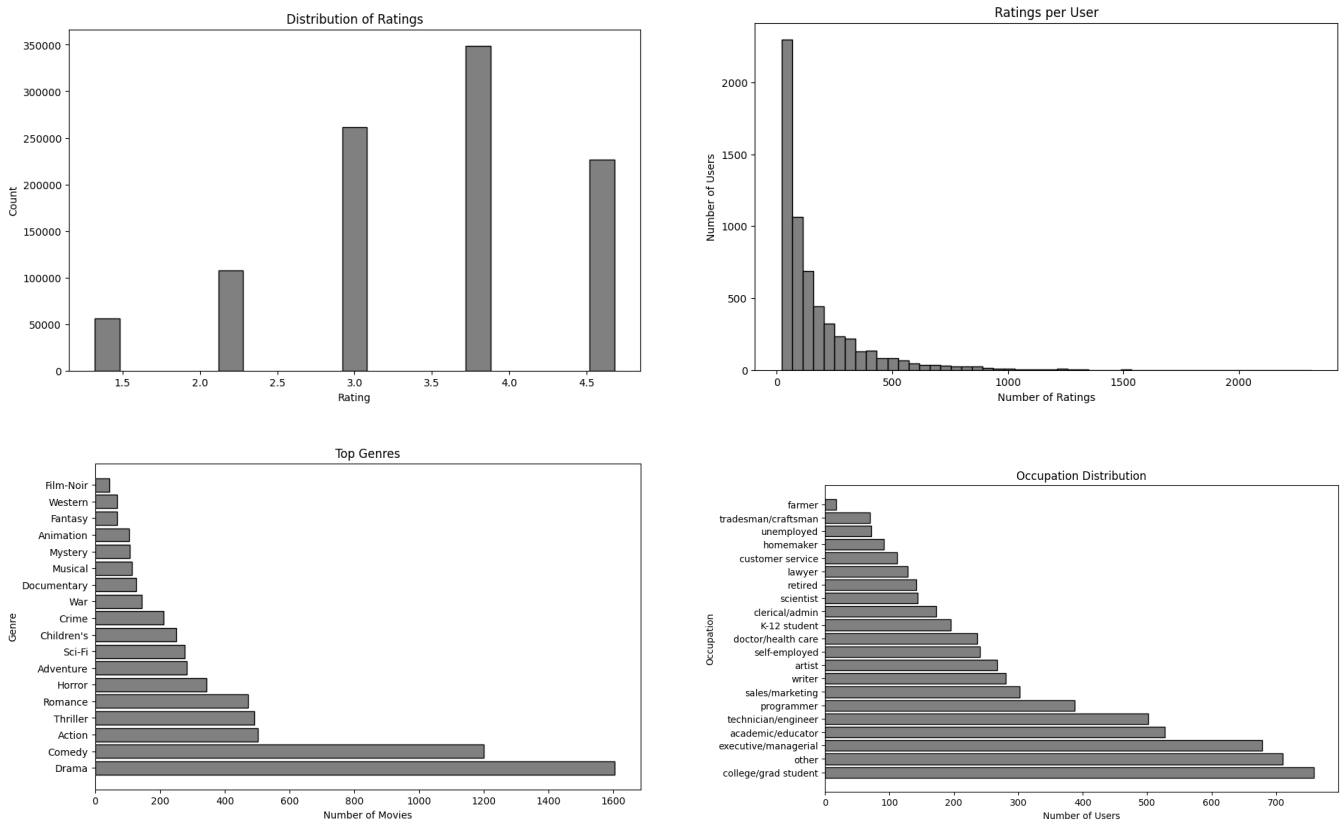

Figure 2: Data analysis plots for movielens dataset (ratings, users and movies)

Table 2: Summary of the Movielens Dataset

Name	Statistics
Number of users	6040
Number of movies	4052
Number of reviews	1000209
Click/browse	6976551
Review	354016
Favorites	72604
Like	244740
Total user actions	7647911
Dataset sparsity	0.958

(McAuley and Leskovec 2013) for the Product Recommendation. The moveielens dataset consists of 3 other tables Users, ratings, Movies which are joined using the userID resulting in a table of total 6040 unique users and highest number of ratings given by the user-4169 with 1157 ratings out of 1 million. Table 2 presents a summary of the dataset, offering an overview of various movie-related attributes. The dataset spans movies from 1939 onwards, encompassing a diverse range of 18 genres, with the distribution of these genres across different years visually represented in Figure 3. Table 1 provides a comprehensive overview of the data generation following the sequencing process. For each user, a sequence is crafted, encompassing movie IDs and their corresponding ratings. Age is categorized into seven distinct groups, while gender is denoted as Male (M) or Female (F). Furthermore, occupation is classified into 21 diverse categories resulting in the length of 498623 rows and 6 columns.

Following the data creation process, each row in the dataset represents a user's interactions, encapsulating the movies they have viewed and their corresponding ratings. Notably, the final movie and its associated rating are isolated from the remaining sequence, serving as the prediction target for the model.

Experiments and Results

Federated Learning

In the realm of Federated Learning (FL), the Adaptive Federated Optimization (FedAvgOpt) algorithm stands out as a prevalent approach for training ML models across a multitude of decentralized devices. This algorithm involves a series of iterative communication rounds between a central server and numerous local devices, commonly referred to as clients. At its core, FedAvgOpt aims to address an optimization problem by harnessing the Optimizers to Clients technique. The foundation of this framework comes from the work by Reddi et al (Reddi et al. 2020). In their study, they

introduced the integration of Yogi, AdaGrad, and Adam optimizers into the federated learning process to train a global model. The results they obtained were exceptionally promising, demonstrating the potential of this approach. Building upon this foundational research, our work delves deeper into the framework of federated optimization utilizing server and client optimizers with an array of algorithms, encompassing BERT and Feed-Forward Neural Networks. By leveraging this framework, we meticulously design novel adaptive federated optimization applications for movie recommendation systems and product recommendation. The intricate parameters employed in the construction of these models are meticulously outlined in Table 3. To our knowledge, our proposed methods represent the first applications of adaptive server optimization in the context of FL. Through extensive experimentation, we have comprehensively evaluated the performance of these methods across a diverse set of benchmark datasets.

Table 3: Federated Adam Averaging parameters

Parameters
Number of Clients
Number of Rounds
Batch Size
Epochs per Round
Learning Rate

In the context of movie recommendation systems, a deep learning architecture is employed, consisting of two dense linear layers following the user, ratings, and movie embedding layers. This architecture incorporates additional features such as occupation, age, and gender to enhance prediction accuracy. The user embedding layer captures the user's preferences and characteristics, while the movie embedding layer encodes the attributes of each movie. The ratings layer represents the user's ratings for previously watched movies. The subsequent dense linear layers leverage these embedded representations to learn complex relationships between users, movies, and their associated features, enabling the model to make personalized movie recommendations tailored to each user's preferences and demographics.

To comprehensively evaluate the performance of the proposed methodology, which leverages the BST model for movie recommendations and BERT for text classification, specific evaluation metrics have been devised for both prediction and classification tasks. The Adam optimizer (Adaptive Moment Estimation) is employed to enhance the effectiveness of the models. This optimizer combines the advantages of RMSProp (Root Mean Square Propagation) and AdaGrad (Adaptive Gradient Algorithm), making it a popular choice for training deep neural networks. The Adam optimizer has demonstrated effectiveness in a wide range of deep learning tasks and is often the optimizer of choice, as shown in equation (1). To address the FL optimization problem, it is essential to minimize the gap between the actual (target) values and the predicted values. To determine the performance of regression models, we commonly use the Mean Squared Error (MSE) metric, which can be expressed mathematically as equation (2). For BST, we make use of the loss function, while for BERT, we use the Binary Cross Entropy Loss (BCELoss). To assess the performance of our model, we employ two evaluation metrics: Accuracy and Mean Absolute Error (MAE). Accuracy measures the proportion of correct predictions, while MAE quantifies the average magnitude of errors in predictions. The mathematical formulations of Accuracy and MAE are presented in equations (3) and (4), respectively. By utilizing these evaluation metrics, we can comprehensively evaluate the performance of our model in both prediction and classification tasks. This allows us to gain insights into the effectiveness of our methodology and identify areas for potential improvement.

$$w_{t+1} = w_t - \alpha \cdot \frac{m_t}{\sqrt{v_t} + \epsilon} - \alpha \cdot \text{weight\_decay} \cdot w_t \quad (1)$$

Where w_{t+1} represents the updated parameter value, w_t is the current parameter value, α is the learning rate, set to 1×10^{-5}, m_t is the moving average of the gradient, $\sqrt{v_t}$ is the square root of the moving average of the squared gradient, ϵ is a small constant used to prevent division by zero, weight_decay is the weight decay coefficient, introducing L2 regularization.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (2)$$

where, n is the number of data points, y_i represents the true target value for the i-th data point and y_hat represents the predicted value for the i-th data point.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (3)$$

The MAE measures the average absolute difference between the predicted values $\hat{y}_i$ and the actual values y_i across all data points in your dataset.

The training pipeline starts by loading the dataset and dividing it into 75% for the federated training process and 25% for testing the global model after weight aggregation. A single round of weights aggregation in both experiments, ensuring an even distribution of the dataset among the specified number of clients, as defined by the hyper-parameter. This choice aims to prevent potential over-fitting, which can occur when using the same dataset in multiple consecutive rounds of training. Such an approach helps maintain a balanced performance and generalization of the model.

During the training of each client model, we evenly split the dataset to create decentralized data (non-iid), ensuring no sharing of data between the global model on the server and the client model on the user's device. The client model is constructed with the same architecture as the global model to facilitate proper weight aggregation and avoid potential system intrusion detection issues. Fed-Average-Opt (Reddi et al. 2020) introduces a new research gap where 2 optimizers are used in client and server side which produces different outcomes. To avoid this problem, this paper utilizes the

Table 4: Regression Prediction Metrics for for Federated Feed-Forward Neural Network using Movielens-1m dataset

Clients s.no	MAE	MAPE	R2Score	MSE	RMSE
1	0.861271	34.193677	0.121443	1.110641	1.053869
2	0.844557	33.979844	0.128833	1.083926	1.041118
3	0.838203	34.173737	0.130822	1.078016	1.038275
4	0.843004	34.873277	0.123515	1.094560	1.046212
5	0.837860	35.227289	0.118945	1.099636	1.048635
6	0.856534	33.510565	0.120854	1.095998	1.046899
7	0.850350	33.967662	0.122866	1.093216	1.045570
8	0.844025	34.068765	0.130675	1.085979	1.042103
9	0.858443	33.897860	0.112239	1.105887	1.051612
10	0.845914	34.077291	0.127015	1.088638	1.043378

Table 5: Federated Adam Averaging hyper-parameters

Parameters	Values
Number of Clients	10
Number of Rounds	1
Batch Size	256
Epochs per Round (local models)	10
Learning Rate	0.01

same Adam as the optimizer during the local model or client training. Once the local model architecture is set up, the decentralized data is further divided into an 80:20 ratio. The 80% portion is utilized to train the local model, while the remaining 20% is reserved to assess each client model's accuracy in prediction or classification. After the training process, we save the model weights, which are then aggregated with weights from other client models and transferred to the global model. Table 13 provides an overview of the fixed hyper-parameters utilized in this paper.

Discussion and Limitations

Tables 11 and 12 display the regression metrics for each client's evaluation. From these tables, we can see that the error rates differ among clients. This indicates that, in addition to the known issue of system heterogeneity in FL, the specific model architecture also plays a role in performance variation. The varying error rates suggest that certain clients perform better than others. This difference is not only due to the expected variations in federated systems but is also influenced by the choices made in designing the transformer model. While system heterogeneity is a recognized challenge in FL, the impact of the model's design on individual client performance adds an extra layer of complexity. It implies that the effectiveness of the chosen transformer architecture differs across clients, affecting the overall performance of the FL system. These findings in this paper highlight the need to consider both system differences and model design when working with FL. Striking the right balance between adapting to inherent system variations and tailoring the model to diverse client characteristics is crucial for achieving optimal performance.

Conclusion and Future work

This work opens up new avenues for research in Federated Learning (FL), particularly in the realm of adaptive optimization. The adaptive federated optimization framework developed here is expected to be a valuable tool for the FL community, facilitating the creation of more efficient and effective FL algorithms. Our research has commenced with the deployment of transformers in limited-scale applications. For Federated Learning to thrive in real-time applications, we must confront the challenges of Data Heterogeneity and Communication Overhead. Data Heterogeneity necessitates robust methods for harmonizing diverse data sources, while Communication Overhead requires streamlined protocols and reduced information exchange. In the future, our research will concentrate on the practical deployment of Federated models within real-time systems. This initiative will provide a comprehensive and clear perspective on the security aspects of Machine Learning systems as well as increased personalization.

References

Ammad-Ud-Din, M.; et al. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888*.

Dong, Z.; et al. 2022. A brief history of recommender systems. *arXiv preprint arXiv:2209.01860*.

Harper, F. M.; and Konstan, J. A. 2023. MovieLens 1M Dataset. [Online]. Available: https://grouplens.org/datasets/movielens/1m/.

Javeed, D.; et al. 2023. Quantum-Empowered Federated Learning and 6G Wireless Networks for IoT Security: Concept, Challenges and Future Directions. *Quantum*, 96: 1.

Jie, Z.; et al. 2023. Personalized federated recommendation system with historical parameter clustering. *Journal of Ambient Intelligence and Humanized Computing*, 14(8): 10555–10565.

Karimireddy, S. P.; et al. 2021. Breaking the centralized barrier for cross-device federated learning. *Advances in Neural Information Processing Systems*, 34: 28663–28676.

Kim, M. C.; and Chen, C. 2015. A scientometric review of emerging trends and new developments in recommendation systems. *Scientometrics*, 104: 239–263.

Li, L.; et al. 2020. A review of applications in federated learning. *Computers & Industrial Engineering*, 149: 106854.

Li, Y.; et al. 2023. Recent Developments in Recommender Systems: A Survey. *arXiv preprint arXiv:2306.12680*.

Mazeh, I.; and Shmueli, E. 2020. A personal data store approach for recommender systems: enhancing privacy without sacrificing accuracy. *Expert Systems with Applications*, 139: 112858.

McAuley, J.; and Leskovec, J. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*.

McMahan, B.; et al. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR.

Muhammad, K.; et al. 2020. Fedfast: Going beyond average for faster training of federated recommender systems.

Nilsson, A.; et al. 2018. A performance evaluation of federated learning algorithms. In *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*.

Occhioni, D.; et al. 2023. Eyeing the Visitor's Gaze for Artwork Recommendation. In *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*.

Reddi, S.; et al. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.