



FedNRM: A Federal Personalized News Recommendation Model Achieving User Privacy Protection

Shoujian Yu¹, Zhenchi Jie¹, Guowen Wu¹, Hong Zhang¹ and Shigen Shen^{2,*}

¹School of Computer Science and Technology, Donghua University, Shanghai, 201620, China

²School of Information Engineering, Huzhou University, Huzhou, 313000, China

*Corresponding Author: Shigen Shen. Email: shigens@zjhu.edu.cn

Received: 24 February 2023; Accepted: 17 April 2023; Published: 23 June 2023

Abstract: In recent years, the type and quantity of news are growing rapidly, and it is not easy for users to find the news they are interested in the massive amount of news. A news recommendation system can score and predict the candidate news, and finally recommend the news with high scores to users. However, existing user models usually only consider users' long-term interests and ignore users' recent interests, which affects users' usage experience. Therefore, this paper introduces gated recurrent unit (GRU) sequence network to capture users' short-term interests and combines users' short-term interests and long-term interests to characterize users. While existing models often only use the user's browsing history and ignore the variability of different users' interest in the same news, we introduce additional user's ID information and apply the personalized attention mechanism for user representation. Thus, we achieve a more accurate user representation. We also consider the risk of compromising user privacy if the user model training is placed on the server side. To solve this problem, we design the training of the user model locally on the client side by introducing a federated learning framework to keep the user's browsing history on the client side. We further employ secure multi-party computation to request news representations from the server side, which protects privacy to some extent. Extensive experiments on a real-world news dataset show that our proposed news recommendation model has a better improvement in several performance evaluation metrics. Compared with the current state-of-the-art federated news recommendation models, our model has increased by 0.54% in AUC, 1.97% in MRR, 2.59% in nDCG@5%, and 1.89% in nDCG@10. At the same time, because we use a federated learning framework, compared with other centralized news recommendation methods, we achieve privacy protection for users.

Keywords: News recommendation; federal learning; privacy protection; personalized attention



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Nowadays, with the popularity of the Internet, the spread of online news is particularly rapid, and various online news websites and news APPs are popping up [1]. Users are facing a huge amount of information every day, and in the case of information explosion, a news recommendation system plays an urgent role, which can make personalized recommendation according to user's browsing history, provide thousands of people with thousands of news recommendations, and improve user experience [2,3].

Deep learning methods are now widely used in the field of natural language processing (NLP), which can characterize text information and extract complex information from text. In most applications, news is always presented as text, and because of this, many research methods have also applied deep learning to the field of news recommendation. Most of these methods use deep learning methods to extract features such as headlines and contents of news, characterize each news item as a vector, then model users based on their browsing news records, and finally estimate the rating for each candidate news item. For example, Wu et al. [4] proposed a news recommendation model named NPA with a personalized attention mechanism, in which a convolutional neural network (CNN) is first used to process a sequence of news headline words and extract the semantic feature of the words by capturing the local context of the words in the news headline [5]. Then based on the idea that different words have different amounts of information when representing news, a personalized attention model is designed to identify important words for different users. Wu et al. [6] proposed an attention-based multi-view learning model called NAML, in addition to using the CNN in the NPA model combined with the attention model in modeling news headlines, this model also considers the subject and category perspectives of news, and uses the attention model to learn the different levels of importance that different perspectives have in modeling different news to obtain the final news representation. Wu et al. [7] proposed a method named NRMS based on the multi-head self-attention mechanism. In the news model, the multi-head self-attention mechanism is used to learn the contextual representation of words, capture the interaction of words at long distances in the text, and use the attention mechanism to select the more important words in the sentence for news representation. With the successful use of some pre-trained language models in natural language processing [8–11], there is a growing amount of research in the field of news recommendation that combines pre-trained models with its own methods, as news usually appears in the form of text. For example, Wu et al. [12] proposed a new news recommendation method with pre-trained language models called PLM-NR to enhance news representation. In this paper, we first capture the deep contextual features of the news using a pre-trained language model (PLM), and then pool the output of the PLM using an attention network to finally obtain a representation of the news. For user models, most of them model users based on their browsing history. For example, Okura et al. [13] proposed to capture user's click history with gated recurrent unit (GRU) network to learn user's embedding. Wu et al. [7] proposed capturing user features from news browsing history using a multi-head self-attention mechanism combined with an attention mechanism in the NRMS model. However, in real life, a user may be a basketball fan, and most of the news he clicks on are sports news, but at a certain period of time, for example, during the U.S. election, even people who are not too interested in politics may pay attention accordingly, so the short-term interest of users should also be taken into account when making news recommendations [14]. In this paper, we capture the long-term preferences of users using the multi-head self-attention layer and the personalized attention layer, capture the short-term preferences of users using the GRU network, and then combine the two preferences of users to obtain their representations.

However, all the methods mentioned above require centralized storage of user's browsing behavior, and user's browsing history belongs to user's privacy, which will result in infringement of user's rights if

user's browsing history is stolen from the server [15,16]. And with the recent increased focus on cyber security around the world, we may not have centralized access to users' data in the future [17]. The federated learning framework is a popular direction recently used to solve user privacy problems in applications [18,19], which of course includes the recommendation system domain [20]. It proposes that the user's information can be saved locally and only a copy of the global recommendation model needs to be saved on the client side. The basic idea of the federated framework is that the client trains locally using local privacy data, and sends the training loss gradient to the server, which aggregates the gradient to update the global model [21]. It then sends the global model with updated parameters to each client, which trains on the updated model. Finally it converges after several iterations. Both [22] and [23] use this federal framework, but both approaches ignore the problem of differentiated attention to news by different users in the user encoder.

Overall, the existing centralized news recommendation models do not take into account the user's privacy data sensitivity problem, while some recent distributed news recommendation models ignore many issues affecting the recommendation accuracy when building news recommendation models, and fail to achieve a better recommendation effect. The federal news recommendation model proposed in this paper not only achieves a better recommendation effect and fully considers the user's personalized recommendation problem, but also protects the user's privacy to a certain extent.

In addition to that, considering the communication load brought by federal learning, in this work, a larger news model is maintained on the server side, a shared lightweight user model is maintained on the server and the client, and the user retains only a portion of the news representation model on the client side. With this model distribution we can reduce communication costs [24]. Based on secure multi-party computation, this paper exchanges a set of news representations involved in a group of user behavior when the client acquires the news representation, hiding the click history of a single user, which makes the security of a single client node guaranteed [25].

The main contributions of this work include:

- (1) In terms of user model, we utilize user ID embeddings to achieve a personalized attention mechanism, thus better capturing the users' long-term interests. We also use GRU network to capture users' short-term interests and combine users' long-term interests with short-term interests to finally get more accurate user representation. Our user interest modeling approach solves two problems simultaneously. One is the problem of differential attention of different users to the same news, and the other is the problem of user's interest shifting over time.
- (2) In terms of news model, we use pre-trained language models as well as attention models to learn contextual information of news headline texts and obtain better news representations. Our news modeling approach solves the problem of different meanings of the same word due to changes in word order in different sentences, or changes in the context of different sentences.
- (3) We introduce a privacy-preserving scheme that protects user privacy while making news recommendations based on a federal framework and incorporating secure multi-party computing.
- (4) We conducted extensive experiments on real datasets to verify the effectiveness of the proposed method.

Next, we show the current related work in the field related to the paper in Section 2, and in Section 3 we detail the model proposed in this paper, which contains two modules: the news recommendation model and the federal news recommendation framework. In Section 4 we show the experiments performed based on the model proposed in Section 3. In the last section we conclude the paper.

2 Related Works

2.1 News Recommendation

The news recommendation problem can be formulated as a problem of matching the candidate news set (i.e., a news set in an impression) with the users' interests. News recommendation models contain three main tasks, namely, how to model the contents of news articles, how to model users' interests and how to represent the correlation between news contents and users' interests. In terms of news representation, many recent deep learning-based news recommendation methods use neural networks to represent the contents of news. For example, Okura et al. [13] used denoising autoencoder to learn news embedding from the textual contents of the news. An et al. [26] proposed to learn contextual word embeddings using CNN model and to select important words for representation into sentence embeddings using an attention layer. Tran et al. [27] proposed to learn the representation of news from multiple attribute features of news, where the attributes contain categories, subcategories, headlines, and knowledge entities in the news datasets, and it applied a self-attention mechanism in extracting features. In terms of user representation, Wu et al. [6] used an attention network to encode user representation from users' click history. It turns out that short-term interest is also important [28], and An et al. [26] proposed a long-term and short-term user representation model (LSTUR) to obtain users' long-term interests through their ID embeddings, combined with GRU to obtain short-term user interests. Hu et al. [29] proposed a graphical neural news recommendation model that constructs graphs and learns the user's embeddings using graph convolutional networks to obtain higher-order features of the user, which also takes into account the user's temporary preferences and combines the long-term and short-term interests of the user. However, these news recommendation models mentioned above need to be trained by storing users' behaviors on the central server of the news platform, but this behavior data may be sensitive to the user. Therefore, there are many recent studies applying privacy protection methods to recommendation systems to alleviate the above problem. News recommendation also belongs to one of the recommendation domains, and recently there are some studies that combine news recommendation with federated learning. For example, Qi et al. [22] proposed a privacy-preserving approach called FedRec for news recommendation model training, using local data to compute model parameters and applying local differential privacy methods to add some noise to the loss gradients and uploading them to the server to update the global model. Yi et al. [23] proposed an efficient federated recommendation framework called Efficient-FedRec that decomposes the news recommendation model into two parts: a large news model unique to the server, and a small user model and news representation model, which is shared by the server and the client. It also applies the idea of secure aggregation to upload gradients, which reduces the communication cost of the client while protecting privacy.

We analyze and compare some of the news recommendation methods mentioned in this paper and summarize the weaknesses of each news recommendation model, as detailed in Appendixes A and B.

2.2 Federated Learning

Federated learning, as described in [30], is a type of distributed machine learning that can be used to protect the privacy of users. It enables model training to be performed on the user's device, keeping sensitive data with the owner of the data, collected locally, trained locally, and the server does not access the actual sensitive data [31]. Each user uploads the gradient of model training to the server, which performs aggregation of the gradient and global model update, and effectively reduces the risk of privacy leakage since the model update mentions contain less user information. Based on this, the federated learning framework has been applied to the training of some models in smartphones [32].

Recently, there are some studies that apply federal learning to the recommendation domain, for example, Ammad et al. [33] proposed federated collaborative filtering (FCF). FCF keeps the user matrix locally on the client side and the item matrix on the server side. The server side updates the item matrix and sends it to the client side, and the client side calculates it based on the updated item matrix and the local user matrix, and finally passes it back to the server side. This approach is very representative and can be extended to other recommendation system areas. Lin et al. [34] proposed a federated recommendation framework for explicit data. In addition to uploading the set of items for user interactions, it proposes that some other items can be randomly sampled and mixed into the set of interacting items for uploading, making the server unable to accurately identify the user's preferences, thus protecting the user's privacy. However, this method adds some noise to the gradient of the model, which reduces the recommendation accuracy. Therefore Liang et al. [35] proposed a denoised recommendation model. It proposes to use a randomly sampled denoising clients to collect the noisy gradients from the ordinary clients, then generate their own gradients and upload them, and the server side collects the encrypted gradients from both clients to reduce the sum of the original gradients. Using this method the server side can mitigate the noise of the gradients. Wu et al. [36] proposed the FedGNN model, which applies the classical federated learning framework to graph neural network recommendation systems where each user stores a local user-item graph locally, and it proposes a user-item graph extension method to find neighboring users with common interactions and exchange their embeddings to extend the local user-item graph without compromising privacy. Wu et al. [37] argued that not all of the user's local data are sensitive and require different levels of data protection, and it proposes using hierarchical information to classify data as private and public to achieve privacy heterogeneity, and that personalized update policies can be implemented on the client side to achieve statistical heterogeneity.

2.3 Secure Multi-Party Computation

The relationship between multi-party computing (MPC) and federated learning is verified in [38], and it proposes that a practical solution for federated learning also belongs to an instance of secure multi-party computing. Secure multi-party computing (SMPC) was proposed by MingZhi Yao in 1982, which refers to the use of private data to participate in confidential computing without revealing their own private data, and to jointly accomplish a certain computing task. In this way, the attacker cannot determine to which party the message belongs [39]. This technology can meet the demand for confidential computation using private data and effectively solve the conflict between "confidentiality" and "sharing" of data. A secure multi-party computation protocol is a distributed protocol that allows multiple untrusted participants to jointly compute an objective function on their own private inputs, so that each participant can compute a result through a defined logic without revealing their own private information [40,41].

There has also been some recent research applying secure multi-party computing to distributed machine learning as a means of securing users' data. For example, when users request news representations, Yi et al. [23] proposed to compute the joint news set of a group of users and request the news representation of the joint news set from the server to avoid privacy leakage of individual users. In addition, Bonawitz et al. [42] proposed a practical secure aggregation protocol for privacy-preserving machine learning based on secure multi-party computation, and this approach can be used for model updating in federal learning. In this paper, we also apply a secure aggregation approach to pass the gradients from the client to the server side.

3 Methodology

In this section, we introduce FedNRM, a federal news recommendation approach for protecting user privacy. We first introduce an improved personalized news recommendation model in the first part, detailing the news encoder, user encoder, and click predictor, respectively. In the second part, we consider the user's privacy issue by combining an efficient federal learning recommendation framework, keeping the user's browsing history locally, training the user model locally, and incorporating a secure multi-party computation method to protect the user's private data. At the same time, we keep the large news model on the server side and only part of the news representation model on the client side to reduce the communication cost.

3.1 Improved News Recommendation Module

In this subsection, we will introduce the improved news recommendation module used in this paper, which contains three parts: news encoder, user encoder and click score predictor. The general framework of the news recommendation model is shown in Fig. 1.

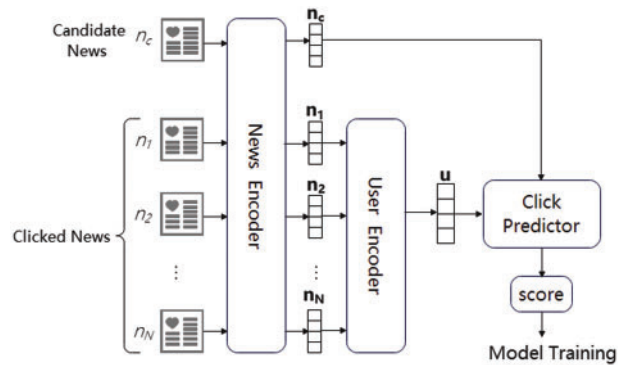


Figure 1: General news recommendation framework

3.1.1 News Encoder

The role of a news model is that given a news item, the news contents contain elements such as category, subcategory, title, abstract and news body, and the news encoder takes one or more of the elements as input from which the news representation is learned. The specific composition of the news model is shown in Fig. 2, where the pre-trained model bidirectional encoder representations from transformers (BERT) is used, and the headline of the news is used as the input, and the output of the BERT model is used as the input of the additive attention network to finally obtain the representation of the news.

The structure of pre-trained language model used in this paper is shown in Fig. 3. The input consists of token embedding, segment embedding and position embedding, which contain different features of the sentence. The BERT model transforms each word in the title sentence into its token embedding, combines the semantic segmentation embedding and the position embedding as the input to the BERT model, and then learns the representation of the word through several encoder layers of the transformer [43]. The encoder layer acts as a feature extractor and uses the accompanying mask language model (MLM) training method to achieve bidirectional encoding of the input sequence text. In addition, the BERT model also gradually adjusts the model parameters through NLP tasks such as text classification to obtain better performance [44]. Compared with the separate self-attention layer,

BERT has a stronger semantic information extraction ability. The self-attention layer can capture the correlation between different words in the news headlines, but it is difficult to capture the order characteristics of words [45]. For example, “A likes B” and “B likes A” actually have different meanings, but the two sentences get the same set of word embedding vectors after the output of self-attention. The BERT model adds position embedding to the input of the word sequence, giving each word a position information and distinguishing the order feature, and also adds segment embedding to separate the sentences, adding residual concatenation and normalization, and finally gets a set of embedding of words. The sentence representations obtained after processing are more closely matched to the original semantics of the sentence.

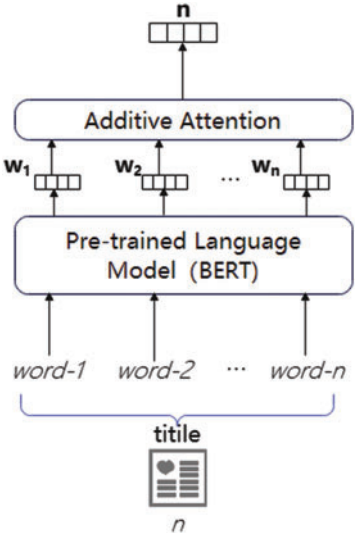


Figure 2: Architecture of the news model

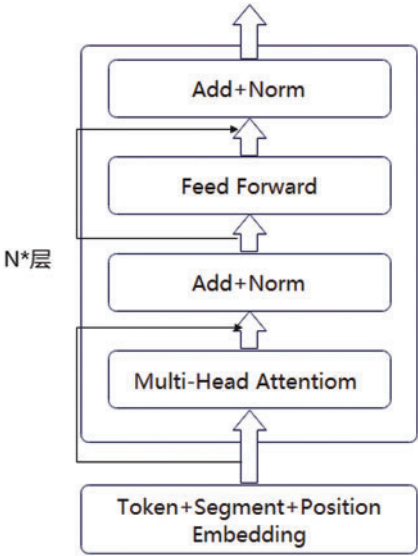


Figure 3: Architecture of the BERT model

We represent the hidden word representation sequence output by PLM as $[\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]$. Finally, the sentence embedding vector \mathbf{n} of news titles is obtained by assigning different weights to different word embedding vectors through the attention pooling network, and it is used as the representation vector of this news item. In the subsequent user model, for the user's browsing history news and the candidate news set, each news is modeled in the same way to obtain the news embedding.

3.1.2 User Encoder

The user encoder is used to obtain the user's representations from the user's browsing history. As shown in Fig. 4, the user's news click sequence is $[n_1, n_2, \dots, n_N]$, and the click sequence is passed through the news encoder to obtain the corresponding news embedding sequence $[\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_N]$. The news embedding sequence is used as the input of the user encoder, and the user embedding \mathbf{u} is output after the user model calculation. In this paper, the user model contains two parts: one part is used to capture the long-term interest of the user and the other part is used to capture the short-term interest of the user. Finally, the two are combined to obtain the representation vector of the user, and the overall structure of the user model is shown in Fig. 4.

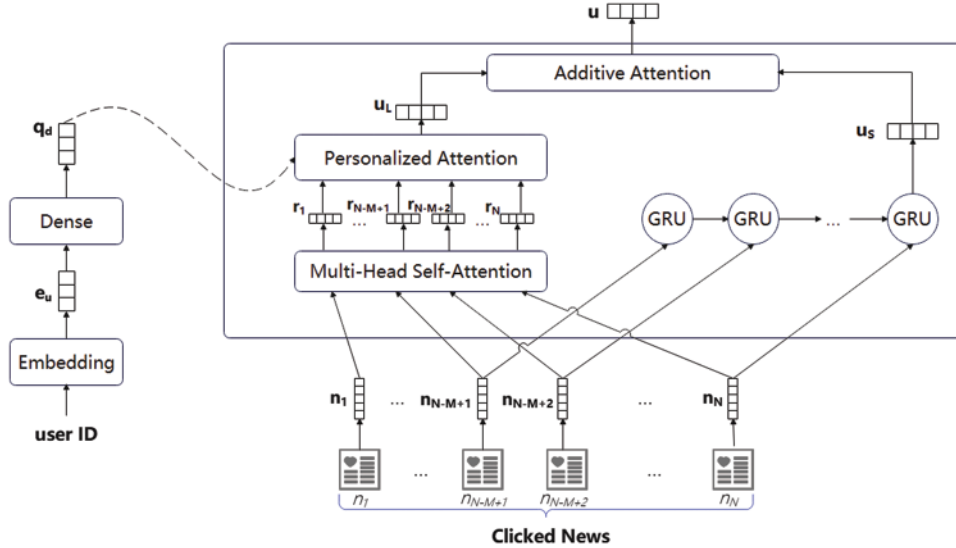


Figure 4: Architecture of the user model

We use multi-head self-attention mechanism and personalized attention mechanism to capture users' long-term interests. Since there is also some correlation between news read by the same user, the multi-head self-attention network can capture the correlation between different news. The news embedding sequence $[\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_N]$ is passed through the multi-head self-attention layer to obtain a new sequence of news representations $[\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N]$. Meanwhile, since the same news has different importance for different users, and each user attaches different importance to different news, this paper uses a personalized attention mechanism, which finds the corresponding user embedding vector e_u by checking the table in the embedding layer for user ID, and gets the news level vector q_d by passing e_u through two fully connected layers, and uses q_d as the query vector for the personalized attention layer, inner product q_d with each news vector r_i , and finally normalizes it by softmax to obtain the weight of each news. The specific steps are as follows:

$$q_d = \text{Tanh}(W_d \times \text{Relu}(V_d \times e_u + v_d) + b_d) \quad (1)$$

$$\alpha_i = \mathbf{r}_i^T \mathbf{q}_d, \quad (2)$$

$$\alpha'_i = e^{\alpha_i} / \sum_{j=1}^N e^{\alpha_j} \quad (3)$$

where \mathbf{W}_d , \mathbf{V}_d , v_d and b_d are parameters of the fully connected layer [4]. Finally, the clicked history news is weighted and summed to obtain a long-term representation \mathbf{u}_L of the user as

$$\mathbf{u}_L = \sum_{j=1}^N \alpha'_j \mathbf{r}_j \quad (4)$$

Previously, many recommendation systems ignored the impact of the time factor in reality [46]. In fact, when making news recommendations, we should not only consider the long-term interests of users, but also pay attention to their short-term interests. For example, a user is a basketball fan, he usually pays attention to sports news, but if some major political news happened recently, such as the U.S. election, then the user may have clicked on a lot of news about the political field recently. At this time, if we only pay attention to the user's long-term interest, then the news related to the U.S. election may not be pushed to the user, which is not beneficial to the user's experience. Since the self-attention model itself is difficult to capture the sequential characteristics of the sequences, this paper uses a gated recurrent unit network (GRU) to capture the short-term interest of the user, which can capture sequence information better [47]. GRU is computationally faster and logically simpler than earlier long short-term memory network (LSTM), and can well capture dependencies on sequences with long time step distances. We take the most recent batch of news vectors viewed by the user as input, and if the length of the most recent news sequence to obtain the short-term interest is set to be M , then the input of GRU is $[\mathbf{n}_{N-M+1}, \mathbf{n}_{N-M+2}, \dots, \mathbf{n}_N]$, and finally the short-term representation \mathbf{u}_s of the user is obtained.

Finally, to combine the long-term representation and short-term representation of the user, as proposed by An et al. [26], there are usually two ways. One way is to directly concatenate the two user representations, and the other is to use the long-term interest representation as the input of the user's short-term interest representation model, and the output of the short-term interest model as the final result [26]. While both approaches combine the two user interests, they ignore the fact that different people place different importance on long-term and short-term interests. Take the example mentioned above, one sports fan may not want to follow some political news related to the US election after reading it, but another may want to learn more about what happens afterwards. We need to consider the differences in psychological preferences of different users. In this paper, considering that different users pay different attention to long-term and short-term interests, an additive attention network is introduced to take long-term and short-term representations of users as inputs, assign different weights to long-term and short-term interests, and finally the output of the attention network is used as the final user representation \mathbf{u} .

With the above approach, our user model solves both the problem of differences in interest focus across users and the problem of shifting user interests over time.

3.1.3 Click Predictor

The purpose of the click score prediction model is to take the candidate news representation and the user representation obtained from the above encoder as input to get the user's rating for the candidate news. The method used in this paper is to do a dot product operation, where the user embedding and the candidate news embedding are dot producted to get the final score [48]. The reason for this operation is the simplicity of the operation. If the news embedding and the user embedding

are concatenated and then the score is computed through the fully connected layer, it will increase the training parameters and increase the training time. Since the pre-trained language model PLM already has hundreds of millions of parameters and is already very time-consuming to train, this method is like putting more pressure on model training [49]. At the same time, it is known experimentally that the difference between the scores obtained by the dot product and the scores output by the fully connected layer is not very large, so the dot product is chosen after a trade-off. Suppose the embedding vector of the candidate news n_c is \mathbf{n}_c , and representation vector of user u is \mathbf{u} , then the rating of user u for the candidate news n_c is

$$s(u, n_c) = \mathbf{u}^\top \mathbf{n}_c \quad (5)$$

3.1.4 Model Training

The training of the model is based on the user's clicking behavior. If a user clicks on a news item, the model can be adjusted accordingly to set a higher ranking score for that user news pair based on the model's score prediction for that news item. If the user does not click on the news, then the user's score for the news pair is correspondingly reduced. The model is trained accordingly for both clicked and non-clicked behaviors in the same impression.

More specifically, according to Wu et al. [4], for each news item clicked by user u , we randomly sample K news items that are not clicked, and we suppose these $K+1$ news items are in the same impression. Referring to [22] and [23], assuming that the user has a total of B_u clicks, the average loss function of the model with parameter set Θ is defined as

$$\mathcal{L}_u(\Theta) = - \sum_{i=1}^{|B_u|} \frac{1}{|B_u|} \mathcal{L}_u^i \quad (6)$$

$$\mathcal{L}_u^i = - \log \left(\frac{\exp(s(u, n_i^c))}{\exp(s(u, n_i^c)) + \sum_{j=1}^K \exp(s(u, n_{i,j}^{nc}))} \right), \quad (7)$$

where n_i^c and $n_{i,j}^{nc}$ are the news clicked and unclicked by users in the same impression respectively, and $s(u, t)$ is the rating of user u for news t .

3.2 Federal News Recommendation Framework

In this subsection, we combine the news recommendation model proposed in the previous section with the efficient federal recommendation framework as presented in [23] to obtain the privacy-preserving personalized news recommendation model FedNRM, and the general structure of the federal news recommendation framework is shown in Fig. 5.

The news recommendation model needs to collect the user's browsing history to model the user while the user's browsing history is usually private data. If the user's browsing history is stored centrally on the server, it is easy to lead to privacy leakage [50]. Therefore, a federal news recommendation framework is introduced, where the user can put the browsing history to the local client, train the model locally, and upload the local loss gradient to the server. Then, the server aggregates the local loss gradient, performs global model update, and sends the updated user model and news representation model to each client [51]. The process above will be iterative until the model converges.

Since the news model is very large, it is very time consuming for the client to request the news model from the server. We develop a method in which the user side only needs to request a small amount of the news representation model from the server, and does not need to request the whole news model. The specific algorithm of the system is shown in Algorithm 1.

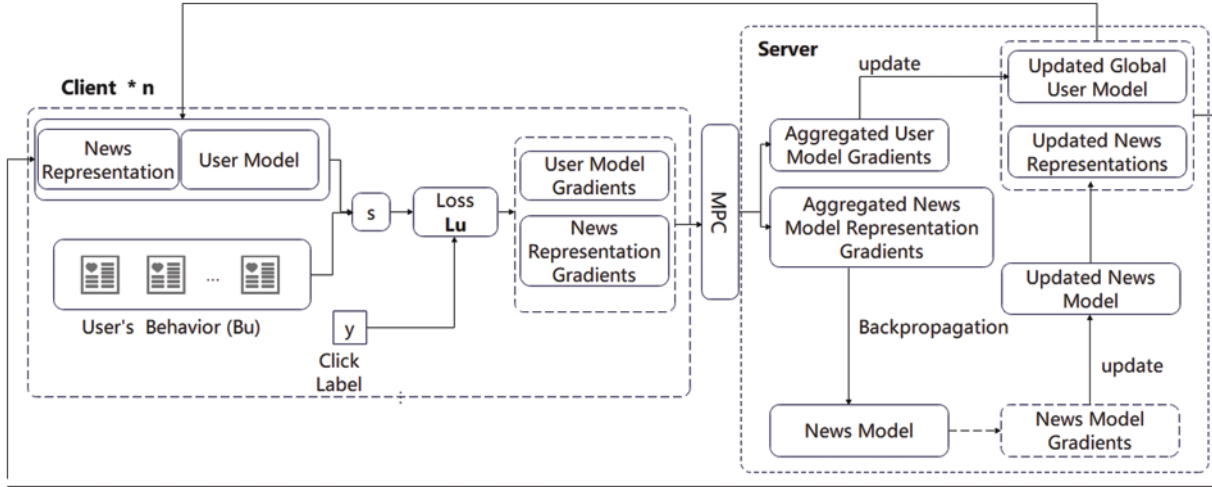


Figure 5: Architecture of the federal news recommendation framework

Algorithm 1: Federated Model Training

```

1  Server executes:
2  Initialize  $w_0$ ;
3  for each round  $t = 1, 2, \dots$  do
4    Sample a set of clients  $u'_s$ ;
5    for each client  $u \in u'_s$  in parallel do
6      Send  $w_{e_s}^t, w_u^t$  to user  $u$ ;
7       $g_{uu}^t, g_{e_s u}^t \leftarrow \text{ClientUpdate}(u, w_{e_s}^t, w_u^t)$ ;
8    end
9     $\bar{g}_u^t \leftarrow \sum_{u \in u'_s} \frac{1}{\sum_{u \in u'_s} |B_u|} \cdot |B_u| \cdot g_{uu}^t$ ;
10    $\bar{g}_{e_s}^t \leftarrow \sum_{u \in u'_s} \frac{1}{\sum_{u \in u'_s} |B_u|} \cdot |B_u| \cdot g_{e_s u}^t$ ;
11    $\bar{g}_n^t \leftarrow \sum_{n_i \in N_s^t} \bar{g}_{e_i}^t \cdot \frac{\partial \mathbf{n}_i}{\partial w_n^t}$ ;
12    $w_u^{t+1} \leftarrow \text{FedAdam}(\bar{g}_u^t)$ ;
13    $w_n^{t+1} \leftarrow \text{FedAdam}(\bar{g}_n^t)$ ;
14    $w_{e_s}^{t+1} \leftarrow w_n^{t+1}$ ;
15  end
16
17  ClientUpdate  $(u, w_{e_s}^t, w_u^t)$ : //Executed on client  $u$ 
18  for each behavior  $j \in B_u$  do
19     $\mathcal{L}_u^j \leftarrow \text{Model Training}(w_{e_s}^t, w_u^t)$ ;
20  end
21   $\mathcal{L}_u \leftarrow -\frac{1}{|B_u|} \sum_{j=1}^{|B_u|} \mathcal{L}_u^j$ ;
22   $g_{uu}^t \leftarrow \frac{\partial \mathcal{L}_u}{\partial w_u^t}$ ;

```

(Continued)

Algorithm 1: Continued

```

23    $g_{e_s u}^t \leftarrow \frac{\partial \mathcal{L}_u}{\partial w_{e_s}^t};$ 
24   return  $g_{uu}^t, g_{e_s u}^t$  to server

```

In general, it is divided into two participants: the server and the client. Assume that the t -th round of training is in progress, the algorithm is divided into three steps: in the first step, the server randomly selects a set of users u_s^t , and then sends the parameter set $w_{e_s}^t$ of the news representation model and the parameter set w_u^t of the user model to each user u in the set. In the second step, the client obtains the average loss function L_u for all the user's click behaviors based on the obtained model parameters, combined with the local user behavior data B_u , after the model training in Section 3.1.4, so as to calculate and obtain the loss gradient g_{uu}^t for the user representation and the loss gradient $g_{e_s u}^t$ for the news representation, and return these two gradients to the server. In the third step, the server performs gradient aggregation based on the g_{uu}^t and $g_{e_s u}^t$ of all sampled users in the set u_s^t to obtain the aggregated gradients \bar{g}_u^t and $\bar{g}_{e_s}^t$. Assume that the user set u_s^t involves a union news set of N_s^t , the server computes the representation \mathbf{n}_i of each news $n_i \in N_s^t$ based on the news model, combines the corresponding news representation gradient $\bar{g}_{e_i}^t$ and the news model parameters w_n^t to obtain the aggregation gradient \bar{g}_n^t of the news model. Then the server uses the FedAdam algorithm [52] to compute the updated model parameter sets w_n^{t+1} and w_u^{t+1} based on the aggregation gradients \bar{g}_n^t and \bar{g}_u^t , and finally infers the news representation based on w_n^{t+1} to obtain the updated news representation parameters $w_{e_s}^{t+1}$. With this, one round of work is completed, we repeat the operation of the first step, and the model converges after multiple iterations.

Besides, considering that users' browsing history is leaked when requesting news representations from the server, this paper will compute a set of users' joint news sets when requesting local news representations. Given a user u_i , assume that he/she requests a local news set of N_i , which is converted into a local vector \mathbf{h}_i whose dimension is equal to the number of all news. The j -th dimension of \mathbf{h}_i is defined as

$$\mathbf{h}_i^j = \begin{cases} 1, & \text{if } n_j \in N_i \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where n_j is the j -th news item in the whole news set [23].

The joint news set is represented as the union of the local vectors of this group of user local vectors, and the news requested from the server in the end is represented as the set of news corresponding to a dimension in the vector \mathbf{h} that is non-zero. The process of a group of users requesting news representations is shown in Fig. 6. In this way, the server only knows the news that interacted with this group of users, but not the individual users, which protects the privacy of users. Specially,

$$\mathbf{h} = \sum_{u_i \in u_s} \mathbf{h}_i \quad (9)$$

where u_s is a set of users.

Through the use of federal frameworks and secure aggregation algorithms, the privacy of users can be effectively protected while reducing the cost of communication.

In a practical news recommendation system application, we need to consider the cost of computation and communication, as well as the loss of device resources on the server side and the client side. Since our federated recommendation model is relatively large and will have high equipment

requirements, we may need to introduce novel electrodes in the future and use novel materials and devices to improve the practicality and scalability of the model [53–56].

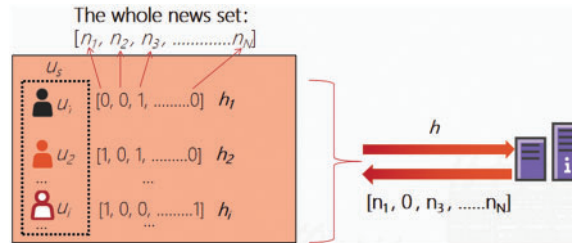


Figure 6: Process of a group of users requesting news representations from the server

4 Experiment

4.1 Dataset and Experimental Settings

We conducted our experiments on a real dataset, namely Adressa [57]. Adressa is a publicly available news dataset from a Norwegian newspaper company, which contains 7 days of sample data. Referring to [23] and [58], we use the first 5 days of news click data as the user’s historical clicks, the 6th day as the training set, and the remaining day as the validation set and the test set, where the validation set accounts for 20% and the test set accounts for 80%. The detailed statistical information of the dataset is summarized in Table 1. It can be seen that the Adressa dataset does not distinguish impression and does not contain negative samples, so for each user’s click, we randomly select 20 news items as negative samples, and treat these 21 news items as news items in an impression, so as to conduct the subsequent experiments. Drawing on the performance metrics previously used by others in the field of news recommendation [7,23,59,60], we use AUC, MRR, nDCG@5 and nDCG@10 as the evaluation metrics for recommendation performance.

In our experiments, for the Adressa dataset, we utilize nb-bert-base proposed in [61] to initialize the BERT model in the news encoder. Referring to the experiment in [23], we set the dimensionality of the news representation to 400 and the maximum length of the user’s click history to 50, i.e., only a maximum of 50 browsing records of the user are kept. In addition, we set the size of the user group mentioned in the previous section to 50. For training, we set the learning rate of the user model and the news model to 0.00005. We use the optimal solutions of the hyper-parameters on the validation set as the hyper-parameters of the test set, and on the test dataset, we repeat each experiment five times and take the mean and standard deviation as the final results of the experiments.

Table 1: Statistical information of the Adressa dataset

Data type	Quantity
News	20,428
Users	640,503
Impressions	-
Positive samples	3,101,991
Negative samples	-

4.2 Baselines

In our experiments, we use the following methods as baselines for comparison:

- (1) DFM: It is a deep fusion model using a combination of fully connected layers of different depths for news recommendation [62].
- (2) DKN: It is a news recommendation method using deep knowledge-aware network [63].
- (3) LSTUR: It is a neural news recommendation method with long-term and short-term user representation, which captures long-term interest of users using their ID embeddings and short-term interest of users using GRU sequences [26].
- (4) NAML: It uses a multi-view attention mechanism for news representation, which enables more accurate news representation through multiple features of news datasets, such as categories, headlines, and news bodies [6].
- (5) NRMS: It uses the self-attention mechanism to represent news features and user features, and uses news word sequences and user browsing news sequences as inputs respectively [7].
- (6) PLM-NR: It uses pre-trained language models to characterize the topics of news [12].
- (7) FCF: It uses federal collaborative filtering methods for news recommendation, and is one of the earlier approaches to apply federal learning to news recommendation [33].
- (8) FedRec: It is a privacy-preserving news recommendation method that applies a federal learning framework and local differential privacy to protect user privacy [22].
- (9) Efficient-FedRec: It is an efficient privacy-preserving news recommendation method [23].

4.3 Performance Evaluation

Our experiments first compare some baseline news recommendation algorithms and also compare the model proposed in this paper with some federal news recommendation methods. Among them, LSTUR, NAML, NRMS and PLM-NR are centralized news recommendation methods, and FCF, FedRec and Efficient-FedRec are federated news recommendation methods.

The recommendation performance of different recommendation methods is shown in [Table 2](#), and the experimental results of the other methods in the baselines, except for Efficient-FedRec, are referred to [\[23\]](#). It can be seen that LSTUR, NAML, NRMS and PLM-NR methods obtain much higher recommendation performance than DFM, a traditional deep learning method. The reason is that LSTUR, NAML, NRMS, and PLM-NR apply more complex models to characterize users as well as news. Among the centralized news recommendation methods, the PLM-NR model has the highest recommendation performance because PLM-NR uses a pre-trained model to characterize the topics of the news and obtains deeper semantic features of the news. Our method FedNRM has some performance improvement compared to other federated learning methods (FCF, FedRec and Efficient-FedRec). FCF has the lowest recommendation performance, on the one hand, because collaborative filtering is still a relatively basic recommendation algorithm with a cold start problem [\[64\]](#), every user and item must be involved in the training process as a way to learn the embedding. At the same time, because the news articles are updated very fast, FCF cannot handle new items. On the other hand, because centralized data is more conducive to model training than decentralized data, it leads to the case that the training effect of FCF model is not ideal. In the FedRec method, the news model uses CNN and self-attention to capture the features of news. Since it is difficult for self-attention to capture the sequential features that constitute the news words, the news model in this paper uses a pre-trained model BERT to capture the features of news, which incorporates the sequential information of news words and introduces residual connectivity as well as standardization methods to obtain better recommendation performance. In the Efficient-FedRec approach, the user model uses a multi-head self-attention mechanism and an additive attention mechanism to characterize the user's

browsing news sequence. In this paper, we add the users' ID information, introduce the personalized attention mechanism to enhance users' long-term representation. We also add the GRU sequence network to the user model to capture the users' short-term interests, and make an additive attention operation between the users' short-term interests and long-term interests to obtain the new user vector. The final experimental results show that this paper's news recommendation model FedNRM obtains better recommendation performance than FedRec and Efficient-FedRec.

Table 2: News recommendation performance of different methods. The performance is of the form $A \pm B$, where A is the average of multiple experiments and B is the corresponding standard deviation

Method	AUC	MRR	nDCG@5	nDCG@10
DFM	59.90 ± 1.20	32.68 ± 0.75	29.69 ± 0.93	36.43 ± 1.11
DKN	73.73 ± 0.48	39.52 ± 1.34	40.98 ± 1.24	47.48 ± 0.86
LSTUR	68.37 ± 2.63	38.76 ± 2.14	38.11 ± 2.39	44.33 ± 2.42
NAML	73.09 ± 1.53	44.27 ± 1.53	43.51 ± 1.89	50.02 ± 1.71
NRMS	75.31 ± 0.94	42.24 ± 0.92	44.66 ± 1.50	48.46 ± 1.19
PLM-NR	78.20 ± 1.28	47.26 ± 1.73	48.41 ± 2.10	54.60 ± 1.64
FCF	51.39 ± 0.74	18.98 ± 1.57	15.42 ± 1.72	22.94 ± 1.30
FedRec	71.73 ± 1.72	41.37 ± 2.21	41.81 ± 2.35	47.18 ± 2.09
Efficient-FedRec	79.00 ± 1.43	45.78 ± 1.59	47.56 ± 1.42	54.03 ± 1.89
FedNRM	79.43 ± 1.49	46.68 ± 1.88	48.79 ± 2.26	55.05 ± 1.56

Our method improves the performance compared to LSTUR, NAML, NRMS, etc., and verifies the effectiveness of our method in achieving personalized news recommendations. In addition, since our method uses the training method of the federal learning framework to save browsing news history on the client side without uploading browsing history news to the server side and train locally, this approach effectively protects the user's browsing history from being stolen. Therefore, our model achieves high recommendation performance while protecting user privacy to a certain extent.

4.4 Effectiveness of Personalized Attention

In this section, ablation experiments are conducted to verify the performance improvement brought by replacing additive attention with personalized attention in the long-term representation in the user model, and the experimental results are shown in Fig. 7. The experiments show that the personalized attention layer used in this model can improve the news recommendation performance, and personalized attention is more accurate in representing the long-term interests of users compared to additive attention. The reason is that in additive attention, the query vector is randomly initialized, while in personalized attention the user ID embedding is introduced as the query vector, and the news embedding sequence output by self-attention is weighted and summed to obtain the long-term interest representation of the user. For the same news, different users may pay attention to different parts of the news, and for different news, users attach different degrees of importance, and the introduction of user ID is beneficial to personalize the representation of users, so as to achieve personalized recommendation.

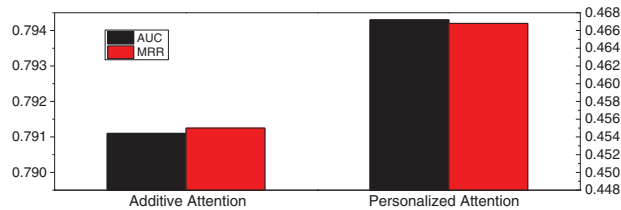


Figure 7: Performance comparison of additive attention and personalized attention

4.5 Effectiveness of Short-Term Representation

In this section, we conduct an ablation experiment to verify the effect of adding the short-term user representation, and the experimental results are shown in Fig. 8. It can be found that the performance metrics of the news recommendation model are improved after adding the GRU sequence model. The reason is that the long-term interest of users can be captured by self-attention, but for self-attention, the user representations obtained from news sequences of different orders may be the same, because it is difficult for self-attention itself to capture the order information of the sequences. However, the GRU sequence network can capture the sequence information of news, and the recent browsing history of users can be used as input to get a short-term representation of users. Although users may belong to a certain type of interest group in the long term, but they may also pay attention to some recent big events in the short term, and these recent news events also need to be pushed to users. If only attention mechanism is considered to represent users' embedding, some information about users' recent reading preferences will be lost, which reduces users' experience.

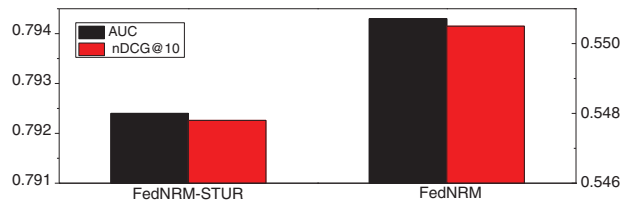


Figure 8: Effectiveness of short-term user representations (STUR)

4.6 Effectiveness of Pre-Trained Language Model

In this section, the effect of adding the pre-trained model (BERT) to characterize the news vector is verified, and the experimental results are shown in Fig. 9. Here, removing the pre-trained model in this ablation experiment means characterizing the word sequence of the news with a non-pre-trained model. The non-pre-trained news representation model used for the control in this paper is divided into four layers, namely word embedding layer, CNN, self-attention layer and attention pooling layer. The experimental results show that using pre-trained language models for news coding brings a considerable improvement in recommendation performance due to the improved text representation capability of BERT with stronger semantic information extraction. As mentioned above, the BERT model adds positional information to the input of word sequences, distinguishes sequential features, and also adds segment embedding to separate sentences, adds residual concatenation and normalization for optimization, and finally obtains the representation vector of news. The news embedding obtained after the pre-trained language model is significantly better than the simple CNN and attention layer superposition, and the user model is also built based on the browsed historical news vector, so the accuracy of the final news recommendation is greatly improved.

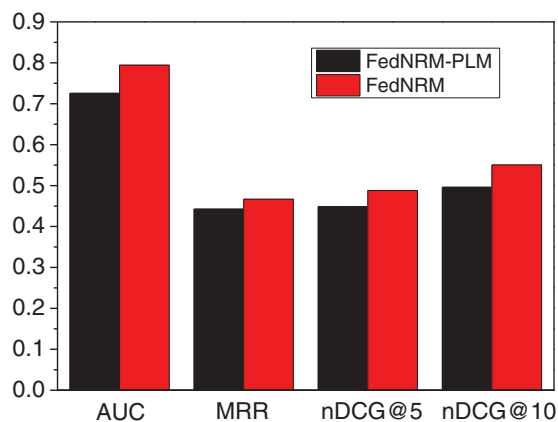


Figure 9: Effectiveness of the pre-trained language model

4.7 Hyper-Parameter Optimization

In this section, we make appropriate adjustments to the parameters in our experiments to summarize the influences of different parameters on the performance and training cost of the news recommendation model.

4.7.1 Size of User Group

Here, the hyper-parameter optimization is performed for the selection of user group size in Section 3.2, and the results are shown in Fig. 10a. It can be found that the time required for training increases linearly with the increase of user group size. This is because the larger the user group is, the larger the size of the joint news set is, and the propagation cost and communication time during training increases accordingly. As shown in Fig. 10b, the experiments also show that the recommendation performance of the model does not change much as the user group size increases. Moreover, the news recommendation performance tends to increase when the user group size is lower than 50, but starts to decrease when the user group size is larger than 50. In real-world recommendation systems, we need to find a balance between news recommendation performance and training cost considering factors such as user experience and computer equipment communication cost. Experiments show that the recommendation performance of the model reaches the highest when the user group size is 50 and the time required for training is within an acceptable range. Therefore, with the combination of training cost and recommendation performance in mind, we choose the user group size as 50 when we do other experiments.

4.7.2 Length of Short-Term News Sequences

Here we optimize the selection of the length M of the sequence of browsed news in the short-term user representation mentioned in Section 3.1.2. The length of the sequence refers to the selection of the number of recently browsed news by users, and the representation of these news is used as the input to the GRU sequence network. The total amount of user browsing history is taken as 50 in the experiment, and the experimental results are shown in Fig. 11. If the number of recently viewed news as the input to GRU is set to M , the performance of news recommendation is optimal when M is set to 20. When M is less than 20, the recommendation performance increases with the increase of M . The reason should be that when M is too small, the number of news samples is not enough to capture the user's interest and the accuracy of the indicated user embedding is not high. On the other hand,

when M is too large, the recommendation performance decreases with increasing M . This is because when M is too large, the browsed news history tends more and more to the long-term interest of the user, since the user cannot always watch the more important news in the short term or the type of temporary interest all the time.

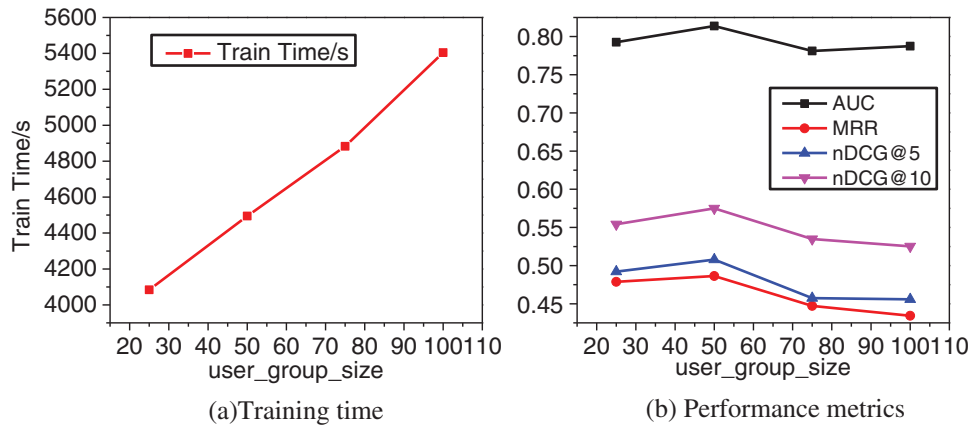


Figure 10: Influence of different size of user group

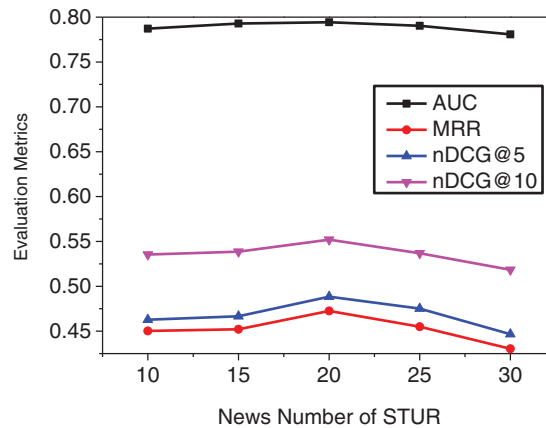


Figure 11: Influence of different length of short-term news sequences

5 Conclusion

In this paper, a federal news recommendation method FedNRM is proposed. Unlike previous news models, we apply a pre-trained model BERT to represent the news vector and obtain a better news representation. Unlike previous user models, our user model not only focuses on the long-term interests of users, but also captures their short-term interests. In addition, this paper introduces userID embedding, applies personalized attention mechanism to long-term interest representation to obtain personalized user interest representation, applies GRU network to short-term interest representation, and then applies additive attention to combine user's long-term interest and short-term interest to obtain more accurate user representation. Our model solves both the problem of differences in interest focus across users and the problem of shifting user interests over time. After that, considering that the user's browsing history contains a large amount of user privacy, we also introduce a federal framework

to save the user's browsing history locally, train the model locally, and then upload the local training loss gradients to the server, which performs the aggregation of gradients and global model updates. The server passes the updated model to the client and converges after iterating many times, which protects the user's privacy to some extent. We conducted experiments on a real dataset and the final results show that our approach achieves good recommendation performance compared to other baselines and previous federal recommendation methods, while preserving user privacy to a certain extent.

Funding Statement: The authors received no funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] L. Li, D. D. Wang, S. Z. Zhu and T. Li, "Personalized news recommendation: A review and an experimental investigation," *Computer Science & Technology*, vol. 26, no. 5, pp. 754–766, 2011.
- [2] Y. Wang, L. Qi, R. Dou, S. Shen, L. Hou *et al.*, "An accuracy-enhanced group recommendation approach based on DEMATEL," *Pattern Recognition Letters*, vol. 167, pp. 171–180, 2023.
- [3] S. Wu, S. Shen, X. Xu, Y. Chen, X. Zhou *et al.*, "Popularity-aware and diverse web APIs' recommendation based on correlation graph," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 2, pp. 771–782, 2023. Early Access. <https://doi.org/10.1109/TCSS.2022.3168595>
- [4] C. H. Wu, F. Z. Wu, M. X. An, J. Q. Huang, Y. F. Huang *et al.*, "NPA: Neural news recommendation with personalized attention," in *Proc. the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, pp. 2576–2584, 2019.
- [5] S. Feng, L. Zhao, H. Shi, M. Wang, S. Shen *et al.*, "One-dimensional VGGNet for high-dimensional data," *Applied Soft Computing*, vol. 135, pp. 110035, 2023.
- [6] C. H. Wu, F. Z. Wu, M. X. An, J. Q. Huang, Y. F. Huang *et al.*, "Neural news recommendation with attentive multi-view learning," in *Proc. the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Macao, China, pp. 3863–3869, 2019.
- [7] C. H. Wu, F. Z. Wu, S. Y. Ge, T. Qi, Y. F. Huang *et al.*, "Neural news recommendation with multi-head self-attention," in *Proc. the 2019 Conf. on Empirical Methods in Natural Language Processing & the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, pp. 6389–6394, 2019.
- [8] Y. M. Cui, W. X. Che, T. Liu, B. Qin and Z. Q. Yang, "Pre-training with whole word masking for Chinese bert," *IEEE/ACM Transactions on Audio, Speech, & Language Processing*, vol. 29, pp. 3504–3514, 2021.
- [9] M. Suzuki, H. Sakaji, M. Hirano and K. Izumi, "Constructing and analyzing domain-specific language model for financial text mining," *Information Processing & Management*, vol. 60, no. 2, pp. 103194, 2023.
- [10] Y. Y. S. Tian, Y. Wan, L. J. Lyu, D. Z. Yao, H. Jin *et al.*, "FedBERT: When federated learning meets pre-training," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–26, 2022.
- [11] S. Rothe, S. Narayan and A. Severyn, "Leveraging pre-trained checkpoints for sequence generation tasks," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 264–280, 2020.
- [12] C. H. Wu, F. Z. Wu, T. Qi and Y. F. Huang, "Empowering news recommendation with pre-trained language models," in *Proc. the 44th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Virtual Event, Online, pp. 1652–1656, 2021.
- [13] S. Okura, Y. Tagami, S. Ono and A. Tajima, "Embedding-based news recommendation for millions of users," in *Proc. KDD*, Halifax, NS, Canada, pp. 1933–1942, 2017.
- [14] T. Z. Zang, Y. M. Zhu, J. Zhu, Y. A. Xu and H. B. Liu, "MPAN: Multi-parallel attention network for session-based recommendation," *Neurocomputing*, vol. 471, pp. 230–241, 2022.

- [15] L. Ravi, V. Subramaniaswamy, M. Devarajan, K. S. Ravichandran, S. Arunkumar *et al.*, “SECRECSY: A secure framework for enhanced privacy-preserving location recommendations in cloud environment,” *Wireless Personal Communications*, vol. 108, no. 3, pp. 1869–1907, 2019.
- [16] Y. Shen, S. Shen, Q. Li, H. Zhou, Z. Wu *et al.*, “Evolutionary privacy-preserving learning strategies for edge-based IoT data sharing schemes,” *Digital Communications and Networks*, 2022. <https://doi.org/10.1016/j.dcan.2022.05.004>
- [17] P. Sun, S. Shen, Z. Wu, H. Zhou and X. Gao, “Stimulating trust cooperation in edge services: An evolutionary tripartite game,” *Engineering Applications of Artificial Intelligence*, vol. 116, pp. 105465, 2022.
- [18] Y. Qu, L. Gao, Y. Xiang, S. Shen and S. Yu, “FedTwin: Blockchain-enabled adaptive asynchronous federated learning for digital twin networks,” *IEEE Network*, vol. 36, no. 6, pp. 183–190, 2022. Early Access. <https://doi.org/10.1109/MNET.105.2100620>
- [19] L. Cui, Y. Qu, G. Xie, D. Zeng, R. Li *et al.*, “Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3492–3500, 2022.
- [20] H. S. Hu, G. Dobbie, Z. Salcic, M. Liu, J. B. Zhang *et al.*, “Differentially private locality sensitive hashing based federated recommender system,” *Concurrency and Computation: Practice & Experience*, 2021. Early Access. <https://doi.org/10.1002/cpe.6233>
- [21] Z. Sun, J. Y. Feng, L. H. Yin, Z. X. Zhang, R. Li *et al.*, “Fed-DFE: A decentralized function encryption-based privacy-preserving scheme for federated learning,” *CMC-Computers Materials & Continua*, vol. 71, no. 1, pp. 1867–1886, 2022.
- [22] T. Qi, F. Z. Wu, C. H. Wu, Y. F. Huang and X. Xie, “Privacy-preserving news recommendation model learning,” CoRR, abs/2003.09592, 2020. [Online]. Available: <https://arxiv.org/abs/2003.09592>
- [23] J. W. Yi, F. Z. Wu, C. H. Wu, R. X. Liu, G. Z. Sun *et al.*, “Efficient federated learning framework for privacy-preserving news recommendation,” in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic, pp. 2814–2824, 2021.
- [24] C. Wang, C. Jiang, J. Wang, S. Shen, S. Guo *et al.*, “Blockchain-aided network resource orchestration in intelligent internet of things,” *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6151–6163, 2023.
- [25] P. Zhang, P. Gan, N. Kumar, C. Hsu, S. Shen *et al.*, “RKD-VNE: Virtual network embedding algorithm assisted by resource knowledge description and deep reinforcement learning in IIoT scenario,” *Future Generation Computer Systems*, vol. 135, pp. 426–437, 2022.
- [26] M. X. An, F. Z. Wu, C. H. Wu, K. Zhang, Z. Liu *et al.*, “Neural news recommendation with long-and short-term user representations,” in *Proc. the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, pp. 336–345, 2019.
- [27] D. H. Tran, Q. Z. Sheng, W. E. Zhang, N. H. Tran and N. L. D. Khoa, “CupMar: A deep learning model for personalized news recommendation based on contextual user-profile and multi-aspect article representation,” *World Wide Web-Internet & Web Information Systems*, 2022. Early Access. <https://doi.org/10.1007/s11280-022-01059-6>
- [28] J. Q. Zhang, D. J. Wang and D. J. Yu, “TLSAN: Time-aware long-and short-term attention network for next-item recommendation,” *Neurocomputing*, vol. 441, pp. 179–191, 2021.
- [29] L. M. Hu, C. Li, C. Shi, C. Yang and C. Shao, “Graph neural news recommendation with long-term and short-term interest modeling,” *Information Processing & Management*, vol. 57, no. 2, pp. 102142, 2020.
- [30] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. AISTATS*, Fort Lauderdale, FL, USA, pp. 1273–1282, 2017.
- [31] N. N. Thilakarathne, G. Muneeswari, V. Parthasarathy, F. Alassery, H. Hamam *et al.*, “Federated learning for privacy-preserved medical internet of things,” *Intelligent Automation & Soft Computing*, vol. 33, no. 1, pp. 157–172, 2022.
- [32] D. Jiang, Y. F. Song, Y. X. Tong, X. Y. Wu, W. W. Zhao *et al.*, “Federated topic modeling,” in *Proc. the 28th ACM Int. Conf. on Information and Knowledge Management*, Beijing, China, pp. 1071–1080, 2019.

- [33] M. Ammad, E. Ivannikova, S. AKhan, W. Oyomno, Q. Fu *et al.*, “Federated collaborative filtering for privacy-preserving personalized recommendation system,” CoRR, abs/1901.09888, 2019. [Online]. Available: <https://arxiv.org/abs/1901.09888>
- [34] G. Y. Lin, F. Liang, W. K. Pan and Z. Ming, “FedRec: Federated recommendation with explicit feedback,” *IEEE Intelligent Systems*, vol. 36, no. 5, pp. 21–29, 2021.
- [35] F. Liang, W. K. Pan and Z. Ming, “FedRec plus plus: Lossless federated recommendation with explicit feedback,” in *Proc. the 35th AAAI Conf. on Artificial Intelligence/33rd Conf. on Innovative Applications of Artificial Intelligence/11th Symp. on Educational Advances in Artificial Intelligence*, Virtual Event, Online, pp. 4224–4231, 2021.
- [36] C. H. Wu, F. Z. Wu, Y. Cao, Y. F. Huang and X. Xie, “FedGNN: Federated graph neural network for privacy-preserving recommendation,” CoRR, abs/2102.04925, 2021. [Online]. Available: <https://arxiv.org/abs/2102.04925>
- [37] J. Z. Wu, Q. Liu, Z. Y. Huang, Y. T. Ning, H. Wang *et al.*, “Hierarchical personalized federated learning for user modeling,” in *Proc. the 30th World Wide Web Conf. (WWW)*, Ljubljana, Slovenia, pp. 957–968, 2021.
- [38] H. F. Zhu, R. S. M. Goh and W. K. Ng, “Privacy-preserving weighted federated learning within the secret sharing framework,” *IEEE Access*, vol. 8, pp. 198275–198284, 2020.
- [39] M. R. Nosouhi, S. Yu, K. Sood, M. Grobler, R. Jurdak *et al.*, “UCoin: An efficient privacy preserving scheme for cryptocurrencies,” *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 242–255, 2023.
- [40] J. Liu, Y. Tian, Y. Zhou, Y. Xiao and N. Ansari, “Privacy preserving distributed data mining based on secure multi-party computation,” *Computer Communications*, vol. 153, pp. 208–216, 2020.
- [41] G. Xu, F. Yun, X. B. Chen, S. Y. Xu, J. Z. Wang *et al.*, “Secure multi-party quantum summation based on quantum homomorphic encryption,” *Intelligent Automation & Soft Computing*, vol. 34, no. 1, pp. 531–541, 2022.
- [42] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan *et al.*, “Practical secure aggregation for privacy-preserving machine learning,” in *Proc. the 2017 ACM SIGSAC Conf. on Computer & Communications Security*, Dallas, TX, USA, pp. 1175–1191, 2017.
- [43] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. naacL-HLT*, Minneapolis, MN, USA, pp. 4171–4186, 2019.
- [44] X. Y. Yin, W. Zhang, W. H. Zhu, S. Liu and T. J. Yao, “Improving sentence representations via component focusing,” *Applied Sciences*, vol. 10, no. 3, pp. 958, 2020.
- [45] Q. P. Guo, X. P. Qiu, X. Y. Xue and Z. Zhang, “Low-rank and locality constrained self-attention for sequence modeling,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2213–2222, 2019.
- [46] L. Kong, G. Li, W. Rafique, S. Shen, Q. He *et al.*, “Time-aware missing healthcare data prediction based on ARIMA model,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022. Early Access. <https://doi.org/10.1109/TCBB.2022.3205064>
- [47] Y. Cheng, H. Sun, H. M. Chen, M. Li, Y. Y. Cai *et al.*, “Sentiment analysis using multi-head attention capsules with multi-channel CNN and bidirectional GRU,” *IEEE Access*, vol. 9, pp. 60383–60395, 2021.
- [48] J. Ni, Z. H. Huang, J. J. Cheng and S. C. Gao, “An effective recommendation model based on deep representation learning,” *Information Sciences*, vol. 542, pp. 324–342, 2020.
- [49] Y. Shen, S. Shen, Z. Wu, H. Zhou and S. Yu, “Signaling game-based availability assessment for edge computing-assisted IoT systems with malware dissemination,” *Journal of Information Security and Applications*, vol. 66, pp. 103140, 2022.
- [50] H. J. Shin, S. Kim, J. Shin and X. K. Xiao, “Privacy enhanced matrix factorization for recommendation with local differential privacy,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.

- [51] J. Sun, T. Y. Chen, G. B. Giannakis, Q. M. Yang and Z. Y. Yang, “Lazily aggregated quantized gradient innovation for communication-efficient federated learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2031–2044, 2020.
- [52] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush *et al.*, “Adaptive federated optimization,” in *Proc. Int. Conf. on Learning Representations (ICLR)*, Virtual Event, Online, 2021.
- [53] M. Zhang, W. L. Wang, G. T. Xia, L. C. Wang and K. Wang, “Self-powered electronic skin for remote human-machine synchronization,” *ACS Applied Electronic Materials*, vol. 5, no. 1, pp. 498–508, 2023.
- [54] Y. Guo, D. F. Yang, Y. Zhang, L. C. Wang and K. Wang, “Online estimation of SOH for lithium-ion battery based on SSA-elman neural network,” *Protection and Control of Modern Power Systems*, vol. 7, no. 1, pp. 40, 2022.
- [55] M. Zhang, Y. S. Liu, D. Z. Li, X. L. Cui, L. C. Wang *et al.*, “Electrochemical impedance spectroscopy: A new chapter in the fast and accurate estimation of the state of health for lithium-ion batteries,” *Energies*, vol. 16, no. 4, pp. 1599, 2023.
- [56] N. Ma, D. F. Yang, S. Riaz, L. C. Wang and K. Wang, “Aging mechanism and models of supercapacitors: A review,” *Technologies*, vol. 11, no. 2, pp. 38, 2023.
- [57] J. A. Gulla, L. M. Zhang, P. Liu, O. Ozgobek and X. M. Su, “The adressa dataset for news recommendation,” in *Proc. IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI)*, Leipzig, Germany, pp. 1042–1048, 2017.
- [58] L. M. Hu, S. Y. Xu, C. Li, C. Yang, C. Shi *et al.*, “Graph neural news recommendation with unsupervised preference disentanglement,” in *Proc. the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Virtual Event, Online, pp. 4255–4264, 2020.
- [59] K. H. Yang, S. S. Long, W. Zhang, J. Q. Yao and J. Liu, “Personalized news recommendation based on the text and image integration,” *Computers Materials & Continua*, vol. 64, no. 1, pp. 557–570, 2020.
- [60] J. Tang, L. S. Zhang and C. Y. Sang, “News recommendation with latent topic distribution and long and short-term user representations,” *Data Analysis and Knowledge Discovery*, vol. 6, no. 9, pp. 52–64, 2022.
- [61] P. E. Kummervold, J. D. Rosa, F. Wetjen and S. A. Brygfjeld, “Operationalizing a national digital library: The case for a Norwegian transformer model,” in *Proc. the 23rd Nordic Conf. on Computational Linguistics (NoDaLiDa)*, Reykjavik, Iceland, pp. 20–29, 2021.
- [62] J. X. Lian, F. Z. Zhang, X. Xie and G. Z. Sun, “Towards better representation learning for personalized news recommendation: A multi-channel deep fusion approach,” in *Proc. the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Stockholm, Sweden, pp. 3805–3811, 2018.
- [63] H. W. Wang, F. Z. Zhang, X. Xie and M. Y. Guo, “Dkn: Deep knowledge-aware network for news recommendation,” in *Proc. the World Wide Web Conf. (WWW)*, Lyon, France, pp. 1835–1844, 2018.
- [64] A. Binbusayyis, “Deep embedded fuzzy clustering model for collaborative filtering recommender system,” *Intelligent Automation & Soft Computing*, vol. 33, no. 1, pp. 501–513, 2022.

Appendix

Appendix A: Comparison of some news recommendation models involved in this paper

Model	News encoder	User encoder
NPA	Word Embedding + CNN + Personalized Attention	Personalized Attention
LSTUR	Word Embedding + CNN + Attention	ID Embedding + GRU

(Continued)

Appendix A: Continued

Model	News encoder	User encoder
NAML	Category Embedding + Dense(Categories) Word Embedding + CNN + Attention(Title) Word Embedding + CNN + Attention(Body)	Attention
NRMS	Word Embedding + Self-Attention + Attention	Self-Attention + Attention
PLM-NR	Pre-trained Language Model + Attention	-
FCF		Federated Collaborative Filtering
FedRec	Word Embedding + CNN + Self-Attention + Attention	Self-Attention + Attention + GRU
Efficient-FedRec	Pre-trained Language Model + Attention	Self-Attention + Attention
FedNRM(our)	BERT + Attention	Self-Attention + Personalized Attention + GRU

Appendix B: Weaknesses of some news recommendation models involved in this paper

Model	Weaknesses
NPA	Ignore the correlation between browsing news, ignore users' short-term interests, and have privacy issues
LSTUR	Ignore the correlation between browsing news, and have privacy issues
NAML	Ignore the correlation between browsing news, ignore users' short-term interests, and have privacy issues
NRMS	Ignore users' short-term interests, and have privacy issues
PLM-NR	Have privacy issues
FCF	Have cold start problems
FedRec	Ignore differences in individual attention
Efficient-FedRec	Ignore differences in individual attention, and ignore users' short-term interests