

Article

MFPCDR: A Meta-Learning-Based Model for Federated Personalized Cross-Domain Recommendation

Yicheng Di  and Yuan Liu * 

School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214126, China

* Correspondence: lyuan1800@jiangnan.edu.cn

Abstract: Cross-domain recommendation systems frequently require the use of rich source domain information to improve recommendations in the target domain, thereby resolving the data sparsity and cold-start problems, whereas the majority of existing approaches frequently require the centralized storage of user data, which poses a substantial risk of privacy breaches. Compared to traditional recommendation systems with centralized data, federated recommendation systems with multiple clients trained collaboratively have significant privacy benefits in terms of user data. While users' interests are often personalized, meta-learning can be used to learn users' personalized preferences, and personalized preferences can help models make recommendations in cold-start scenarios. We use meta-learning to learn the personalized preferences of cold-start users. Therefore, we offer a unique meta-learning-based federated personalized cross-domain recommendation model that discovers the personalized preferences for cold-start users via a server-side meta-recommendation module. To avoid compromising user privacy, an attention mechanism is used on each client to find transferable features that contribute to knowledge transfer while obtaining embeddings of users and items; each client then uploads the weights to the server. The server accumulates weights and delivers them to clients for update. Compared to traditional recommendation system models, our model can effectively protect user privacy while solving the user cold-start problem, as we use an attention mechanism in the local embedding module to mine the source domain for transferable features that contribute to knowledge transfer. Extensive trials on real-world datasets have demonstrated that our technique effectively guarantees speed while protecting user privacy.

Keywords: meta-learning; federated learning; cross-domain recommendation; attention



Citation: Di, Y.; Liu, Y. MFPCDR: A Meta-Learning-Based Model for Federated Personalized Cross-Domain Recommendation. *Appl. Sci.* **2023**, *13*, 4407. <https://doi.org/10.3390/app13074407>

Academic Editor: Jose Machado

Received: 16 February 2023

Revised: 22 March 2023

Accepted: 29 March 2023

Published: 30 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the Internet era, recommendation systems play a significant function, and in recent years, they have played an increasingly vital part in mobile apps. Following a period of study, the performance of recommendation systems has been greatly improved. Traditional recommendation systems are often based on a single domain for recommendations; however, user interests are rarely single and often extend to different application areas. The cold-start problem is a challenge for recommendation systems. Additionally, because traditional recommendation systems need to upload private data to a central server for training, this inevitably impacts on the privacy of the user.

To safeguard user privacy, we need a model that does not communicate data across the source and target domains. Federated learning can learn a shared global model from multiple independent clients but without sharing data, and it can be effective in protecting user privacy. However, real-world information is frequently diverse, and the size and sparsity of data on different clients may be inconsistent, and learning a single global model cannot meet real-world data requirements.

Existing traditional recommendation systems suffer from user privacy problems, cold-start problems, and user personalization problems. However, personalized federated learning allows each client to fine-tune the model based on their own data. The main

research problem of our paper is how to prevent user privacy leakage in a cold-start scenario of a cross-domain recommendation system while achieving personalized recommendations for users. How to prevent user privacy leakage is the focus of our research. We combine federated learning with recommendation systems to protect users' private data while using meta-learning for personalized recommendations, an approach that addresses the problem of privacy breaches in traditional recommendation systems. The novelty of our approach is that we use a meta-network to generate transferable embeddings for cold-start users, rather than building a model on each client. Cold-start users do not have interaction items in the target domain, so they need to make use of interaction items in the source domain, and we use an attention mechanism [1] in the process of obtaining user and item embeddings to help find transferable features that contribute to knowledge transfer. Based on these, we propose a novel meta-learning based federated personalized cross-domain recommendation model called MFPCDR to solve the user cold-start problem while protecting the user's private data.

The MFPCDR's core concept is as follows: (i) The central server is responsible for maintaining a globally shared recommendation model, and the user behavior data are stored on the client machine. We upload a random set of local model weights to the server to ensure that user data does not leave the local domain to effectively protect user privacy. The central server then updates the parameters and returns the updated parameters to the client in order to update the global recommendation model. (ii) After obtaining the embeddings, the central server obtains the personalized transformed user embeddings by obtaining the interest preferences between the source and target domains via the meta-recommendation module, i.e., by discovering the potential relationships between the interest preferences of overlapping users in the domain. (iii) After using the meta-recommendation module, we can obtain the personalized converted user embeddings of cold-started users in the target domain and then propose new users in the target domain based on the prediction scores, thereby solving the cold-start problem while maintaining user privacy.

The following are the primary contributions of this paper.

- (1) We propose MFPCDR, a federated personalized cross-domain recommendation model based on meta-learning that employs a federated learning technique to keep user data on the client computer in order to secure the user's private data effectively.
- (2) We use meta-networks in the meta-recommendation module to learn personalized transferable embeddings of cold-start users, enabling personalized transfer of user preferences across domain recommendations to solve the user cold-start problem.
- (3) We used an attention mechanism in the local embedding module to mine the source domain for transferable features that contribute to knowledge transfer and better obtain item embeddings.
- (4) We conducted extensive experiments on real-world datasets to demonstrate the effectiveness of our approach in terms of recommendation performance and privacy preservation.

2. Related Work

In this part, we address important prior research, such as meta-learning, federated learning for recommendation, and cross-domain recommendation.

2.1. Meta-Learning

Meta-learning, often known as learning to learn, is the act of enhancing learning algorithms across many learning stages with the goal of rapidly acquiring new concepts or abilities with limited sample data. The meta-learning paradigm may be generalized to a new task following training on many tasks. In the base learning process, the learning algorithm solves a task defined by the dataset and the goal. Optimizer-based meta-learning [2], model-based meta-learning [3], metric-based meta-learning [4], data augmentation-based meta-learning, and reinforcement learning based meta-learning [5] are the five current

classifications of meta-learning. Meta-learning has been shown to be effective in image classification [6], recommendation systems, and small sample learning [7].

Boris N. Oreshkin et al. [8] suggested metric scaling to enhance the performance of meta-learning in handling few-sample classification problems, but pairwise comparisons can be computationally expensive when the task becomes large. Finn et al. [9] proposed separating the gradient descent process from the optimized loss function task data to improve the generalization capability of meta-learning, but this suffers from overfitting problems. Ravi S et al. [10] proposed combining meta. Gordon J et al. [11] gave a probabilistic interpretation of meta-learning and proposed new algorithmic examples. Few of them target recommendation systems, which are a complex task.

A number of recent studies have introduced meta-learning to recommendation systems. Zhang et al. [12] propose a sequential meta-learning approach for any differentiable model, which solves the problem of overfitting and forgetting due to the small size of new data containing little information about long-term user preferences, but it does not learn the preferences of new users well. Wei et al. [13] offer the CLOVER meta-learning framework to ensure that meta-learning recommendation models are thorough and fair. Mecos is a meta-learning-based cold-start sequential recommendation framework proposed by Zheng et al. [14] to alleviate the item cold-start problem in sequential recommendation. Mecos can extract user preferences from limited interactions and match target cold-start items with potential users, but it tends to ignore cold items. Yang et al. [15] proposed MetaKRes, a collaborative meta-knowledge augmented recommendation system that uses convolutional encoders to transfer knowledge obtained from knowledge graphs and user preferences, but is unable to personalize recommendations for users.

In contrast to these approaches, we personalize recommendations to cold-start users by learning personalized transferable embeddings of cold-start users through a meta-network in the meta-recommendation module on the server.

2.2. Cross-Domain Recommendation

CDR leverages greater source domain information to increase recommendation performance in sparse domains in order to tackle the long-standing data sparsity problem in recommendation systems. The cold-start user recommendation workflow across domains is shown in Figure 1, where the model learns features for overlapping users in the source and target domains to obtain a mapping of user preferences and then makes recommendations in the target domain based on the mapping of user preferences. CDR jointly mines information from two datasets to better construct their relevance and present more meaningful recommendation results. Cross-domain recommendation systems can be classified into two broad categories based on the type of method: mapping-based cross-domain recommendation techniques and multi-domain recommendation algorithms trained cooperatively. In the collaborative multi-domain cross-domain recommendation technique CMF [16], for instance, data from the source and target domains are combined for matrix decomposition, and knowledge transfer is achieved by exchanging intermediate vectors, but the hidden vectors it decomposes are all consistent; therefore, it cannot achieve efficient knowledge transfer. CoNet [17] considers cross-domain recommendation in a neural network by introducing a cross-connection to achieve dual knowledge transfer; however, a substantial quantity of data is necessary to train the network, and the quality of the suggestions is contingent on the historical datasets. HYPNER [18] improves the quality of recommendations by combining collaborative filtering-based and content-based filtering methods.

Learning user preferences across domains are the topic of a second set of CDR approaches. In the mapping-based cross-domain recommendation approach, CST [19] migrates user and item knowledge into the auxiliary data and migrates the principal coordinates of people and things discovered in the auxiliary data matrix into the target domain; however, it consumes more memory and is slower to compute. OCA [20] addresses the effects of data sparsity through ordered clustering. EMCDR [21] determines that the same

user representation in multiple domains or the same object representation are related; hence, it represents this relationship using a display mapping function but amplifies the sparsity of the data when learning the linkage. In real life, where only a small percentage of users or items are shared between two domains, SSCDR [22] proposes a semi-supervised approach that uses data from non-shared users to learn the mapping function; however, semi-supervised methods often require training the network from scratch, which can be time-consuming. Some approaches explicitly model user preferences. CATN [23] learns cross-domain layer correlations by extracting multiple layers for each user and item from the review document and using an attention mechanism, but it cannot capture location information and cannot learn sequential relationships in a sequence. EnsVAE [24] uses embedding-based representations and stacked ensemble learning techniques to reduce user preference for items that are often rated by users. Chen et al. [25] consider the issue of user privacy in the context of cross-domain. Drif et al. [26] proposed an ensemble variational autoencoder framework for recommendation to improve the accuracy of recommendation systems. Because user preferences are personalized, Zhu et al. [27] propose to use meta-networks to model personalized user preferences. We are inspired by this to model user preferences in the model.

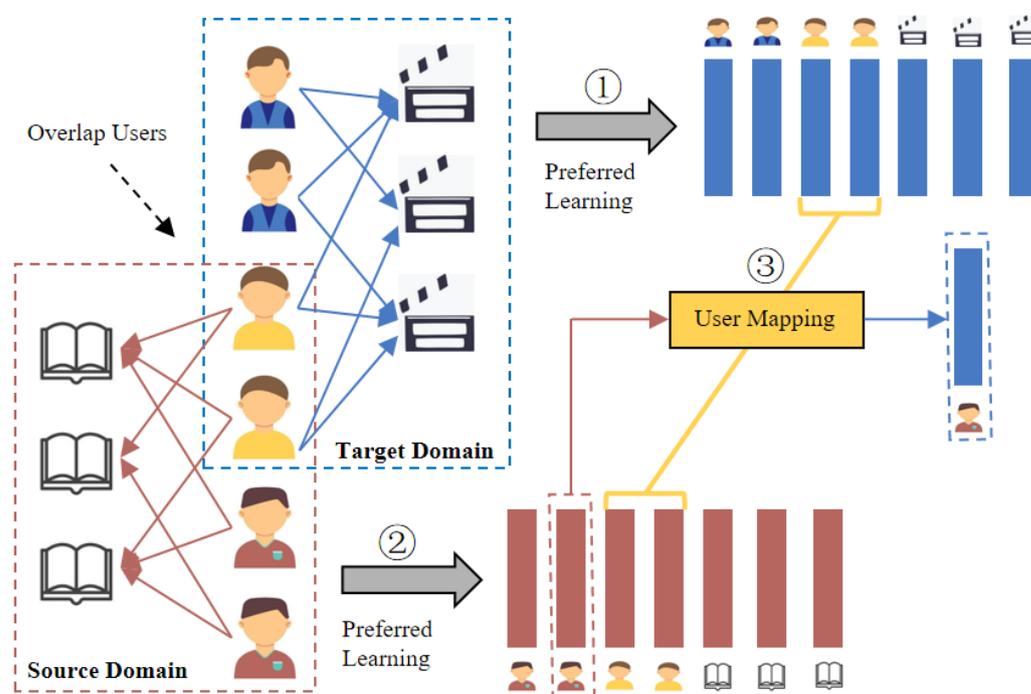


Figure 1. Workflow diagram for cold-start user cross-domain recommendation.

However, the CDR methods mentioned above, which require data to be stored on a central server, do not take privacy protection into account, and there is a risk of attack in their uploading of user behavioral data to the central server. Users' behavioral data are sensitive, and doing this poses a significant risk of a privacy breach.

2.3. Federated Learning for Recommendation System

Matrix decomposition is widely used because it can learn latent factors. He et al. [28] argue that matrix decomposition cannot learn deep information in latent features. Therefore, we propose the NCF framework to increase the capacity to mine deep user knowledge by replacing the inner product with a neural network capable of learning arbitrary functions from the input; however, it is difficult to query the side features of items. Ammad et al. [29] first explored the federated environment to explore recommendation systems, implementing a personalized recommendation system based on implicit user feedback under federation learning. In their model the main machine semester model is distributed to the client for

training and then model updates are performed on the server without the user data leaving the client, but data noise is added and the degree of preference is not expressed.

Several recent studies have advanced the application of federation learning to recommendation systems in a deeper way. Lin et al. [30] propose the FedRec framework for rating prediction with explicit feedback in the context of generalized federated recommendation frameworks by federating a number of decomposition-based recommendation models in a batch and random manner, randomly sampling unrated items, and assigning virtual ratings. This approach can obfuscate the server from accurately identifying the user's preferences for items to protect the user's privacy but introduces additional noise in the interaction, increasing computational and access costs. Liang et al. [31] improved FedRec by proposing a lossless federated recommendation method, FedRec++, which proposes to use a randomly sampled denoised client to collect noisy gradients from a normal client and then upload them. FedFast [32] optimizes the selection of client updates and the aggregation of parameters to achieve convergence more quickly. MetaMF [33] learns a small personalized model for each client through a meta-learning approach while learning the same updates as the selected clients on the server side. Collaborative signals and a meta-recommendation system are built, and then a scoring prediction model is built on the client side to generate item embeddings to achieve accurate recommendations.

The above approach applies federal learning to recommendation systems to effectively protect user privacy, and we learn from the federal learning approach and apply it to our model.

3. Model

3.1. Problem Formulation

The objective of the federated personalized cross-domain recommendation model based on meta-learning is to forecast the ratings of goods by cold-start users in the target domain. There is a source domain and a target domain, each with a user set $U = \{u_1, u_2, \dots, u_M\}$, and an item set $T = \{t_1, t_2, \dots, t_N\}$, and a rating matrix R , where M and N are, respectively, the number of users and items. For ease of differentiation, we denote the source domain user set, item set, and rating matrix as U^s, T^s, R^s , where $R^s = y_{i,j}^s$ denotes a user's rating of an item, and the user set, item set, and rating matrix in the target domain as U^t, T^t, R^t . We define the overlapping users between the two domains as $U^l = U^s \cap U^t$, and there are no shared items between the two domains, and therefore T^s and T^t do not intersect.

We convert users and items into embeddings, where the embedding of user u_i^d and item t_i^d is denoted as $p_u^d \in R^k$ and $q_v^d \in R^k$, where k represents the dimension of embedding and $d \in \{s, t\}$ is the domain label. For each user u_i , the series of sequential interactions in the origin is given as $S_{u_i} = \{t_1^s, t_2^s, \dots, t_n^s\}$, where N is the number of interaction items and N is less than M . U^c is represented as only users who have ratings in U^s , using U^l as a vehicle for knowledge transfer from the source domain to the target domain, and therefore forecasting the target domain's U^c 's rating. Let us take a real-world example to make the problem we are solving more understandable. We want to build a book recommendation system to recommend books of interest to users. However, we do not have a user's book interest dataset; we only have a user's music interest dataset, and there are some overlapping users in the user's book interest dataset and the user's music interest dataset. We take the user's music interest dataset as the source domain and the book interest dataset as the target domain. Both the music interest dataset and the book interest dataset have a user set, an item set, and a rating matrix. An overlapping user is a user who exists in both the music interest dataset and the book interest dataset but only has a rating for the music interest dataset. We need to use the overlapping users' ratings on the music interest data to learn the preferences of the overlapping users, and then we can use the overlapping users' preferences to make recommendations to them in the book interest data set, thus solving the cold-start problem. This is similar to shopping with Amazon: after we

have browsed some books that we are interested in, Amazon will recommend other items that are related to our interests.

3.2. Local Embedding Module

In the local embedding layer, we use a federated learning approach inspired by the approach proposed by Yan et al. [34], where the client is each user involved in the training of the model and the central server needs to learn to update a local learning model to obtain a representation vector of users and items. Yan et al. used the GMF method, a matrix decomposition-based algorithm for recommender systems, to make predictions by decomposing the user-item interaction matrix into the product of two low-rank matrices. We use GMF as a representation learning model to obtain user embeddings and item embeddings. GMF adds a fully connected layer for better learning of non-linear information compared to the traditional MF [35]. Unlike the normal GMF approach, we use an attention mechanism to learn item embeddings with weight. In each training session, a random number of clients are given the representation learning model for local training by the central server. The clients utilize their data to locally train the model and then submit the model's parameters to the central server for updating. The central server accumulates model parameter information and updates the global model.

During the initialization phase, the central server picks a proportion of clients at random and distributes the model to them for training without the data leaving the local region; the clients simply train the model and submit the model parameters. The local model training phase is then completed by using GMF and obtaining embeddings of users and projects. Using the example of user u and item t from the source domain, the user embedding and item embedding may be denoted as $p_u^s \in R^k$ and $q_v^s \in R^k$ respectively, where k represents the dimension of the embedding. Similarly, the user embedding and item embedding in the target domain can be denoted as $p_u^t \in R^k$ and $q_v^t \in R^k$, respectively.

We inner-product the user embedding and the item embedding and afterwards input them into the fully connected layer of the GMF to produce the prediction scores, in the method indicated in Figure 2, with the following equation:

$$\hat{y}_{u,v}^s = h^T(p_u^s \otimes q_v^s) \tag{1}$$

where h^T indicates the weight of the fully connected layer.

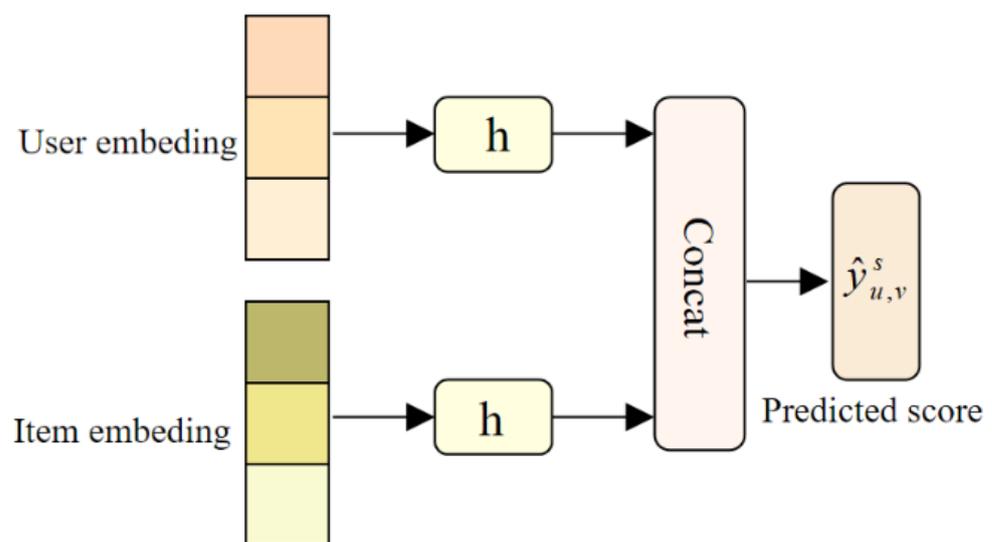


Figure 2. The process of generating predicted scores.

In the global model aggregation phase, taking the user side as an example, the weights and other parameters on the client side must be uploaded to the central server for aggre-

gation. The central server uses Fedavg [36] as the aggregation algorithm to federate the parameters, and the aggregation process can be expressed as follows:

$$W = \frac{\sum_{u \in U_{train}} |D_u| \cdot W_u}{\sum_{u \in U_{train}} |D_c|} \quad (2)$$

where W_u denotes the weight of the user u on the server side, U_{train} denotes the set of users participating in this round of training, D_c denotes the user dataset of the local model, and D_u denotes the dataset of user u . The central server distributes the aggregated model weights to the client models participating in this round before moving on to the subsequent round of training until the model converges.

3.3. Meta-Recommendation Module

In the meta-recommendation module, we must discover a user's personalized preference function and collect the user's personalized transferable preferences from interacting users. Since cold-start users lack interaction words in the target domain, we must discover transferable characteristics that contribute to knowledge transfer by utilizing interaction terms from the source domain. The attention technique [37] enables various sections to contribute differently to a single compressed representation. We used the idea of attention mechanism, firstly, to calculate the attention weights for each feature pair and their similarity, in the form of a dot product. Then, for each feature, its similarity to all other features is normalized by Softmax [38] to obtain the attention weight of that feature to the other features. Next, the interaction vector is calculated and the sum of all features is weighted using the attention weights to obtain the attention-weighted interaction vector. We do this because we discover that, for the target domain, irrelevant items contribute little to learning the user's personalized preference function; hence, the attention fraction in the item embedding is learnt via an attention network, which may be represented by the following formula:

$$o_{i,j} = \frac{\exp(o'_{i,j})}{\sum_{t_j^s \in S_{u_i}} \exp(o'_l)} \quad (3)$$

$$o'_{i,j} = g(q_v, \theta)$$

where $o_{i,j}$ is the attention score, normalized by Softmax to help find useful interaction items for cold-start users. $g(\cdot)$ denotes the attention network, θ denotes the parameters of the attention network, and the attention network is represented by feedforward networks.

Different items contribute differently to the learning of personalized transferable preferences, and we apply the attention method to item embedding by conducting a weighted summation, and the transferable embedding of users' personalized preferences can be represented by the formula as follows:

$$p_{u_i} = \sum_{t_j^s \in S_{u_i}} o_{i,j} q_v^s \quad (4)$$

where $p_{u_i} \in R^k$ denotes the personalized transferable embedding of user u_i , $o_{i,j}$ is the attention score of the item, and the attention score size is used as a judge of the importance of the item embedding q_v^s in predicting the personalized preference function. Afterwards, we can use the personalized transferable features as an input to the meta-network to learn the personalized preference function.

Users' preferences are often personalized, with different users having different preferences. We find that users' preference features in the source domain are transferable; for example, users who like martial arts novels tend to choose martial arts movies to watch when they watch movies. We use a meta-network to learn users' personalized transferable

preferences, taking as input the transferable embeddings of users' personalized preferences, and the meta-network can be represented by the following equation:

$$m_{u_i} = f(p_{u_i}, \varphi) \tag{5}$$

where $f()$ is the meta-network and φ is the parameters of the meta-network. We use a feed-forward network to represent the meta-network and express the personalized preference function by the formula as follows:

$$h_{u_i}(m_{u_i}, \partial) \tag{6}$$

where m_{u_i} is the vector and ∂ is the parameter of the personalized preference function. We learn from the approach of Zhu et al. and use a linear layer as the personalization preference function. The personalization preference function is responsible for generating different user embeddings for different users, and we personalize the converted user embeddings for cold-start users by means of the formula as follows:

$$\hat{p}_u = h_{u_i}(m_{u_i}, p_{u_i}^s) \tag{7}$$

where \hat{p}_u represents the personalized transmitted embedding of the cold-start user and $p_{u_i}^s$ represents the embedded user u_i in the source domain. In the prediction module, we can then utilize the cold-start user's transferred embedding and the item embedding in the target domain to generate suggestions for the cold-start user.

3.4. Prediction Module

After the local embedding module and the meta-recommendation module, we may generate predictions for cold-start users based on their personalized transformed user embedding and the item embedding in the target domain and then after the inner product of the following equation:

$$\hat{y}_{u,v} = \hat{p}_u \otimes q_v^t \tag{8}$$

where \hat{p}_u is the user embedding after conversion of the cold-start user to a personalized user and q_v^t is the item embedding in the target domain. We provide cold-start users with suggestions based on their predicted ratings in the target domain.

3.5. Loss

To enable the client to train the model on the source domain, we create the following loss function on the local embedding module:

$$L_r = \frac{1}{|D_{train}|} \sum_{u,v \in R} \|y_{u,v}^s - \hat{y}_{u,v}^s\|_2^2 + \lambda_u \sum_u \|p_u^s\|_2^2 + \lambda_v \sum_v \|q_v^s\|_2^2 \tag{9}$$

where D_{train} represents the number of user-item interactions in the source domain, λ_u represents the user-side regularization term, and λ_v represents the item-side regularization term. We use back propagation and stochastic gradient descent for effective training of federation learning on dispersed data.

On the meta-recommendation module, in order to train the meta-network, we refer to the method used by Fu et al. [39] to minimize the distance using a mapping-oriented approach and define the loss function as follows:

$$L_m = \sum_{u_i \in U^t} \|\hat{p}_u - p_u^t\|^2 \tag{10}$$

where \hat{p}_u represents the user embedding following the cold-start user personalization transformation, whereas p_u^t represents the user embedding in the target domain. The

mapping-oriented minimization of distance will bring the cold-start user’s personalized transformed user embedding closer to the user embedding in the target domain.

3.6. Overall Procedure

Figure 3 illustrates the general model architecture of MFPCDR. The training procedure consists of three modules: the local embedding module, the meta-recommendation module, and the prediction module, as shown in Algorithm 1.

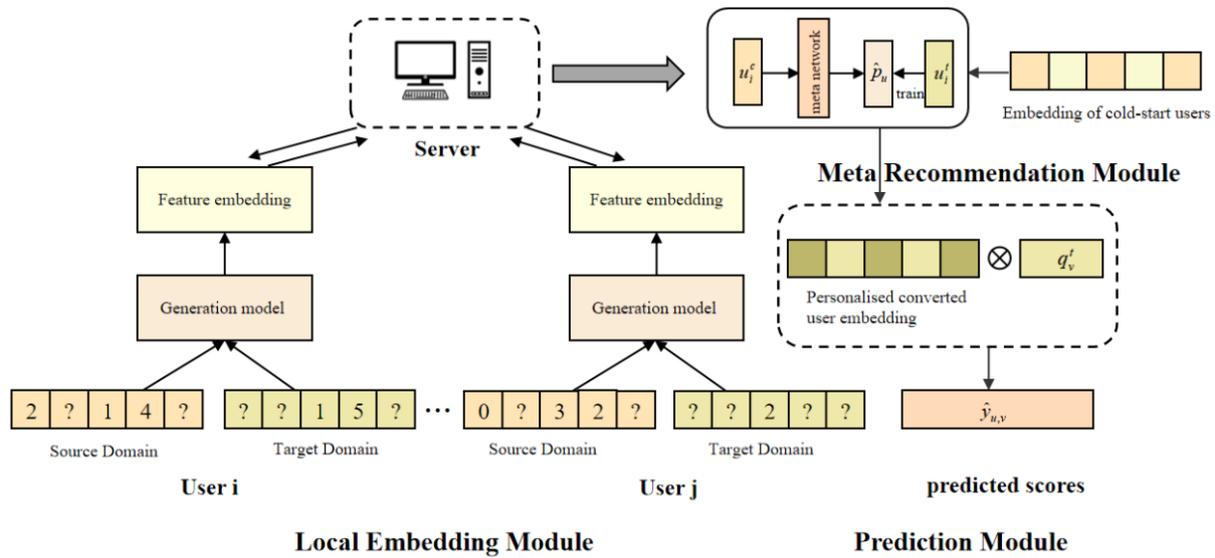


Figure 3. The framework of MFPCDR model.

The local embedding module: this is primarily accountable for learning the possible spaces of the source and target domains. The client trains the model in the source domain using Equation (9), and, following training using a federated learning technique, we may receive the respective user and item embeddings in the source and target domains, as well as the user’s personalized transferable user embedding.

The meta-recommendation module: this is mainly responsible for training the meta-network to learn personalized transformed user embeddings; the meta-network is optimized by Equation (10).

The prediction module: after obtaining the personalized transfer embeddings of cold-start users and the item embeddings of the target domain, we apply an internal product model to forecast the ratings of cold-start users.

Algorithm 1: A meta-learning-based model for federated personalized cross-domain recommendation.

Input: $U^s, U^t, T^s, T^t, R^s, R^t, U^l$

Output: $\hat{y}_{u,v}$

Local Embedding Module:

1. The client trains the model by optimizing Equation (9) and learns the representation vectors for users and items in the source and target domains p_u^s, q_v^s, p_u^t , and q_v^t , respectively.
2. Learn the personalized transferable feature embedding p_{u_i} of user u_i .

Meta-Recommendation Module:

3. Learn a meta-network $f(\varphi)$ by minimizing Equation (10) to obtain personalized converted user embeddings \hat{p}_u .

Prediction Module:

4. The prediction score is calculated by the inner product model after obtaining the tailored conversion user embedding for the cold-start user \hat{p}_u and the item embedding in the target domain q_v^t for the cold-start user.
-

4. Experiments

In this section, we conduct experiments to evaluate the effectiveness of MFPCDR. We answer the following research questions: **RQ1:** How does MFPCDR perform in cold-start scenarios compared to other models? **RQ2:** How well does MFPCDR work in more realistic recommendation scenarios? **RQ3:** How does the proportion of overlapping users affect the model? **RQ4:** Does varying sparsity in the source domain influence MFPCDR performance? **RQ5:** What is the classification performance of MFPCDR?

4.1. Datasets

To compare our model to a baseline, we conducted trials with the Amazon review datasets. The Amazon datasets were chosen because Amazon’s data volume is large enough to provide more samples with diversity, and the Amazon datasets is real, stable, and repeatable enough to provide enough information to verify the effectiveness of the system. The Amazon datasets can be divided into sub-datasets such as “Books”, “Electronics”, and “Movies and TV” according to product categories, which can effectively help us conduct cold-start recommendation experiments across domains: e.g., the source domain is the “Book” domain and the target domain is the “Movie and TV” domain. For other e-commerce datasets, the MFPCDR model is also applicable as long as the samples are diverse and have overlapping users. Three major areas were selected as the three domains: “Movies and Television”, “Books”, and “CDs and Vinyl”. In each domain, we deleted entries that had no review language and then filtered out people with fewer than 10 interactions and goods with fewer than 30 interactions. People with fewer than 10 interactions were deleted because they provided too few reviews or feedback to provide enough information for analysis, and items with fewer than 30 interactions were removed because the item’s sales or market performance was insufficient to support analysis, or they were deemed unpopular or irrelevant. These filtering criteria ensure that only high-quality data are used for analysis, improving the credibility of the data and the accuracy of the results.

Since these three domains are highly interrelated, three cross-domain recommendation scenarios can be constructed in pairs: “Books” to “Movies and TV”, “Movies and TV” to “CDs and Vinyl”, and “Books” to “CDs and Vinyl”. In each scenario, the one with more users is selected as the source domain, and the one with fewer users is selected as the target domain. We randomly chose 30% of the users as cold-start users whose interactions in the target domain were unknown to the model and were used to verify and test the model, while 70% of the users were used to train the model. To simulate varying amounts of overlapping users, we evaluated our model by randomly picking overlapping users from the training set and adjusting the proportion η to $\eta \in \{5\%, 10\%, 20\%, 50\%, 60\%\}$. The situations for cross-domain recommendations are detailed in Table 1.

Table 1. Statistics of the Amazon datasets.

Scenario	Domain	Dataset	Overlap Users	$\eta = 5\%$	$\eta = 10\%$	$\eta = 20\%$	$\eta = 50\%$	$\eta = 60\%$
Scenario 1	S	Books	6053	149	302	604	1517	1815
	T	Movies and TV						
Scenario 2	S	Movies and TV	2839	67	135	276	673	803
	T	CDs and Vinyl						
Scenario 3	S	Books	1692	38	81	160	412	487
	T	CDs and Vinyl						

4.2. Baseline Methods

We have compared this to the following classic and contemporary benchmarks.

- CMF: A migration learning method that uses other relational data related to the predicted relationship to improve prediction performance when doing relational learning tasks.

- EMCDR: A framework for cross-domain mapping and recommendation that uses a multi-layer perceptron to capture non-linear mapping functions across domains, followed by the learning of mapping functions using entities with adequate data to enhance robustness.
- CDLFM [40]: The cross-domain latent feature mapping model takes into account the similarity link between users in terms of rating behavior and provides a matrix decomposition that includes user similarity and three similarity measures. A domain-based gradient boosting tree approach was used for cross-domain knowledge transfer. Mapping functions are learned for cold-start users utilizing latent attributes of users in the auxiliary domain with similar ratings to cold-start users.
- DFM [39]: A deep fusion model based on comments and content is utilized to preserve more semantic information by expanding the stacking noise reduction auto-encoder and then transferring user latent factors across the two domains using a multi-layer perceptron.
- ANR [41]: A comprehensive aspect-based model that performs aspect-based representation learning on users and things using an attention-based component to describe the aspect process underlying the user's evaluation strategy. After training the model on the target domain, related source domain comments are utilized to provide recommendations.

4.3. Experimental Setup

We used PyTorch to implement our proposed MFPCDR model, preprocessing all user and project documents in the datasets to remove deactivated words and words with high document frequency. We used grid search to adjust the hyperparameters of all methods and Adam to optimize the model, and we set the learning rate to {0.001,0.005,0.01,0.05,0.1,0.5}, the embedding dimension range for users and items to {8,16,32,64}, and the batch size test range to {64,128,256,512}. The activation function was set to sigmoid, and the performance of all methods was reported after five runs. We found that the MFPCDR model works well when the Adam learning rate is set to 0.01, the user and item embeddings are set to 32, and the batch size is set to 128.

For assessment metrics, we considered RMSE, MSE, MAE, and Top-K. MSE is the abbreviation of mean square error, which is the average value of the square of the difference between the predicted value and the actual value. It is a measure of the difference between the predicted value and the actual value. The smaller the MSE, the smaller the difference between the predicted result and the real result, indicating that the model's prediction effect is better. RMSE is less sensitive to outliers than MSE because it operates on the square root. MAE maximizes user satisfaction but does not minimize prediction error. Top-K measures how many items in a given recommendation the recommendation algorithm matches with the user's true preferences, often with a long tail of users problem. The MSE is sensitive to outliers and computationally efficient, and it can be effectively applied to our model. Therefore, we use MSEs as performance indicators; they are used as assessment metrics in many related works, and the formula for the performance indicator MSE is:

$$MSE = \frac{1}{N} \sum_{(i,j,y_{i,j}) \in R_{test}} (y_{i,j} - \hat{y}_{i,j})^2 \quad (11)$$

where R_{test} is the test set used for performance comparison or the cold-start user validation set used for parameter selection, N is the number of ratings in the test set, $y_{i,j}$ is the true rating of item t_j by user u_i , and $\hat{y}_{i,j}$ is the predicted rating of item t_j by user u_i . According to the definition of MSE, a smaller MSE number suggests higher model performance.

4.4. Model Comparison (RQ1 and RQ2)

According to the MSE, we examine the performance of several models with varying amounts of overlapping users. Figure 4 is a graphical representation of the experimen-

tal results, and Table 2 provides a summary of the outcomes for all three cross-domain recommendation tasks. It can be seen from the data that MFPCDR performs better than other baseline models in three cross-domain recommendation scenarios. The mean MSE of MFPCDR in the first scenario is 1.086, and the median is 1.081. The mean MSE of MFPCDR in the second scenario is 1.126, and the median is 1.128. The mean MSE of MFPCDR in the third scenario is 0.8896, and the median is 0.895. The results yielded the following observations.

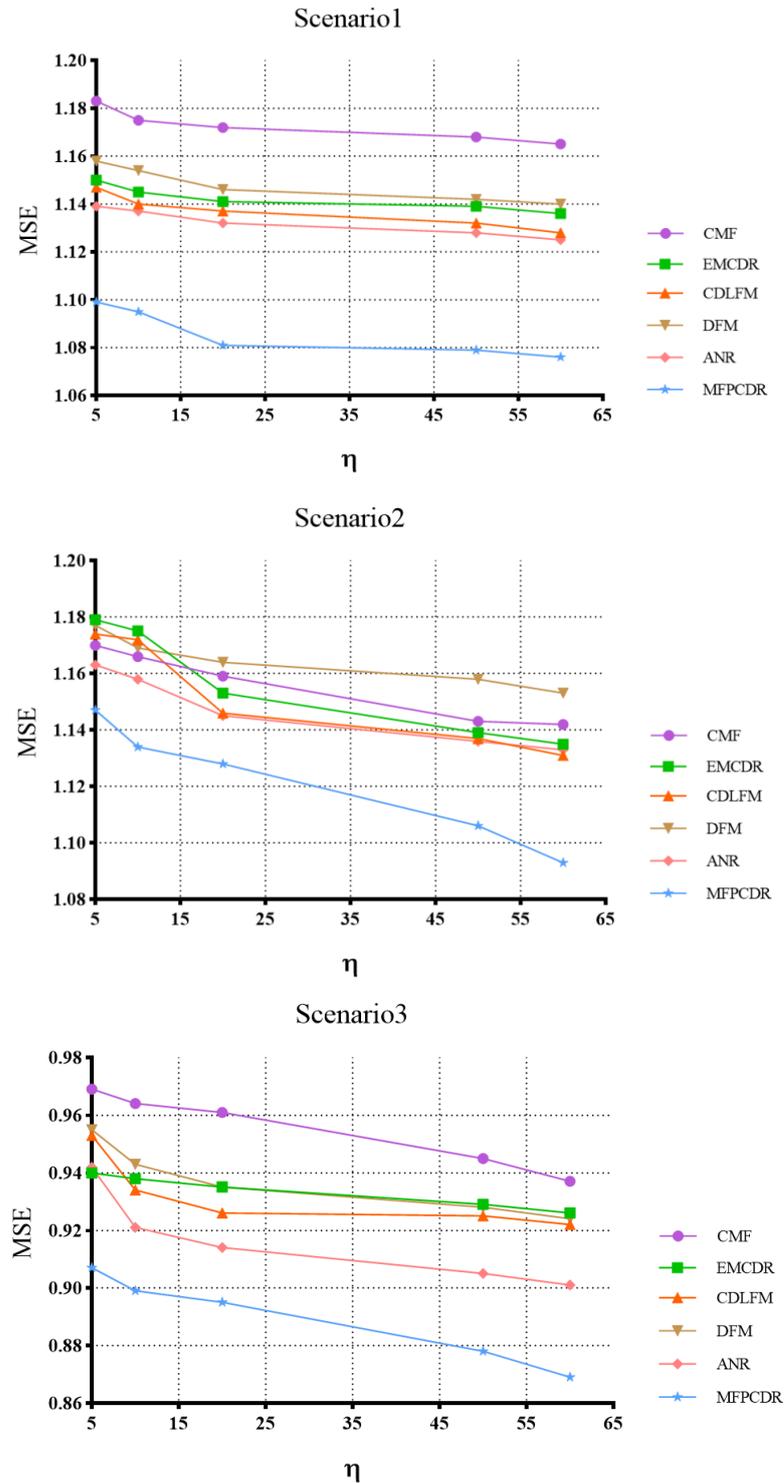


Figure 4. Impact of the proportion of overlapping users (η represents the proportion of overlapping users).

Table 2. Comparison of MSE in three cross-domain recommendation scenarios.

Scenario	Scenario 1					Scenario 2					Scenario 3					
	$S \rightarrow T$	Books \rightarrow Movies and TV					Movies and TV \rightarrow CDs and Vinyl					Books \rightarrow CDs and Vinyl				
Methods	η	5%	10%	20%	50%	60%	5%	10%	20%	50%	60%	5%	10%	20%	50%	60%
CMF		1.183	1.175	1.172	1.168	1.165	1.170	1.166	1.159	1.143	1.142	0.969	0.964	0.961	0.945	0.937
EMCDR		1.150	1.145	1.141	1.139	1.136	1.179	1.175	1.153	1.139	1.135	0.940	0.938	0.935	0.929	0.926
CDLFM		1.147	1.140	1.137	1.132	1.128	1.174	1.172	1.146	1.137	1.131	0.953	0.934	0.929	0.925	0.922
DFM		1.158	1.154	1.146	1.142	1.140	1.177	1.169	1.164	1.158	1.153	0.955	0.943	0.935	0.928	0.924
ANR		1.139	1.137	1.132	1.128	1.125	1.163	1.158	1.145	1.136	1.133	0.942	0.921	0.914	0.905	0.901
MFPCDR		1.099	1.095	1.081	1.079	1.076	1.147	1.134	1.128	1.106	1.093	0.907	0.899	0.895	0.878	0.869
Improve%		3.51	3.69	4.50	4.34	4.35	1.37	2.07	1.48	2.64	3.53	3.71	2.38	2.07	2.98	3.55

Firstly, MFPCDR outperformed all methods for all cross-domain recommendation tasks, and MFPCDR outperformed all methods with different proportions of overlapping users. In Scenario 1, when the proportion of overlapping users reaches approximately 20%, the fold starts to level off gradually. This phenomenon is also observed in Scenario 2 and Scenario 3 when the proportion of overlapping users reaches approximately 10%. We find that it is at the beginning of the model, as the proportion of overlapping users increases, that the model is more likely to capture the personalized transferable preferences of users, which helps our model solve the user cold-start problem. Additionally, as the proportion of overlapping to users continues to increase, the discounting starts to level off, which we find is due to several reasons. Firstly, it is due to the overlap between users who may be interested in the same products; secondly, when the proportion of overlapping users increases, more users in the MFPCDR model will generate similar behaviors; and, finally, when the proportion of overlapping users is high, the MFPCDR model is susceptible to the influence of popular recommendations. Compared to ANR, MFPCDR improved performance in MSE by 3.53% to 4.35% when the proportion of overlapping users was 60%. As the proportion of overlapping users becomes larger, the performance improvement also becomes larger. Both MFPCDR and ANR outperform CMF, demonstrating the effectiveness of the deep learning approach. CMF performs the worst of all evaluations because it learns the user representation by decomposing the union matrix only. The DFM retains more semantic information for the model through the autoencoder. However, neither of them achieves optimal results.

According to our experiments, DFM shows degraded results compared to EMCDR. What we found is that the stacked noise reduction autoencoder takes as input the original rating vector, which may have over 100,000 dimensions in the real-world Amazon dataset.

In this case, the training parameters to be optimized are too large, which makes it difficult for the model to converge and produces poor results. Although ANR is not a model designed for cross-domain recommendation scenarios, it works better compared to other baselines, and we found that it is its approach to modelling the aspectual process behind the user–evaluation–item approach that improves the competitiveness of the model. In the vast majority of cases, MFPCDR achieved better results than ANR, suggesting that our use of meta-learning to learn personalized user preferences in cross-domain recommendations is more appropriate than representing user and item vectors at the layer level. We find that the MFPCDR model works better than other models because it uses an attention mechanism on the item embedding by performing a weighted summation, which reduces the weight given to learning the user’s preferences for useless items and thus improves the model’s effectiveness. Our successful integration of federal learning methods in a cross-domain recommendation system is effective in protecting user privacy. One can also observe that all models typically produce worse results in Scenario 2 than in Scenarios 1 and 3, due to the lower data density in Scenario 2. Our models are applicable to different regions or domains, but the effectiveness of the recommendations depends on the quality of the data, feature

selection, and user feedback. The data should adequately reflect the interests and behavior of users. Different domains and regions may have different characteristics. For example, in a stock exchange recommendation system, trading history and stock fundamentals indicators may be important features, while in a location-based recommendation system, user location and attached shops may be important features, and applying MFPCDR in these scenarios requires appropriate feature selection.

From the experimental results, it can be seen that the baseline model is sensitive to the proportion of overlapping users in the training data, but our MFPCDR shows better performance, especially in the initial stage, when the proportion of overlapping users is 5%, which indicates that our model has good recommendation results in real scenarios. As the proportion of overlapping users decreases, the cross-domain recommendation model cannot be trained well due to the lack of training instances to learn user preferences, which can lead to poorer results. Our proposed MFPCDR largely mitigates the effect of the proportion of overlapping users on the model because it uses meta-learning to learn users' personalized transferable preferences with less reliance on overlapping users. This also makes our model more adaptable to real-world recommendation scenarios because it uses a federated learning approach in learning embeddings, which keeps user data local to protect user privacy. Because our meta-recommendation module needs to learn from interacting users, our model is not suitable for extreme cold-start situations. Additionally, because a federated learning approach is used to protect user privacy, this inevitably introduces noise that reduces the effectiveness of the model training.

4.5. Impact of the Size of Overlapping Users (RQ3)

To investigate the influence of the number of overlapping users, further tests were performed at five percentage levels of 5%, 10%, 20%, 50%, and 60% in three cross-domain recommendation situations. Figure 4 depicts the outcomes. Observations indicate that MSE lowers as the fraction of overlapping users increases, which means that the larger the proportion of overlapping users, the more user preferences the model can learn and the better the model accuracy. The setting of the proportion of overlapping users directly affects the number of users from which the system learns similar preferences for cross-domain recommendations. Clearly, a greater amount of overlapping users allows the model to better understand the preferences of the cross-domain recommendation domain, thus improving the accuracy of the recommendations. Intuitively, the curve of our proposed MFPCDR model is relatively flat.

4.6. Effect of Source Domain Sparsity (RQ4)

To examine the impact of source domain sparsity on MFPCDR performance, we altered the sparsity of the source domain by sampling from the original dataset so that the sparsity sp could be modified to $\{1, 1/2, 1/4, 1/8\}$, where $sp = 1$ signifies an unsampled original dataset. Figure 5 depicts the results of applying the MFPCDR model to the setups of Scenario 1, Scenario 2, and Scenario 3 with the fraction of overlapping users in the training dataset set to 60% in the interest of fairness and consistency. We can observe from the findings that data sparsity decreases model performance. The graph demonstrates that the MSE performance curve with MFPCDR is flatter in all three circumstances, which indicates that it has better sparsity perception and can be better adapted to real-world scenarios. We found that this is attributable to the attention mechanism used by MFPCDR in obtaining the personalized transferable feature embeddings of the user, where different items contribute differently to obtaining the personalized preferences of the user, and it is the use of this approach that allows the MFPCDR model to handle sparse data better.

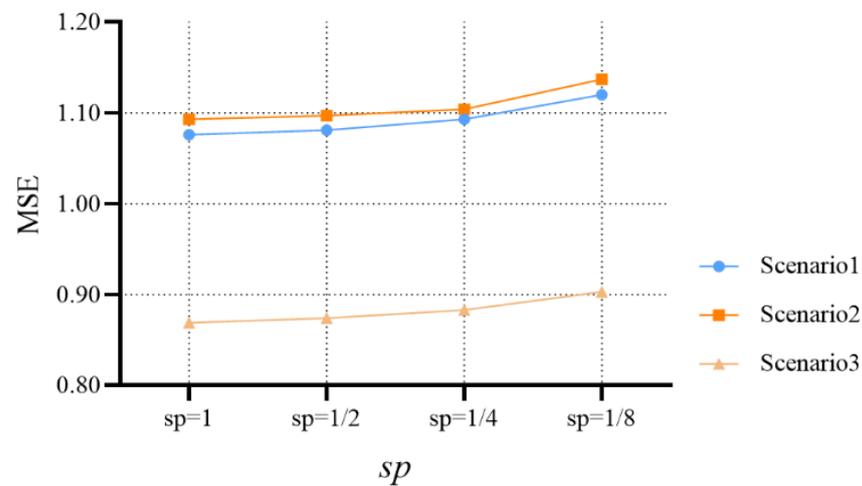


Figure 5. Effect of the sparsity degree sp of the source domain.

4.7. Classification Performance Evaluation of MFPCDR (RQ5)

To evaluate the classification performance of the MFPCDR model, we chose Precision, Recall and F-measure as important metrics to assess the model performance. Their mathematical formulas are defined as follows:

$$\begin{aligned}
 Precision &= \frac{TP}{TP+FP} \\
 Recall &= \frac{TP}{TP+FN} \\
 F-Measure &= \frac{2*Precision*Recall}{Precision+Recall}
 \end{aligned}
 \tag{12}$$

where TP indicates the number of samples that the model correctly predicts as positive, FP indicates the number of samples that the model incorrectly predicts as negative, and FN indicates the number of samples that the model incorrectly predicts as negative. *Precision* measures the ratio of correct recommendations made by the recommendation system to the actual correct recommendations among all recommendations, while *Recall* measures the coverage of all possible correct recommendations by the recommendation system. *F-Measure* is a combination of *Precision* and *Recall*, which provides an overall assessment. *F-Measure* measures the balance between *Precision* and *Recall*, and it is influenced by both *Precision* and *Recall*, so it can reflect the comprehensive performance of the recommendation system well.

We compared the MFPCDR model to five baseline models with a source domain of “Books” and a target domain of “CDs and Vinyl” and a 60% overlap in the proportion of users, and the experimental results for all methods are reported in Table 3. To make the results clearer, we show the experimental results in the form of bar charts in Figure 6.

Table 3. Precision, Recall, and F-measure experimental results table.

Methods	Scenario	Books → CDs and Vinyl		
	Index	Precision	Recall	F-Measure
	CMF	0.70	0.61	0.65
	EMCDR	0.74	0.63	0.68
	CDLFM	0.78	0.67	0.72
	DFM	0.81	0.69	0.75
	ANR	0.85	0.73	0.79
	MFPCDR	0.89	0.78	0.83
	Improve%	4.7%	5.4%	5.0%

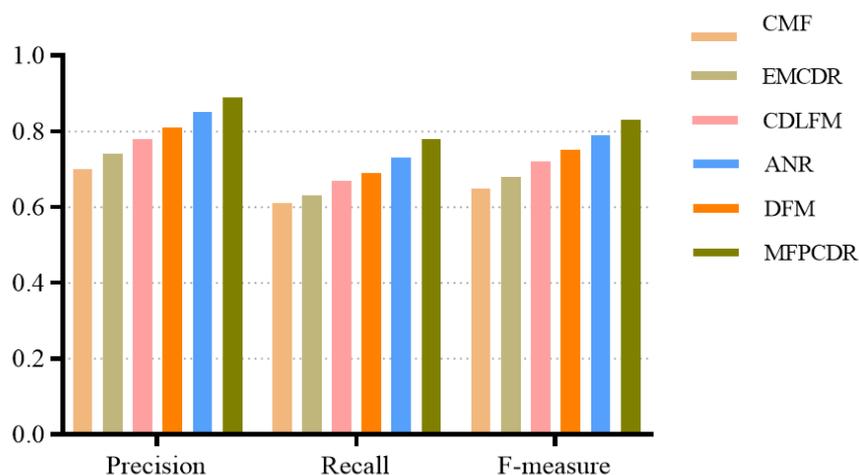


Figure 6. Precision, Recall, and F-measure experimental results graph.

As can be seen from Table 3, our MFPCDR model improves by 4.7% in the Precision metric compared to the best model in the baseline, indicating that the MFPCDR model is better at recommending items of interest to users; it improves by 5.4% in the Recall metric, indicating that the MFPCDR is better at capturing user interest preferences than the baseline model; and it improves by 5.0% in the F-measure metric improved by 5.0%, indicating that the MFPCDR model can cover the items that users are really interested in while ensuring the accuracy of the recommendation results.

5. Conclusions

The objective of this study is to solve the cold-start problem in recommendation systems and the privacy preservation challenge in cross-domain recommendation systems. To this end, we propose a novel meta-learning-based framework for federated personalized cross-domain recommendation systems, namely MFPCDR, in which a federated learning approach is used to train the model on the local embedding module; user behavior data are stored on the client machine to participate in the training, and upon completion of training, the data and settings are transferred to a central server for update without the user behavior data leaving the client machine, thus protecting the user's privacy data. Compared with traditional recommendation systems, we combine federated learning methods with recommendation systems to effectively prevent user privacy leakage. We use meta-learning, also on the meta-recommendation module, to effectively learn the user's personalized preferences to solve the user cold-start problem. After implementing the local embedding module, we may gain user and project embedding as well as a customized conversion of user embedding for cold-start users, after which we can obtain the transfer of cold-start users through the meta-recommendation module. In the prediction module, the items and embeddings of the target domain are inner-produced in order to calculate the prediction scores of the cold-start users. We promote cold-start users based on their prediction scores, thus resolving the cold-start issue in recommendation systems. We study the efficacy of our proposed MFPCDR on real-world datasets, and extensive experimental findings indicate the efficacy of MFPCDR.

We intend to continue to investigate privacy-preserving solutions for cross-domain recommendations in the future in order to improve speed while safeguarding user privacy even further.

Author Contributions: Conceptualization, Y.D.; methodology, Y.D.; software, Y.D.; validation, Y.D. and Y.L.; formal analysis, Y.D.; investigation, Y.D.; resources, Y.L.; data curation, Y.D.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D. and Y.L.; visualization, Y.D.; supervision, Y.L.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant number 61972182.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data generated or used during the current study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; p. 30.
2. Lee, K.; Maji, S.; Ravichandran, A.; Soatto, S. Meta-learning with differentiable convex optimization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10657–10665.
3. Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M.W.; Pfau, D.; Schaul, T.; Shillingford, B.; De Freitas, N. Learning to learn by gradient descent by gradient descent. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; p. 29.
4. Thrun, S.; Pratt, L. Learning to learn: Introduction and overview. In *Learning to Learn*; Springer: Boston, MA, USA, 1998; pp. 3–17. [[CrossRef](#)]
5. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; p. 29.
6. Khodadadeh, S.; Boloni, L.; Shah, M. Unsupervised meta-learning for few-shot image classification. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; p. 32.
7. Wang, Y.X.; Ramanan, D.; Hebert, M. Meta-learning to detect rare objects. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9925–9934.
8. Oreshkin, B.; Rodríguez López, P.; Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018; p. 31.
9. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
10. Ravi, S.; Larochelle, H. *Optimization as a Model for Few-Shot Learning*; International Conference on Learning Representations; OpenReview: Amherst, MA, USA, 2016.
11. Gordon, J.; Bronskill, J.; Bauer, M.; Nowozin, S.; Turner, R.E. Meta-learning probabilistic inference for prediction. *arXiv* **2018**, arXiv:1805.09921.
12. Zhang, Y.; Feng, F.; Wang, C.; He, X.; Wang, M.; Li, Y.; Zhang, Y. How to retrain recommender system? A sequential meta-learning method. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, China, 25–30 July 2020; pp. 1479–1488.
13. Wei, T.; He, J. Comprehensive fair meta-learned recommender system. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 1989–1999.
14. Zheng, Y.; Liu, S.; Li, Z.; Wu, S. Cold-start sequential recommendation via meta learner. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 35, pp. 4706–4713.
15. Wang, S.; Yang, L.; Gong, J.; Zheng, S.; Du, S.; Liu, Z.; Philip, S.Y. MetaKRec: Collaborative Meta-Knowledge Enhanced Recommender System. *arXiv* **2022**, arXiv:2211.07104.
16. Singh, A.P.; Gordon, G.J. Relational learning via collective matrix factorization. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 650–658.
17. Hu, G.; Zhang, Y.; Yang, Q. Conet: Collaborative cross networks for cross-domain recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 667–676.
18. Darvishy, A.; Ibrahim, H.; Sidi, F.; Mustapha, A. HYPNER: A hybrid approach for personalized news recommendation. *IEEE Access* **2020**, *8*, 46877–46894. [[CrossRef](#)]
19. Pan, W.; Xiang, E.; Liu, N.; Yang, Q. Transfer learning in collaborative filtering for sparsity reduction. In Proceedings of the AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010; Volume 24, pp. 230–235.
20. Gulzar, Y.; Alwan, A.A.; Abdullah, R.M.; Abualkishik, A.Z.; Oumrani, M. OCA: Ordered Clustering-Based Algorithm for E-Commerce Recommendation System. *Sustainability* **2023**, *15*, 2947. [[CrossRef](#)]
21. Man, T.; Shen, H.; Jin, X.; Cheng, X. Cross-domain recommendation: An embedding and mapping approach. In Proceedings of the IJCAI, Melbourne, Australia, 19–25 August 2017; Volume 17, pp. 2464–2470.

22. Kang, S.; Hwang, J.; Lee, D.; Yu, H. Semi-supervised learning for cross-domain recommendation to cold-start users. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1563–1572.
23. Zhao, C.; Li, C.; Xiao, R.; Deng, H.; Sun, A. CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, China, 25–30 July 2020; pp. 229–238.
24. Drif, A.; Zerrad, H.E.; Cherifi, H. Ensvae: Ensemble variational autoencoders for recommendations. *IEEE Access* **2020**, *8*, 188335–188351. [[CrossRef](#)]
25. Chen, C.; Wu, H.; Su, J.; Lyu, L.; Zheng, X.; Wang, L. Differential Private Knowledge Transfer for Privacy-Preserving Cross-Domain Recommendation. In Proceedings of the ACM Web Conference 2022, Virtual, France, 25–29 April 2022; pp. 1455–1465.
26. Drif, A.; Zerrad, H.E.; Cherifi, H. Context-awareness in ensemble recommender system framework. In Proceedings of the 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Kuala Lumpur, Malaysia, 12–13 June 2021; pp. 1–6.
27. Zhu, Y.; Tang, Z.; Liu, Y.; Zhuang, F.; Xie, R.; Zhang, X.; Lin, L.; He, Q. Personalized transfer of user preferences for cross-domain recommendation. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Tempe, AZ, USA, 21–25 February 2022; pp. 1507–1515.
28. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
29. Ammad-Ud-Din, M.; Ivannikova, E.; Khan, S.A.; Oyomno, W.; Fu, Q.; Tan, K.E.; Flanagan, A. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv* **2019**, arXiv:1901.09888.
30. Lin, G.; Liang, F.; Pan, W.; Ming, Z. Fedrec: Federated recommendation with explicit feedback. *IEEE Intell. Syst.* **2020**, *36*, 21–30. [[CrossRef](#)]
31. Liang, F.; Pan, W.; Ming, Z. Fedrec++: Lossless federated recommendation with explicit feedback. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 4224–4231.
32. Muhammad, K.; Wang, Q.; O’Reilly-Morgan, D.; Tragos, E.; Smyth, B.; Hurley, N.; Geraci, J.; Lawlor, A. Fedfast: Going beyond average for faster training of federated recommender systems. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, CA, USA, 6–10 July 2020; pp. 1234–1242.
33. Lin, Y.; Ren, P.; Chen, Z.; Ren, Z.; Yu, D.; Ma, J.; Rijke, M.D.; Cheng, X. Meta matrix factorization for federated rating predictions. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, China, 25–30 July 2020; pp. 981–990.
34. Yan, D.; Zhao, Y.; Yang, Z.; Jin, Y.; Zhang, Y. FedCDR: Privacy-preserving federated cross-domain recommendation. *Digit. Commun. Netw.* **2022**, *8*, 552–560. [[CrossRef](#)]
35. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [[CrossRef](#)]
36. McMahan, E.B.; Moore, D.; Ramage, S.; Hampson, B.A.; Arcas, Y. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
37. Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; Chua, T.S. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv* **2017**, arXiv:1708.04617.
38. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
39. Fu, W.; Peng, Z.; Wang, S.; Xu, Y.; Li, J. Deeply Fusing Reviews and Contents for Cold Start Users in Cross-Domain Recommendation Systems. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 1–27 February 2019; Volume 33, pp. 94–101.
40. Wang, X.; Peng, Z.; Wang, S.; Yu, P.S.; Fu, W.; Hong, X. Cross-domain recommendation for cold-start users via neighborhood based feature mapping. In *International Conference on Database Systems for Advanced Applications*; Springer: Cham, Switzerland, 2018; pp. 158–165.
41. Chin, J.Y.; Zhao, K.; Joty, S.; Cong, G. ANR: Aspect-based neural recommender. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 147–156.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.