PDF Download
3651168.pdf
28 December 2025
Total Citations: 8
Total Downloads: 721

Latest updates: https://dl.acm.org/doi/10.1145/3651168

Published: 29 July 2025
Online AM: 04 March 2024
Accepted: 28 February 2024
Revised: 17 January 2024
Received: 05 June 2023

Citation in BibTeX format

RESEARCH-ARTICLE

Communication-Efficient Federated Neural Collaborative Filtering with Multi-Armed Bandits

WAQAR ALI, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

MUHAMMAD AMMAD-UD-DIN

XIANGMIN ZHOU, RMIT University, Melbourne, VIC, Australia

YAN ZHANG, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

JIE SHAO, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

Open Access Support provided by:

University of Electronic Science and Technology of China

RMIT University

ACM Transactions on Recommender Systems, Volume 4, Issue 1 (March 2026)
https://doi.org/10.1145/3651168
EISSN: 2770-6699

Communication-efficient Federated Neural Collaborative Filtering with Multi-armed Bandits

WAQAR ALI, Sichuan Artificial Intelligence Research Institute, Yibin, China and Yibin Park, University of Electronic Science and Technology of China, Yibin, China
MUHAMMAD AMMAD-UD-DIN, Comparables.ai Oy, Helsinki, Finland
XIANGMIN ZHOU, School of Computing Technologies, RMIT University, Melbourne, Australia
YAN ZHANG, University of Electronic Science and Technology of China, Chengdu, China
JIE SHAO, University of Electronic Science and Technology of China, Chengdu, China and Sichuan Artificial Intelligence Research Institute, Yibin, China

Federated learning (FL) has received much attention in privacy-preserving and responsible recommender systems. Recent studies have shown promising results while federating widely used recommendation methods such as collaborative filtering. A major barrier when bringing FL into production is that the model complexity or the volume of gradients to be transmitted over the communication channel grows linearly as the number of items in a particular system increases. To address this challenge, we propose a communication-efficient neural collaborative filtering method for federated recommender systems. First, to align our solution with other deep neural architectures, we construct standard neural collaborative filtering in federated settings. Second, to solve the underlying model complexity challenge, a multi-armed bandit framework is used that intelligently selects a smaller set of payloads for each iteration of federated model training. The item selection is based on a carefully designed reward function that determines which portion of the overall payloads would be optimal for a particular user. The FL model only comprising of the selected items is transmitted over the network. The FL users train their local models in the regular federated learning way utilizing the payload-efficient global model, requiring no additional optimizations. The results show that using only 10% of the model's payload, our method can achieve recommendation performance comparable with the standard federated neural collaborative filtering.

CCS Concepts: • **Information systems** → **Collaborative filtering**; • **Computing methodologies** → *Reinforcement learning;*

Additional Key Words and Phrases: Neural collaborative filtering, communication-efficient federated learning, multi-armed bandits, payload optimization, recommender systems

This work is supported by the National Natural Science Foundation of China (No. 62276047), Australian Research Council Discovery Project (No. DP200101175), and Sichuan Science and Technology Program (No. 2022YFWZ0006).

Authors' addresses: W. Ali, Sichuan Artificial Intelligence Research Institute, Yibin, China, 644000 and Yibin Park, University of Electronic Science and Technology of China, Yibin, China, 644000; e-mail: waqar@uestc.edu.cn; M. Ammad-ud-din, Comparables.ai Oy, Helsinki, Finland, 00100; e-mail: mu.ammad.ud.din@comparables.ai; X. Zhou, School of Computing Technologies, RMIT University, Melbourne, Australia, 3000; e-mail: xiangmin.zhou@rmit.edu.au; Y. Zhang, University of Electronic Science and Technology of China, Chengdu, China, 611731; e-mail: yixianqianzy@gmail.com; J. Shao (Corresponding author), University of Electronic Science and Technology of China, Chengdu, China, 611731 and Sichuan Artificial Intelligence Research Institute, Yibin, Sichuan, China, 644000; e-mail: shaojie@uestc.edu.cn.

ACM 2770-6699/2025/07-ART2
https://doi.org/10.1145/3651168

ACM Reference Format:
Waqar Ali, Muhammad Ammad-ud-din, Xiangmin Zhou, Yan Zhang, and Jie Shao. 2025. Communication-efficient Federated Neural Collaborative Filtering with Multi-armed Bandits. *ACM Trans. Recomm. Syst.* 4, 1, Article 2 (July 2025), 28 pages. https://doi.org/10.1145/3651168

1 INTRODUCTION

Recommender system (RS) serves as a valuable and efficient solution to assist individuals in navigating the progressively intricate information landscape of both business and everyday life. As one of the principal information retrieval tools, these systems aim to find items for users, fulfilling the commercial or personal demands for information search and exploration. Recommendation technologies are becoming the lifeblood of business models in the current digital era. Practically, these systems learn the tastes and preferences of users from prior interaction data and forecast their future interests. Creating a balance between personalization gains and privacy risks is a key challenge for designing trustworthy RS. Additionally, with the introduction of the **general data protection regulation (GDPR)** [13] in 2018[1] and China's recent regulations for the use of recommendation algorithms,[2] both the service providers and users have become more concerned about privacy. For data operators and companies working with machine intelligence, it becomes necessary to ensure the protection of personal data collected for recommendation purposes. This concern brought considerable interest from the research community that intensified the efforts to find privacy-preserving solutions.

Collaborative learning [30] is a widely adopted and efficient strategy used in practical recommendation engines. However, a major drawback in collaborative filtering models is sharing personal information with third-party servers for collaborative training. This raises significant technical challenges for service providers. To address this issue, numerous privacy protection techniques have been proposed to hide users' preferences while generating recommendations [11, 34]. These frameworks are designed to retain the data utility for recommendation services while protecting confidential information before sharing them with outside servers. Privacy protection in recommender systems has gained an extensive research attention in recent years [2, 27, 36, 48, 56]. Existing techniques can be broadly classified into three categories: data anonymization [11, 34], cryptographic techniques [6], and differential privacy [42, 55]. Most of these frameworks have their own technical limitations, such as computational overhead, difficulties in anonymizing personal identities, and generating top-k recommendations.

Federated learning (FL), a privacy-by-design machine learning paradigm, has gained significant attention due to its ability to derive valuable knowledge from distributed data sources without the need for physically transferring the data to a central server. FL in recommender systems works in a collaborative training fashion, where multiple clients train their local models and share their model weights with a central server for aggregation into a global model. A key advantage of FL is its capacity to learn a centralized model by leveraging decentralized data sources. A number of traditional recommender algorithms such as **collaborative filtering (CF)** and matrix factorization have been adapted in federated settings [1, 3, 9, 56]. However, state-of-the-art neural models such as **neural collaborative filtering (NCF)** [14] are still facing considerable challenges in federated settings. The inherent design features of FL make it well suited for such practical applications. Specifically, FL enables the collaborative learning process is secure and resilient to data tempering. In FL, each collaborator keeps its personal data for local training and only shares the gradients of

[1] https://gdpr-info.eu/
[2] http://www.cac.gov.cn/2022-01/04/c_1642894606364259.htm

its local model with the federated server for global model learning. Generally, during the FL process of weight transmission (local model updates from collaborators to federated server, or global submission from federated server to participants), the transmission cost termed as “payload" is inescapable. It becomes a critical bottleneck for large-scale recommender systems. Indeed, the computational and communication capabilities of terminal devices in federated networks vary in terms of hardware, network connectivity, and power. Terminal devices and networks were not designed to bear the payloads for billions of items. Therefore, it is important to develop a communication-efficient federated model for RS that iteratively sends small-size model updates as part of the training process. Although existing federated collaborative filtering [30, 34] and many similar federated recommendation models [4, 56] have achieved great success as privacy-preserving recommendation models, a key practical concern is still neglected that existing network bandwidth and terminal devices are unable to handle large-scale gradients for practical recommendation engines.

Typically, there are bandwidth restrictions between edge nodes and federated server. The frequency with which edge nodes forward and receive model updates requires a huge amount of bandwidth. For instance, the AlexNet model has parameters that are around 60M in size. The frequent transmission of local and global model updates causes the communication channel overcrowded. This is one of the key barriers while working in federated settings. Larger systems such as YouTube, Amazon, or Facebook need more space, memory, and bandwidth to avoid model outdates and mitigate concept drift. In addition to these major challenges, balancing model payload is a major practical concern while bringing federated learning in production. As the number of elements grows, the model payload increases linearly. For example, assuming a CF model with 25 latent factors and precision as 64 bits, the payload can be estimated as $\# \, items \times 25 \times 64/8$. In this context, for 100K items the payload volume would be around 19 MB, and for 1 M and 10 M items the payload volume could reach up to 190 MB and 1.9 GB, respectively, for each communication round. It becomes more challenging when the number of items or users increases. For large-scale **federated recommender systems (FRS)**, the payload transfer between participating clients and the FL server involves a substantial communication cost. The increasing payloads are a critical challenge not only for service operators but also for end-users where terminal devices were not designed to bear such a large computation and communication cost. The end-users do not concern about the server-side item counts and federated recommender algorithm.

In this work, we first construct standard NCF model in federated settings called **federated neural collaborative filtering (FNCF)**. Then, we present a novel payload optimization technique for federated recommender systems entitled as **federated neural collaborative filtering with multi-armed bandits (FNCF-MAB)**. Our motivation of selecting multi-armed bandits for payload optimization is based on the recent success of bandits based model for multi-constraints optimization [8, 12, 18] and the generalized exploration and exploitation ability of bandit models. Additionally, considering the theoretical significance of classical Bayesian probability for bandit models [12, 32], we employ Bayesian Thompson sampling to balance the exploration-exploitation and boost the optimization ability of bandit model. One advantage of this sampling strategy is that it does not require a prior or predefined set of dependent parameters that can affect the tradeoff of exploration and exploitation. Adopting NCF for recommendation generation is to justify that our proposal can be utilized to other deep learning based recommendation frameworks. Figure 1 illustrates how our optimization method uses an intelligent strategy to broadcast only a small amount of gradients rather than the complete payload during each federated iteration. A multi-armed bandit model that includes a novel reward and regret strategy created especially for federated neural collaborative filtering facilitates the selection process. By transmitting only the relevant gradients, this strategy assures efficient and effective communication, improving performance and lowering channel overhead. For this purpose, we construct a posterior distribution that helps the model to

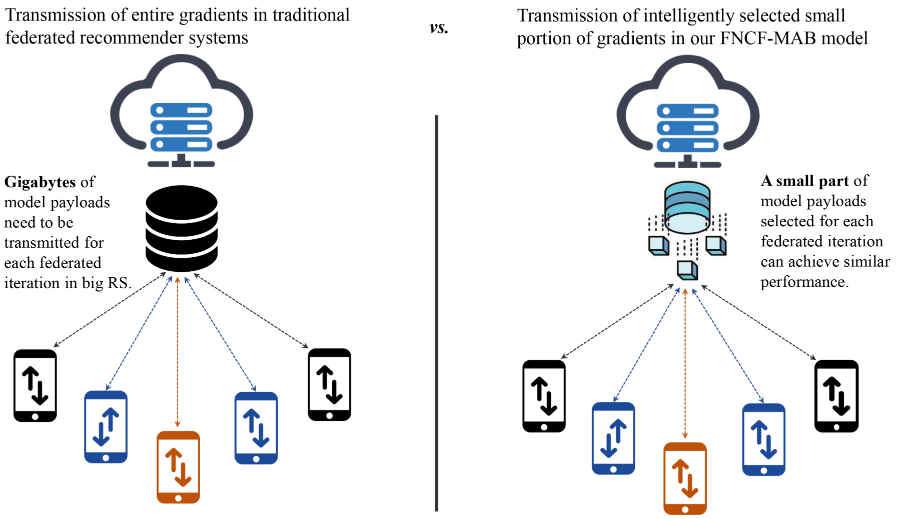

Transmission of entire gradients in traditional federated recommender systems
vs.
Transmission of intelligently selected small portion of gradients in our FNCF-MAB model
Gigabytes of model payloads need to be transmitted for each federated iteration in big RS.
A small part of model payloads selected for each federated iteration can achieve similar performance.

Fig. 1. An overview of the proposed FNCF-MAB model that intelligently selects a small portion of gradients, while existing FL-based recommender systems transmit entire gradients that impose a huge cost on communication channels.

sample the next set of items. Additionally, NCF in the federated settings allows the clients to train lightweight local models and instead of personalized data only model gradients need to transfer to the federated server to form the global model for the next federated iteration. The users perform standard model updates as part of the FL process, thus avoiding additional optimization steps. Extensive experiments on four benchmark datasets prove the significance of our proposal for real-world applications. In summary, our contributions can be summarized as follows:

- *Federated implementation of NCF*: We implement standard NCF model in federated settings with optimal payload requirements for each federated iteration. This is highly beneficial for practical recommenders operating in resource-constrained environments. To the best of our knowledge, we are among the pioneers to explore the adaptation of NCF for the federated recommender system and demonstrate the feasibility of transporting minimum gradients at each federated iteration.
- *Communication efficient FNCF*: We introduce a novel multi-armed bandit solution for reducing communication overhead in federated neural collaborative filtering. The proposed FNCF-MAB model is the first of its kind in the literature that mitigates a major practical challenge for bringing FL into large-scale practical recommenders.
- *Lightweight local models*: We ensure intelligent selection and transmission of only relevant model gradients instead of the entire payload. Furthermore, the adoption of NCF in federated settings allows clients to train lightweight local models, eliminating the need for transmitting personalized data. This streamlined approach avoids additional optimization steps and minimizes computational burdens on client devices.
- *Proof of concept*: We perform extensive experiments on four large-scale rating datasets to validate the impact of our proposal. The proposed FNCF model is consistent in federated settings, and the recommendation performance is comparable with the original NCF model. Meanwhile, our analysis found a worthy outcome, the convergence behavior of the proposed FNCF-MAB model on 10% communication cost is highly comparable with the model working with full payload.

The remaining parts of the article are structured as follows: An overview of the recent research on privacy protection and federated learning based recommender systems is presented in Section 2. Section 3 explains the proposed FNCF-MAB model, relevant concepts, and formal definitions. The experimental setup, major findings, and a discussion of the results are presented in Section 4. Finally, Section 5 wraps up our findings and discusses possible future directions.

2 RELATED WORK

Since the early 2010s, recommender systems have shown great progress in digital trade and become an integral part of our daily lives. Responsible and privacy-preserved recommendation services are essential aspects and are directly linked with the commercial success of a recommender system [44, 56]. Existing privacy-preserving CF techniques are mostly based on cryptographic, random perturbations, or differential privacy frameworks. The cryptographic models are usually based on homomorphic encryption [6, 20] which is effective but expensive in terms of computational cost. On the other hand, techniques based on random perturbations [41] are cheap, but the perturbations may cause quality issues for recommendation engines.

2.1 Responsible Recommendation Services

The term “responsible recommenders" has broader meanings including but not limited to the practice of designing, developing, and deploying recommendation solutions with a good intention to empower businesses, fairly impact customers and society, and allow companies to engender trust and scale AI with confidence. With the newly proposed constitutional requirements, the need is indispensable, organizations’s concerns about bringing fairness, trust, privacy, and security in personalization services raising a lot of unsolved questions for the research community. For example, the European Union has taken serious actions related to the automation and the responsible use of recommendation algorithms [39]. The Digital Services Act from December 2020 and the Artificial Intelligence Act from April 2021 can be considered in this regard. There are considerable examples in academic literature [36, 45, 56] that address the need of responsible recommendation services. Companies across the world are defusing intelligence in the form of recommendation algorithms to boost their sales, increase client satisfaction, strengthen goodwill, and manage risk factors. With GDPR directives and recently imposed China’s recommender system regulations 2022, the commercial use of recommendation algorithms is not easy as in the past. The companies that employ recommender system technologies and similar content decision algorithms in their Apps and websites have to follow a strict set of protocols.

2.2 Federated Learning

Federated learning, as introduced by Google [22, 23], presents a promising approach for privacy protection in the context of machine learning. The goal is to collaboratively train a high-quality global model without compromising the privacy of client-side data. Yang et al. [50] presented a comprehensive analysis of FL algorithms, including their fundamental principles, applications across diverse domains, and implementation challenges. Additionally, researchers have emphasized the use of federated learning for privacy protection in mobile edge computing [8, 15, 31, 36]. Zheng et al. [55] highlighted the unique advantages of federated learning over differential privacy for training general-purpose machine learning models. Recognizing the practical significance of FL, industry expectations for the communication-efficient FL models (e.g., References [1, 25, 40, 43, 47, 57]) are increasing. Transmitting and exchanging large volumes of gradients, especially for deep learning models, can impose significant challenges in terms of bandwidth, latency, and energy consumption. Therefore, communication-efficient FL models are crucial for ensuring the practicality and scalability of FL in real-world applications.

2.3 Federated Recommender Systems

Federated learning has emerged as a promising approach for developing secure and privacy-preserving recommender systems by enabling collaborative training of a global model across distributed clients without compromising user privacy. Several studies have explored FL-based solutions for recommender systems [2, 26, 38]. Early works in FRS mainly focused on implementing existing recommendation models in federated settings, such as basic formulation of standard CF model [3], implementation of NCF in a federated environment [38], and meta matrix factorization for federated rating prediction [28]. While these initial studies paved the way for FRS, they did not address the practical challenges of high-dimensional embeddings, computational costs, and bandwidth constraints that arise in large-scale deployments.

Subsequent research raised concerns on various aspects such as communication efficiency, reliability, security, and training speed. For example, Zhang et al. [54] proposed a framework to mitigate data poisoning attacks, safeguarding the integrity of the federated training process in FRS. Recently, Perifanis et al. [37] introduced a privacy-preserving federated point of interest recommendation framework that incorporates social influence factors to improve recommendation accuracy while keeping privacy values. Likewise, Huang et al. [16] proposed a federated deep reinforcement learning framework for daily schedule recommendation, enabling personalized recommendations for daily activities. Imran et al. [17] proposed a resource-efficient federated recommender system that adapts to dynamic and diversified user preferences, ensuring the system's adaptability to evolving user needs. Similarly, Zhang et al. [53] introduced a lightweight federated recommender with less computational overhead on factorization process.

Despite extensive research efforts, high communication costs of recommendation models in FL settings remain a common deficiency in existing techniques on FRS. To address this open challenge, our work focuses on reducing the communication costs for large-scale, real-world FRS deployments. We aim to significantly improve communication efficiency and reduce network overhead while maintaining recommendation performance. By leveraging multi-armed bandits as a novel optimization method, we select only the relevant gradients for transmission. This allows minimizing bandwidth usage for large-scale FRS deployments.

2.4 Practical Issues

Until now, the FL techniques in RS do not sufficiently consider the computing capability of the participants involved in the learning process. At the abstract level, it seems good to train a global model in the federated environment, but practically sometimes it becomes impossible for the edge devices and network infrastructure to bear the cost of rapid updates for gigabytes of model payloads. Recent academic literature [1, 25, 40, 57] has witnessed that the concerns to address the massive communication overhead problem during FL training is getting research attention in many domains. Techniques in mainstream FL research, addressing communication-efficient FL training can be classified into two general categories: reducing federated iterations and data compression. For reducing federated iterations, consider a standard federated training process that normally takes hundreds of thousands of iterations to converge. The simple solution advised in many works [7, 57] is to allow clients do multiple iterations locally before making a global model update. Although the experiments reflect that FedAvg can massively increase the convergence speed with respect to wall-clock time due to the reduction of communication rounds, it has no theoretical justification to perform well for the FRS case. Therefore, rapid local model updates might be unnecessary for FRS. Contrariwise, another approach common in academic literature [5, 46] is to reduce the size of data transfer or simply data compression. These techniques (either quantization or sparsification) aim to represent the original data by a low-precision alternative with a smaller size. Adopting these frameworks on image data can considerably reduce the size of gradients needed to transmit in a federated environment. However, it is impractical to bring compression models for FRS due to the nature of learning and adaptive needs. While sparsification inevitably introduces extra overhead during the federated training process, including sampling, encoding, decoding, compressing, and decompressing, it is essential to consider its potential impact on overall training efficiency.

2.5 Design Goals

With consideration for practical concerns and existing deficiencies, as outlined in Sections 2.3 and 2.4, the main objective of this research is to address the practical barriers hindering the implementation of FRS in production. Specifically, we aim to achieve the following goals:

- *Privacy protection.* Privacy protection is one of the primary objectives for the context in which we build this research. The recommendation model should respect the individuals' preference behavior and protect the information collected for collaborative learning. The framework we designed ensures privacy (NCF in federated settings, as justified in Section 4.5) with optimal operational cost under resource constraints environment.
- *Communication efficient FRS.* In the classic federated recommender settings, a global model copies the entire set of gradients to all the clients for local model updates. Practically, client-side devices are not computationally efficient to manage aggregation process for millions of items. Also, the in-between communication network and allocated bandwidth are not designed to meet rapid model updates. Therefore, we aim to design a communication-efficient federated recommender (as justified in Section 4.8) that could be employed in production.
- *Generalized federated recommender.* We aim to construct a generalized federated recommender solution with the ability to adapt to any kind of item recommendation. Therefore, we choose NCF as a base recommender and implement in federated settings (formally illustrated in Section 4.4). The model can react appropriately to previously unseen, fresh data chosen from the same distribution as the models' initial input. Simply, we aim to bring a model for federated recommender systems that has the ability to preserve privacy, and is practically acceptable for large-scale recommenders, and generalized.

3 PAYLOAD OPTIMIZATION METHOD

In this section, we describe our proposed FNCF-MAB model in detail. First, we state some preliminary concepts related to collaborative filtering, neural collaborative filtering, and federated neural collaborative filtering. Then, we formulate the payload optimization problem with multi-armed bandits and **Bayesian Thompson sampling (BTS)**. A list of common notations used in the article is given in Table 1.

3.1 Problem Formulation

Generally, given a set of N users $U = \{1, 2, 3, \ldots, N\}$ and a set of M items $I = \{1, 2, 3, \ldots, M\}$, the users' feedback for items can be represented by an $N \times M$ preference matrix R where r_{ui} is the recorded preference given by user u for item i. For standard matrix factorization model [24] any rating preference can be estimated as

$$r_{ui} \sim p_u^\top q_i. \tag{1}$$

A classical CF model obtains r_{ui} through a linear combination of latent user-factors (also termed as embedding vectors) p_u for $u = 1, 2, 3, \ldots, N$ users and item-factors q_i for $i = 1, 2, 3, \ldots, M$ items collected in factor matrices P and Q, respectively. The NCF extends the **generalized matrix factorization (GMF)** approach by integrating an additional multi-layer feed-forward neural network to model the non-linear relationships between user–item preferences while learning the latent factor P and Q. As a result, the NCF model infers a preference matrix by concatenating P

Table 1. A List of Common Notations Used

Notations	Description
U	A set of users of size N: $U = \{1, 2, 3, \ldots, N\}$
I	A set of items of size M: $I = \{1, 2, 3, \ldots, M\}$
R	Rating (or preference) matrix in the form of $N \times M$
r_{ui}, $\hat{r}_{ui}$	Actual ratings and predicted ratings given by user u for item i, respectively
P, Q	Matrices of user and item factors, respectively
p_u	A factor vector for user u: $p_u \in P$
q_i	A factor for item i: $q_i \in Q$
∇p_u^t, ∇Q_u^t	The direction and magnitude of change needed to update p_u and Q_u, respectively, at time t
Q^*	A part of factor matrix Q selected for transmission
∇Q	Gradients of the factor matrix Q used for FL (model updates)
∇Q^*	A part of Q's gradient matrix selected for transmission (reduced payload)
T	Number of federated iterations
η	Learning rate
θ	Parameters of the neural network that are used to learn P and Q
$\mathcal{G}$	A set of users participated in FL training process for each federated iteration
$\mathcal{B}$	Batch size
E	Number of epochs

and Q obtained from a GMF component and a NCF component.[3] One advantage of using NCF in federated settings is that it can handle high-dimensional and sparse user–item data, which is common in real-world recommender systems. Moreover, NCF is scalable and can be easily adopted in production, making it well suited for the distributed and decentralized environment. Let each user $u \in U$ have an embedding vector p_u that represents its latent factor. Similarly, each item $i \in I$ has an embedding vector q_i. We use $\hat{r}_{ui}$ as predicted preference score (or rating) between user u and item i, which can be formulated as

$$\hat{r}_{ui} = \Omega(p_u, q_i), \tag{2}$$

where Ω is a scoring function used to predict r_{ui}. To model preference signals characterized by large number of non-linear relationships, NCF utilizes a **multi-layer perceptron (MLP)** to learn the function Ω. The prediction function Ω can be defined as follows:

$$\Omega(p_u, q_i) = a_{out}\left(\mathbf{h}^T \Psi(p_u \odot q_i)\right), \tag{3}$$

where a_{out}, $\mathbf{h}$, Ψ, and $\odot$ denote the activation functions, the weights of the output layer, the MLP function, and element-wise dot product of vectors, respectively. To restrict the model output as (0, 1), we use sigmoid function as a_{out}. For MLP function Ψ, let f_m, W_m, and b_m denote the activation function, the weight matrix, and the bias vector in the kth layer of the perceptron, respectively. For the NCF model with L hidden layers, the MLP function Ψ can be defined as follows:

$$\Psi(x) = \phi_L\Big(\ldots\left(\phi_2(\phi_1(x))\right)\ldots\Big), \tag{4}$$

where $\phi_m(x) = f_m(W_m^T x + b_m)$. For optimal performance, we use Rectifier (ReLU) as the activation function f_m in all perceptron layers.

Additionally, the GMF component is added to the NCF architecture by replacing the MLP layer with a multiplication layer, which essentially performs an element-wise multiplication on the user and item embedding vectors. Then, we project the weight from the multiplication layer to the output layer to be a fixed unit matrix (consisting of all ones) of dimension $K \times 1$ with a linear activation function (practically computing an inner product between user and item embedding). Integrating

[3] For simplicity throughout the manuscript, we use NCF to refer to the joint model.

GMF and MLP networks allows to learn the combination of linear and non-linear relationships from the data. Finally, the output of GMF and MLP networks are concatenated and connected with the sigmoid activation output layer.

3.2 NCF in Federated Settings

In federated learning settings, the NCF model can be used by allowing each user $u \in U$ to train a lightweight model locally using their local user–item interaction data. These local models then sent local model updates ∇Q_u to a central server, where they are aggregated to generate a global model. Each client is considered as an agent that holds private interactions and trains the model locally. The FL server maintains global parameters and controls the distributed learning process. In recommender systems, the local updates only affect a small part of the item profile for the global model, which slows down convergence if the standard FedAvg algorithm [33] is exactly used like in other FL applications. Therefore, the FedAvg algorithm must consider the specificity of the updates in the item profile, where the updates for the items that a user has not interacted, must be adjusted as a client-side update process.

Formally, given P and Q denote the latent matrices of users and items, respectively, where $p_u \in P$ and $q_i \in Q$ represent a particular user or item in matrices P and Q. Let $\mathcal{D}_u$ denote the training dataset of a particular user u, which consists of item-score (i, r_{ui}). If user u has interacted with item i, then $r_{ui} = 1$, and $r_{ui} = 0$ otherwise. It is important to consider that for each user u, a large number of items remain uninteracted in I. Thus, the negative instances in $\mathcal{D}_u$ are sampled at a ratio of $b : 1$, meaning that for each positive instance, there are b negative instances. In the federated settings for the NCF model, each user u stores p_u, local model updates ∇Q, and $\mathcal{D}_u$ locally, while global model parameters θ are stored on the FL server. To train the recommender model, we use the **binary cross-entropy (BCE)** loss to measure the difference between the predicted scores of the model and the ground-truth scores on the training dataset. Equation (5) formally shows the loss function $\mathcal{L}_u$ for predicted ratings $\hat{r}_{ui}$ and actual ratings scores r_{ui}:

$$\mathcal{L}_u = -\sum_{i, r_{ui} \in \mathcal{D}_u} \Big(r_{ui} \log \hat{r}_{ui} + (1 - r_{ui}) \log (1 - \hat{r}_{ui}) \Big). \tag{5}$$

For each federated iteration t, a subset of users $\mathcal{G} \subset U$ is randomly selected by the central server to participate in updating global model parameters θ^t. After computing $\mathcal{L}_u$, each user locally updates the user factors p_u using a learning rate η. Gradients (∇p_u^t and ∇Q_u^t) are derived to represent the direction and magnitude of the change needed to improve the model's performance. ∇p_u^t and ∇Q_u^t are derived from the loss function $\mathcal{L}_u$. These gradients are essential in adjusting the user factors and local model updates in a way that minimizes the loss function for the NCF model. The federated learning approach protects users' private data by keeping it on their devices. In our federated settings, the p_u update for the next federated iteration $t + 1$ can be defined as follows:

$$p_u^{t+1} = p_u^t - \eta \nabla p_u^t. \tag{6}$$

The selected user uploads ∇Q_u^t to the federated server for global model update. Finally, after collecting all the uploaded gradients from the selected users, the federated server updates θ by aggregating the uploaded gradients using the process:

$$Q^{t+1} = Q^t - \eta \sum_{u \in \mathcal{G}} \nabla Q_u^t. \tag{7}$$

$$\theta^{t+1} = \theta^t - \eta \sum_{u \in \mathcal{G}} \nabla \theta^t. \tag{8}$$

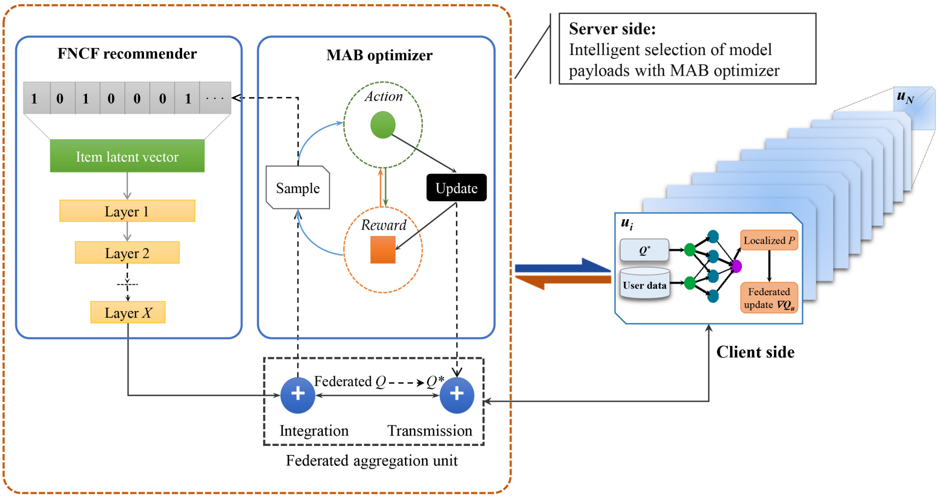


Fig. 2. Structure of the proposed communication-efficient federated neural collaborative filtering model. We employ a multi-armed bandit approach to intelligently select optimal gradients. The server-side structure is split into three key components: (1) FNCF recommender that works in federated settings for the corresponding NCF approach, (2) MAB optimizer intelligently feeds the federated aggregation unit and FNCF recommender for global model updates, and (3) federated aggregation unit controls the distributed learning process. The client-side network trains the local model on standard federated settings.

A general presentation of gradients aggregation at server-side for global model updates is given in Equation (8). We use a modified version of FedAvg algorithm [23]. As illustrated in Figure 2, each user has an independent local recommender and personalized interaction data $\mathcal{G}$ for local training. The federated server maintains a global recommender, publicly accessible item information and a **federated aggregation unit (FAU)** responsible for managing the distributed learning process. Consider the model parameters into three categories, the user vectors P, the item vectors Q, and MLP weighting parameters denoted as θ ($\nabla\theta$ reflects the direction and magnitude of adjustments). At first, the user vectors are initialized randomly and are never communicated to the other users, while the item vectors and model parameters are initialized on the server for initiating the global training process. The FNCF recommender takes user preferences from Q to integrate them into item latent vectors through the MLP network. The FNCF recommender feeds the learned preferences to FAU to interact with the MAB optimizer and transmit over the network for local model updates. The workflow inside FNCF recommender can be seen in Algorithm 1.

The user vectors are kept private, while model parameters θ and Q^* are shared and need to be aggregated at FAU at the server side. This is obtained by simply utilizing the FedAvg algorithm for users joined a specific federated round t. The integration part at FAU, as shown in Figure 2, averages the global parameters in the subsequent round and transmits them to the set of selected users. The process repeats until the global model converges on the model updates received from the local models running at the client side. For client-side model updates, when a user receives an intelligently selected part of gradients ∇Q^* the model is ready to execute local training to feed the next federated iteration. We consider the scenario when a user trains the local model with four negative instances per positive sample. The users' local data $\mathcal{D}_u$ is then divided into batches of size $\mathcal{B}$ by utilizing stochastic gradient descent approach.

Our federated implementation is generic and can be applied to other types of common recommendation frameworks, such as feature-based and graph-based models.

ALGORITHM 1: FNCF recommender: MLP based federated neural collaborative filtering model.

```
Input: $T$, $\eta$, $\mathcal{D}_u$
Output: Optimized model parameter $\theta$ and item factors $Q$
Client side: Function ClientUpdate $(u, \theta^t, Q^t)$
    User $u$ executes:
    $\mathcal{B}$ = Split $\mathcal{D}_u$ into batches of size $\mathcal{B}$
    foreach batch $b \in \mathcal{B}$ do
        $p_u \leftarrow p_u - \frac{1}{|\mathcal{G}|}\eta\nabla_{p_u}\Omega(\theta^t, Q^t, p_u)$
        $\nabla Q_u^t \leftarrow \nabla_Q \Omega(\theta^t, Q^t, p_u)$
        $\nabla \theta_u^t \leftarrow \nabla_\theta \Omega(\theta^t, Q^t, p_u)$
    end
    return $\nabla\theta_u^t, \nabla Q_u^t$
end
Client side:
    Initialize $P_u$ independently
    Fetch model parameters $\theta$ and item factors $Q$
Server side:
foreach round $t = 1, 2, 3, \ldots, T$ do
    $\mathcal{G} = u \in U$ available for local updates
    foreach user $u \in \mathcal{G}$ do
        $\nabla\theta_u^t, \nabla Q_u^t$ = ClientUpdate($u$, $\theta^t$, $Q^t$)
    end
    $\theta^{t+1} \leftarrow \theta^t - \eta\frac{1}{|\mathcal{G}|}\sum_u \nabla\theta_u^t$
    $Q^{t+1} \leftarrow Q^t - \eta\frac{1}{|\mathcal{G}|}\sum_u \nabla Q_u^t$
end
```

3.3 MAB Optimizer

Instead of transmitting entire payloads over the network, we devise an optimizer termed MAB. This optimizer carefully selects gradients (a part of overall payloads) from the global model for local updates, as illustrated in Figure 2. Numerous challenges arise in choosing the optimal set of payloads in a complex federated environment. First, due to the FL architecture and privacy considerations, the server-side aggregation unit lacks access to users' identities. This absence of user information complicates the determination of relevant items for the gradient selection process. To address this challenge, our MAB optimizer utilizes Bayesian Thompson sampling and a dynamic reward allocation mechanism for informed decision-making. Instead of relying solely on user identities (as in standard collaborative filtering approaches) or item groupings, it dynamically selects potentially relevant items based on their expected contributions to the model updates. By modeling the expected reward, the MAB optimizer effectively balances the tradeoff between exploration and exploitation, maximizing learning performance while minimizing communication costs. Additionally, the online training process of FNCF differs significantly from the standard offline training in conventional NCF models. Federated updates ∇Q continuously stream in from participating users.

These settings promptly address scalability concerns in large-scale FRS, particularly optimizing communication overhead. Our proposed MAB optimizer effectively tackles scalability issues by skillfully managing computational resources within the FL setting. As the number of participating nodes or dataset size scales, the MAB optimizer strategically optimizes payload transmission by dynamically allocating resources based on varying computational demands. For instance, in scenarios with an increased number of participating nodes, the MAB optimizer intelligently adjusts its payload optimization strategies employing underlying Bayesian Thompson sampling. This adaptability showcases the MAB optimizer's capacity to scale seamlessly and manage escalating computational demands in federated learning environments. For a comprehensive exploration of the MAB components, please refer to Section 3.3.1 and Section 3.3.2, outlining the workflow presented in Algorithm 2.

Extending the previous line of research [19], we propose an intelligent sampling strategy (briefly explained in Section 3.3.1) in the MAB optimizer. Initially, the FNCF recommender requests the set of items from the MAB optimizer, based on the action-update-reward cycle, the optimizer samples a set of potential item vectors to feed into the recommendation model for global model training and submits optimal payloads ∇Q^* to FAU for integration and transmission to a set of users $\mathcal{G}$. Also, for each iteration t the feedback s_t is used to compute the optimal reward score r_t. To address the online sequential nature of FL training, the optimizer employs the Bayesian Thompson sampling, which guides the selection of items to determine the optimal Q^* for the FL process. It leverages Bayesian inference techniques to explore and exploit different options, ultimately making informed decisions on which items to include in the gradient selection. Additionally, a carefully designed reward function (briefly explained in Section 3.3.2) quantifies the utility of samples based on the observed outcomes from a group of users $\mathcal{G}$ for each federated iteration. The workflow inside the optimizer is outlined in Algorithm 2. Formally, the MAB optimizer operates as a structured tuple <M_s, S, A, R>, representing key elements within a federated round t as follows:

(1) *Item:* M_s is a subset of M items, and in a bandit model the subset M_s is considered as available potential arms.
(2) *State:* $S = [S^1, S^2, \ldots, S^{M_s}]$ is a set of states containing the feedback collected by the optimizer from the federated environment. Specifically, a state $S^j = [s_1^j, s_2^j \ldots, s_t^j]$ obtained from the federated environment contains s_t^j that includes the feedback on an item j (for $j = 1, \ldots, M_s$) received from the set of users $\mathcal{G}$ at a particular iteration t. We denote s_t^j as the feedback mechanism that incorporates the local model updates ∇Q^*.
(3) *Action:* The set of potential actions, denoted as $A = [A^1, A^2, \ldots, A^M]$, is suggested by the bandit model. Figure 2 illustrates that the bandit model selects a subset of items M_s based on a probability distribution and performs specific actions from the action set A. Each action A^j corresponds to a sequence of individual actions a_t^j, performed by the bandit model for selecting relevant item j (where $j = 1, \ldots, M$). These recommended items are included in Q^* for a particular federated iteration t.
(4) *Reward:* The reward function, denoted as $\mathbf{R} = [R^1, R^2, \ldots, R^{M_s}]$, plays a crucial role to select a correct proportion of gradients to feed the FL process. It represents the observation signal that the system collects for a selected set of items. The reward function $\mathbf{R} : S \times A \rightarrow R$ maps the state–action pairs to corresponding reward values. It serves as a feedback signal that the system uses to update its model and make decisions for future recommendations. Specifically, each reward sequence R^j, where j ranges from 1 to M_s, captures the rewards obtained for item j in each iteration t. These rewards, denoted as r_t^j, are determined based on the feedback provided by the user after the optimizer takes action a_t^j. The reward estimation process typically involves Equation (14) or a similar equation to estimate the reward value based on the user's feedback.

There are $\mathcal{G}$ clients connected in our federated network with a fixed set of K arms. For a client $u \in \mathcal{G}$, the arm $k \in K$ generates local rewards r_k^u at iteration t independently from a sampling distribution constructed through a sampling strategy formally explained in Section 3.3.1. The MAB optimizer as a key component of the global training process, with the same K set of arms (referred to as global arms), interacts with the local models, where a global reward R^j estimated through a reward function outlined in Algorithm 2 and formally explained in Section 3.3.2. The global reward can be thought of as the virtual averaged reward for all $\mathcal{G}$ clients pulled the same arm k at iteration t. We observed that a global reward is a form of an average of local rewards; still, the global rewards are not directly accessible for the client-side local models.

3.3.1 Bayesian Thompson Sampling. Considering the theoretical significance of classical Bayesian probability for bandit models and recommender systems [12, 32, 35, 51], we employ BTS with Gaussian priors to balance exploration–exploitation tradeoffs and boost the optimization ability of our bandit model. BTS offers several advantages that enhance the gradient selection process in our FNCF model. First, BTS enables us to adapt to changing dynamics in user preferences and item characteristics. By continuously updating the posterior distributions based on observed rewards, BTS allows the model to dynamically adjust our payload selection strategy. This adaptability ensures that our system can effectively respond to shifts in user behavior and evolving item trends. As a result, the recommendation performance remains robust and aligned with the latest user preferences. Second, BTS handles uncertainty and exploration efficiently. Through the use of Gaussian priors and posterior distributions, BTS provides a probabilistic framework that captures the uncertainty in item rewards. This allows us to balance exploration and exploitation effectively. By exploring different options in a principled manner, we can discover potentially valuable recommendations that might have been overlooked in a purely exploitative approach. This capability is especially valuable in scenarios where user preferences are uncertain or rapidly changing, as it allows us to discover and adapt to emerging trends.

By integrating BTS into the gradient selection process, we devise a mechanism to sample the subsequent set of items from the posterior distributions. These samples are then used to optimally select Q^*. The model assumes (as given in Equation (9)) that the item reward distribution follows a normal distribution with an unknown mean μ and fixed precision ($\tau = \frac{1}{\text{variance}} = 1$). This Bayesian approach ensures that the selection is guided by both historical knowledge and the latest observed rewards, leading to potentially relevant payload selection,

$$p\left(R^j \mid \mu^j, 1\right) \sim \mathcal{N}\left(\mu^j, 1\right). \tag{9}$$

For an item j, the prior probability for unknown μ^j is also believed to be normally distributed with parameter μ_θ and precision τ_θ. The posterior probability distribution of μ^j can be obtained by solving the standard Bayes theorem [51] such as

$$\begin{gathered} p\left(\mu^j\right) \sim \mathcal{N}\left(\mu_\theta, \frac{1}{\tau_\theta}\right), \\ p\left(\mu^j \mid R^j, \mu_\theta, \tau_\theta\right) \sim \mathcal{N}\left(\hat{\mu}^j_\theta, \frac{1}{\hat{\tau}^j_\theta}\right), \end{gathered} \tag{10}$$

where the updates $\hat{\mu}^j_\theta$ and $\hat{\tau}^j_\theta$ for μ^j are calculated as shown in Equation (11) and Equation (12), respectively. Formal discussions and relevant proofs can be found in many existing works such as [19, 51],

$$\hat{\mu}^j_\theta = \frac{\tau_\theta \mu_\theta + n^j Z_t\left(a^j_t\right)}{\tau_\theta + n^j}, \tag{11}$$

and

$$\hat{\tau}^j_\theta = \tau_\theta + n^j \tau, \tag{12}$$

ALGORITHM 2: Payload optimization with MAB optimizer: workflow for intelligent selection of optimal gradients, and distributed learning control.

```
Input: Parameter $\theta$, item factors $Q$, $\nabla Q$, and $T$
Output: Optimized item gradients $\nabla^j Q^*$
Initialization: Set number of items to sample $M_s$
Initialize Bayesian Thompson sampling (BTS) bandit model parameters
Initialize local model updates $\nabla^j Q = 0 \; \forall j = 1, 2, 3 \ldots M_s$ matrix
Execution:
foreach round $t = 1, 2, 3, \ldots, T$ do
    Select $M_s$ items from $BTS$, having the maximum sampled values organized by their rewards as
      specified in Equation (9) and Equation (10)
    Extract the item factor $Q$ considering items in $M_s$, represented as $Q^*_t$
    Forward $Q^*_t \longrightarrow$ FAU for user-side local updates
    Collect item-factors: $\forall j = 1, 2, 3 \ldots M_s; \nabla^j Q^*_t \longleftarrow$ user-side users
    if Number of gradient updates $\leq |\mathcal{G}|$ then
        Update $\theta$ for $t + 1$ federated iteration as stated in Equation (8)
        Update $v^j_t$ using $\nabla^j Q^*_t$ as given in Equation (15)
        foreach item $j = 1, 2, 3, \ldots, M_s$ do
            Compute rewards $r^j_t$ using Equation (14)
            Update BTS parameters with $r^j_t$ using Equation (11), Equation (12), and Equation (13)
            Update $\nabla^j Q^* = \nabla^j Q^*_t$
        end
    end
end
```

where n^j represents the count of j selected in optimal set Q^*, while $Z_t(a^j_t)$ (as defined in Equation (13)) denotes the corresponding value of action a^j at the federated round t,

$$Z_t\left(a^j_t\right) = \frac{1}{n^j} \sum_{i=1}^{n^j} r^j_t, \tag{13}$$

where r^j_t as given in Equation (14) is the reward obtained at federated iteration t when action a^j_t is taken. Essentially, for each federated iteration t, we update two parameters $\hat{\mu}^j_\theta$ and $\hat{\tau}^j_\theta$ (as defined in Equation (11) and Equation (12)) of the selected item $j \in M_s$. Next, we perform sampling for μ^j from the current state of knowledge in the posterior distribution, following the procedure outlined in Equation (10). Subsequently, we select the items with the highest sampled values, prioritizing them based on their expected rewards as indicated in Equation (9). We employ a similar approach for the selection strategy reported in various existing studies [8, 18, 19]. Furthermore, extensive research has consistently shown that BTS outperforms non-Bayesian sampling strategies, leading to significant improvements in performance [19, 35, 51].

3.3.2 Reward Function. The reward estimation is a core concept in bandit-based solutions. Here, we explain the reward function designed for the proposed solution in the MAB optimizer. For each iteration t, the sampling model explained in Section 3.3.1 recommends M_s items, selected as part of Q^* item factors to receive feedback $s^j_t; j \in M_s$ (model updates or gradients denoted by $\nabla^j Q^*_t \in R^{K \times M_s}$) from all of the users. For each item j, a scalar feedback r^j_t shows the users' likeliness for the respective item. It is optimized by integrating the immediate and gradual rate of change in the gradients, jointly,

$$r_t^j = (1 - \gamma t) \left| v_t^j - \nabla^j Q_t^* \right| + \frac{\gamma}{t} \sum_{k=1}^{K} \left| \nabla^j Q_{t-1} - \nabla^j Q_t^* \right|, \tag{14}$$

where γ is the regularization term. The quantities $\nabla^j Q_{t-1}$ and $\nabla^j Q_t^*$ are the gradients of item j from the $t - 1$ and t iterations, respectively. As stated by ADAM [21], v_t^j records an exponential decay of the past squared gradients for an item j, given as

$$v_t^j = \frac{\beta_2 v_{t-1}^j + (1 - \beta_2) \left(\nabla^j Q_t^* \right)^2}{(1 - \beta_2)}, \tag{15}$$

where $0 < \beta_2 < 1$.

Inspired by the stochastic gradient approach, the optimizer computes a composite reward to optimize the selection of gradients at each federated iteration. We propose a reward formulation that considers dynamic changes in the gradients. Given an item j in the federated iteration t, the composite reward r_t^j for this item is computed as a combination of two terms. The first term $(1 - \gamma t)| \ v_t^j - \nabla^j Q_t^* \ |$ monitors the behavior of change in gradients. Here, v_t^j corresponds to the accumulated weighted squared gradients for item j, capturing the collective similarity of gradients observed in previous iterations. The term $\nabla^j Q_t^*$ represents the gradients of item j in the current iteration, indicating the current direction of optimization. As the difference between the gradual changes and the current gradients decreases, this term encourages the selection of items that contribute to stable convergence in the underline MLP architecture used for FNCF. The second term $\frac{\gamma}{t} \sum_{k=1}^{K} |\nabla^j Q_{t-1} - \nabla^j Q_t^*|$ focuses on immediate changes during the initial iterations. It sums the absolute differences between the gradients of item j in the previous iteration Q_{t-1} and the current iteration Q_t^*. By doing so, it emphasizes items that exhibit significant changes in the gradients, enabling exploration and adaptation in the early stages of optimization.

We integrate this composite reward formulation into our FNCF model for federated settings. By considering both immediate and gradual changes, our approach aims to strike a balance between exploration and exploitation, promoting convergence while allowing for adaptation to dynamic user preferences. The regularization parameter γ allows for tuning the tradeoff between immediate and gradual changes, with higher values favoring immediate changes and lower values prioritizing long-term stability. By incorporating these principles of bandit models and composite rewards, our FNCF model effectively selects gradients for recommendation generation in a federated manner. It combines the advantages of capturing gradual trends in the gradients while also adapting to immediate changes, leading to improved recommendation performance over time. Furthermore, the formulation of the composite reward aligns with the goal of minimizing regret, ensuring that our approach seeks to maximize the expected reward by optimizing the selection of gradients in the federated optimization process.

Precisely, the proposed bandit approach brings forth several advantageous aspects for practical implementation. First, by employing NCF as the foundation for an item recommendation, our approach demonstrates the potential for generalization across various deep learning based recommender systems. This highlights the versatility and applicability of FNCF-MAB beyond a specific domain. Second, the optimized selection of gradients in FNCF-MAB occurs at the server side that eliminates the need for additional computational burden on user devices, leading to a reduction in network latency. Third, despite utilizing only a fraction (10%) of the overall payloads, the recommendation performance of FNCF-MAB remains remarkably comparable to the baseline techniques such as NCF and FNCF, which employ 100% of payloads. This witnesses the effectiveness of our proposal. Moreover, our approach offers the advantage of seamless integration within the existing

Table 2. Summary of the Characteristics and Statistics of Datasets Used for Experimental Evaluation

Dataset	Users	Items	Ratings	Density	Scale
Ml-1M	6,040	3,706	1,000,209	4.4684%	1 to 5
Ml-100K	943	1,682	100,000	6.3047%	1 to 5
FilmTrust	1,508	2,071	35,497	1.1366%	0.5 to 4
YahooMusic	15,400	1,000	311,704	2.0241%	1 to 5

FL architecture and recommendation pipeline. It does not require any customization on the user side, as it performs the local training process in a generic FL style. This streamlines the adoption of FNCF-MAB, facilitating its plug-in and plug-out capability without disrupting the overall FL framework.

Taking these observations into account, the proposed FNCF-MAB model stands as a promising solution to address the communication overhead challenges in large-scale federated recommenders. By leveraging the power of bandit-based payload optimization, our approach offers the potential to significantly reduce communication costs while maintaining competitive recommendation performance. This not only benefits the efficiency and scalability of federated recommendation systems but also contributes to enhancing user satisfaction in real-world applications.

4 EXPERIMENTAL EVALUATION

In this section, we describe the experimental settings and results to validate the effectiveness of our proposed model. Experimental settings such as datasets, evaluation metrics, model configurations and baseline algorithms are given. The primary aim of our experiments is to emphasize that despite employing a restricted set of weights for client-side model updates, the performance of the selected recommender (NCF in our case) remains comparable to the full-length model. Notably, the chosen recommender demonstrates commendable performance in centralized (NCF), federated (FNCF), and communication efficient bandit (FNCF-MAB) settings. In particular, we aim to answer the following research questions:

RQ 1: Does the recommendation performance of the bandit-based model FNCF-MAB (with 10% payloads) is compared with the original NCF and FNCF models (with 100% payloads)?

RQ 2: Does the proposed FNCF-MAB model converge equally to NCF or FNCF to witness the true federated implementation for a specific task?

RQ 3: What is the impact of varying levels of payload reduction on the recommendation performance within the FNCF-MAB model?

RQ 4: How does the proposed FNCF-MAB model influence the client-side memory usage compared with the FNCF model for each federated iteration?

RQ 5: How much communication cost and bandwidth can be secured to apply gradient reduction for each federated iteration and does it affect model convergence?

4.1 Datasets

To validate the significance of the proposed bandit model for large-scale recommender systems in production, four publicly available benchmark datasets are used. The selection is made while considering the sparsity of datasets, which is a key practical challenge in large-scale systems. The selected datasets are highly sparse and diversified in nature. We preprocessed all datasets to extract the implicit feedback only. Therefore, the timestamp and other irrelevant features are dropped during the preprocessing phase. The statistics of each dataset after preprocessing are given in Table 2.

Additionally, to witness the varying dynamics of the selected datasets, we represent user distribution as per the amount of user–item interactions for each dataset in Figure 3. The horizontal

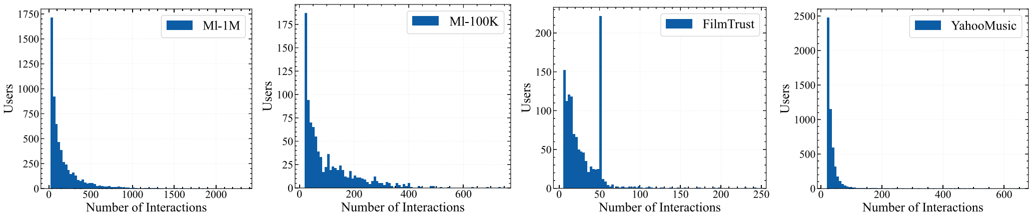


Fig. 3. Distributions of users with varying numbers of interactions in the Ml-1M, Ml-100K, FilmTrust, and YahooMusic datasets, respectively.

axis represents the number of ratings, while the vertical axis indicates the corresponding number of users. Our analysis reveals that a significant proportion of users in the datasets exhibit a limited number of ratings, which we classify as "inactive users." Conversely, a small fraction of "active users" exhibit a substantially higher number of ratings. For instance, in the Ml-1M dataset, approximately 75.28% of the users have less than 100 ratings, around 22.46% have ratings ranging between 100 and 300, and only 2.26% of the users have more than 300 ratings. An overview of these datasets is as follows:

- *Ml-1M:* The MovieLens 1 million dataset (Ml-1M) is a diverse collection of data provided by GroupLens Lab.[4] It consists of approximately 1 million ratings (1,000,209) given by 6,040 users for 3,952 movie items. Each user has rated at least 20 items on a scale of 1 to 5. This dataset serves as a widely used benchmark for evaluating recommendation performance.
- *Ml-100K:* This dataset is also provided by GroupLens Lab. It consists of 100,000 ratings given by 943 users for 1,682 different movies. Each user has rated at least 20 movies on a scale of 1 to 5. Furthermore, the dataset includes additional user information such as age, gender, and profession.
- *FilmTrust:* The FilmTrust[5] dataset was obtained by the FilmTrust website in June 2011. It consists of user–item ratings along with user-user trust information, which is particularly useful for neighborhood-based methods. The dataset exhibits a sparsity of approximately 98.86%. Additional statistics and user-wise item interactions are given in Table 2 and Figure 3.
- *YahooMusic:* Yahoo research[6] has collected billions of user ratings for musical pieces, including information on song attributes such as hidden patterns, artists, albums, and release time. As part of this extensive collection, a dataset with 311,704 ratings from 15,400 users has been made available for academic purposes.

4.2 Evaluation Protocols

For evaluation purposes, we adopt a widely used leave-one-out strategy in deep learning based recommender systems [1, 14, 30, 38, 49]. In this strategy, the training set comprises all user interactions except for the most recent one, ensuring that the model learns from a comprehensive history of user–item interactions. On the other hand, the test set consists of only the latest interaction for each user, allowing us to evaluate the model's ability to predict the user's most recent preference. By separating the training and test sets in this manner, the model's performance with varying users' interests can be evaluated. For ranking predictions, we follow the standard settings outlined in the original paper of NCF [14]. Specifically, we randomly select 100 items that the user

[4] https://grouplens.org/datasets/movielens/
[5] https://guoguibing.github.io/librec/datasets.html
[6] https://webscope.sandbox.yahoo.com/

has not interacted with and then provide ranking predictions for the test item among these 100 samples. To ensure comprehensive evaluations, we perform experiments for 300 federated iterations, allowing most clients to be naturally assessed through the internal random sampling process.

To access the model's convergence and prediction ability, we used **binary cross-entropy (BEC)** loss function. The BCE loss is commonly used for binary classification tasks and is suitable when dealing with independent binary predictions. A formal definition of loss function $\mathcal{L}$ is given in Equation (16),

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)), \tag{16}$$

where N is the total number of samples, y_i represents the ground-truth label for sample i, and $\hat{y}_i$ represents the predicted probability for sample i. To evaluate the recommendation performance, we use common ranking evaluation measures, i.e., **hit ratio (HR)** and **normalized discount cumulative gain (NDCG)** [29, 38, 51, 52]. HR is used to evaluate the precision of the recommender system, i.e., whether the test item is contained in the top-N list. NDCG measures the ranking accuracy of the recommender system, i.e., whether the test item is ranked at the top of the list. The formal definitions of both measures are given in Equation (17) and Equation (18), respectively,

$$HR@k = \frac{n}{k}, \tag{17}$$

where k represents the number of recommendations made to the particular user u and n represents the number of correctly recommended items within the top k recommendations list,

$$\begin{aligned} DCG@k &= \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log_2(i+1)}, \\ NDCG@k &= \frac{DCG@k}{IDCG@k}, \end{aligned} \tag{18}$$

where k represents the position at which we perform the evaluation (we use $k = 10$). It determines the number of items considered when calculating the gain. Also, r_i represents the relevance score of the item at position i in the ranked list. To make the performance metrics comparable, we further normalize them using the theoretically best achievable metrics for each dataset. We compute the theoretically best metrics by recommending items from the test set of each user.

4.3 Implementation Details

We implement the proposed solution with PyTorch. For convenience, code, dataset, and reproducibility guidelines are available at https://github.com/waqar-uestc/fncf_mab. The architecture of the neural collaborative model is made up of two sub-networks: (1) MLP and (2) GMF. Both MLP and GMF networks are used to learn the latent factors in the form of user and item embeddings. We employ a 16-hidden layer for MLP with a ReLU activation function. While restricting the size of the latent factors to 4, we set the architecture of the NCF model to be $\left[8, 16, 8\right]$, where the first and the last layer is the concatenation MLP and GMF individual layers. The last layer of GMF is obtained by multiplying the user embedding vector with the item embedding vector while the last layer of MLP is obtained by concatenation of the user embedding vector with the item embedding vector. For the user embedding layer, the input channel is the number of unique users N for each of the experimental datasets and the output channel is the number of factors (set to 4), likewise for the item embedding layer the input channel is number of unique items M of the datasets and output channel is the number of factors (set to 4). The sigmoid activation function is applied after the last layer to transfer the current task as a series of binary classification tasks.

Table 3. NCF's Hyper-Parameter Values Used in Our Experiments

Model	K	α	Batch size	β_1	β_2	η	ϵ
NCF	4	4	32	0.1	0.99	0.05	1e-8

K represents the number of latent factors, λ is the L2-regularization term, negative samples = α accounting for the implicit confidence parameter and batch size. β_1, β_2, η, and ϵ are the parameters of the ADAM optimizer.

For the FL setting, we train the model for 300 federated rounds and set the client's sample number to be 500. For each local client setting, the model configuration has a batch size 32, learning rate of 5e-2, and a payload reduction of 0.90 (i.e., just 10% gradients are used to independently transmit). Additionally, we optimize the model with a BCE loss to learn optimal weights, where we take four negative instances per positive sample. The Adam optimizer was used to train the model in all the three settings. To ensure consistent settings across all three methods, we employed the same hyper-parameter configurations for NCF, FNCF, and FNCF-MAB (as mentioned in Table 3), which have been identified as optimal in previous studies [3, 10]. Specifically, the number of federated rounds required to update the global model in the case of FNCF and FNCF-MAB was set to 300 for each dataset. Additionally, we set the FNCF-MAB specific hyper-parameters as $(\mu_\theta, \tau_\theta) = (0, 10000)$, and the regularization term for the reward as $\gamma = 0.999$.

4.4 Performance in Federated Environment (RQ 1)

It is important to state before performance evaluation that we are not going to set a new state-of-the-art performance level for recommenders in production, but we aim to highlight how the neural collaborative filtering outperforms in a federated environment. Also by the performance, we mean that despite a significant payload reduction (up to 90%), the outcome in terms of HR and NDCG is almost comparable with the baseline technique (i.e., NCF and FNCF that utilize full payloads) under specific settings. The aim of our initial experiment is twofold: (i) highlight our findings regarding the comparison of proposed FNCF-MAB with the NCF, FNCF, and FNCF-MAB models and (ii) compare FNCF-MAB with up to 90% payload reduction with the centrally trained version of NCF and FNCF models. Our findings presented in Table 4 help readers to understand that we obtain significant recommendation outcomes while decreasing 90% of the model payloads. Additionally, because of our self-imposed settings to use lightweight models, we deliberately force all candidate models to have the same factor size, learning rate, step-size, and embeddings. Therefore, the reflecting values of the corresponding performance indicators such as HR and NDCG are not super fancy.

Table 4 compares the recommendation performance of FNCF-MAB (with 10% model payloads) with NCF (centrally trained model) and FNCF (with 100% model payloads). The values represent the mean value of the corresponding measure of the test set recommendation performance from 300 FL iterations. The proposed FNCF-MAB method is consistently comparable to both of the baseline methods for recommendation performance. It is worth considering that the loss on recommendation performance with FNCF-MAB is quite tolerable while the load on the communication channel is considerably low compared with the baseline models. For load calculation on the communication channel, we assume a system with an FAT32 data representation, where each item is represented by 32 bits and each user is represented by 32 bits. The load on the communication channel can be approximated as the product of the user count and the item count, multiplied by the size of each data unit in bytes. This provides an estimate of the total amount of data that needs to be transmitted during a federated iteration. The corresponding values for each dataset in Table 4 are given in MB. For the ML-1M dataset, around 5% loss in ranking performance occurs while we secured 88%

Table 4. Recommendation Performance with 90% Reduction in Payloads for Each Federated Iteration

Datasets	Measure	NCF	FNCF	FNCF-MAB	Impact %
Ml-1M	Hit_Ratio	0.61365	0.56108	0.51295	−8.58%
	NDCG	0.34784	0.32680	0.30964	−5.25%
	Load	—	86.88096	9.63405	88.91%
Ml-100K	Hit_Ratio	0.61103	0.58764	0.57299	−2.49%
	NDCG	0.35421	0.33508	0.31457	−6.12%
	Load	—	7.06690	0.77721	89.01%
FilmTrust	Hit_Ratio	0.86309	0.85342	0.84712	−0.74%
	NDCG	0.82541	0.81173	0.80617	−0.68%
	Load	—	10.15720	1.08583	89.31%
YahooMusic	Hit_Ratio	0.63620	0.61254	0.59042	−3.61%
	NDCG	0.40371	0.38469	0.36647	−4.74%
	Load	—	18.34543	2.01804	88.99%

The values represent the mean value of the corresponding measure from 300 FL iterations. The proposed FNCF-MAB method is consistently comparable with both the baseline techniques trained with the full model payloads.

load on the communication channel. A similar trend can be observed on other three datasets. It is a potential advantage for large-scale systems in production.

4.5 Convergence Analysis (RQ 2)

It is worth considering that despite of 90% communication cost (payloads) reduction, as illustrated in Figure 4 the convergence behavior of the proposed FNCF-MAB model is consistent after limited rounds of initial iterations. The smooth reduction in loss curves presented in Figure 4 on all four datasets is evident that not only federated implementation of the standard NCF is stable but the FNCF-MAB model is also consistent with intelligently selected (only 10%) payloads. We observe that the loss curve stabilizes after a few initial iterations, indicating that our implementation quickly converges to a stable and optimal solution. This stability in the loss curve further emphasizes the reliability and robustness of our federated learning approach.

Additionally, recommendation indicators in terms of NDCG and HR highlight that the actual recommendation capability of neural collaborative filtering remains there despite of working with a limited portion of gradients intelligently selected through the multi-armed bandit model. Figure 4 compares the convergence behavior in terms of recommendation performance with the baseline models. The original NCF model is slightly better than FNCF and FNCF-MAB, because it is fully centralized and utilizes the entire data (i.e., taking advantage of full model payloads), while our proposed federated version of NCF (i.e., FNCF) is closely comparable with NCF. Finally, our optimized solution, the FNCF-MAB method (using only 10% communication cost) is also highly comparable with both models using the same recommendation technique.

4.6 Impact of Gradients Reduction (RQ 3)

The impact of varying levels of payload reduction on the recommendation performance within the FNCF-MAB model is a critical aspect. Figure 5 provides insights into the average recommendation performance at different levels of gradient reduction (20%, 40%, 60%, 80%, 90%, and 95%) within the FNCF-MAB model. The performance is compared against two baseline methods: the NCF model (centrally trained) and the FNCF model (utilizing 100% payloads). Figure 5 reveals a consistent

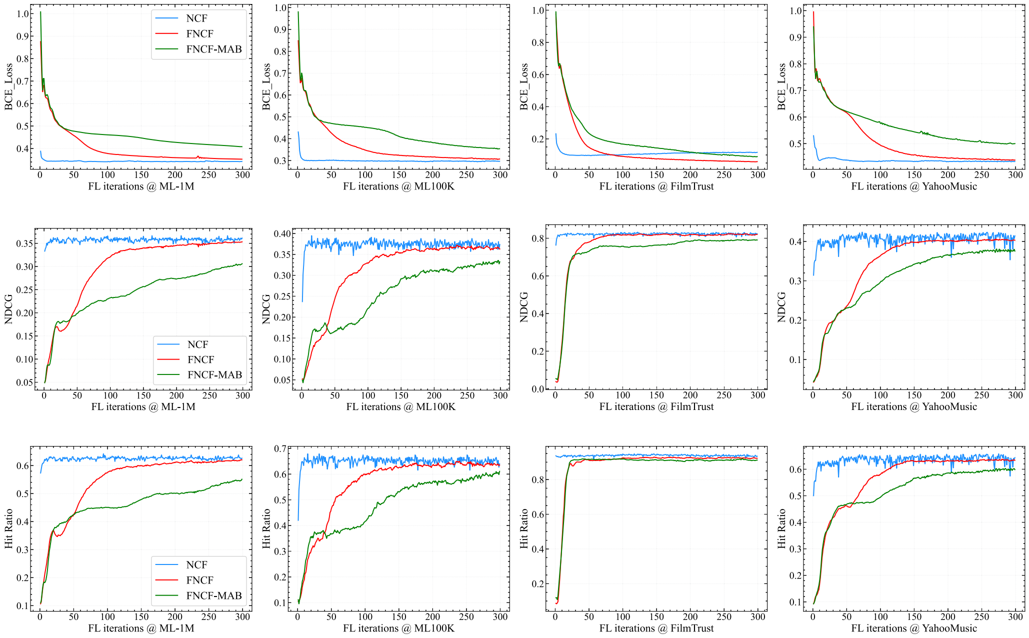


Fig. 4. The convergence behavior of the proposed FNCF-MAB model and baseline techniques on four selected datasets. The experiments performed for 300 FL iterations. The plots highlight that the model becomes consistent after around 100 FL iterations across all the datasets.

NDCG value for the two baseline models (NCF and FNCF), while demonstrating a varying average NDCG value for the FNCF-MAB model across different levels of gradient reduction on all four datasets. Our findings highlight that the recommendation performance of FNCF-MAB remains remarkably comparable up to 60% payload reduction. Even for large datasets such as ML-1M and YahooMusic, the recommendation performance remains satisfactory up to 90% payload reduction. The main advantage of running a federated model at reduced payloads, such as 90%, lies in the significant reduction in network bandwidth and communication cost. By transmitting and processing only a fraction of the original model's payload, the FNCF-MAB model achieves a notable reduction in resource consumption, including memory usage and communication load. This reduction in overhead is highly beneficial for practical recommenders operating in resource-constrained environments. In practical recommender systems, where network bandwidth and communication cost are crucial considerations, the FNCF-MAB model offers a compelling solution. By striking a balance between reduced payloads and comparable recommendation performance, it enables more efficient and cost-effective federated learning. The slight degradation in recommendation performance at higher payload reduction levels is generally acceptable and can be further fine-tuned based on specific application requirements.

4.7 Impact on Memory Consumption (RQ 4)

The proposed FNCF-MAB model has a substantial impact on client-side memory usage compared with the FNCF model for each federated iteration. The aim of our research was not to establish a new state-of-the-art accuracy level, but to explore the feasibility of executing a common recommendation model (NCF) in a federated environment. Our findings demonstrate that the FNCF-MAB

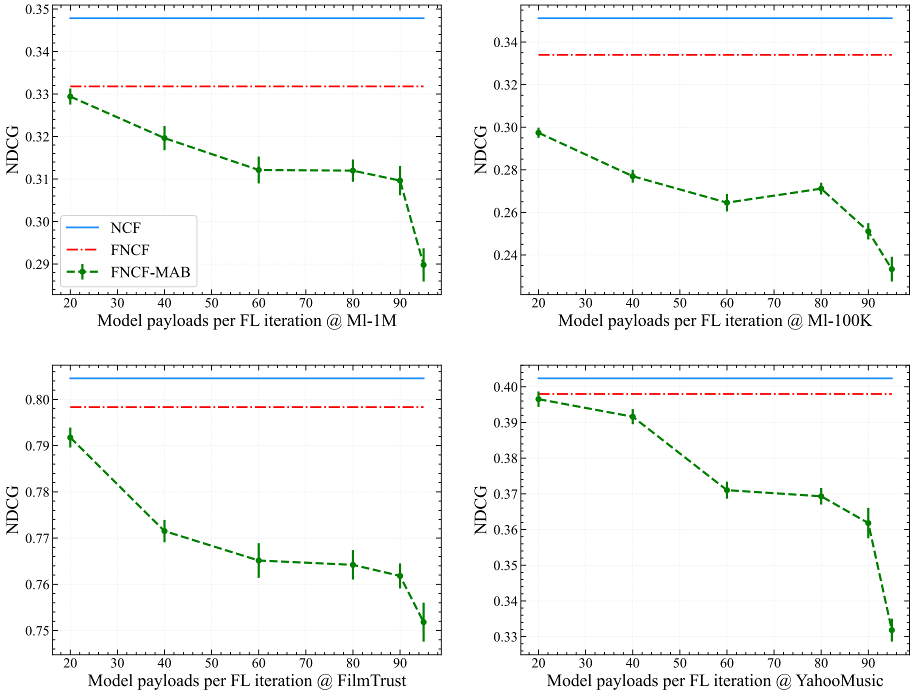


Fig. 5. Impact of payload reduction on recommendation performance in the FNCF-MAB model. Highlighted the average recommendation performance of NCF, FNCF, and FNCF-MAB at varying levels of payload reduction, ranging from 20% to 95%. The findings demonstrate that the FNCF-MAB model achieves comparable recommendation performance.

model, with only 10% of the model payload, comparably performs to both the centrally trained NCF model and the FNCF model with 100% payloads. Specifically, in this section, we highlight the effectiveness of reduced client-side memory consumption. As illustrated in Figure 6, the FNCF model with 100% payloads, imposes a significant burden on client-side memory, ranging from approximately 6–8 MB for each federated iteration for the Ml-100K dataset.

Remarkably, as depicted in Figure 6, the proposed FNCF-MAB model exhibits significantly lower memory requirements compared with the FNCF model for two test datasets. This reduction in memory usage is advantageous for several reasons. First, it allows for more efficient utilization of client-side resources, making the model more accessible to a wider range of devices with limited memory capabilities. Second, it mitigates the potential impact on system performance by reducing memory overhead. Additionally, the reduced memory footprint facilitates faster model deployment and reduces the cost associated with transmitting and storing model payloads. Overall, the advantageous reduction in client-side memory consumption provided by the FNCF-MAB model strengthens its suitability for federated recommender systems, enabling efficient execution on resource-constrained client devices while maintaining comparable recommendation performance to models with 100% payloads.

4.8 Communication Cost versus Recommendation Performance (RQ 5)

The communication cost and bandwidth required to apply gradient reduction for each federated iteration and its impact on model convergence are key aspects of our investigation. To shed light

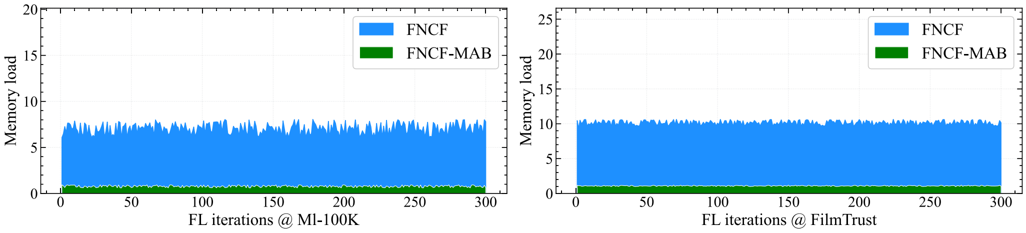
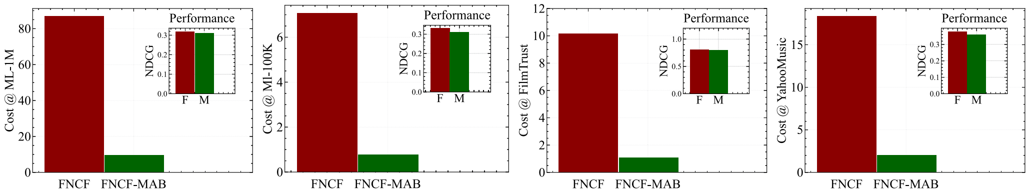


Fig. 6. Client-side memory usage for each FL iteration for the FNCF and FNCF-MAB models. The graph showcases the memory consumption, measured in megabytes, associated with the two models. The FNCF model, with 100% payloads, puts a significant burden on client-side memory, while the FNCF-MAB model demonstrates significantly lower memory requirements.



(a) ML-1M (b) Ml-100K (c) FilmTrust (d) YahooMusic

Fig. 7. Comparative analysis of communication cost reduction and impact on performance between conventional (FNCF) and intelligent (FNCF-MAB) implementations of selected recommendation model in federated settings. The outer pair of bars represents cost, average amount of gradients transmitted in megabytes for each model across four datasets. The inner pair of bars shows the recommendation performance in average NDCG for both models. The x-axis labels F and M denote the FNCF and FNCF-MAB models, respectively.

on how the communication cost and bandwidth are effectively managed, Figure 7 depicts the average load on the communication channel for both the FNCF and FNCF-MAB models. This figure serves as a nested bar plot, where the outer pair of bars represents the average amount of gradients transmitted (in megabytes) for each FL iteration across the four selected datasets. In contrast, the inner pair of bars compare the average recommendation performance in terms of NDCG. By examining the outer bars, we can clearly observe the communication cost of the FNCF and FNCF-MAB models. Importantly, the results indicate that our proposed FNCF-MAB model achieves a remarkable reduction of approximately 90% in network bandwidth compared with the FNCF model. This substantial reduction in communication cost highlights the efficiency and resource optimization achieved by the FNCF-MAB model.

Moreover, the inner bars in Figure 7 demonstrate the comparative recommendation performance between the two models. Notably, the FNCF-MAB model only sacrifices less than 5% of the recommendation accuracy compared with the FNCF model. This tradeoff indicates that we secured a significant amount of network bandwidth while maintaining a high level of recommendation performance. There are several advantages of saving 90% network bandwidth at the cost of only 5% recommendation accuracy. First, it reduces the burden on network infrastructure and mitigates potential bottlenecks, ensuring smoother and more efficient communication during federated learning. Second, the reduced communication cost enables faster transmission of model updates, leading to quicker convergence and improved overall training efficiency. Additionally, it illuminates a major challenge for bringing FL in large-scale (millions of users times billions of items) systems in production.

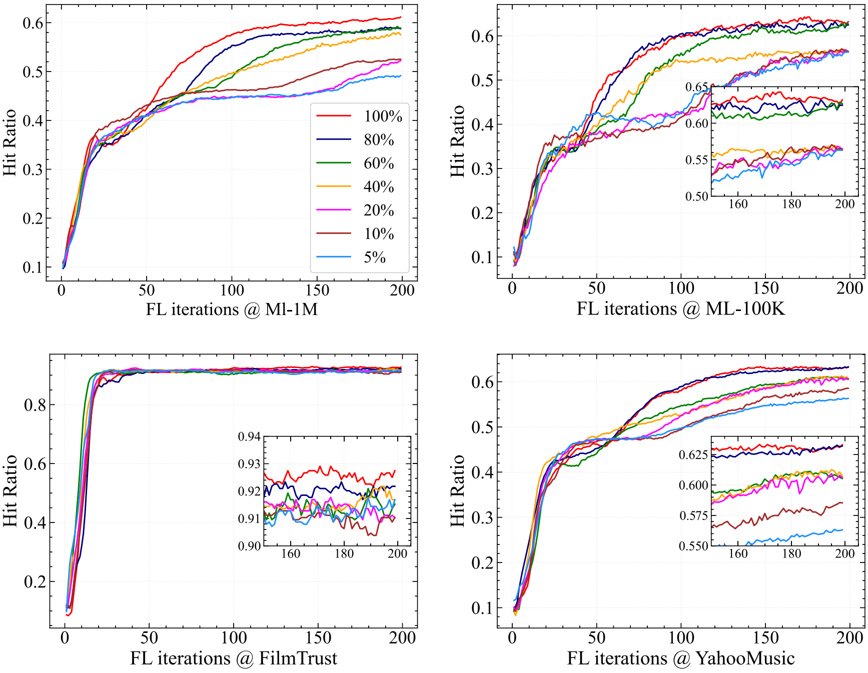


Fig. 8. Ablation analysis for varying payload reduction. The plots depict that the hit ratio remains relatively stable, showing minimal fluctuation as the payload decreases from 100% to 5%. The results indicate consistent recommendation performance across all four datasets.

4.9 Ablation Analysis

The ablation analysis aims to assess the impact of varying levels of payload reduction on recommendation performance. This investigation is crucial for several reasons. First, the ratio of payloads is a key input factor in the bandit model. It is important to understand how different levels of payload reduction affect the recommendation performance. It provides insights into the model's behavior and effectiveness. Second, the practical implementation of federated recommender systems often involves the need for resource efficiency, making it essential to evaluate the performance tradeoffs associated with payload reduction. Third, demonstrating the stability of the solution across different datasets reinforces the generalizability and reliability of the proposed bandit model.

In our ablation analysis, we systematically examined six levels of payload reduction: 20%, 40%, 60%, 80%, 90%, and 95%. The recommendation performance was evaluated using the hit ratio @10 metric, which measures the accuracy of the top-10 recommended items. The results, as depicted in Figure 8, are highly promising. We analyzed the model's performance at each level of payload reduction across all four datasets. The trend lines displayed in Figure 8 highlight the consistent and stable recommendation accuracy observed even at high payload reduction levels, such as 80% to 95%. Such stability at higher payload reduction is significant for large-scale systems in production. First, it assures that the model can maintain a satisfactory recommendation performance while significantly reducing the network bandwidth and communication cost associated with transmitting the payload. This efficiency advantage is highly beneficial in resource-constrained settings or scenarios where communication resources are expensive or limited. Second, the stability of the model across different datasets enhances its applicability and versatility, making it a reliable choice for a wide range of privacy-preserved recommendation solutions.

Our findings provide compelling evidence of the FNCF-MAB stability and consistent performance across multiple datasets. These findings are crucial for decision-makers and practitioners involved in developing and deploying federated recommender systems. By understanding the impact of payload reduction on recommendation performance, they can make informed choices and strike an optimal balance between resource efficiency and the desired recommendation accuracy.

5 CONCLUSION

Privacy-preserving and responsible recommender systems are highly demanded and creating new opportunities. Federated learning, has received significant research attention in recent years. We argue that for large-scale federated recommender systems, as the number of items grows, the volume of gradients over the communication channel increases rapidly. Training large federated recommendation models becomes practically challenging with limited resources, additionally demanding network throughput and data bandwidth. In this study, we suggest a communication-efficient neural collaborative filtering approach for federated recommender systems. To solve the underlying model complexity challenge, the proposed method uses a well-known multi-armed bandit framework that intelligently selects a smaller set gradients in each iteration of federated model training. The FL users train their local models in the regular federated learning way utilizing the payload-efficient global model, requiring no additional optimization steps. We illustrate the effectiveness of our proposal with four benchmark recommendation datasets. The results show that using only 10% of the model's payload, our method achieves recommendation performances comparable to standard federated neural collaborative filtering that uses the 100% model's payload.

The given research is subjectively performed with a well-known deep learning based recommendation model NCF, because it is a state-of-the-art recommendation method in several practical applications. In future, we would like to extend our research with other recommendation techniques. Also, we will extend our proposal by incorporating more optimization models to further reduce communication overhead for per federated iteration.

REFERENCES

[1] Alaa Awad Abdellatif, Naram Mhaisen, Amr Mohamed, Aiman Erbad, Mohsen Guizani, Zaher Dawy, and Wassim Nasreddine. 2022. Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data. *Fut. Gener. Comput. Syst.* 128 (2022), 406–419.

[2] Waqar Ali, Rajesh Kumar, Zhiyi Deng, Yansong Wang, and Jie Shao. 2021. A federated learning approach for privacy protection in context-aware recommender systems. *Comput. J.* 64, 7 (2021), 1016–1027.

[3] Muhammad Ammad-ud-din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. arXiv:1901.09888. Retrieved from https://arxiv.org/abs/1901.09888

[4] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Antonio Ferrara, and Fedelucio Narducci. 2022. User-controlled federated matrix factorization for recommender systems. *J. Intell. Inf. Syst.* 58, 2 (2022), 287–309.

[5] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. 2018. SIGNSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18).* 559–568.

[6] Rabeya Bosri, Mohammad Shahriar Rahman, Md. Zakirul Alam Bhuiyan, and Abdullah Al Omar. 2021. Integrating blockchain with artificial intelligence for privacy-preserving recommender systems. *IEEE Trans. Netw. Sci. Eng.* 8, 2 (2021), 1009–1018.

[7] Zachary Charles and Jakub Konečný. 2021. Convergence and accuracy trade-offs in federated learning and meta-learning. In *Proceeding of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS'21).* 2575–2583.

[8] Shuzhen Chen, Youming Tao, Dongxiao Yu, Feng Li, and Bei Gong. 2021. Distributed learning dynamics of multi-armed bandits for edge intelligence. *J. Syst. Arch.* 114 (2021), 101919.

[9] Yongjie Du, Deyun Zhou, Yu Xie, Jiao Shi, and Maoguo Gong. 2021. Federated matrix factorization for privacy-preserving recommender systems. *Appl. Soft Comput.* 111 (2021), 107700.

[10] Adrian Flanagan, Were Oyomno, Alexander Grigorievskiy, Kuan Eeik Tan, Suleiman A. Khan, and Muhammad Ammad-ud-din. 2020. Federated multi-view matrix factorization for personalized recommendations. In *Proceedings of the European Conference of Machine Learning and Knowledge Discovery in Databases (ECML PKDD'20), Part II.* 324–347.

[11] Mohamed R. Fouad, Khaled M. Elbassioni, and Elisa Bertino. 2014. A supermodularity-based differential privacy preserving algorithm for data anonymization. *IEEE Trans. Knowl. Data Eng.* 26, 7 (2014), 1591–1601.

[12] Dorota Glowacka. 2019. Bandit algorithms in recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys'19).* 574–575.

[13] Bryce Goodman and Seth R. Flaxman. 2017. European union regulations on algorithmic decision-making and a "right to explanation." *AI Mag.* 38, 3 (2017), 50–57.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW'17).* 173–182.

[15] Ruei-Hau Hsu, Yi-Cheng Wang, Chun-I Fan, Bo Sun, Tao Ban, Takeshi Takahashi, Ting-Wei Wu, and Shang-Wei Kao. 2020. A privacy-preserving federated learning system for android malware detection based on edge computing. In *Proceedings of the 15th Asia Joint Conference on Information Security (AsiaJCIS'20).* 128–136.

[16] Wei Huang, Jia Liu, Tianrui Li, Tianqiang Huang, Shenggong Ji, and Jihong Wan. 2023. FedDSR: Daily schedule recommendation in a federated deep reinforcement learning framework. *IEEE Trans. Knowl. Data Eng.* 35, 4 (2023), 3912–3924.

[17] Mubashir Imran, Hongzhi Yin, Tong Chen, Quoc Viet Hung Nguyen, Alexander Zhou, and Kai Zheng. 2023. ReFRS: Resource-efficient federated recommender system for dynamic and diversified user preferences. *ACM Trans. Inf. Syst.* 41, 3 (2023), 65:1–65:30.

[18] Daniel Jiang, Haipeng Luo, Chu Wang, and Yingfei Wang. 2021. Multi-armed bandits and reinforcement learning: Advancing decision making in e-commerce and beyond. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21).* 4133–4134.

[19] Farwa K. Khan, Adrian Flanagan, Kuan Eeik Tan, Zareen Alamgir, and Muhammad Ammad-ud-din. 2021. A payload optimization method for federated recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems (RecSys'21).* 432–442.

[20] Jinsu Kim, Dongyoung Koo, Yuna Kim, Hyunsoo Yoon, Junbum Shin, and Sungwook Kim. 2018. Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption. *ACM Trans. Priv. Secur.* 21, 4 (2018), 17:1–17:30.

[21] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15).*

[22] Jakub Konecný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. arXiv:1610.02527. Retrieved from https://arxiv.org/abs/1610.02527

[23] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. arXiv:1610.05492. Retrieved from https://arxiv.org/abs/1610.05492

[24] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 447–456.

[25] Mounssif Krouka, Anis Elgabli, Chaouki Ben Issaid, and Mehdi Bennis. 2022. Communication-efficient and federated multi-agent reinforcement learning. *IEEE Trans. Cogn. Commun. Netw.* 8, 1 (2022), 311–320.

[26] Zitao Li, Bolin Ding, Ce Zhang, Ninghui Li, and Jingren Zhou. 2021. Federated matrix factorization with privacy guarantee. *Proc. VLDB Endow.* 15, 4 (2021), 900–913.

[27] Wenmin Lin, Hui Leng, Ruihan Dou, Lianyong Qi, Zhigeng Pan, and Md. Arafatur Rahman. 2023. A federated collaborative recommendation model for privacy-preserving distributed recommender applications based on microservice framework. *J. Parallel Distrib. Comput.* 174 (2023), 70–80.

[28] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Dongxiao Yu, Jun Ma, Maarten de Rijke, and Xiuzhen Cheng. 2020. Meta matrix factorization for federated rating predictions. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR'20).* 981–990.

[29] Zhaohao Lin, Weike Pan, Qiang Yang, and Zhong Ming. 2023. A generic federated recommendation framework via fake marks and secret sharing. *ACM Trans. Inf. Syst.* 41, 2 (2023), 40:1–40:37.

[30] Shuchang Liu, Shuyuan Xu, Wenhui Yu, Zuohui Fu, Yongfeng Zhang, and Amélie Marian. 2021. FedCT: Federated collaborative transfer for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'21).* 716–725.

[31] Wentao Liu, Xiaolong Xu, Dejuan Li, Lianyong Qi, Fei Dai, Wanchun Dou, and Qiang Ni. 2023. Privacy preservation for federated learning with robust aggregation in edge computing. *IEEE Internet Things J.* 10, 8 (2023), 7343–7355.

[32] Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. 2015. Bayesian incentive-compatible bandit exploration. In *Proceedings of the 16th ACM Conference on Economics and Computation, EC'15, Portland, OR, USA, June 15-19, 2015.* 565–582.

[33] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated learning of deep networks using model averaging. arXiv:abs/1602.05629.

[34] Linh Nguyen and Tsukasa Ishigaki. 2019. Collaborative multi-key learning with an anonymization dataset for a recommender system. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'19).* 1–9.

[35] Yi Ouyang, Mukul Gagrani, Ashutosh Nayyar, and Rahul Jain. 2017. Learning unknown markov decision processes: A thompson sampling approach. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems.* 1333–1342.

[36] Cong Peng, Debiao He, Jianhua Chen, Neeraj Kumar, and Muhammad Khurram Khan. 2021. EPRT: An efficient privacy-preserving medical service recommendation and trust discovery scheme for ehealth system. *ACM Trans. Internet Techn.* 21, 3 (2021), 61:1–61:24.

[37] Vasileios Perifanis, George Drosatos, Giorgos Stamatelatos, and Pavlos S. Efraimidis. 2023. FedPOIRec: Privacy-preserving federated poi recommendation with social influence. *Inf. Sci.* 623 (2023), 767–790.

[38] Vasileios Perifanis and Pavlos S. Efraimidis. 2022. Federated neural collaborative filtering. *Knowl. Based Syst.* 242 (2022), 108441.

[39] Sebastian Felix Schwemer. 2021. Recommender systems in the EU: From responsibility to regulation. In *Proceedings of the FAccTRec Workshop.*

[40] Jun Sun, Tianyi Chen, Georgios B. Giannakis, Qinmin Yang, and Zaiyue Yang. 2022. Lazily aggregated quantized gradient innovation for communication-efficient federated learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 4 (2022), 2031–2044.

[41] Ashwini Tonge and Cornelia Caragea. 2019. Privacy-aware tag recommendation for accurate image privacy prediction. *ACM Trans. Intell. Syst. Technol.* 10, 4 (2019), 40:1–40:28.

[42] Soumya Wadhwa, Saurabh Agrawal, Harsh Chaudhari, Deepthi Sharma, and Kannan Achan. 2020. Data poisoning attacks against differentially private recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR'20).* 1617–1620.

[43] Qinyong Wang, Hongzhi Yin, Tong Chen, Junliang Yu, Alexander Zhou, and Xiangliang Zhang. 2022. Fast-adapting and privacy-preserving federated recommender system. *VLDB J.* 31, 5 (2022), 877–896.

[44] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2022. A survey on session-based recommender systems. *ACM Comput. Surv.* 54, 7 (2022), 154:1–154:38.

[45] Shoujin Wang, Ninghao Liu, Xiuzhen Zhang, Yan Wang, Francesco Ricci, and Bamshad Mobasher. 2022. Data science and artificial intelligence for responsible recommendations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'22).* 4904–4905.

[46] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. 2018. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS'18).* 1306–1316.

[47] Qiong Wu, Xu Chen, Tao Ouyang, Zhi Zhou, Xiaoxi Zhang, Shusen Yang, and Junshan Zhang. 2023. HiFlash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association. *IEEE Trans. Parallel Distrib. Syst.* 34, 5 (2023), 1560–1579.

[48] Zongda Wu, Guiling Li, Qi Liu, Guandong Xu, and Enhong Chen. 2018. Covering the sensitive subjects to protect personal privacy in personalized recommendation. *IEEE Trans. Serv. Comput.* 11, 3 (2018), 493–506.

[49] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Guandong Xu, and Nguyen Quoc Viet Hung. 2022. On-device next-item recommendation with self-supervised knowledge distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'22).*

[50] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.* 10, 2 (2019), 12:1–12:19.

[51] Quan Yuan, Gao Cong, Kaiqi Zhao, Zongyang Ma, and Aixin Sun. 2015. Who, where, when, and what: A nonparametric bayesian approach to context-aware recommendation and search for Twitter users. *ACM Trans. Inf. Syst.* 33, 1 (2015), 2:1–2:33.

[52] Wei Yuan, Hongzhi Yin, Fangzhao Wu, Shijie Zhang, Tieke He, and Hao Wang. 2023. Federated unlearning for on-device recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM'23).* 393–401.

[53] Honglei Zhang, Fangyuan Luo, Jun Wu, Xiangnan He, and Yidong Li. 2023. LightFR: Lightweight federated recommendation with privacy-preserving matrix factorization. *ACM Trans. Inf. Syst.* 41, 4 (2023), 90:1–90:28.

[54] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Quoc Viet Hung Nguyen, and Lizhen Cui. 2022. PipAttack: Poisoning federated recommender systems for manipulating item promotion. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM'22).* 1415–1423.

[55] Huadi Zheng, Haibo Hu, and Ziyang Han. 2020. Preserving user privacy for machine learning: Local differential privacy or federated machine learning? *IEEE Intell. Syst.* 35, 4 (2020), 5–14.

[56] Pan Zhou, Kehao Wang, Linke Guo, Shimin Gong, and Bolong Zheng. 2021. A privacy-preserving distributed contextual federated online learning framework with big data support in social recommender systems. *IEEE Trans. Knowl. Data Eng.* 33, 3 (2021), 824–838.

[57] Yuhao Zhou, Qing Ye, and Jiancheng Lv. 2022. Communication-efficient federated learning with compensated overlap-FedAvg. *IEEE Trans. Parallel Distrib. Syst.* 33, 1 (2022), 192–205.

Received 5 June 2023; revised 17 January 2024; accepted 28 February 2024