# A Federated Recommender System for Online Services

Ben Tan, Bo Liu, Vincent Zheng, Qiang Yang*
WeBank, China
btan,brodyliu,vincentz,qiangyang@webank.com

## ABSTRACT

Due to privacy and security constraints, directly sharing user data between parties is undesired. Such decentralized data silo issues commonly exist in recommender systems. In general, recommender systems are data-driven. The more data it uses, the better performance it obtains. The data silo issues is a severe limitation of the recommender's performance. Federated learning is an emerging technology, which bridges the data silos and builds machine learning models without compromising user privacy and data security. We design a recommender system based on federated learning. It is known as the federated recommender system. The system implements plenty of popular algorithms to support various online recommendation services. The algorithm implementation is open-sourced. We also deploy the system on a real-world content recommendation application, achieving significant performance improvement. In this demonstration, we present the architecture of the federated recommender system and give an online demo to show its detailed working procedures and results in content recommendations.

## CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Security and privacy** → **Software and application security**.

## KEYWORDS

Recommender Systems, Federated Learning

**ACM Reference Format:**
Ben Tan, Bo Liu, Vincent Zheng, Qiang Yang. 2020. A Federated Recommender System for Online Services. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3383313.3411528

## 1 INTRODUCTION

The recommender system (RecSys) plays an important role in real-world online applications. It has become an indispensable tool for coping with information overload and is one of the most successful applications of machine learning technologies in business. Protecting private data in RecSys is of critical importance. In order to provide reasonable recommendations, the RecSys need to know as much as possible from the user. In general, the more data is used in the RecSys, the better recommendation performance can be obtained [8]. Private user data, including the demographic information, the purchase history, the recommendation feedback, and so on, is always collected by the RecSys or transmitted to the RecSys from other data providers [6, 16]. However, in recent years, several acts protecting the privacy and security have come out, such as the General Data Protection Regulation (GDPR) [15]. Data collection and integration is becoming difficult. The protection of privacy and security is an essential part of current RecSys.

Federated learning [7, 17] is an emerging technology for decentralized and secure machine learning. It protects the data privacy of parties during the joint training of machine learning models. User private data is stored locally at each party. Only the intermediate results, *e.g.*, parameter updates, are used to communicate with other parties. Federated learning allows knowledge to be shared among multiple parties without compromising user privacy and data security. In this demonstration, we implement a recommender system based on federated learning. It is known as the federated recommender system (FedRecSys). The objective is to further optimize the recommendation performance by leveraging more data in a privacy-preserving manner. Our system accomplishes this goal by enabling the recommenders to collaboratively train a predictive model among multiple parties. The private data of each party is kept on their data repositories. The system has secure multi-party computation protocols based on homomorphic encryption [12] and secret sharing [9]. We also design and implement popular recommendation algorithms under the federated learning setting with secure computation protocols. These algorithms include the general matrix factorization (MF) [2, 4], singular value decomposition (SVD)[11], factorization machine (FM) [10], wide&deep learning [3]. They can support different recommendation scenarios, such as explicit and implicit feedback. In this demonstration, we also present the architecture of the FedRecSys, and give a demo to show the system's working procedure and results in the content recommendation.

## 2 SYSTEM ARCHITECTURE

The goal of the FedRecSys is to train a recommender model on data from multiple parties without revealing the private information of each party. As shown in Fig. 1, the system is composed of a data layer, an algorithm layer, a service layer, and an interface layer. In the following, we discuss how they support various online applications such as content recommendation, product recommendation, online advertising, *etc.*

The Data Layer facilitates independent data management for each party. The parties accumulate the terabyte-scale behavior data in the distributed storage, including HIVE [14] and HDFS [13]. A behavior record indicates a user's feedback for an item in a specific context. All parties store the user's profile, item's content, and context information in their own HDFS and participate in the

*Ben Tan and Bo Liu contributed equally to this research.


*RecSys '20, September 22–26, 2020, Virtual Event, Brazil*

ACM ISBN 978-1-4503-7583-2/20/09.
https://doi.org/10.1145/3383313.3411528

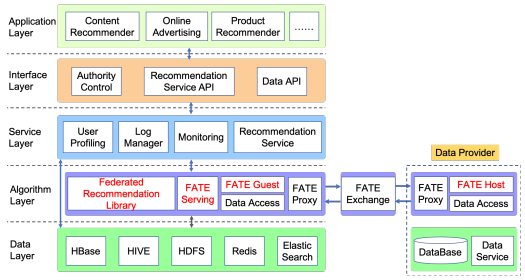


**Figure 1: The architecture of the Federated Recommender System.**

collaborative federated modeling. Our system adopts Redis [1] and Elasticsearch [5] to memorize and query the past behaviors of the specific user in real-time.

The Algorithm Layer is the most important part in FedRecSys, which now has general matrix factorization, SVD, factorization machine, wide&deep learning under the federated learning setting. The algorithms are built on FATE [1], which is an open-source project and provides a secure computing framework to support the federated AI ecosystem. The algorithms in this layer have been merged into the FATE project and open-sourced [2]. They can be used for everyone. The detailed introduction and implementation of the algorithms are also given in the project. In this layer, the parties build recommender together in a privacy-preserving manner. The "Data Access", "FATE Proxy" and "FATE Exchange" components aim to perform secure instance mapping, encrypt model parameters and transmit them between parties. The "Federated Recommendation Library", "FATE Guest" and "FATE Host" do the model updating and optimization. To protect user privacy and data security, the "FATE Guest" and "FATE Host" keep each party's own data and update the partial model locally. In general, "FATE Host" is the data provider party, the "FATE Guest" is the data consumer party, like a recommender. "FATE Severing" provides online prediction by combining the partial prediction results from the "FATE Host" and "FATE Guest".

The Service Layer is composed of four components, including user profiling, log manager, monitoring, and recommendation services. Given the user profiles and the recommender, the service layer decides the preferable items in a coarse-to-fine manner. In particular, the system filters the majority of items according to the user behaviors and ranks the remaining items using recommender models. Meantime, the log manager records users' behavior and write into the data layer for further analysis. Finally, the monitoring system guarantees the robustness by tracking the status of the system in real-time and issuing warnings when necessary.

In the Interface Layer, various online applications, including content, product, and ads recommendation, interact with the aforementioned layers via a unified interface layer. This layer protects the security of the data and the system by controlling the authorities for all participants. Furthermore, the interface layer allows the applications to store, modify, and query data securely from the service layer. Finally, the interface layer abstracts the communication between applications and the service layer.

[1]https://github.com/FederatedAI/FATE
[2]https://github.com/FederatedAI/FedRec

## 3 APPLICATION AND DEMONSTRATION

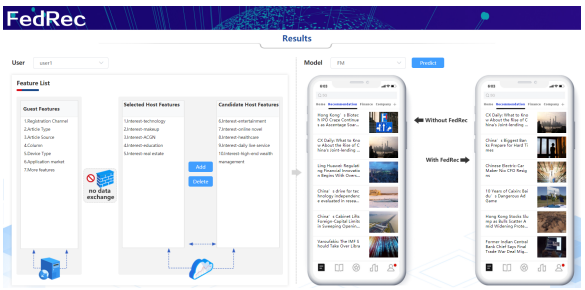


**Figure 2: The screenshot of the content recommender demo. (Link)**

In this section, we introduce a FedRecSys for content recommendation. A content application on mobile-device and a data provider desire to build a recommender model collaboratively in a privacy-preserving manner. The recommender has the device information, the browsing history of users, the metadata of the contents. The data provider has demographic information and user interests. Integrating user information from the data provider helps the recommender know more about the user and can make more reasonable recommendations. The system is running on real-world application, improving the click-through rate by 11% and increasing the average reading time by 22%. Besides, as shown in Fig. 2, we provide a demo, which is publicly available online at https://ad.webank.com/fedrecdemo/index.html?type=en. In the demo, we demonstrate how the auxiliary information determines the ranking results on an individual user. One can select a specific user, choose auxiliary user interest features from the data provider, and make predictions with an algorithm such as FM. The demo shows the results on the screens. By comparing the results, we can see the difference when the recommendations are made with or without auxiliary user interest features. For example, given auxiliary data indicating that a user is interested in "Wealth Management", our FedRecSys can rank finance articles higher than other non-finance articles. In the demo, we also show the iterative parameter updating procedure and the loss curve for popular recommendation algorithms, such as FM.

## REFERENCES

[1] Josiah L Carlson. 2013. *Redis in action.* Manning Publications Co.
[2] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2019. Secure federated matrix factorization. *arXiv preprint arXiv:1906.05108* (2019).
[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems.* 7–10.
[4] Dashan Gao, Ben Tan, Ce Ju, Vincent W Zheng, and Qiang Yang. 2020. Privacy Threats Against Federated Matrix Factorization. *arXiv preprint arXiv:2007.01587* (2020).
[5] Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine.* " O'Reilly Media, Inc.".

[6] Arjan JP Jeckmans, Michael Beye, Zekeriya Erkin, Pieter Hartel, Reginald L Lagendijk, and Qiang Tang. 2013. Privacy in recommender systems. In *Social Media Retrieval*. 263–281.

[7] Qinbin Li, Zeyi Wen, and Bingsheng He. 2019. Federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693* (2019).

[8] Weike Pan. 2016. A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing* 177 (2016), 447–453.

[9] Mohassel Payman and Zhang Yupeng. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. *IACR Cryptology ePrint Archive* (2017), 396.

[10] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 10th IEEE International Conference on Data Mining*. 995–1000.

[11] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 1–35.

[12] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. 1978. On data banks and privacy homomorphisms. *Foundations of Secure Computation* 4, 11 (1978), 169–180.

[13] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. The Hadoop Distributed File System. In *IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST 2012, Lake Tahoe, Nevada, USA, May 3-7, 2010*. 1–10. https://doi.org/10.1109/MSST.2010.5496972

[14] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. 2009. Hive: a warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment* 2, 2 (2009), 1626–1629.

[15] European Union. 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). (2016).

[16] Jun Wang and Qiang Tang. 2015. Recommender Systems and their Security Concerns. *IACR Cryptology ePrint Archive* 2015 (2015), 1108.

[17] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (2019), 12.