Federated News Recommendation with Fine-grained Interpolation and Dynamic Clustering

Sanshi Lei Yu
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
meet.leiyu@gmail.com

Qi Liu*
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
qiliuql@ustc.edu.cn

Fei Wang
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
wf314159@mail.ustc.edu.cn

Yang Yu
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
yflyl613@mail.ustc.edu.cn

Enhong Chen
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
cheneh@ustc.edu.cn

ABSTRACT

Researchers have successfully adapted the privacy-preserving Federated Learning (FL) to news recommendation tasks to better protect users' privacy, although typically at the cost of performance degradation due to the data heterogeneity issue. To address this issue, Personalized Federated Learning (PFL) has emerged, among which model interpolation is a promising approach that interpolates the local personalized models with the global model. However, the existing model interpolation method may not work well for news recommendation tasks for some reasons. First, it neglects the fine-grained personalization needs at both the temporal and spatial levels in news recommendation tasks. Second, due to the cold-user problem in real-world news recommendation tasks, the local personalized models may perform poorly, thus limiting the performance gain from model interpolation. To this end, we propose FINDING (Federated News Recommendation with **F**ine-grained **In**terpolation and **D**ynamic Cluster**ing**), a novel personalized federated learning framework based on model interpolation. Specifically, we first propose the fine-grained model interpolation strategy which interpolates the local personalized models with the global model in a time-aware and layer-aware way. Then, to address the cold-user problem in news recommendation tasks, we adopt the group-level personalization approach where users are dynamically clustered into groups and the group-level personalized models are used for interpolation. Extensive experiments on two real-world datasets show that our method can effectively handle the above limitations of the current model interpolation method and alleviate the heterogeneity issue faced by traditional FL.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Security and privacy** → **Privacy protections**.

KEYWORDS

news recommendation; personalized federated learning; model personalization

ACM Reference Format:
Sanshi Lei Yu, Qi Liu, Fei Wang, Yang Yu, and Enhong Chen. 2023. Federated News Recommendation with Fine-grained Interpolation and Dynamic Clustering. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614881

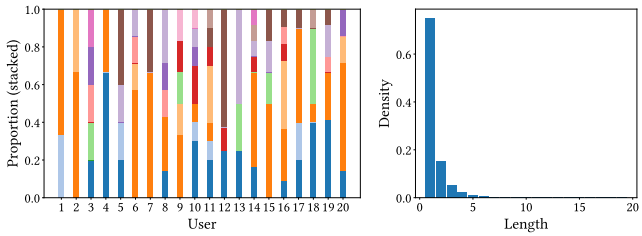
1 INTRODUCTION

News recommendation is crucial for online news platforms to help users find the news that interests them. Traditional news recommendation methods require centralized storage of user behavior, which is highly privacy-sensitive [32]. Federated Learning (FL) [27] is a distributed training framework that allows multiple clients to jointly train a deep learning model without ever sending raw data to the central server. To protect user privacy in the training and development of a news recommendation model, researchers have successfully adapted the privacy-preserving federated learning to news recommendation tasks [24, 29, 30, 43].

*Corresponding Author.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614881



(a) (b)

Figure 1: Data distributions of the MIND dataset. (a): category distribution of users' history news. Each color denotes a news category. (b): length distribution of users' training samples.

However, these federated news recommendation methods implicitly assume that a single global model can be a good fit for all users. This may not be true since the participants usually have different personal preferences or behavioral patterns, namely the non-IID data problem [20]. Fig. 1a shows the category distributions of the clicked news of 20 randomly-selected users in the real-world MIND dataset (to be introduced later), from which we can see the different preferences of the users. Therefore, in federated news recommendation settings, the averaged model may be far from the true global optimum [33] because of the phenomenon of client drift [17]. As a result, these methods typically suffer from performance degradation.

Personalized Federated Learning (PFL) aims to train personalized models tailored to each client and has shown great potential in dealing with this data heterogeneity problem. The general idea of PFL is to find a good balance between the global model, which usually generalizes well to the *common* data, and the local models, which capture the individual preferences of users and may achieve better performance on the *individual* data. In this literature, model interpolation [26] is an intuitive and widely-used strategy for federated personalization, where the interpolation between the local and global models is used for local inference. In this way, both the generalization of the global model and the personalization of the local models can be well preserved. This interpolation approach is commonly regarded as a promising FL personalization strategy by the FL community [2, 8, 28].

However, this model interpolation strategy has some limitations when applied to news recommendation tasks. First, it neglects the fine-grained personalization needs of the local models. In fact, for a federated news recommendation model based on model interpolation, the personalization needs, i.e., the interpolation weights of the local models, vary at both the temporal and spatial levels:

- **At the temporal level:** the news recommendation model may have tons of parameters, and training a federated news recommendation model would mean a long-term process. Then the importance of the global and local models would change over time, as would the personalization needs. To be specific, when the training is just starting, the global model is more useful since it contains the information from the whole data, thus better capturing the underlying data pattern and generalizing well; however, when the training is converging, the local models should be given more weights as they can better reflect the distributions of the local data and model the users' individual preferences.
- **At the spatial level:** as shown in Fig. 2, a news recommendation model generally has a hierarchical structure, where different layers in the model have different roles and thus different personalization needs. Specifically, the lower layers in the model are generally used to capture the more primitive features from the input data. For example, the word embedding layer maps the word tokens into the embedding space; the self-attention layer in a news encoder captures the complex relationships between words. These layers are expected to behave similarly for different users. On the contrary, the upper layers do more to capture individual user preferences. For example, the layers in user encoders generally learn the user representations, which should involve more personalization.

Second, this model interpolation method suffers from the cold-user problem [34] in real-world news recommendation scenarios. For real-world news platforms, a large fraction of users are called *cold* users with little or no interaction data. Fig. 1b illustrates the length distributions of users' training samples in the real-world MIND dataset, where we can observe that most users have less than five training samples. With insufficient training data, the parameters in the local models can only be updated a few times by gradient descent and most parameters will be nearly the same as the randomly initialized value. Therefore, the local personalized model itself may perform poorly, thus limiting the performance gain from model interpolation.

To address the above limitations of the current model interpolation method, we propose FINDING (Federated News Recommendation with **F**ine-grained **In**terpolation and **D**ynamic Cluster**ing**), a novel personalized federated learning framework based on model interpolation for news recommendation. It consists of two components, i.e., the fine-grained model interpolation and the group-level personalization with dynamic clustering.

The first part of FINDING, i.e., the fine-grained model interpolation strategy, is designed to satisfy the personalization needs at both the temporal and spatial levels. It includes both time-aware and layer-aware interpolation. The time-aware interpolation mechanism works by assigning personalization weights based on training progress. The layer-aware interpolation, on the other hand, takes into account the position of the model parameters and applies the varying personalization weights layer-wisely.

The group-level personalization with dynamic clustering, the second part of FINDING, targets the cold-user problem in news recommendation tasks. Specifically, the data of a cold user may be insufficient to train an ideal local personalized model, but the data of a group of users with similar preferences may be. Inspired by Clustered Federated Learning (CFL) [31], we propose the group-level personalization approach where the users are clustered into groups and those within a group collaboratively train a shared group-level personalized model, which will be interpolated with the global model. To obtain more accurate clusterings, we periodically run the user clustering algorithm based on user representations.

The main contributions of our work are as follows:

- We propose the fine-grained model interpolation method for federated news recommendation, which is both time-aware and layer-aware. It takes into account both the progress of the federated training and the position of model parameters, providing better model personalization in federated learning settings.
- We propose the group-level personalization strategy by dynamically clustering users into groups and interpolating the global model with the group-level personalized models. It can effectively alleviate the cold-user problem in news recommendation tasks.
- We conduct thorough experiments as well as the visualization experiment on real-world datasets to verify the effectiveness of our approach.

2 RELATED WORK

2.1 Federated News Recommendation

News recommendation is an important technique to alleviate the information overload problem and has been widely studied in recent years [5, 16, 19, 22, 35, 38, 44]. The existing works mainly focus on learning the accurate representations of users and news. For example, An et al. [3] propose to learn both the long-term and short-term user representations with the GRU network. Wu et al. [38] employ the Multi-head Self-attention module to model the complex relationships between news words and user behavior. Wu et al. [37] use the Attention module to learn the news representations from multiple views.

These traditional news recommendation methods typically rely on centrally stored user behavior data for model training, which can lead to serious privacy concerns. To protect user privacy, Qi et al. [29] propose FedNewsRec, which adapts federated learning to news recommendation tasks. Besides, they propose to apply local differential privacy to protect the private information in gradients. In Efficient-FedRec [43], the authors decompose the news recommendation model into a large news model maintained in the server and a light-weight user model shared by the server and clients. In this way, the computational and communication costs of the clients can be greatly reduced. Uni-FedRec [30] and PrivateRec [24] are unified privacy-preserving federated news recommendation frameworks that focus on the practical training and serving of a federated news recommendation model. However, these existing federated news recommendation methods ignore the data heterogeneity issue in federated settings, and thus usually have weaker performance than the centralized models with the same model structure.

2.2 Personalized Federated Learning

Personalized federated learning aims to train personalized models for clients to handle the data heterogeneity under the federated learning settings [4, 13, 23, 26, 31, 36]. For example, Hanzely and Richtárik [13] introduce a new optimization formulation for training federated learning models that simultaneously optimizes the loss of local models and the difference between the local models. CFL [31] recursively bi-partitions the users by the local model updates in a top-down way. Wang et al. [36] propose to tune some or all parameters of the trained global model by retraining the model on local data. In FedPer [4], the base layers of a deep model are trained centrally while the top layers (i.e., personalization layers) are trained locally. Mansour et al. [26] propose three approaches for personalization, namely user clustering, data interpolation, and model interpolation. Among the three methods, the first two require some meta-features of the clients which may raise privacy concerns. The last one is a promising approach for personalization, but it is not well suited for news recommendation tasks since it ignores the fine-grained personalization needs and the cold-user problem in news recommendation, which are what our framework tries to handle.

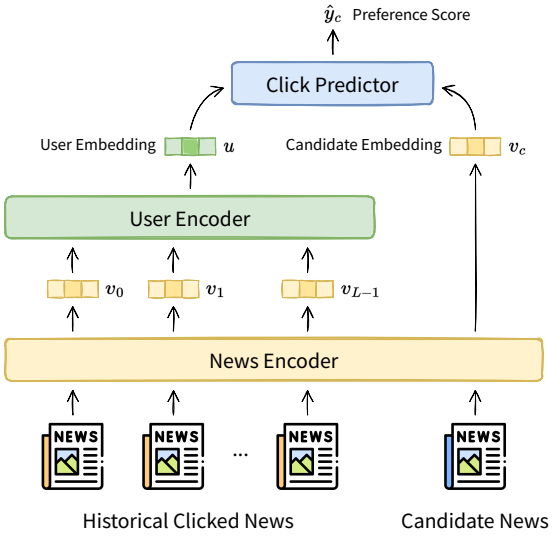


Figure 2: General news recommendation model structure

3 PRELIMINARIES

3.1 General News Recommendation Model

In this section, we describe the general structure of news recommendation models, which is widely used in the news recommendation literature. As shown in Fig. 2, the news recommendation model generally consists of three core modules: news encoder, user encoder, and click predictor.

3.1.1 News Encoder. The news encoder is used to learn the news representations from news content. In the regular news recommendation task, both the historical clicked news and the current candidate news are fed into the news encoder to obtain their vector representations. The news encoder is usually the most important part of a news recommendation model. It can be implemented by various model structures, e.g., CNN and Attention [37], Self-attention [38], BERT [39].

3.1.2 User Encoder. The user encoder is used to learn the user representations from users' historical clicked news. Denoting the representations of a user's clicked news as $\{\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{L-1}\}$, these L vectors are input to the user encoder to get the user representation u, which is used in the later preference prediction part. For example, NRMS [38] uses both Additive Attention and Self-attention

to learn user representations. LSTUR [3] proposes to capture user preferences with the GRU network.

3.1.3 Click Predictor. After obtaining the user representation $\boldsymbol{u}$ and the candidate news representation $\boldsymbol{v}_c$, the task of the click predictor is to predict the preference score based on the two representations. The most popular method is to use the dot product between the two vectors, which is simple but effective. Some more complex structures, e.g., Multilayer Perceptron (MLP) [14], are also used to capture the complex relationship between them.

3.2 Personalized FL with Model Interpolation

In this section, we will introduce the model interpolation [26] method in personalized federated learning literature since our framework is based on it.

As its name suggests, the basic idea of model interpolation is to use the interpolation of the global and local models to exploit both the generalization of the former and the personalization of the latter. Formally, the federated training process at round t can be formulated as follows:

$$\boldsymbol{w'}^t_{l_i} = \lambda \boldsymbol{w}^{t-1}_{l_i} + (1-\lambda)\boldsymbol{w}^{t-1}_g \tag{1a}$$

$$\boldsymbol{w}^t_{l_i} = \boldsymbol{w'}^t_{l_i} - \eta \nabla \ell(\boldsymbol{w'}^t_{l_i}, d_i) \tag{1b}$$

$$\boldsymbol{w}^t_g = \boldsymbol{w}^{t-1}_g - \eta \sum_{i \in S_t} \frac{|d_i|}{\sum_{k \in S_t} |d_k|} \nabla \ell(\boldsymbol{w'}^t_{l_i}, d_i) \tag{1c}$$

where $\boldsymbol{w'}^t_{l_i}$ represents the temporary interpolated model parameters of client i at round t. $\boldsymbol{w}^t_{l_i}$ denotes the local model parameters of client i after round t and $\boldsymbol{w}^t_g$ is the global model parameters after round t. $\ell(\boldsymbol{w}, d_i)$ is the loss of model $\boldsymbol{w}$ on the local data of client i, i.e., d_i. η is the learning rate and S_t denotes the set of selected clients at round t. $\lambda \in [0, 1]$ is the interpolation coefficient, which controls how much the model is personalized: a higher value of λ means a more personalized model. If $\lambda = 1$, each client independently trains a local model with only the local data, i.e., extremely personalized models. $\lambda = 0$ corresponds to the original federated learning method, i.e., no personalization is involved.

Obviously, the interpolation coefficient λ is an important parameter and should be set appropriately. In the original work [26], the authors set λ per client by minimizing the empirical risk of the interpolated model, which is formulated as:

$$\begin{aligned} \lambda^t_i &= \arg\min_{\lambda} \ \ell(\boldsymbol{w'}^t_{l_i}, d_i) \\ &= \arg\min_{\lambda} \ \ell(\lambda \boldsymbol{w}^{t-1}_{l_i} + (1-\lambda)\boldsymbol{w}^{t-1}_g, d_i) \end{aligned} \tag{2}$$

However, this is not practical for real-world news recommendation tasks, as trying different λs would lead to high computational cost, considering that the news recommendation model may have tons of parameters. In practice, we usually need to treat it as a hyper-parameter to be tuned. In this work, we propose the fine-grained interpolation approach which selects the value of λ in a better way.

4 METHODOLOGY

In this section, we present the FINDING framework. First, we introduce the details of our fine-grained model interpolation method. Next, we describe the group-level personalization strategy in our framework. The workflow of FINDING is described in Fig. 4 and Alg. 1. In Appendix A, we extend our framework with Homomorphic Encryption (HE) to make it more privacy-preserving.

Since FINDING is a general personalized federated learning framework for news recommendation that focuses on the federated training process instead of the specific model structure, we do not impose any restrictions on the choice of the concrete news recommendation model structure. Any centralized news recommendation model following the pattern specified in Section 3.1 can be easily adapted to our federated training framework.

4.1 Fine-grained Model Interpolation

To satisfy the personalization needs at both the temporal and spatial levels, we propose the fine-grained model interpolation strategy, which includes two types of interpolation: time-aware and layer-aware interpolation.

4.1.1 Time-aware Interpolation. The federated training of a news recommendation model is a long-term process in which the importance of the global and local models changes over time. Thus the personalization needs of the local models vary at the temporal level. To this end, we design the *time-aware interpolation* mechanism. Specifically, at the beginning of the training, we give more weight to the global model for its good generalization. Then, as the training progresses, we gradually increase the interpolation weights of the local models. In this way, the interpolated model first learns the *commonality* as a good starting point. Then it gradually focuses on the *individuality* of the local data, being more personalized for the users. Formally, the personalization coefficient λ varies with the number of training rounds: the longer the training processes, the higher λ should be. This can be formulated as:

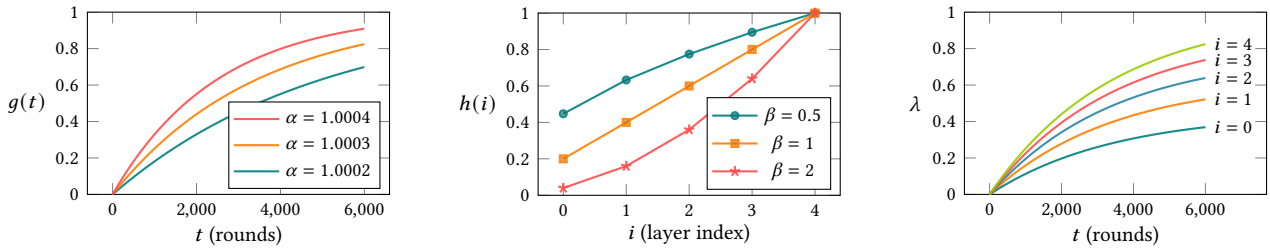
$$\lambda \propto g(t) : [0, \infty) \rightarrow [0, 1] \tag{3}$$

where t is the global training rounds and $g(t)$ is an increasing mapping function. Technically $g(t)$ can be any function that satisfies these properties. Here we formulate it empirically in the form of the exponential function:

$$g(t) = 1 - \alpha^{-t} \quad (\alpha > 1) \tag{4}$$

where α is a hyper-parameter. It controls how fast the λ increases with the training rounds t. A larger α means that λ will increase faster. Fig. 3a shows the graph with different values of α.

4.1.2 Layer-aware Interpolation. Since a news recommendation model typically has a hierarchical structure as shown in Fig. 2, the different layers in the model may have different roles and different personalization needs. To handle the different personalization needs at the spatial level, we design the *layer-aware interpolation* strategy. Specifically, for the lower layers, e.g., the layers in the news encoder of a news recommendation model, we tend to share the model parameters among the clients since the lower layers usually capture the data pattern from the raw input which reflects fewer personal preferences. In contrast, for the upper layers, e.g., the layers in the user encoder, we give more interpolation weights as they learn the user representations which involves more personalization. Formally, the personalization coefficient λ depends on the layer depth: the



(a) Time-aware interpolation: $\lambda \propto g(t)$ **(b) Layer-aware interpolation:** $\lambda \propto h(i)$ **(c)** $\lambda(t,i) = g(t)h(i)$

Figure 3: The illustration of the fine-grained model interpolation. It consists of two types of interpolation: time-aware and layer-aware interpolations. In these figures, $g(t) = 1 - \alpha^{-t}$, $h(i) = (\frac{i+1}{N})^{\beta} = (\frac{i+1}{5})^{\beta}$. **In (c),** $\alpha = 1.0003$, $\beta = 0.5$.

shallower a layer is, the higher λ should be. This is formulated as:

$$\lambda \propto h(i) : [0, N-1] \rightarrow [0, 1] \tag{5}$$

where N is the number of layers in the model. i is the layer index (0-indexed and starting from the bottom layer). $h(i)$ is another increasing mapping function. Similarly, we design it in the form of the power function as an example:

$$h(i) = (\frac{i+1}{N})^{\beta} \quad (\beta > 0) \tag{6}$$

where β is a hyper-parameter. It controls to what extent the λ for the upper layer is greater than that for the lower layer. A larger β would mean a larger difference in λ between the layers. Fig. 3b shows the graph with different values of β when $N = 5$.

4.1.3 Fine-grained Model Interpolation. Integrating the two types of interpolation, we propose the *fine-grained model interpolation* strategy. Specifically, we formulate λ as a function of the number of training rounds t and the layer index i:

$$\begin{aligned} \lambda(t,i) &= g(t)h(i) \\ &= (1-\alpha^{-t})(\frac{i+1}{N})^{\beta} \end{aligned} \tag{7}$$

Fig. 3c shows an example graph of λ at different training rounds for different layer indexes. It gives us a glimpse of how our method can satisfy the fine-grained personalization needs of model parameters: first, as the training progresses, all the model parameters are assigned increasing interpolation weights to satisfy the temporal level personalization needs; second, comparing the curves of the parameters at different layers, the upper layers have higher interpolation weights than the lower layers, thus satisfying the spatial level personalization needs.

4.2 Group-level Personalization with Dynamic User Clustering

To overcome the cold-user issue and improve the quality of the personalized models for interpolation, motivated by Clustered Federated Learning (CFL) [31], we propose the group-level personalization strategy. Specifically, instead of maintaining a user-level personalized model for each user, we cluster the users into groups based on their personal preferences and train the group-level personalized models shared by users in the same group. The later model interpolation process is the same, only now it is the group-level

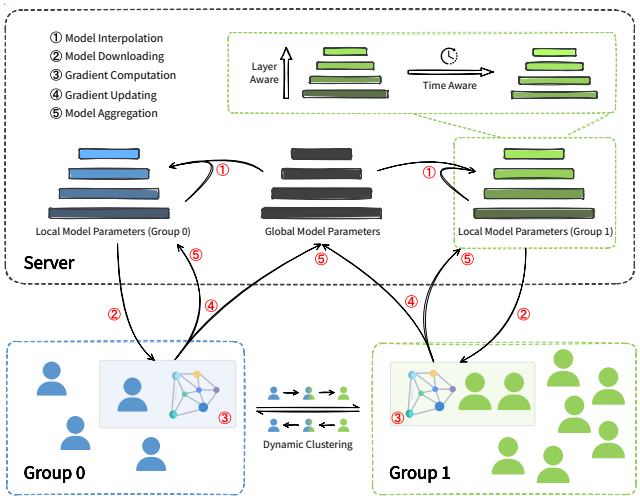


Figure 4: The framework of FINDING. The users are dynamically clustered into groups and the group-level personalized models are interpolated with the global model in a fine-grained way, i.e., both time-aware and layer-aware. The color of each layer in local model parameters indicates the degree of personalization. Step 1–5 demonstrate the process of a federated training round.

local models that are interpolated with the global model, instead of the user-level local models.

The current works in CFL literature mainly cluster the users by model parameters [42, 46] or gradients [31]. However, the size of a news recommendation model can be large and clustering over the parameters or gradients would impose an unacceptable computational cost. Unlike these methods, we cluster the users by user representations, which are much more efficient for clustering since they are just 1D vectors of length no more than a few hundred. Besides, they are directly learned from the interacted news representations and could naturally reflect user preferences.

Specifically, when the federated training process just starts, each client will download the global model from the server and run it on its own data to get the user vector representations. They will upload the vectors to the server for server-side clustering. Here we

Algorithm 1 The FINDING algorithm. $\boldsymbol{w}_{l_i}^t$ denotes the local model parameters of group i after round t and $\boldsymbol{w}_g^t$ represents the global model parameters. K is the number of groups. $\ell(\boldsymbol{w}, d_i)$ is the loss of model $\boldsymbol{w}$ on local data of user i, i.e., d_i. The users are clustered every T rounds. C is the number of users selected in each round.

Training

1: initialize $\boldsymbol{w}_g^0$
2: $\boldsymbol{w}_{l_0}^0, \boldsymbol{w}_{l_1}^0, \ldots, \boldsymbol{w}_{l_{K-1}}^0 \leftarrow \boldsymbol{w}_g^0$
3: $\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots \leftarrow \text{InferUserVector}(\boldsymbol{w}_g^0, \{d_0, d_1, \ldots\})$
4: $m \leftarrow \text{Cluster}(\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots)$ ▷ m maps users to groups
5: **for** each round $t = 1, 2, \ldots$ **do**
6: **for** each group $i = 0, 1, \ldots, K-1$ **do**
7: $\boldsymbol{w'}_{l_i}^t \leftarrow \lambda \boldsymbol{w}_{l_i}^{t-1} + (1-\lambda)\boldsymbol{w}_g^{t-1}$ ▷ λ from Eq. (7)
8: **end for**
9: $S_t \leftarrow$ (randomly select C users)
10: $\boldsymbol{w}_g^t \leftarrow \boldsymbol{w}_g^{t-1} - \eta \sum_{i \in S_t} \frac{|d_i|}{\sum_{k \in S_t} |d_k|} \nabla \ell(\boldsymbol{w'}_{l_{m(i)}}^t, d_i)$
11: **for** each group $i = 0, 1, \ldots, K-1$ **do**
12: $S_{t,i} \leftarrow \{j \in S_t \mid m(j) = i\}$
13: $\boldsymbol{w}_{l_i}^t \leftarrow \boldsymbol{w'}_{l_i}^t - \eta \sum_{j \in S_{t,i}} \frac{|d_j|}{\sum_{k \in S_{t,i}} |d_k|} \nabla \ell(\boldsymbol{w'}_{l_i}^t, d_j)$
14: **end for**
15: **if** $t \,\%\, T = 0$ **then** ▷ re-cluster periodically
16: $\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots \leftarrow \text{InferUserVector}(\boldsymbol{w}_g^t, \{d_0, d_1, \ldots\})$
17: $m \leftarrow \text{Cluster}(\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots)$
18: $\boldsymbol{w}_{l_0}^t, \boldsymbol{w}_{l_1}^t, \ldots, \boldsymbol{w}_{l_{K-1}}^t \leftarrow$ (right-hand side of Eq. (8))
19: **end if**
20: **end for**

Evaluation

1: **for** each user $i = 0, 1, \ldots$ in evaluation set **do**
2: **if** i exists in training set **then** ▷ has personalized model
3: $\text{Evaluate}(\boldsymbol{w}_{l_{m(i)}}, d_i)$
4: **else** ▷ has no personalized model, assign one
5: $\boldsymbol{u}_i \leftarrow \text{InferUserVector}(\boldsymbol{w}_g, \{d_i\})$
6: $g_i \leftarrow \text{FindClosestGroup}(m, \boldsymbol{u}_i)$
7: $\text{Evaluate}(\boldsymbol{w}_{l_{g_i}}, d_i)$
8: **end if**
9: **end for**

adopt the K-means [25] algorithm for clustering, leaving the more advanced clustering algorithms for future research, since K-means has shown satisfactory performance in our experiments. The group number K is a hyper-parameter that should be tuned according to the demographic characteristics of users in specific scenarios. Besides the global model, the server also maintains K local models for the groups, which are first initialized by copying the global model. In each round, the server interpolates the K local models as specified by Eq. (1a) and Eq. (7) (Alg. 1 Line 6–8). Then it randomly selects some online users from each group such that the number of selected users in a group is proportional to the group size. These selected users will download the interpolated models of the group they belong to and perform the normal back-propagation process. When their gradients are uploaded to the server, all the gradients are used (i.e., weighted average) to update the global model as specified by Eq. (1c) (Alg. 1 Line 10), while only the gradients from users of a group are used to update the local model of the corresponding group (Alg. 1 Line 11–14).

However, the initial clustering result may not be accurate since it is from user vectors computed by a randomly initialized global model. As training progresses, the user vector representations become more accurate in modeling users' personal preferences, and thus more suitable for clustering. For this reason, we run the clustering algorithm every T rounds to allow users to move between the groups, i.e., a *dynamic user clustering* mechanism (Alg. 1 Line 15–19). Note that while computing the user representations it is the global model that is used, instead of the local models of each group. Using the local personalized models may limit the potential clustering changes, since a specialized personalized model will also produce specialized user vectors.

Another point worth mentioning is that after each re-clustering, the correspondence between local models and users will change, so we need to update the local model parameters properly. The approach we take is to reinitialize the local models of the new groups as a linear combination of the old local models. Formally, we first define the *transition matrix* as a square matrix $\mathbf{M} \in \mathbb{N}^{K \times K}$ whose i-th row and j-th column element m_{ij} is the number of users who have moved from group i to group j on a re-clustering. Then the K local models are updated as follows:

$$\begin{bmatrix} \boldsymbol{w}_{l_0} \\ \boldsymbol{w}_{l_1} \\ \vdots \\ \boldsymbol{w}_{l_{K-1}} \end{bmatrix}^\intercal \leftarrow \begin{bmatrix} \boldsymbol{w}_{l_0} \\ \boldsymbol{w}_{l_1} \\ \vdots \\ \boldsymbol{w}_{l_{K-1}} \end{bmatrix}^\intercal \mathbf{M}\, \text{diag}\left(\sum_{i=1}^{K-1} m_{ij}\right)^{-1} \tag{8}$$

where $\boldsymbol{w}_{l_i}$ represents the local model parameters of group i ($0 \le i \le K-1$).

To illustrate what the formula does, here is a simple example: after the re-clustering, if the 80%, 10% and 10% of the users in the new group 0 are from the original group 0, group 1 and group 2, respectively, then the local model of group 0 is reinitialized as:

$$\boldsymbol{w}_{l_0} \leftarrow 0.8\,\boldsymbol{w}_{l_0} + 0.1\,\boldsymbol{w}_{l_1} + 0.1\,\boldsymbol{w}_{l_2} \tag{9}$$

After the model training, we use the local personalized models of each group, instead of the global model, for evaluation. For the user without the personalized model (i.e., a user that exists only in the evaluation set), we compute its user vector using the global model, assign it to a group that is closest to the user vector representation, and use the corresponding local model for inference.

5 EXPERIMENTS

5.1 Experimental Settings

5.1.1 Datasets. We conduct thorough experiments on two public datasets, namely Adressa[1] and MIND[2]. Adressa [12] is a public dataset released by a newspaper company in Norway, which includes news articles in Norwegian in connection with anonymized users. Following the previous works [15, 29], we construct historical clicks from the data of the first 5 days. The training set is built from the clicks of the 6th day. We randomly sample 20% of the clicks from

[1] https://reclab.idi.ntnu.no/dataset/
[2] https://msnews.github.io/. It has two versions and we use the smaller one for speed.

Table 1: Results of different methods on two datasets (in percent). The numbers in the parentheses are the p values of the t-test, where the alternative hypothesis is that FINDING performs better than the corresponding baseline.

		Adressa				MIND			
		AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
NRMS	Centralized	**72.67**	**29.39**	35.66	41.16	66.11	**31.59**	**34.76**	41.00
	Vanilla FL	71.13 (0.01)	26.10 (0.00)	32.03 (0.00)	37.49 (0.00)	65.04 (0.00)	30.78 (0.00)	33.58 (0.00)	40.04 (0.01)
	FedProx	71.25 (0.02)	27.30 (0.01)	32.55 (0.00)	38.33 (0.02)	65.14 (0.01)	30.49 (0.00)	33.42 (0.00)	39.75 (0.01)
	FedPer	71.39 (0.02)	27.64 (0.01)	34.02 (0.01)	39.17 (0.02)	65.43 (0.00)	31.03 (0.02)	34.06 (0.02)	40.41 (0.03)
	SCAFFOLD	71.50 (0.04)	27.66 (0.03)	34.59 (0.02)	39.28 (0.05)	65.48 (0.04)	30.81 (0.01)	33.95 (0.04)	40.26 (0.05)
	pFedMe	71.73 (0.05)	27.83 (0.04)	34.32 (0.01)	40.17 (0.02)	65.27 (0.01)	30.73 (0.00)	33.56 (0.01)	40.19 (0.04)
	CFL	71.60 (0.02)	27.79 (0.02)	34.62 (0.03)	40.04 (0.02)	65.32 (0.01)	30.92 (0.04)	33.80 (0.01)	40.39 (0.05)
	FINDING	72.51	28.89	**35.81**	**41.28**	**66.14**	31.30	34.62	**41.03**
NAML	Centralized	**80.44**	**33.79**	**42.16**	47.93	67.17	**31.88**	**35.30**	41.60
	Vanilla FL	78.71 (0.00)	32.84 (0.02)	41.04 (0.02)	46.75 (0.00)	66.01 (0.00)	30.96 (0.00)	34.38 (0.01)	40.70 (0.00)
	FedProx	78.69 (0.02)	33.26 (0.03)	41.74 (0.04)	47.01 (0.00)	66.15 (0.00)	31.16 (0.01)	34.41 (0.01)	40.66 (0.01)
	FedPer	79.01 (0.02)	33.11 (0.02)	41.88 (0.05)	47.43 (0.04)	66.78 (0.03)	31.56 (0.05)	34.92 (0.04)	41.02 (0.02)
	SCAFFOLD	79.44 (0.05)	33.22 (0.04)	41.34 (0.02)	47.15 (0.01)	66.42 (0.03)	31.37 (0.02)	34.69 (0.03)	40.94 (0.03)
	pFedMe	79.17 (0.04)	32.98 (0.03)	41.73 (0.03)	47.68 (0.03)	66.16 (0.01)	31.41 (0.05)	34.28 (0.00)	40.57 (0.00)
	CFL	79.44 (0.04)	33.12 (0.04)	41.60 (0.04)	47.58 (0.03)	66.23 (0.00)	31.25 (0.01)	34.50 (0.01)	40.94 (0.01)
	FINDING	80.35	33.59	42.13	**48.06**	**67.26**	31.85	35.19	**41.64**

the last day's data for validation and the rest for testing. MIND [40] is a news recommendation dataset collected from anonymized behavioral logs of Microsoft News[3] website. It contains approximately 160k English news articles and more than 15 million impression logs generated by 1 million users.

5.1.2 Compared Methods. Since our FINDING framework is a novel personalized federated training framework, we compare it with the traditional centralized training method and the existing personalized federated training methods. Specifically, we choose the following baseline methods:

- **Centralized** denotes the plain centralized training method.
- **Vanilla FL** [27] is the vanilla adaptation of federated learning to news recommendation tasks.
- **FedProx** [21] addresses the heterogeneity issue with a proximal term that adjusts local model updates.
- **FedPer** [4] trains the base layers of a deep model centrally, while the top layers (i.e., the personalization layers) are trained locally. In our experiments, we consider the layers in news encoders as base layers and those in user encoders as top layers.
- **SCAFFOLD** [17] proposes to tackle the client drift problem in federated learning with control variates.
- **pFedMe** [9] makes use of the Moreau envelope function which helps decompose the personalized model optimization from global model learning.
- **CFL** [31] iteratively splits the users into groups based on the similarity of the gradient updates.

Then for each method, we select the following base news recommendation models for instantiation:

- **NRMS** [38] uses both Multi-head Self-attention and Additive Attention to learn users and news representations.
- **NAML** [37] is a multi-view model to learn unified news representations from news titles, bodies and categories.

For a fair comparison, we always compare the methods with the same news recommendation model. The code is available at https://github.com/yusanshi/FINDING.

5.1.3 Hyper-parameters Settings. After searching the value of α and β in Eq. (7), the interpolation function we use is $\lambda(t,i) = g(t)h(i) = (1-1.0003^{-t})(\frac{i+1}{N})^{0.5}$. We randomly select 50 users in each round. The number of groups for clustering, i.e., K, is searched in $\{2,4,8,\ldots,64\}$ and was finally set to 8 as a trade-off between the performance and computational cost. We run the clustering algorithm every 500 rounds. The Adam [18] optimizer is used and the learning rate is set to 0.0001. The dimension of the user vectors and news vectors is 300.

5.2 Performance Comparison

Following the previous works [38, 40, 45], we choose these evaluation metrics: AUC, MRR, nDCG@5, and nDCG@10. We repeat each experiment 5 times and report the average results. The results are shown in Table 1, where we can observe that FINDING significantly outperforms the vanilla federated learning method and achieves a comparable performance to the traditional centralized training methods. We attribute this to the model personalization in our work, which can effectively handle the data heterogeneity issue and achieve the performance gain. Moreover, we find that FINDING outperforms other PFL methods and the improvement is significant ($p < 0.05$). This is probably because FINDING can provide more fine-grained and complete personalization than them. For example, compared to FedPer, FINDING has the group-level

[3] https://microsoftnews.msn.com/

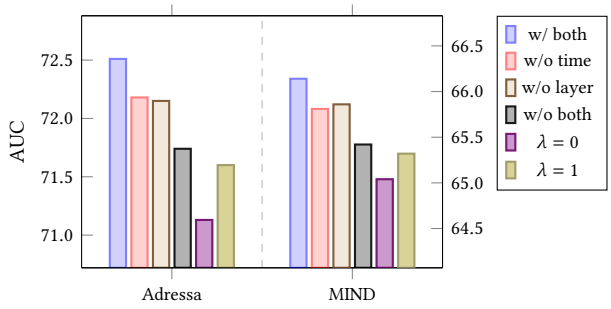


Figure 5: Ablation study of the fine-grained interpolation. *time* and *layer* denote the time-aware interpolation and layer-aware interpolation, respectively.

personalization which can alleviate the cold-user problem and help to train better local personalized models; compared to CFL, FINDING has the fine-grained model interpolation mechanism which ensures that the local personalized models are not over-specialized. In short, FINDING keeps all the privacy benefits of the federated learning architecture with almost no performance degradation.

5.3 Further Analysis

5.3.1 Ablation Study of Fine-grained Model Interpolation. The fine-grained model interpolation in our framework involves two types of interpolation. Specifically, the interpolation coefficient λ is formulated as $\lambda(t,i) = g(t)h(i) = (1-\alpha^{-t})(\frac{i+1}{N})^{\beta}$, where $g(t)$ is the time-aware interpolation and $h(i)$ is the layer-aware interpolation. To analyze the effectiveness of the two types of interpolation, we conduct an ablation study by removing them separately and simultaneously. For a fair comparison, when one is removed, the hyper-parameter for the other is retuned; when both are removed, λ becomes a fixed value and is searched in $[0, 1]$. Moreover, we add the experiments where λ is 0 and 1 (in fact, if $\lambda = 0$ FINDING will degenerate to Vanilla FL). NRMS is selected as the base news recommendation model to instantiate the FINDING method.

The results are shown in Fig. 5, from which we have some observations. First, the results of the first four experiments suggest that both time-aware and layer-aware interpolation are playing a positive role. Second, when the interpolation coefficient λ is a fixed value (the last three experiments), we find that an elaborately searched λ (the fourth experiment) does achieve better performance. This indicates the effectiveness of the model interpolation strategy. However, the fixed λ in the plain model interpolation method neglects the fine-grained interpolation needs, which are exactly what FINDING is trying to fulfill.

5.3.2 Visualization of Dynamic User Clustering. To further investigate how the dynamic user clustering works, we randomly select 400 users and visualize their group changes on each re-clustering in Fig. 6. From the figure, we can observe lots of group changes in the first few clusterings. However, as time goes on, there are fewer and fewer users moving among groups. This demonstrates that the clustering results are becoming more and more stable. In other words, the user vectors are becoming more and more accurate in modeling user preferences, since the data source for clustering is

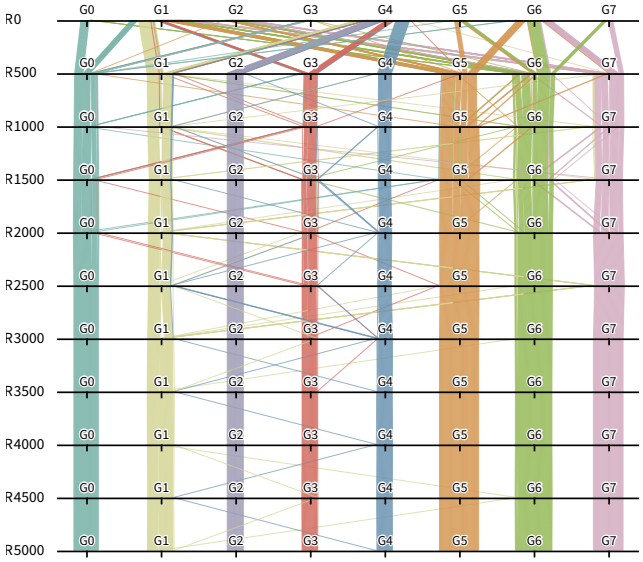


Figure 6: The visualization of the user clustering changes. Gi denotes the i-th group and Rj means round j. Each line represents a user's group change history. The lines are colored by the groups the users are finally in. From top to bottom, the bands of the lines show the flow of users between groups.

the user vectors inferred from the global model. Then, if we cluster only once in the beginning, the clustering result will remain inaccurate because of the inaccurate user vectors; if we cluster only once after some time (e.g., on model convergence), the clustering result would be accurate but we could not benefit from the model personalization before clustering. In conclusion, periodical clustering is necessary for accurate user clustering, which will help to learn a good personalized model.

5.3.3 Connection with Other Methods. Thanks to the great flexibility introduced by the parameter λ (the interpolation coefficient) and K (the number of groups for clustering), some existing methods can be seen as degenerate cases of FINDING if the parameters satisfy certain conditions:

- $\lambda = 0$ or $K = 1$: means Vanilla FL, since no personalization is involved and all users share the same model parameters.
- $\lambda = 1$: corresponds to Clustered Federated Learning (CFL) method, although there is a difference in the clustering mechanism, i.e., *iterative clustering* in CFL and *dynamic clustering* in FINDING.
- $\lambda = h(i) = \mathbb{1}_{\{x|x\geq X\}}(i)$, K = #users: degenerates to FedPer [4], where the base layers of a deep model are trained centrally while the top layers are trained locally.
- $\lambda = g(t) = \mathbb{1}_{\{x|x\geq X\}}(t)$, K = #users: is the fine-tuning approach by Wang et al. [36], where the central model is fine-tuned on the local data.

In summary, FINDING can be seen as a generalization of some existing methods. It provides more granular control over the model personalization, which can explain its superiority over the degenerate methods.

6 CONCLUSION

In this paper, we investigated the problem of how to address the data heterogeneity issue, namely the non-IID data problem, in federated news recommendation tasks. As a solution, we proposed FINDING, a novel personalized federated learning framework based on model interpolation for news recommendation. We first proposed the fine-grained model interpolation strategy, which is both time-aware and layer-aware. It can satisfy the fine-grained personalization needs of model parameters. Then, to tackle the cold-user problem and learn a better personalized model, we adopted the group-level personalization approach by dynamically clustering users into groups and using the group-level personalized models for interpolation. Our FINDING framework can be seen as a generalization of some existing PFL methods but with enhanced personalization. We conducted extensive experiments on real-world datasets which show that our method can effectively handle the limitations of the existing PFL methods and alleviate the data heterogeneity issue in federated news recommendation settings.

ACKNOWLEDGMENTS

This research was supported by grant from the National Key Research and Development Program of China (No. 2021YFF0901003).

A PRIVACY PRESERVING WITH HOMOMORPHIC ENCRYPTION

In this section, we present FINDING-HE, an extension of FINDING with Homomorphic Encryption (HE) for better privacy preserving.

A.1 Introducing Homomorphic Encryption

Although federated learning can effectively protect user privacy as the raw data are not uploaded to the server, the updated gradients can still leak privacy [11]. Also, in our framework, the user vector representations are uploaded for clustering, which may lead to privacy concerns. Thanks to the development of cryptography, our framework can be further equipped with the Homomorphic Encryption (HE) [10] technique for better privacy protection. It is a form of encryption that supports performing computations on encrypted data without first decrypting it. Specifically, an encryption scheme is said to be *homomorphic* if the following equation holds:

$$D(E(m_1) \oplus E(m_2)) = m_1 \otimes m_2, \ \forall m_1, m_2 \in M \tag{10}$$

where E is the encryption algorithm and D is the corresponding decryption algorithm. M is the set of all possible messages. $\oplus$ and $\otimes$ are the operators [1]. If for any $\otimes$, there exists a combination of D, E and $\oplus$ such that the above equation holds, the scheme is also known as Fully Homomorphic Encryption (FHE). With it, the following is possible:

$$D(f(E(\boldsymbol{v}_1), E(\boldsymbol{v}_2), \ldots, E(\boldsymbol{v}_n))) = g(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n) \tag{11}$$

where $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ are the client-side data. g is the arbitrary computation we want and f is the corresponding computation performed on the encrypted data on the server side. Thus, the users can first encrypt the data before uploading. Then the server performs the corresponding computation on the encrypted data such that the users can download the results, decrypt them and get what they want. For averaging the gradients, $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ would be the gradients and g is the weighted averaging operation. For clustering the users, $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ are the user vectors and g is the clustering algorithm that returns the group identity of each user.

Algorithm 2 The FINDING-HE algorithm. p_k and s_k denote the public and secret keys, respectively. E and D are the encryption and decryption algorithms, respectively. The other symbols are the same as in Alg 1. Please refer to FedMF [6] for details not described here (e.g., key generation) since the adoption of Homomorphic Encryption in FINDING-HE is similar to that in FedMF.

initialize $\boldsymbol{w}_g^0$
initialize p_k and s_k
$\boldsymbol{w}_g^0 \leftarrow E(\boldsymbol{w}_g^0, p_k)$
$\boldsymbol{w}_{l_0}^0, \boldsymbol{w}_{l_1}^0, \ldots, \boldsymbol{w}_{l_{K-1}}^0 \leftarrow \boldsymbol{w}_g^0$
$\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots \leftarrow$ InferUserVector$(D(\boldsymbol{w}_g^0, s_k), \{d_0, d_1, \ldots\})$
$m \leftarrow$ Cluster-HE$(\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots)$ ▷ m maps users to groups
for each round $t = 1, 2, \ldots$ **do**
  for each group $i = 0, 1, \ldots, K-1$ **do**
    $\boldsymbol{w'}_{l_i}^t \leftarrow \lambda \boldsymbol{w}_{l_i}^{t-1} + (1-\lambda)\boldsymbol{w}_g^{t-1}$ ▷ λ from Eq. (7)
  end for
  $S_t \leftarrow$ (randomly select C users)
  $\boldsymbol{w}_g^t \leftarrow \boldsymbol{w}_g^{t-1} - \eta \sum_{i \in S_t} \frac{|d_i|}{\sum_{k \in S_t} |d_k|} E(\nabla \ell(D(\boldsymbol{w'}_{l_{m(i)}}^t, s_k), d_i), p_k)$
  for each group $i = 0, 1, \ldots, K-1$ **do**
    $S_{t,i} \leftarrow \{j \in S_t \mid m(j) = i\}$
    $\boldsymbol{w}_{l_i}^t \leftarrow \boldsymbol{w'}_{l_i}^t - \eta \sum_{j \in S_{t,i}} \frac{|d_j|}{\sum_{k \in S_{t,i}} |d_k|} E(\nabla \ell(D(\boldsymbol{w'}_{l_i}^t, s_k), d_j), p_k)$
  end for
  if $t \,\%\, T = 0$ **then** ▷ re-cluster periodically
    $\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots \leftarrow$ InferUserVector$(D(\boldsymbol{w}_g^t, s_k), \{d_0, d_1, \ldots\})$
    $m \leftarrow$ Cluster-HE$(\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots)$
    $\boldsymbol{w}_{l_0}^t, \boldsymbol{w}_{l_1}^t, \ldots, \boldsymbol{w}_{l_{K-1}}^t \leftarrow$ (right-hand side of Eq. (8))
  end if
end for

A.2 Implementation of FINDING-HE

The adoption of Homomorphic Encryption in FINDING-HE is similar to that in FedMF [6]. We utilize a Python package, *python-paillier* library[4] for the linear operations (e.g., averaging the gradients). The K-means clustering algorithm with homomorphic encryption is implemented following the work of Wu et al. [41]. Since the focus of our work is not privacy protection, we leave the workflow of FINDING-HE in Alg. 2.

Due to the cryptographic property of HE, the integration of HE will keep the computational results unchanged (or at a certain precision) [7]. It increases the computational cost, but also enhances privacy protection. In our experiments, FINDING-HE produces an almost indistinguishable result from FINDING in terms of the evaluation metrics, and consumes about 10 times more training time. In practice, we may need to make trade-offs between privacy protection and computational efficiency based on specific needs.

[4] https://github.com/data61/python-paillier

REFERENCES

[1] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Comput. Surv.* (2018).

[2] Alekh Agarwal, John Langford, and Chen-Yu Wei. 2020. Federated Residual Learning. *CoRR* (2020). arXiv:2003.12880

[3] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long- and Short-term User Representations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*.

[4] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated Learning with Personalization Layers. *CoRR* (2019). arXiv:1912.00818

[5] Trapit Bansal, Mrinal Kanti Das, and Chiranjib Bhattacharyya. 2015. Content Driven User Profiling for Comment-Worthy Recommendations of News and Blog Articles. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*.

[6] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2021. Secure Federated Matrix Factorization. *IEEE Intell. Syst.* (2021).

[7] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. 2017. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*.

[8] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2020. Adaptive Personalized Federated Learning. *CoRR* (2020). arXiv:2003.13461

[9] Canh T. Dinh, Nguyen Hoang Tran, and Tuan Dung Nguyen. 2020. Personalized Federated Learning with Moreau Envelopes. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[10] Caroline Fontaine and Fabien Galand. 2007. A Survey of Homomorphic Encryption for Nonspecialists. *EURASIP J. Inf. Secur.* (2007).

[11] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting Gradients - How easy is it to break privacy in federated learning?. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

[12] Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The Adressa dataset for news recommendation. In *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*.

[13] Filip Hanzely and Peter Richtárik. 2020. Federated Learning of a Mixture of Global and Local Models. *CoRR* (2020). arXiv:2002.05516

[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 2017, Perth, Australia, April 3-7, 2017*.

[15] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph Neural News Recommendation with Unsupervised Preference Disentanglement. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*.

[16] Zhenya Huang, Binbin Jin, Hongke Zhao, Qi Liu, Defu Lian, Bao Tengfei, and Enhong Chen. 2023. Personal or general? a hybrid strategy with multi-factors for news recommendation. *ACM Transactions on Information Systems* (2023).

[17] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020*.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Joseph A. Konstan, Bradley N. Miller, David A. Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. 1997. GroupLens: Applying Collaborative Filtering to Usenet News. *Commun. ACM* (1997).

[20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* (2020).

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*.

[22] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI 2010, Hong Kong, China, February 7-10, 2010*.

[23] Qi Liu, Jinze Wu, Zhenya Huang, Hao Wang, Yuting Ning, Ming Chen, Enhong Chen, Jinfeng Yi, and Bowen Zhou. 2023. Federated User Modeling from Hierarchical Information. *ACM Trans. Inf. Syst.* (2023).

[24] Ruixuan Liu, Fangzhao Wu, Chuhan Wu, Yanlin Wang, Yang Cao, Lingjuan Lyu, Weike Pan, Yun Chen, Hong Chen, and Xing Xie. 2022. PrivateRec: Differentially Private Training and Serving for Federated News Recommendation. *CoRR* (2022). arXiv:2204.08146

[25] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* (1982).

[26] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. 2020. Three Approaches for Personalization with Applications to Federated Learning. *CoRR* (2020). arXiv:2002.10619

[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*.

[28] Daniel W. Peterson, Pallika Kanani, and Virendra J. Marathe. 2019. Private Federated Learning with Domain Adaptation. *CoRR* (2019). arXiv:1912.06733

[29] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-Preserving News Recommendation Model Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*.

[30] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2021. UniFedRec: A Unified Privacy-Preserving News Recommendation Framework for Model Training and Online Serving. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic*.

[31] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems* (2020).

[32] Hyejin Shin, Sungwook Kim, Junbum Shin, and Xiaokui Xiao. 2018. Privacy Enhanced Matrix Factorization for Recommendation with Local Differential Privacy. *IEEE Trans. Knowl. Data Eng.* (2018).

[33] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2021. Towards Personalized Federated Learning. *CoRR* (2021). arXiv:2103.00710

[34] Michele Trevisiol, Luca Maria Aiello, Rossano Schifanella, and Alejandro Jaimes. 2014. Cold-start news recommendation with domain-dependent browse graph. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*.

[35] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*.

[36] Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. 2019. Federated Evaluation of On-device Personalization. *CoRR* (2019). arXiv:1910.10252

[37] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*.

[38] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

[39] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering News Recommendation with Pre-trained Language Models. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*.

[40] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*.

[41] Wei Wu, Jian Liu, Huimei Wang, Jialu Hao, and Ming Xian. 2020. Secure and efficient outsourced k-means clustering using fully homomorphic encryption with ciphertext packing technique. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[42] Ming Xie, Guodong Long, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. 2020. Multi-Center Federated Learning. *CoRR* (2020). arXiv:2005.01026

[43] Jingwei Yi, Fangzhao Wu, Chuhan Wu, Ruixuan Liu, Guangzhong Sun, and Xing Xie. 2021. Efficient-FedRec: Efficient Federated Learning Framework for Privacy-Preserving News Recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*.

[44] Yang Yu, Fangzhao Wu, Chuhan Wu, Jingwei Yi, and Qi Liu. 2022. Tiny-NewsRec: Effective and Efficient PLM-based News Recommendation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*.

[45] Chuang Zhao, Hongke Zhao, Ming He, Jian Zhang, and Jianping Fan. 2023. Cross-domain recommendation via user interest alignment. In *Proceedings of the ACM Web Conference 2023*. 887–896.

[46] Fengpan Zhao, Yan Huang, Akshita Maradapu Vera Venkata Sai, and Yubao Wu. 2020. A Cluster-based Solution to Achieve Fairness in Federated Learning. In *IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking, ISPA/BDCloud/SocialCom/SustainCom 2020*.