Published: 13 July 2025

Citation in BibTeX format

SIGIR '25: The 48th International ACM SIGIR Conference on Research and Development in Information Retrieval
July 13 - 18, 2025
Padua, Italy

Conference Sponsors:
SIGIR

DEMONSTRATION

NodeRec+: A Lightweight Framework for Federated Recommender Systems

DIARMUID O'REILLY-MORGAN, University College Dublin, Dublin, Leinster, Ireland

ERIKA DURIAKOVA, University College Dublin, Dublin, Leinster, Ireland

ELIAS Z TRAGOS, University College Dublin, Dublin, Leinster, Ireland

NEIL J HURLEY, University College Dublin, Dublin, Leinster, Ireland

AONGHUS LAWLOR, University College Dublin, Dublin, Leinster, Ireland

SIGIR '25: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (July 2025)
https://doi.org/10.1145/3726302.3730138
ISBN: 9798400715921

NodeRec+: A Lightweight Framework for Federated Recommender Systems

Diarmuid O'Reilly-Morgan*
Insight Centre for Data Analytics,
University College Dublin
Dublin, Ireland
diarmuid.oreillymorgan@insight-centre.org

Erika Duriakova
Insight Centre for Data Analytics,
University College Dublin
Dublin, Ireland
erika.duriakova@insight-centre.org

Elias Tragos
Insight Centre for Data Analytics,
University College Dublin
Dublin, Ireland
elias.tragos@insight-centre.org

Neil Hurley
Insight Centre for Data Analytics,
University College Dublin
Dublin, Ireland
neil.hurley@insight-centre.org

Aonghus Lawlor
Insight Centre for Data Analytics,
University College Dublin
Dublin, Ireland
aonghus.lawlor@insight-centre.org

Abstract

Data privacy is a critical concern in today's data-driven world. To this end, Federated Learning (FL) has been researched extensively, as it allows sensitive data to be kept secure on local devices while training global Machine Learning (ML) models across multiple devices. Several FL topologies have been introduced to address different users' needs. FL fits well within the Recommender Systems (RS) domain, with decentralised large-scale datasets and user privacy issues, as it improves personalised recommendations and accuracy while keeping users' sensitive data secure.

In this paper, we introduce NodeRec+, the prototype of an FL framework, geared towards enabling the design and evaluation of decentralised RS. NodeRec+ allows quick and easy composition of various FL topologies in simple Python code and comes ready integrated with a number of RS models and evaluation protocols, enabling rapid benchmarking of models in various FL topologies. In contrast to most similar frameworks that are either used for simulations or focused solely on a single FL topology, NodeRec+ provides a framework to build real decentralised networks of nodes that can work jointly using several FL topologies in RS settings. A demonstration video showcasing the NodeRec+ framework can be found at https://www.youtube.com/watch?v=IIo3Cr56YyI. Code is provided at https://github.com/SEDIMARK-UCD/noderec.

CCS Concepts

• **Computing methodologies → Distributed artificial intelligence**; • **Information systems → Recommender systems**.

*Corresponding author.

SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730138

Keywords

Federated Learning, Decentralised Learning, Recommender Systems.

ACM Reference Format:
Diarmuid O'Reilly-Morgan, Erika Duriakova, Elias Tragos, Neil Hurley, and Aonghus Lawlor. 2025. NodeRec+: A Lightweight Framework for Federated Recommender Systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3726302.3730138

1 Introduction

In recent years, data privacy has become a key concept in the design of next-generation distributed Machine Learning (ML) models. To this end, Federated Learning (FL) has been introduced as a way of keeping user's data locally. Differing from traditional data-centered distributed ML approaches, FL enables users to use local data to build local ML models and use aggregation strategies to collaboratively build a global ML model, omitting the need to share raw user data [18]. Since the introduction of centralised (server-based) FL in 2016 [12], FL has come to cover multiple topologies, including (i) centralised and (ii) decentralised topologies, wherein (i) central servers are used to coordinate the global training process, while in (ii) users communicate with each other directly without the need for a central server [4].

Recommender Systems (RS) leverage ML to generate personalised content based on user preferences. To this end, Deep neural network (DNN) based RS have become popular in the past decade, achieving very high accuracy in personalised recommendations [16]. However, such models require access to large amounts of sensitive user information. As a result, RS based on the FL paradigm have become a rapidly emerging research direction. This is due to issues surrounding both privacy and scalability that arise within centrally located RS. Very few tools exist specifically targeted toward testing FL-based RS at scale, with many research works instead relying upon locally run simulations. Moreover, even fewer frameworks exist that support distributed RS in a variety of FL topologies.

In this paper, we present NodeRec+, an asynchronous FL framework that integrates the RecBole library [20] for easy training and evaluation of RS models, with a flexible definition of the training protocol and FL topology. In particular, the strength of NodeRec+ lies in its ability to support multiple FL topologies to advance new research into FL RS. To this end, NodeRec+ evolves around several key concepts:

- **Modular Design:** NodeRec+'s modular design allows for plug-and-play of multiple FL communication topologies paired with a wide array of aggregation strategies and RS modules.
- **Support for multiple FL topologies:** Currently, NodeRec+ supports centralised (server-based) and decentralised (P2P based) FL topologies. However, the modular design allows extending NodeRec+ with additional FL topologies, such as hierarchical and cluster-based FL topologies.
- **Communication Protocol:** NodeRec+ is built on a simple HTTP request-response architecture. This allows for simulations to be carried out across different devices and platforms in an asynchronous fashion, closely mimicking real-world applications.
- **Recommender Systems Integration:** NodeRec+ integrates the RecBole [20] library for easy training and evaluation of recommender models, with a flexible definition of the training protocol and topology.

The rest of this paper is organised as follows: in Section 2 we discuss how NodeRec+ fits within the landscape of existing FL and RS frameworks. Then in Section 3 we describe the functionalities of NodeRec+ in greater detail. We then demonstrate NodeRec+ empirically in Section 4 and finally conclude in Section 5.

2 Related Work

In the context of centralised FL for general ML, several frameworks have emerged in recent years. Flower [5] primarily targets centralised FL workflows, is implemented in grpc, and supports many specialised FL algorithms out of the box, while providing reasonably good integration for most deep learning frameworks. The architecture includes the representation of the client (the entity for local training) and server (the orchestrator of the FL process). However, because of this basic architecture embedded in the Flower framework, other FL topologies are not directly supported. Similarly, the FedScale [9] framework provides a benchmarking suite to evaluate FL algorithms. This framework provides a wide array of datasets from several domains such as computer vision or natural language processing. The framework also includes one dataset for recommender systems, however, it focuses on centralised FL and as such it does not support other FL topologies out of the box.

In the context of decentralised FL for ML, [4] provides a detailed overview of mature FL frameworks, focusing in particular on FL frameworks supporting decetralised topologies. These include frameworks, such as Tensorflow Federated (TFF) [3], which extends Tensorflow for FL topologies, PySyft [21] which supports PyTorch as well as Tensorflow, or FedML [7] supporting both synchronous and asynchronous strategies. Several other frameworks have been released to date, where most support both centralised and decentralised FL (see [4] for more details). From the point of view of supporting multiple FL topologies, DecentralizePY [6] shares some similarities with NodeRec+, also basing itself around a composable node component, however DecentralizePy inherits its ML modules from PyTorch v1.10.2 without direct support for RS modules. Moreover, the above frameworks either focus on more general ML models, a limited number of pre-implemented RS models, or require users to build their own ML/RS models.

One of the key challenges in modern RS research is reproducibility. In recent years, several frameworks have been released in the RS community to address reproducibility issues and improve fair comparisons of novel RS models. The most advanced RS frameworks offer not only a large set of up-to-date RS models but also improve reproducibility by including features such as data filtering/splitting and hyper-parameter tuning capabilities. To this end several frameworks have been released, such as the DaisyRec [15], RecBole [19, 20], and Elliot [1]. However, none of the above RS frameworks integrate FL topologies, despite the large number of modern RS modules developed in the context of user privacy and FL [2]. One recent library, RF2 [11], integrates a number of pytorch models through the DeepCTR [14] library and allows users to run simulations, but only within a standard centralised FL topology. We propose NodeRec+, which is solely utilised for the RS domain and integrates the RecBole framework [20], which comes with over 60 pre-built RS algorithms and 20+ RS datasets, allowing for rapid evaluations of novel RS models in FL settings.

3 The NodeRec+ Framework

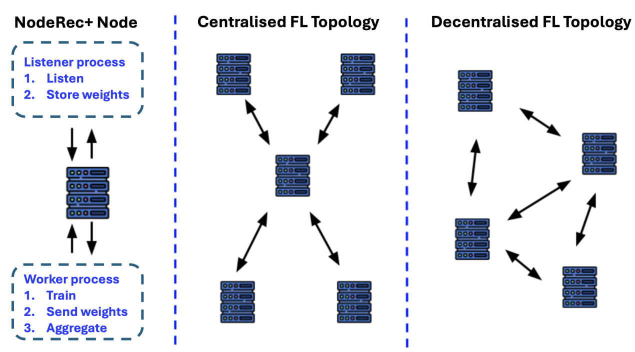

Figure 1: Depiction of the listener and worker processes that run in unison on a NodeRec+ node (left), and how nodes are combined into Centralised FL and Decentralised FL topologies (middle and right).

3.1 Concepts

As discussed above, NodeRec+ provides a framework for FL recommender systems, supporting multiple FL topologies while allowing the user to define custom topologies in a simple way. A high level depiction of the NodeRec+ framework is provided in Figure 1. NodeRec+ is built on several key concepts:

Modular Design: Similarly to DecentralizePy [6], NodeRec+ adopts a simple structure where the primary object is a Node (see Section 3.2) which communicates with other nodes over a decentralised network based on some pre-defined topology.

Support for multiple FL topologies: Using several basic primitives - (train, aggregate, send weights) - the user can write the basic logic for the FL training process of each node, also specifying the topology structure via the list of IP addresses to which each node has access. For instance, within a centralised FL topology, one node will act as a central server and listen and run periodic aggregation, while several other training nodes will run training processes and then communicate with the node that acts as a central server.

Communication Protocol: Communication is handled asynchronously using a request-response format, over HTTP. The listener process continuously listens in the background for incoming HTTP connections, and both deposits received weights within a queue for aggregation and communicates updated weights back to clients' processes. The worker process handles sending new weights when they are available.

Recommender Systems Integration: The provided implementation of NodeRec+ integrates the models and evaluation protocols from the Python library RecBole, enabling a number of different recommender systems experiments to be run. A user can specify a model, evaluation, and dataset configuration as per the general use of the RecBole library, and NodeRec+ will run the provided protocol in a distributed manner, and in a specified FL topology.

3.2 NodeRec+ Node

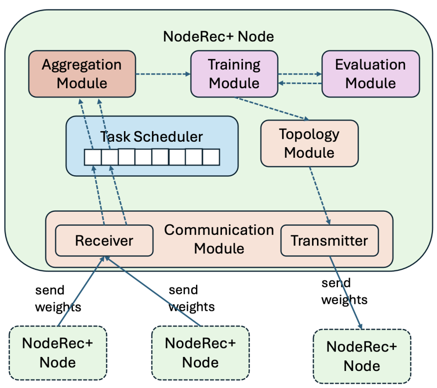

Figure 2: Internal structure of a NodeRec+ node.

Figure 2 shows the high-level internal structure of the NodeRec+ node. NodeRec+ uses the notion of a single node for all types of topologies, even for centralised federated learning, moving past the standard notion of server and client. A NodeRec+ node consists of six internal modules:

- **Communication module**: the Communication module handles the communication between the various NodeRec+ nodes. It consists of two sub-modules, namely (i) the Receiver and (ii) the Transmitter. The Receiver is the sub-module that receives the weights from the neighbouring Nodes that participate in the learning process. The Transmitter is the sub-module that after the local training process is completed, sends the updated weights to the selected target nodes. The communication module is built using HTTP REST as discussed above.
- **Task Scheduler**: this is the main module that handles the input from the peer NodeRec+ Nodes when new model weights are received. The module then schedules the processing of the weights from the Aggregation module according to the aggregation algorithm used. The current implementation of NodeRec+ runs the federated learning process in a fully asynchronous way, meaning that there is no notion of a global training "epoch" and nodes periodically aggregate the weights they receive, as long as a minimum number of weights has been received.
- **Aggregation Module**: this is the module that handles the aggregation of the weights received from peer nodes based on the aggregation algorithm used by the FL process.
- **Training Module**: the training module handles the local training of the model by using the updated aggregated weights from the Aggregation module.
- **Evaluation Module**: the evaluation module performs the model evaluation and sends back the metric results to the training module.
- **Topology Module**: this module handles the topology of the FL network and is mainly responsible for selecting the target nodes to send weights to. Users can configure any type of topology they want, varying from standard server-based, or centralised FL (by having all client nodes send weights to a single target node) to fully decentralised FL (by having all nodes send weights to everyone else in the network).

3.3 Extensibility

NodeRec+ is developed in a simple, modular fashion, such that it can be easily extended and integrated with other machine learning frameworks. For example, the provided implementation of NodeRec+ integrates the models and evaluation protocols from the Python library RecBole, enabling a number of different recommender systems experiments to be run. A user can specify a model, evaluation and dataset configuration as per the general use of the RecBole library, and NodeRec+ will seamlessly run the provided protocol in a distributed manner, and in a specified FL topology.

4 Empirical Demonstration/Evaluation

In this section, we demonstrate NodeRec+'s performance by deploying several different models in two basic FL topologies. In our demonstration video[1], we provide a hands-on demonstration of how to use NodeRec+, including the user interface shown in Figure 3.

4.1 Experimental Setup

We set up both centralised and decentralised FL protocols, with both 5 and 10 nodes. We evaluate on the ML100k[2] dataset, where we randomly assign users and their interactions to one of either 5 or 10 nodes [3]. We deploy 3 popular recommender systems models from the RecBole library, BPR [13], NGCF [17] and LightGCN [8], using their default hyperparameters. As NodeRec+ operates asynchronously, it is difficult to define the concept of a global epoch. After each local training epoch, a node logs its metrics on its local

[1]https://www.youtube.com/watch?v=IIo3Cr56YyI
[2]http://grouplens.org/datasets/movielens/100K/
[3]In the FL settings, the train/test splits within the same nodes were randomised and therefore these splits may have been different between the different FL settings, however for reproducibility purposes, NodeRec+ has the option to create fixed splits across FL settings.

Model	Topology	N_nodes	Precision@10	Max Iteration
BPR	CentralisedFL	10	0.3363	3767
	CentralisedFL	5	0.3421	1749
	DecentralisedFL	10	0.3380	2478
	DecentralisedFL	5	0.3405	948
	Local	0	0.3389	84
LightGCN	CentralisedFL	10	0.3173	3890
	CentralisedFL	5	0.3291	3536
	DecentralisedFL	10	0.3141	3448
	DecentralisedFL	5	0.3323	2350
	Local	0	0.3528	264
NGCF	CentralisedFL	10	0.3266	1271
	CentralisedFL	5	0.3391	793
	DecentralisedFL	10	0.3220	765
	DecentralisedFL	5	0.3436	512
	Local	0	0.3515	110

Table 1: Model Performance Comparison across Different Topologies

test set, and as such we report the maximum of a rolling average of the mean Precision@10 attained by all nodes in the network using local node iteration as an approximate point in time, with window size 50. We also report the local iteration at which that maximum is achieved. Additionally, we run localised implementations of the models for 500 epochs each, reporting the result from the best epoch. Observing that LightGCN and NGCF both struggle to reach localised accuracy in a naive implementation, we extend NodeRec+ with a FedProx [10] loss with $\mu = 0.0001$ for LightGCN and $\mu = 0.01$ for NGCF as regularization, and increase the learning rate for NGCF.

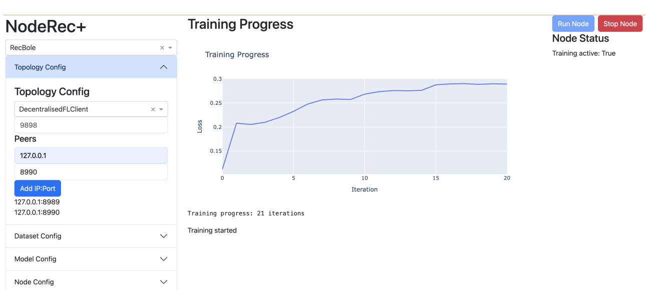

Figure 3: The NodeRec+ interface provided in the demo.

4.2 Results

We present the evaluation results in Table 1. Overall, the results show that NodeRec+ can approach localised metrics for the three given models. In terms of the speed of convergence, we can note that increasing the number of nodes across the two topologies increases the time it takes to approach the performance of the localised models. Further, we can observe that on this current task, NGCF takes the least number of iterations to reach the highest accuracy (in the case of decentralised FL on 10 nodes, it takes 6.96 times more iterations than the Local model). This is particularly important, as the overall number of iterations is directly proportional to the communication overhead of the FL model. In contrast, we can observe BPR run using the decentralised FL topology on 10 nodes takes 29.6 times more iterations than the local model to reach the highest accuracy. This result provides quick insights into the efficiency of various RS models in FL settings.

5 Conclusion and Discussion

In this demonstration paper, we have presented NodeRec+, a lightweight framework for developing and benchmarking novel RS models in the FL scenario. We introduced a modular FL framework capable of supporting multiple FL topologies and paired it with the RecBole library, specifically targeted towards the reproducibility and benchmarking of new models in the area of RS. NodeRec+ has been built with usability in mind so that users can easily and quickly set it up to run real decentralised learning either locally on a computer/server or remotely across distant servers. The framework will be open-sourced so that it can be easily used by researchers to test their models in realistic environments. The framework is currently used within the EU project SEDIMARK [4] running decentralised learning processes on distributed data.

In future work, we aim to further modularise NodeRec+ so that specialised distributed learning algorithms can be more readily integrated into the code and more FL topologies can be supported, such as graph-based and hierarchical FL topologies. We also plan to test NodeRec+ more thoroughly in multiple device setups and across multiple machines.

Acknowledgment

This work has been supported and funded by the EU Horizon Europe SEDIMARK (SEcure Decentralised Intelligent Data MARKetplace) project (Grant no. 101070074), and by the Taighde Éireann – Research Ireland through the Insight Centre for Data Analytics under grant number SFI/12/RC/2289_P2.

[4]https://sedimark.eu/

References

[1] Anelli, V.W., Bellogín, A., Ferrara, A., Malitesta, D., Merra, F.A., Pomo, C., Donini, F.M., Di Noia, T.: Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp. 2405–2414 (2021)

[2] Asad, M., Shaukat, S., Javanmardi, E., Nakazato, J., Tsukada, M.: A comprehensive survey on privacy-preserving techniques in federated recommendation systems. Applied Sciences **13**(10), 6201 (2023)

[3] Authors, T.T.F.: TensorFlow Federated: Machine Learning on Decentralized Data. https://www.tensorflow.org/federated (2023), version 0.x

[4] Beltrán, E.T.M., Pérez, M.Q., Sánchez, P.M.S., Bernal, S.L., Bovet, G., Pérez, M.G., Pérez, G.M., Celdrán, A.H.: Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. IEEE Communications Surveys & Tutorials (2023)

[5] Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Kwing, H.L., Parcollet, T., Gusmão, P.P.d., Lane, N.D.: Flower: A friendly federated learning research framework. arXiv preprint arXiv:2007.14390 (2020)

[6] Dhasade, A., Kermarrec, A.M., Pires, R., Sharma, R., Vujasinovic, M.: Decentralized learning made easy with decentralizepy. In: Proceedings of the 3rd Workshop on Machine Learning and Systems. pp. 34–41 (2023)

[7] He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., et al.: Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518 (2020)

[8] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 639–648 (2020)

[9] Lai, F., Dai, Y., Singapuram, S., Liu, J., Zhu, X., Madhyastha, H., Chowdhury, M.: Fedscale: Benchmarking model and system performance of federated learning at scale. In: International conference on machine learning. pp. 11814–11827. PMLR (2022)

[10] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)

[11] Maeng, K., Lu, H., Melis, L., Nguyen, J., Rabbat, M., Wu, C.J.: Towards fair federated recommendation learning: Characterizing the inter-dependence of system and data heterogeneity. In: Proceedings of the 16th ACM Conference on Recommender Systems. pp. 156–167 (2022)

[12] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)

[13] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. p. 452–461. UAI '09, AUAI Press, Arlington, Virginia, USA (2009)

[14] Shen, W.: Deepctr: Easy-to-use,modular and extendible package of deep-learning based ctr models. https://github.com/shenweichen/deepctr (2017)

[15] Sun, Z., Fang, H., Yang, J., Qu, X., Liu, H., Yu, D., Ong, Y.S., Zhang, J.: Daisyrec 2.0: Benchmarking recommendation for rigorous evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(7), 8206–8226 (2022)

[16] Wang, Q., Yin, H., Chen, T., Yu, J., Zhou, A., Zhang, X.: Fast-adapting and privacy-preserving federated recommender system. The VLDB Journal **31**(5), 877–896 (2022)

[17] Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. pp. 165–174 (2019)

[18] Zeng, D., Liang, S., Hu, X., Wang, H., Xu, Z.: Fedlab: A flexible federated learning framework. Journal of Machine Learning Research **24**(100), 1–7 (2023)

[19] Zhao, W.X., Hou, Y., Pan, X., Yang, C., Zhang, Z., Lin, Z., Zhang, J., Bian, S., Tang, J., Sun, W., et al.: Recbole 2.0: towards a more up-to-date recommendation library. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4722–4726 (2022)

[20] Zhao, W.X., Mu, S., Hou, Y., Lin, Z., Chen, Y., Pan, X., Li, K., Lu, Y., Wang, H., Tian, C., Min, Y., Feng, Z., Fan, X., Chen, X., Wang, P., Ji, W., Li, Y., Wang, X., Wen, J.: Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In: CIKM. pp. 4653–4664. ACM (2021)

[21] Ziller, A., Trask, A., Lopardo, A., Szymkow, B., Wagner, B., Bluemke, E., Nounahon, J.M., Passerat-Palmbach, J., Prakash, K., Rose, N., et al.: Pysyft: A library for easy federated learning. Federated Learning Systems: Towards Next-Generation AI pp. 111–139 (2021)