Efficient-FedRec: Efficient Federated Learning Framework for Privacy-Preserving News Recommendation

Jingwei Yi[1], Fangzhao Wu[2], Chuhan Wu[3], Ruixuan Liu[4], Guangzhong Sun[1], Xing Xie[2]

[1]University of Science and Technology of China [2]Microsoft Research Asia
[3]Tsinghua University [4]Renmin University of China

`yjw1029@mail.ustc.edu.cn {wufangzhao,wuchuhan15}@gmail.com`
`ruixuan.liu@ruc.edu.cn`
`gzsun@ustc.edu.cn xingx@microsoft.com`

Abstract

News recommendation is critical for personalized news access. Most existing news recommendation methods rely on centralized storage of users' historical news click behavior data, which may lead to privacy concerns and hazards. Federated Learning is a privacy-preserving framework for multiple clients to collaboratively train models without sharing their private data. However, the computation and communication cost of directly learning many existing news recommendation models in a federated way are unacceptable for user clients. In this paper, we propose an efficient federated learning framework for privacy-preserving news recommendation. Instead of training and communicating the whole model, we decompose the news recommendation model into a large news model maintained in the server and a light-weight user model shared on both server and clients, where news representations and user model are communicated between server and clients. More specifically, the clients request the user model and news representations from the server, and send their locally computed gradients to the server for aggregation. The server updates its global user model with the aggregated gradients, and further updates its news model to infer updated news representations. Since the local gradients may contain private information, we propose a secure aggregation method to aggregate gradients in a privacy-preserving way. Experiments on two real-world datasets show that our method can reduce the computation and communication cost on clients while keep promising model performance.

1 Introduction

With the explosion of online information, the large quantities of news generated every day may overwhelm users and make them difficult to find the news they are interested in. To tackle this problem, many news recommendation methods (An et al., 2019; Wang et al., 2020; Wu et al., 2019a; Qi et al., 2021c) have been proposed to display news according to users' personalized interests. These methods are usually composed of two core modules, i.e., user model and news model. The user model is used to learn user representations from user historical click behaviors. For example, Wang et al. (2018) use a candidate-aware attention network as the user model to help capture user interests in candidate news. The news model is used to learn news representations from news content. For example, Wu et al. (2019c) apply multi-head self attention network to capture the interactions between words in news model. With the success of pre-trained language models (PLM) in NLP, a few PLM-empowered news recommendation methods have been proposed and achieve remarkable performance. For example, Wu et al. (2021b) apply pre-trained language models to enhance news modeling. However, these methods require centralized storage of user behaviors, which are highly privacy-sensitive (Shin et al., 2018). Collecting private user data has raised many concerns (Wu et al., 2019d). Moreover, due to the adoption of some data protection regulations such as GDPR[1], it might not be able to analyze centralized user data in the future.

Federated learning (McMahan et al., 2017) is a privacy-preserving method to train models on the private data decentralized on a large number of clients. In federated learning, each user keeps a local copy of model, and compute local model gradients with their local private data. A central server coordinates the clients and aggregates local gradients to update the global model. Recently, Qi et al. (2020) proposed a FedRec method to train news recommendation models using federated learning. However, the model sizes of many existing news recommendation methods are large, especially their news models. For example, PLM-NR (Wu et al., 2021b) has 110.7M parameter in total, 110M of which are in the news model (BERT-Base version).

---

[1]https://gdpr-info.eu

Thus, the communication and computation costs of FedRec can be too high for clients with rather limited computation resource.

In this paper, we propose an efficient federated learning framework for privacy-preserving news recommendation named Efficient-FedRec[2]. In our framework, we decompose the news recommendation model into a large news model and a light-weight user model. Instead of training and communicating the whole model, in our approach the clients only request the user model and the representations of news involved in their local behaviors from the server. The clients locally compute the gradients of the user model and news representations on their local data, and send them to the server for aggregation. The central server uses the aggregated user model gradients to update its maintained global user model, and update the news model based on the aggregated news representation gradients. The updated news model is further used to infer updated news representations. The above process is repeated for multiple rounds until the model gets converges. In order to protect user privacy in model training, we develop a secure aggregation protocol based on the multi-party computation framework for privacy-preserving gradient aggregation. We exchange the news representations in the union news set involved by a group of user behaviors to protect the click history of a specific user. We conduct plenty of experiments on two real-world datasets and the results show that our approach can effectively reduce the computation and communication cost on clients for federated news recommendation model training.

The main contributions of this work include:

- We propose an efficient federated learning framework for privacy-preserving news recommendation, which can effectively reduce the computation and communication cost on the user side.

- We develop an effective and efficient secure aggregation protocol to protect user privacy in model training.

- We conduct thorough experiments on two real-world datasets to verify the effectiveness and efficiency of our approach.

2 Related Works

2.1 Neural News Recommendation

Personalized news recommendation is an important technique to alleviate the information overloading problem and improve user reading experience. Many deep learning based recommendation methods have been proposed (Wu et al., 2019b; Okura et al., 2017; Zhu et al., 2019; Qi et al., 2021b,a). They usually contain two core modules, i.e., user model and news model. For example, An et al. (2019) propose to use a CNN to learn contextual word embedding and an attention layer to select informative words. They combine long-term interests and short-term interests of users by using user id embeddings and a GRU network in user model. These methods learn news representations based on shallow NLP models, which is hard to well capture the news semantic information. Recently, pre-trained language models (PLM) achieve great success in NLP (Devlin et al., 2019; Liu et al., 2019; Bao et al., 2020). A few PLM-empowered news recommendation methods have been proposed. For example, Wu et al. (2021b) propose PLM-NR to empower news modeling by applying pre-trained language. They replace the news encoder in previous methods with pre-trained language models, and get stable improvement on news recommendation task. However, all the above methods train models based on centralized data, which is highly privacy-sensitive. Such kind of collections and analysis of private data have led to privacy concerns and risks (Shin et al., 2018). Besides, the adoption of some data protection regulations, such as GDPR[3], gives news platforms restrictions and high pressure of using user data to prevent user data leakage. Different from these methods, we do not use centralized storage for training in our framework, which can better preserve user privacy.

2.2 Federated Learning

Federated Learning (McMahan et al., 2017) is an effective method for privacy-preserving model training. It enables several users to collaboratively train models without sharing their data to a central server. In federated learning, users first request the latest updated model from central server, and compute local gradients with their local private data. Central server aggregates the gradients to update the global model and distributes the updated global model to

---

[2]https://github.com/yjw1029/Efficient-FedRec

[3]https://gdpr-info.eu

user local devices. Since the local gradients may leak some private information of users (Bhowmick et al., 2018; Melis et al., 2019), several privacy protection methods are applied, such as secure multiparty computation (MPC) (Knott et al., 2020), differential privacy (DP) (Ren et al., 2018), and homomorphic encryption (HE) (Aono et al., 2017).

Recently, several works have proposed to leverage federated learning in recommendation scenario. Ammad et al. (2019) propose federated collaborative filtering (FCF). In FCF, users use their private rate data to compute gradients of user embeddings and item embeddings. The user embeddings are updated locally by the gradients of user embeddings, and the gradients of item embeddings are aggregated to update global item embeddings. Chai et al. (2020) propose secure federated matrix factorization (FMF). FMF is similar to FCF but updates user embeddings and item embeddings according to matrix factorization algorithm. However, FCF and FMF are not suitable for news recommendation scenarios, since they represent items with ID embeddings and there is much fresh news generated every day. Qi et al. (2020) propose a privacy-preserving method for news recommendation model training. In FedRec, users use their local data to compute gradients of the model parameters. A group of randomly sampled users sends their local gradients to the central server to update the global model. However, the communication and computation cost of FedRec is unacceptable for user devices with limited resource due to the large size of news recommendation models, especially their news models. In this paper, we propose Efficient-FedRec to reduce the overhead on clients. We decompose the news recommendation model into a large news model maintained in server and a light-weight user model shared between clients and server. A small number of news representations and user model are communicated.

3 Methodology

In this section, we introduce our Efficient-FedRec method for privacy-preserving news recommendation. We first introduce the problem formulation and news recommendation framework. Then we introduce the details of our Efficient-FedRec framework. The details of secure aggregation are demonstrated in the last subsection.
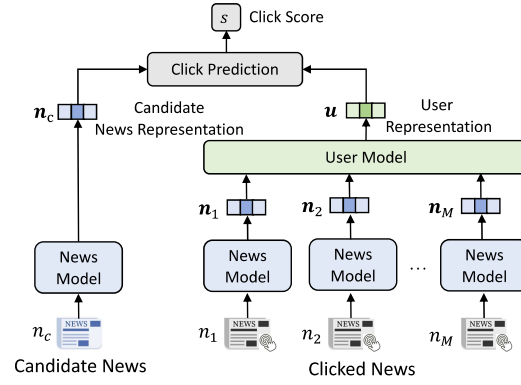

Figure 1: News recommendation framework.

3.1 Problem Formulation

Denote $\mathcal{U} = \{u_1, u_2, ... u_P\}$ as user set, where P is the user number. Given a user u, his private behaviors $\mathcal{B}_u$ are locally stored on his devices. In our approach, we denote all news in user behaviors of user u as $\mathcal{N}_u$. The news recommendation model is decomposed into a news model with parameter set Θ_n and a user model with parameter set Θ_u. The server maintains the news models and generates news representations with parameter set Θ_e, and keeps a global user encoder. The goal is to collaboratively train an accurate news recommendation model without leaking users' private information.

3.2 News Recommendation Framework

In this subsection, we introduce the news recommendation framework, which is shown in Figure 1. It is composed of two core modules, i.e., news model and user model.

News Model Given a news n, the news model is used to learn news representations **n** from news contents. It can be implemented by various model structures. Several existing news recommendation methods use shallow NLP models. Wu et al. (2019c) use a combination of multi-head self attention network and additive attention network, An et al. (2019) use a combination of CNN network and additive attention network. With the success of pre-trained language models (PLM) in NLP, a few methods start to apply pre-trained language models in news model. Wu et al. (2021b) propose PLM-NR, which uses a combination of pre-trained language model and additive attention network as news model. In our Efficient-FedRec, we apply the news model of PLM-NR (Wu et al., 2021b).
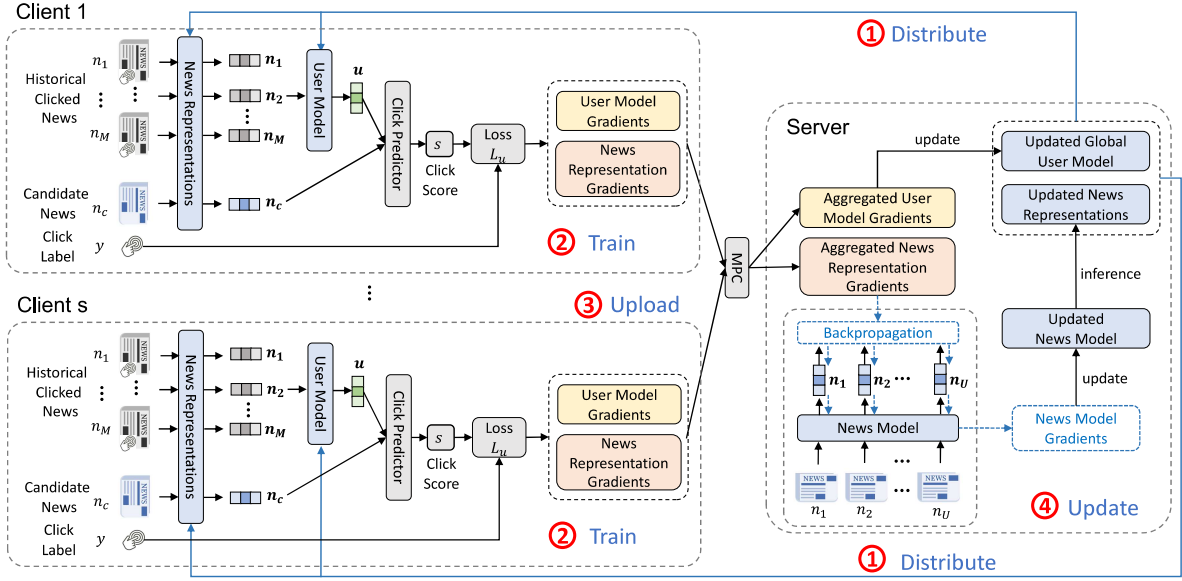
2816

Figure 2: The framework of Efficient-FedRec.

User Model The user model is used to learn user representations from user historical clicked news. Denoting the news representations of user historical clicked news $[\mathbf{n}_1, \mathbf{n}_2, ...\mathbf{n}_M]$ as input, the user model computes user representations $\mathbf{u}$ as output. It can be implemented by several model structures. Wang et al. (2018) use candidate-aware attention, and An et al. (2019) combines user id embeddings and GRU network. In our Efficient-FedRec, we apply the user model of NRMS (Wu et al., 2019c), which uses a combination of multi-head self attention network and additive attention network.

3.3 Framework of Efficient-FedRec

In this subsection, we introduce the framework of our Efficient-FedRec. Each user who participates in model training is called a client. In our framework, client behaviors are locally stored on their devices, which prevents the risks of data leakage. Since data of a single user is not enough to train an intelligent news recommendation model, our framework enables multiple clients to collaboratively train a news recommendation model. To lower the communication and computation overhead on the client side, we decompose the news recommendation model into a large news model maintained on server and a light-weight user model shared on both server and clients. At the t-th round, the model updating contains four steps, i.e., distributing user model and news representations, training local user model and news representations, gradient aggregation and global model updating. The framework of

our Efficient-FedRec is shown in Figure 2.

The first step is distributing user model and news representations. Since the news model is heavy and users only need the news representations to predict click scores, in our framework users request a small number of news representations in their behaviors and user model instead of the whole model from the central server. However, directly requesting the news representations of news in user behaviors $\mathcal{N}_u$ will leak user private information. In our work, we randomly sample a group of clients, who exchange the representations of union news set involved by a group of user behaviors through a secure aggregation protocol (introduced in Section 3.4). Denoting the group of clients as $\mathcal{U}_s = \{u_1, u_2...u_s\}$, the union news set is computed as $\mathcal{N}_s = \cup_{u_i \in \mathcal{U}_s} \mathcal{N}_i$. Thus the server only knows the news accessed by a group of clients. Finally, users keep a local copy of user model Θ_u^t, and news representations of union news set $\Theta_{e_s}^t$. It is noted news representations of union news set is much smaller than the news model (analyzed in Section 4.3), which alleviates the communication cost.

The second step is training local user model and news representations. Given a client u, we use his historical clicked news representations to compute user representations $\mathbf{u}$ through the local user encoder. For a candidate news n_c, we use the candidate news representation $\mathbf{n}_c$ and the user representation $\mathbf{u}$ to compute a click score s through a click predictor, which is dot-product in our framework. Following the previous work (Wu et al., 2019c; An

2817

et al., 2019; Qi et al., 2020), we utilize categorical cross-entropy loss for training. More specifically, for every clicked candidate news, we sample K non-clicked news in the same impression. Denote the label of the i-th news and user u is y^i and the prediction score is s^i, the loss of a training sample is computed as follows:

$$\mathcal{L}_u^j = -\sum_{i=1}^{K+1} y^i \times log(\frac{exp(s^i)}{\sum_{k=1}^{K+1} exp(s^k)}). \quad (1)$$

The final loss is the average loss of all training samples in $\mathcal{B}_u$, which is computed as follows:

$$\mathcal{L}_u = -\frac{1}{|\mathcal{B}_u|}\sum_{j=1}^{|\mathcal{B}_u|} \mathcal{L}_u^j \quad (2)$$

Denote the local gradients of user encoder as g_{vu}^t and the local gradients of news representations as $g_{e_s u}^t$, which are computed as follows:

$$g_{vu}^t = \frac{\partial \mathcal{L}_u}{\partial \Theta_u^t}, \quad g_{e_s u}^t = \frac{\partial \mathcal{L}_u}{\partial \Theta_{e_s}^t}. \quad (3)$$

In this step, since clients only compute the user model, the computation cost on clients is alleviated.

The third step is gradient aggregation. The server needs to compute the weighted sum of gradients of user model, news representations, and the sample number from the randomly sampled user group $\mathcal{U}_s$. Since the local gradients may contain some private information (Bhowmick et al., 2018; Melis et al., 2019), we apply the secure aggregation to compute the summations (introduced in Section 3.4). The aggregated gradients of user model and news representations are denoted as $\overline{g}_v^t$ and $\overline{g}_{e_s}^t$, which are formulated as follows:

$$\overline{g}_v^t = \frac{1}{\sum_{u\in\mathcal{U}_s}|\mathcal{B}_u|}\sum_{u\in\mathcal{U}_s}|\mathcal{B}_u|\cdot g_{vu}^t,$$
$$\overline{g}_{e_s}^t = \frac{1}{\sum_{u\in\mathcal{U}_s}|\mathcal{B}_u|}\sum_{u\in\mathcal{U}_s}|\mathcal{B}_u|\cdot g_{e_s u}^t, \quad (4)$$

It is noted that each user only sends the gradients of news representations in the union news set $\mathcal{N}_s$, which is much smaller than the news model. Thus we alleviate the communication overhead.

The final step is global model updating. The global user model and news model are updated separately. The global user model is directly updated by gradients of user model through FedAdam

algorithm (Reddi et al., 2021) as follows:

$$\Delta_u^t = \beta_1\Delta_u^{t-1} + (1-\beta_1)\overline{g}_u^t,$$
$$v_u^t = \beta_2 v_u^{t-1} + (1-\beta_2)\Delta_u^{t\,2},$$
$$\Theta_u^{t+1} = \Theta_u^t + \eta\frac{\Delta_u^t}{\sqrt{v_u^t + \tau}}, \quad (5)$$

where η is the learning rate, β_1, β_2 and τ are parameters of FedAdam. The news model is updated through a backpropagation training process. For each news in the union news set $n_i \in \mathcal{N}_s$, the central server has its content and the gradients of its news representation $\overline{g}_{e_i}^t \in \overline{g}_{e_s}^t$. We use the news content as input and compute its news representation $\mathbf{n}_i$ through the news model. The gradients of news model $\overline{g}_n^t$ are computed as follows:

$$\overline{g}_n^t = \sum_{n_i\in\mathcal{N}_s} \overline{g}_{e_i}^t \cdot \frac{\partial \mathbf{n}_i}{\partial \Theta_n^t}, \quad (6)$$

where Θ_n^t is the parameters of news model at the t-th round. We use Adam optimizer to updated new model, which in computed as follows:

$$\Delta_n^t = \beta_1\Delta_n^{t-1} + (1-\beta_1)\overline{g}_n^t,$$
$$v_n^t = \beta_2 v_n^{t-1} + (1-\beta_2)\Delta_n^{t\,2},$$
$$\Theta_n^{t+1} = \Theta_n^t + \eta\frac{\Delta_n^t}{\sqrt{v_n^t + \tau}}, \quad (7)$$

where η is the learning rate, β_1, β_2 and τ are hyper parameters of Adam. We further use the updated news model to infer news representations. Finally, the updated news representations and user encoder are distributed to all clients.

3.4 Secure Aggregation

In this subsection, we first introduce secure aggregation proposed by Bonawitz et al. (2017), and then introduce how we apply it to our framework for secure gradients aggregation and news representations distributing. The secure aggregation is mainly based on multi-party computation (MPC). It aims to let central server compute weighted sum of vectors without accessing the local vectors of each client in federated learning scenario. Denoted the local vectors of clients as $\{\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_n\}$, the secure aggregation computes $\mathbf{v} = \sum_{i=1}^n \mathbf{v}_i$ in a privacy-preserving way. Meanwhile, it solves the user drop problem on mobile devices.

As we introduce in Section 3.3, we use the secure aggregation twice. The first time is to compute the union news set $\mathcal{N}_s$ of a group of users. Given a user

u_i, we first transform his local news set $\mathcal{N}_i$ into a local vector $\mathbf{h}_i$, of which dimension equals the number of all news. The $\mathbf{h}_i$ is defined as follows:

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

where $\mathbf{h}_i^j$ is the j-th dimension of $\mathbf{h}_i$, and n_j is the j-th news in the total news set. We apply secure aggregation to compute the sum of vectors $\mathbf{h} = \sum_{u_i \in \mathcal{U}_s} \mathbf{h}_i$. The inverse transformation of Eq 8 is used to compute the union news set $\mathcal{N}_s$ from $\mathbf{h}$. The sampled group of users then request the news representations in the union news set $\mathcal{N}_s$ from central server.

The second time is to securely aggregate gradients. Each user flattens their local weighted gradients of news representations $|\mathcal{B}_u| \cdot g_{e_s u}^t$, local gradients of user model $|\mathcal{B}_u| \cdot g_{vu}^t$ and their sample number $|\mathcal{B}_u|$ to a vector, and applies secure aggregation to compute the summation. It is noted that only the news in the union news set has the gradients of news representations.

4 Experiments

In this section, we demonstrate the efficiency and effectiveness of our Efficient-FedRec. We conduct several experiments to answer the following research questions:

- **RQ1:** How does our method perform compared with baseline methods?

- **RQ2:** Are the communication and computation overhead significantly reduced compared with baseline methods?

- **RQ3:** How does the news model size influence the performance and overhead of our framework?

- **RQ4:** How does the user group size influence the risk of user information leakage and the effectiveness of our method?

- **RQ5:** How does the user number influence the performance of our framework?

4.1 Dataset and Experimental Settings

We conduct thorough experiments on two public datasets, i.e., MIND[4] and Adressa[5]. MIND[6] (Wu

[4] https://msnews.github.io/
[5] http://reclab.idi.ntnu.no/dataset/
[6] We use the small version of MIND for fast experiments.

	MIND	Adressa
#news	65,238	20,428
#users	94,057	640,503
#impressions	230,117	-
#positive samples	347,727	3,101,991
#negative samples	8,236,715	-

Table 1: Statistics of MIND and Adressa datasets.

et al., 2020b) is a public dataset collected on Microsoft News website in six weeks. Adressa (Gulla et al., 2017) is publicly released by Adresseavisen, a local newspaper company in Norway. Following (Qi et al., 2020) and (Hu et al., 2020), we use the 6-th day's click to build training dataset and construct historical clicks from the first 5 days' samples. We randomly sample 20% clicks from the last day's clicks for validation and the rest clicks for testing. The historical clicks of validation and testing dataset are constructed from the first 6 days' samples. Since Adressa does not contain negative samples, we randomly sample 20 news for each click for testing. The detailed dataset statistics are summarized in Table 1. Following many previous news recommendation works (Wu et al., 2020b; An et al., 2019; Qi et al., 2020; Wu et al., 2021a, 2020a), we use AUC, MRR, nDCG@5 and nDCG@10 as evaluation metrics.

In our experiments, we apply BERT-Base (Devlin et al., 2019) for MIND and nb-bert-base (Kummervold et al., 2021) for Adressa to initialize the pre-trained language model in news encoder. The dimension of news representations is 400. To mitigate overfitting, we apply dropout in user model. The dropout rate is 0.2. The learning rate is 0.00005. The number of negative samples associated with each positive sample is 4. The user group size is 50 on both MIND and Adressa. All hyper-parameters are selected according to results on the validation set. We repeat each experiment 5 times independently, and report the average results with standard deviations.

4.2 Performance Evaluation (RQ1)

In this section, we compare our Efficient-FedRec framework for privacy-preserving news recommendation with several baseline methods, including news recommendation methods with centralized storage: (1) DFM (Lian et al., 2018), a multi-channel deep fusion model for news recommendation; (2) DKN (Wang et al., 2018), a knowledge-aware news recommendation method;

Method	MIND				Adressa			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
DFM	60.67±0.20	28.08±0.13	29.93±0.13	35.68±0.13	59.90±1.20	32.68±0.75	29.69±0.93	36.43±1.11
DKN	64.72±0.19	30.53±0.13	33.01±0.15	38.70±0.16	73.73±0.48	39.52±1.34	40.98±1.24	47.48±0.86
LSTUR	66.90±0.08	32.45±0.07	35.11±0.07	40.82±0.07	68.37±2.63	38.76±2.14	38.11±2.39	44.33±2.42
NAML	66.10±0.25	31.91±0.23	34.52±0.26	40.21±0.24	73.09±1.53	44.27±1.53	43.51±1.89	50.02±1.71
NRMS	66.67±0.21	32.25±0.09	49.88±0.11	40.74±0.11	75.31±0.94	42.24±0.92	44.66±1.50	48.46±1.19
CenRec	66.92±0.17	32.30±0.11	35.05±0.13	40.78±0.14	72.85±1.53	40.82±1.73	41.62±2.24	47.54±1.47
PLM-NR	67.79±0.29	33.16±0.18	36.08±0.21	41.81±0.21	78.20±1.28	47.26±1.73	48.41±2.10	54.60±1.64
FCF	50.02±0.24	22.37±0.18	22.77±0.17	29.02±0.17	51.39±0.74	18.98±1.57	15.42±1.72	22.94±1.30
FedRec	66.54±0.18	31.96±0.07	34.54±0.09	40.30±0.09	71.73±1.72	41.37±2.21	41.81±2.35	47.18±2.09
FedRec(BERT)	67.45±0.10	32.80±0.10	35.44±0.16	41.35±0.14	78.60±1.82	43.81±0.95	45.76±0.89	52.64±1.68
Efficient-FedRec	67.44±0.20	32.79±0.06	35.62±0.06	41.35±0.07	79.08±1.18	45.09±1.87	47.13±2.35	53.85±1.69

Table 2: Results of different news recommendation methods.

(3) LSTUR (An et al., 2019), using user id embedding to capture user long-term interests, and GRU network to capture short-term interests; (4) NAML (Wu et al., 2019a), learning news representations via multi-view learning; (5) NRMS (Wu et al., 2019c), using two self-attention networks for better news and user modeling; (6) Cen-Rec (Qi et al., 2020), a central version of FedRec; (7) PLM-NR (Wu et al., 2021b), applying pre-trained language model to empower the performance of news recommendation. For fair comparison, we use the user model in NRMS. privacy-preserving news recommendation methods: (8) FCF (Ammad et al., 2019), federated collaborative filtering for recommendation; (9) FedRec (Qi et al., 2020), privacy-preserving method for news recommendation model training. For fair comparison, we do not add differential privacy; (10) FedRec(BERT), applying FedRec to train PLM-NR in a privacy-preserving way. our method: (11) Efficient-FedRec, using our Efficient-FedRec framework to train PLM-NR in a privacy-preserving and efficient way. The experimental results of all these methods are shown in Table 2.

We have several observations from Table 2. First, comparing our Efficient-FedRec with SOTA news recommendation methods with centralized storage (DKN, NAML, NRMS, LSTUR and PLM-NR), our method achieves comparable performance. Moreover, our method does not need users to share their behavior data. Therefore, it validates our method can train accurate news recommendation models and meanwhile protect user privacy. Second, our method performs better than FCF. This is because FCF is not suitable for news recommendation, since there are severe cold-start problems in news recommendation scenario (Qi et al., 2020; Wu et al., 2020b). Third, our Efficient-FedRec outper-
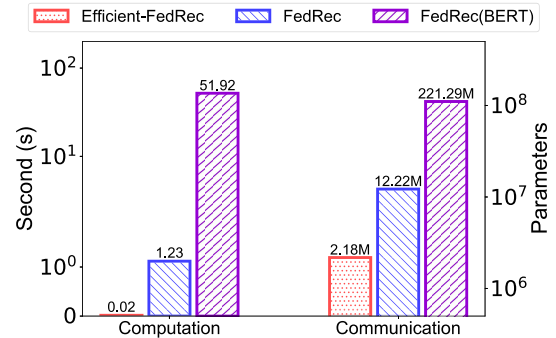

Figure 3: The communication cost and computation cost of privacy-preserving methods on MIND.

forms FedRec. This is because we use pre-trained language model in news model, which can help better understand the semantics of news contents. Forth, comparing our Efficient-FedRec with Fe-dRec(BERT), our Efficient-FedRec achieves comparable performance. This is because our method has the same gradients as FedRec(BERT) if dropout and batch normalization are not applied in news model. Finally, FedRec(BERT) and Efficient-FedRec perform worse than PLM-NR, and FedRec performs worse than CenRec. This is probably because user behaviors are non-i.i.d, which may make it difficult for federated learning to achieve good results (McMahan et al., 2017; Wang et al., 2019).

4.3 Efficiency Analysis (RQ2)

In this subsection, we analyze the communication and computation cost of our Efficient-FedRec on MIND. The average size of the union news set is 1,320 per round, the gradient and parameter size of which is 1.06M. We assume users leverage CPU for calculation. Figure 3 shows the average computation time and the communication overhead of each user per round of several privacy-preserving

BERT	AUC	Efficient-FedRec			FedRec		
		Comm. Cost (client)	Comp. Cost (client)	Comp. Cost (server)	Comm. Cost (client)	Comp. Cost (client)	Comp. Cost (server)
Tiny	64.21	2.18M	0.02s	2.05s	10.01M	0.69s	0.01s
Mini	65.55	2.18M	0.02s	3.20s	23.74M	2.44s	0.01s
Small	65.92	2.18M	0.02s	5.88s	59.32M	9.03s	0.01s
Medium	67.05	2.18M	0.02s	6.39s	84.54M	19.55s	0.01s
Base	67.44	2.18M	0.02s	6.74s	221.29M	51.92s	0.02s
Large	67.50	2.18M	0.02s	8.81s	673.28M	117.04s	0.04s

Table 3: Results of different news models on MIND.

(a) Union news set size. (b) Convergence round. (c) Overall comm. cost. (d) Secure aggregation time.
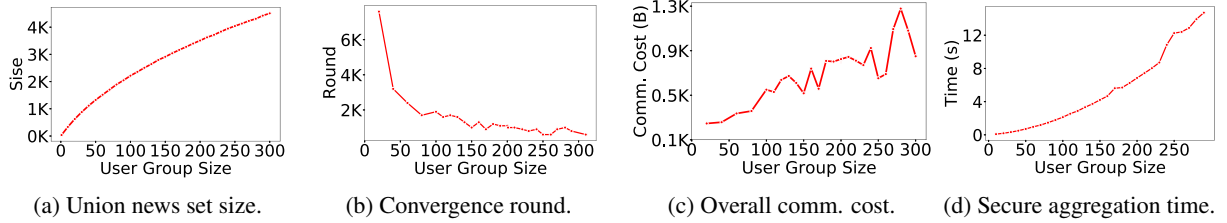
Figure 4: Impact of user group size on MIND.

methods. From Figure 3, we have several observations. First, the average computation time of Efficient-FedRec is lower than those of FedRec and FedRec(BERT). This is because in our framework users do not need to compute the news model, which lowers the computation overhead. Second, the communication overhead of Efficient-FedRec is much lower than the overhead of FedRec and FedRec(BERT). This is because in our framework users request and send the gradients and parameters of user model and a small number of news representations, which is much smaller than the gradients and parameters of the whole model.

4.4 The Influence of News Model Size (RQ3)

In this subsection, we apply different size of BERTs in news model to study the influence of the news model size on MIND. The computation cost of clients is tested on CPU, while the computation cost of server is tested on GPU, which is reasonable since clients are usually with limited computation resource. The result are shown in Table 3, where we have several observations. First, the recommendation performance increases with the news model size, which shows the effectiveness of applying large news model. Second, the communication and computation cost of our method on clients are lower than FedRec. This is because in Efficient-FedRec clients only compute the user model and request the user model and the repre-

sentations of news involved in their local behaviors. Additionally, the gap of the overhead between Efficient-FedRec and FeRec becomes larger with larger news model, which demonstrates the superiority of our method in using large news models. Third, the computation overhead of our method on server is larger than FedRec. It is because in our framework the news model is trained on central server. However, the overall computation time of Efficient-FedRec is lower than FedRec. This is because the server can use powerful GPU clusters to update the news model. It is noted that we simulate client computation cost with 100% CPU utilization. The computation time on real-time devices will be larger than the results reported in Table 3.

4.5 Influence of User Group Size (RQ4)

In this section, we study the influence of user group size on union news set size, convergence round, overall communication cost and secure aggregation time. The results are shown in Figure 4. As shown in Figure 4a, with the increasing of user group size, the size of union news set increases. When user group size is 40, the average size of union news set is 1,115, which is 10 times larger than the average size of user local news set, i.e., 114. Therefore, when user group size is large enough, it is hard for server to recover interacted news of users. Then, we study the impact of user group size on communication cost. Since larger user group
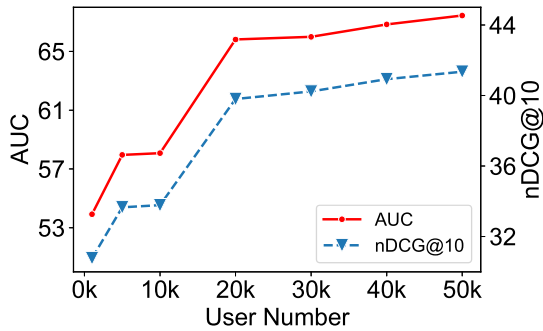
Figure 5: Results of different user numbers on MIND.

size leads to larger union news set size, the communication cost of per user increases. In Figure 4b we also find larger user sizes can make the model converge faster. The impact of user group size on overall communication cost is shown in Figure 4c, which is influenced by group user size, communication cost per user and convergence round. It is shown the overall communication cost increases with larger user group size. Finally, we study the impact of user group size on secure aggregation time (per user). As shown in Figure 4d, the computation cost of secure aggregation increases with larger user group size. The computation time of secure aggregation is 0.71s when user group size is 50. Considering privacy protection ability, communication cost and secure aggregation cost, we set user group size as 50 in our experiment on MIND.

4.6 Influence of User Number (RQ5)

In this subsection, we study the influence of the number of users who participate in model training. We randomly sample different numbers of users from MIND. The experimental results are shown in Figure 5. We can observe the performance increases with higher user numbers, which validates the idea of training news recommendation collaboratively with a large size of users. Moreover, it shows our Efficient-FedRec can effectively explore useful information from multiple user behaviors.

5 Conclusion

In this paper, we propose an efficient federated learning framework for privacy-preserving news recommendation named Efficient-FedRec. We decompose the news recommendation model into a large news model maintained by server and a lightweight user model. Users request news representations and user model from the central server and compute gradients with user local data. The cen-

tral server aggregates gradients to update the user model and news model. The updated news model is further used to infer news representation by server. In order to protect the private information in user local gradients, we apply secure aggregation to aggregate gradients. In order to protect user interacted news history, we exchange the news representations in the union news set involved by a group of user behaviors. Experiments on two real-world datasets validate our method can effectively reduce both communication and computation cost on user side while keep the model performance.

Ethical Statements

User Information Protection in Dataset In this paper, we conduct experiments on two public datasets, i.e., MIND and Adressa. MIND dataset was released in (Wu et al., 2020b). It is a public English news recommendation dataset. In this dataset each user was de-linked from the production system when securely hashed into an anonymized ID using onetime salt mapping to protect user privacy. We have agreed with Microsoft Research License Terms[7] before downloading this dataset and complied with these license terms when using this dataset. Adressa dataset was released in (Gulla et al., 2017). It is a public Norwegian news recommendation dataset. The users in this dataset are anonymized to protect user privacy. We follow the dataset license[8] when using this dataset. Thus, all the datasets used in our paper are public datasets where user privacy information is well protected.

Influence of User Group Size The user groups consist of randomly sampled users in each round to update model according to our framework. We conduct experiments to analyze the influence of user group size, and the results are summarized in Section 4.5. The experimental results show that as long as the user group size is properly large, which is usually easy to satisfy in practical applications, the information of user interacted news can be well protected.

---

[7]https://msnews.github.io/
[8]https://creativecommons.org/licenses/by-nc-sa/4.0/

References

Muhammad Ammad, E. Ivannikova, S. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and A. Flanagan. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. *ArXiv*, abs/1901.09888.

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 *ACL*, pages 336–345.

Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. 2017. Privacy-preserving deep learning via additively homomorphic encryption. *TIFS*, 13(5):1333–1345.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *ICML*, pages 642–652.

Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. 2018. Protection against reconstruction and its applications in private federated learning. *ArXiv*, abs/1812.00984.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *CCS*, pages 1175–1191.

Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2020. Secure federated matrix factorization. *IEEE Intelligent Systems*.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The adressa dataset for news recommendation. In *WI*, page 1042–1048.

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 *ACL*, pages 4255–4264.

B. Knott, S. Venkataraman, A.Y. Hannun, S. Sengupta, M. Ibrahim, and L.J.P. van der Maaten. 2020. Crypten: Secure multi-party computation meets machine learning. In *Proceedings of the NeurIPS Workshop on Privacy-Preserving Machine Learning*.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne" Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29.

Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2018. Towards better representation learning for personalized news recommendation: a multi-channel deep fusion approach. In *IJCAI*, pages 3805–3811.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *SP*, pages 691–706.

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*, pages 1933–1942.

Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021a. Personalized news recommendation with knowledge-aware interactive matching. In *SIGIR*, page 61–70.

Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021b. PP-Rec: News recommendation with personalized user interest and time-aware news popularity. In *ACL*, pages 5457–5467.

Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-preserving news recommendation model learning. In *EMNLP 2020*, pages 1423–1432.

Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie, and Yongfeng Huang. 2021c. HieRec: Hierarchical user interest modeling for personalized news recommendation. In *ACL*, pages 5446–5456.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Brendan McMahan. 2021. Adaptive federated optimization. *ICLR*.

Xuebin Ren, Chia-Mu Yu, Weiren Yu, Shusen Yang, Xinyu Yang, Julie A McCann, and S Yu Philip. 2018. High-dimensional crowdsourced data publication with local differential privacy. *TIFS*, 13(9):2151–2166.

Hyejin Shin, Sungwook Kim, Junbum Shin, and Xiaokui Xiao. 2018. Privacy enhanced matrix factorization for recommendation with local differential privacy. *TKDE*, 30(9):1770–1782.

Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *ACL*, pages 836–845.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *WWW*, pages 1835–1844.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *JSAC*, 37(6):1205–1221.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with attentive multi-view learning. In *IJCAI*, pages 3863–3869.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. Npa: neural news recommendation with personalized attention. In *KDD*, pages 2576–2584.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019c. Neural news recommendation with multi-head self-attention. In *EMNLP-IJCNLP*, pages 6390–6395.

Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021a. User-as-graph: User modeling with heterogeneous graph pooling for news recommendation. In *IJCAI*, pages 1624–1630.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020a. User modeling with click preference and reading satisfaction for news recommendation. In *IJCAI*, pages 3023–3029.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021b. Empowering news recommendation with pre-trained language models. In *SIGIR*, page 1652–1656.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020b. MIND: A large-scale dataset for news recommendation. In *ACL*, pages 3597–3606.

Zhanpeng Wu, Yipeng Zhou, Di Wu, Min Chen, and Yuedong Xu. 2019d. Tamf: Towards personalized time-aware recommendation for over-the-top videos. In *NOSSDAV*, pages 43–48.

Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. Dan: Deep attention neural network for news recommendation. In *AAAI*, volume 33, pages 5973–5980.