Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

A Federated Learning-Based Movie Recommendation System with SHAP-Enhanced Explainability for User Privacy Preservation

Dr. Manju Pawar
Department of Artificial Intelligence and Data Science
Zeal College of Engineering and Research
Pune, India

Hrishikesh Prabhakar Patil
Department of Artificial Intelligence and Data Science
Zeal College of Engineering and Research
Pune, India

Dnyaneshwar Sanjay Nichal
Department of Artificial Intelligence and Data Science
Zeal College of Engineering and Research
Pune, India

Kalpesh Kailas Ahire
Department of Artificial Intelligence and Data Science
Zeal College of Engineering and Research
Pune, India

***Abstract*—Widespread adoption of streaming platforms and e-commerce services has put user data privacy at the forefront of system design concerns. Conventional recommendation architectures route all interaction data through a central server, creating single points of failure that attract data breaches and regulatory scrutiny. This paper introduces a movie recommendation framework built on Federated Learning (FL), where model training is conducted locally on each user's device and only encrypted parameter updates are forwarded to a global aggregation server. Content-based similarity is established through TF-IDF feature extraction applied to movie descriptions, with cosine distance used to rank item relationships. Updates from participating clients are merged via the Federated Averaging (FedAvg) protocol. A notable aspect of this work is the deliberate inclusion of SHAP (SHapley Additive exPlanations) to address the opacity that often plagues federated models—users receive clear, feature-level justifications for why specific titles are suggested. Experiments on a 20,000-record movie dataset show the system produces stable, relevant recommendations even when client data distributions differ substantially (non-IID settings). The resulting architecture is privacy-respecting, interpretable, and readily scalable.**

***Index Terms*—Federated Learning, Movie Recommendation System, Data Privacy, TF-IDF, Cosine Similarity, FedAvg, Explainable AI, SHAP.**

I. Introduction

Recommendation engines are now woven into the fabric of nearly every digital service people use daily—from Netflix queues and Amazon product pages to Spotify playlists and social feeds. Their value lies in cutting through information overload, surfacing content that matches individual tastes without requiring explicit effort from users. Yet beneath this convenience lies a structural problem that has grown harder to ignore: the data hunger of centralized systems.

Nearly all production-grade recommenders still operate on the assumption that user behaviour data—watch history, ratings, search terms—must be pooled on a server before any useful model can be learned. This assumption made engineering sense when storage and compute were centralised, but it has become a liability as breach incidents multiply and regulations such as GDPR impose strict obligations on data controllers. Users are also more informed than they once were; surveys consistently show growing discomfort with opaque data collection practices, and some actively avoid platforms they perceive as harvesting too much personal information [6], [14].

Federated Learning reframes the problem. Rather than shipping raw data to a server, FL inverts the pipeline: the model travels to the data. Each participant trains a local model on its own device and sends back only a compressed parameter update. The server combines these updates—without ever inspecting the underlying records—to refine a shared global model, which it then broadcasts back for the next round of local training [1], [4]. Privacy is substantially strengthened because the server never holds raw interaction logs.

That said, FL is not a silver bullet. Two challenges stand out in the recommendation setting. First, user data across devices is rarely drawn from the same distribution—a teenager's watch history looks nothing like a retired professor's—and this heterogeneity (commonly called non-IID data) can slow convergence and degrade model quality [15]. Second, the aggregated global model inherits the opacity of most machine learning systems: it can produce a recommendation but cannot easily say why. This interpretability gap erodes trust, particularly for users who want to understand or contest a suggestion.

To tackle both issues, this paper proposes a recommendation framework that pairs FL with Explainable Artificial Intelligence (XAI). TF-IDF converts movie metadata into numerical representations; cosine similarity and KNN then identify the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

closest matches to a user's expressed interest. FedAvg handles the aggregation step, and SHAP analyses are run post-hoc to expose which features drove each recommendation. The result is a system that is simultaneously privacy-aware, reasonably accurate across heterogeneous clients, and transparent enough for users to interrogate.

II. System Architecture

The architecture separates responsibility cleanly between the edge and the cloud. On the edge sit individual client nodes—each representing a user's device—that hold their own local copies of interaction data. No fragment of this data is transmitted outward in raw form. Sitting above the clients is a central aggregation server whose sole job is to combine model updates and distribute the improved global model back to participants [1], [5].

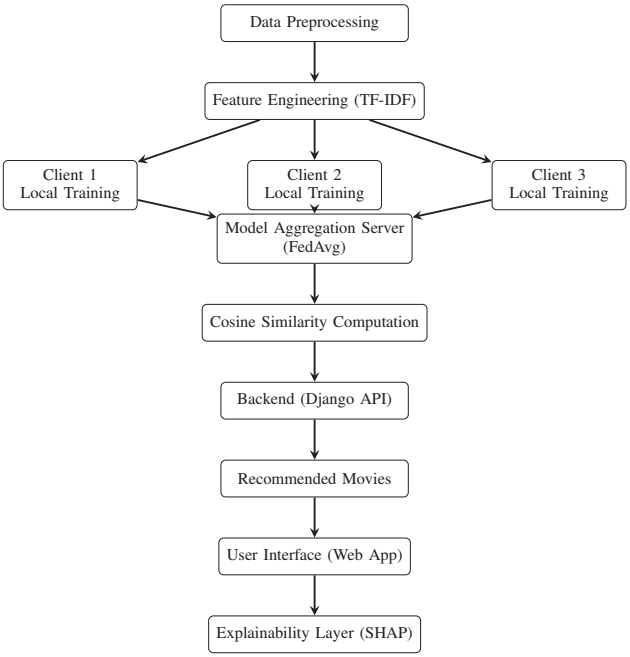

Fig. 1. System Architecture of the Proposed Federated Recommendation Framework.

Data preparation happens first. Movie records are cleaned, deduplicated, and restructured; the overview and genre fields are then fed into a TF-IDF vectorizer that converts free text into sparse numerical matrices. These matrices stay local: each client receives only its own data partition.

Local training proceeds independently on each client. A client fits its model to local data, computes the resulting weight deltas, adds differential privacy noise when configured to do so, and packages the update for transmission. Importantly, the update contains no record-level information—only gradient or weight statistics that characterise what the model learned, not what the data contained.

The server collects updates from all participating clients in a given round, applies FedAvg to compute a weighted average of the received parameters, and broadcasts the refreshed global model. This cycle repeats over multiple rounds until performance stabilises. Once the global model is ready, cosine similarity over the TF-IDF space is used to generate ranked movie lists for each user query. A Django-based API layer mediates between this recommendation engine and the web front-end, which displays titles, poster images, and relevance scores. SHAP analyses are integrated at the output stage to render feature contribution charts alongside each recommendation.

III. System Workflow

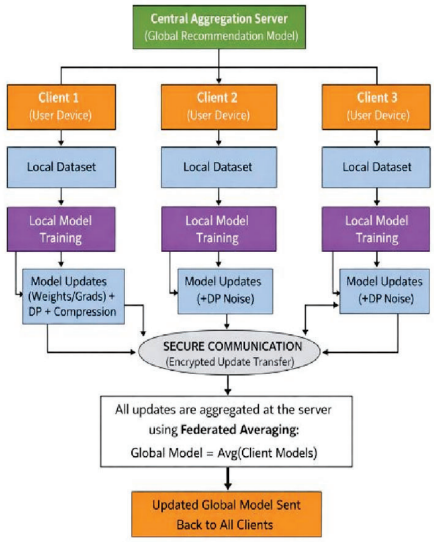

Fig. 2. System workflow .

1) **Feature Extraction:** Raw movie descriptions are tokenised and weighted using TF-IDF. Term importance is calibrated across the entire corpus, so common stopwords receive low weight while genre-defining vocabulary is amplified. Each movie ends up as a dense vector in a shared feature space.
2) **Client-Side Local Training:** Using its local data partition, each client trains a similarity model. The learning objective is to capture which movies tend to co-occur in user histories and to align that signal with content-feature proximity. Raw viewing records never leave the device.
3) **Model Update Generation:** After training, each client packages its weight deltas. These updates are intentionally stripped of any personally identifiable signal; they represent statistical patterns, not individual preferences.
4) **Secure Transmission:** Updates are encrypted before leaving the client. This guards against interception during transit and ensures that even a compromised aggregation server cannot reconstruct individual training examples.
5) **Global Model Aggregation:** The server applies FedAvg, weighting each client's contribution proportion-

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

ally to its dataset size. The merged parameters form a global model that reflects knowledge from all participants without privileging any single one.

6) **Model Distribution:** The updated global weights are pushed back to all clients, which use them as the starting point for the next training round. This iterative refinement continues until the global model converges.
7) **Recommendation Generation:** At query time, KNN retrieves the K movies closest to the query vector under cosine distance. The top-N results are ranked by similarity score and presented to the user.
8) **Explainability Integration:** SHAP values are computed for the final ranking to identify which TF-IDF features—keywords, genres, production context—carried the most weight. These are surfaced in the UI so users can interrogate the rationale behind each suggestion.

IV. Methodology

A. Research Design and Approach

The study follows an experimental design aimed at building and evaluating a recommendation system under realistic federated conditions. The core question is whether content-based FL can match the recommendation quality of centralised approaches while keeping data on-device. Model training is distributed: each simulated client works from its own data shard, produces local updates, and shares only those updates with the aggregation server. FedAvg combines them into a global model at the end of each round.

Transparency sits alongside privacy as a design goal. Users often distrust systems they cannot interpret, so SHAP is embedded in the recommendation pipeline to produce post-hoc explanations. The explanations are meant to be actionable: a user who sees that a recommendation was driven primarily by genre and director keywords can immediately judge whether that reasoning matches their intuition.

The overall design integrates decentralised training, secure parameter aggregation, similarity-based retrieval, and post-hoc explainability into a single coherent pipeline.

B. Dataset Description

The dataset was obtained from Kaggle [19] and draws on metadata originally compiled from IMDb, covering films released before July 2017. The primary file, `movies_metadata.csv`, contains structured fields such as title, genre, and release year, alongside free-text overviews for each title. After removing records with missing overviews, duplicated entries, and irrelevant columns, the working corpus comprised roughly 20,000 movies. Genre analysis showed that drama, comedy, and action titles dominate the distribution, with thriller and romance forming a secondary tier. Word-frequency examination of the overview field confirmed that high-frequency terms like "life," "world," and "story" are present across genres, while more discriminative terms—"heist," "dystopian," "animated"—cluster within specific categories. This vocabulary structure is what TF-IDF is designed to exploit, suppressing the ubiquitous terms and amplifying the genre-specific ones. The textual attributes—overview and genre—served as the primary inputs to the feature extraction stage.

C. Tools, Algorithms, and Models Used

Python was chosen as the main programming language because of its extensive support for machine learning, data analysis, and research-oriented development. Its rich ecosystem of libraries made it suitable for implementing the complete recommendation framework. NumPy and Pandas were used for data handling, preprocessing, and transformation tasks, including cleaning records, managing missing values, and preparing structured datasets for training. Scikit-learn was employed to implement TF-IDF feature extraction and cosine similarity computation, which formed the core of the recommendation engine.

To simulate decentralized model training, TensorFlow Federated (TFF) and Flower were used. These frameworks provide tools for creating federated learning experiments with multiple virtual clients on a single machine. This setup allowed the behaviour of distributed training to be tested without requiring separate physical devices or network infrastructure during the development stage.

The recommendation process was based on a content-driven filtering approach. TF-IDF was applied to convert movie descriptions into numerical vectors by assigning weights to words according to their importance within a movie overview compared with their frequency in the full dataset. Words that appeared frequently in one description but were uncommon across the corpus received higher importance values. This helped the model distinguish between movies more effectively.

After vector generation, cosine similarity was used to measure the closeness between two movie vectors. The resulting score ranged from 0 to 1, where higher values indicated stronger similarity. Since cosine similarity depends on vector orientation rather than text length, it was well suited for movie descriptions of varying sizes. To generate recommendations, the K-Nearest Neighbours (KNN) method was used to identify the top-N closest movies in the TF-IDF feature space based on cosine similarity scores.

For privacy-preserving model training, the Federated Averaging (FedAvg) algorithm was adopted. In each communication round, every client trained the model locally using its own dataset and sent only updated model parameters to the server. The server combined these updates to create a new global model. Client contributions were weighted according to the number of local training samples, allowing larger datasets to have a proportionally greater influence. This process helped maintain stable global learning even when client datasets varied in size and content.

To improve interpretability, SHapley Additive exPlanations (SHAP) were applied after model training. SHAP estimates the contribution of each feature to the final prediction by measuring how the output changes when individual features are included or removed. Positive values indicate features that increase recommendation relevance, while negative values represent features that reduce it. This allowed users to understand

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

why a particular movie was recommended and which factors influenced the decision.

The web backend was developed using Django REST Framework, which managed communication between the recommendation engine and the user interface through API services. Matplotlib was used to generate cosine similarity graphs and SHAP explanation plots. These visual outputs were integrated into the interface to present recommendation results in a clear and understandable form. A summary of all tools, libraries, and algorithms used in the proposed system is presented in Table I.

TABLE I
TECHNOLOGY STACK USED IN THE PROPOSED SYSTEM

Component	**Technology / Library**
Programming Language	Python
FL Frameworks	TensorFlow Federated (TFF), Flower
ML Libraries	NumPy, Pandas, Scikit-learn
Feature Extraction	TF-IDF
Similarity Measure	Cosine Similarity
Recommendation	K-Nearest Neighbours (KNN)
Aggregation Algorithm	Federated Averaging (FedAvg)
Explainability	SHAP
Backend	Django REST API
Visualisation	Matplotlib

D. Replication

Reproducibility was treated as a first-class concern throughout development. The system can be rebuilt by anyone with access to the same Kaggle dataset by following the preprocessing steps described above (deduplication, missing-value removal, TF-IDF vectorisation with identical hyperparameters) and configuring the FL simulation with the same number of clients and communication rounds. All libraries used are open-source and platform-agnostic, so the implementation places no dependency on proprietary software or specialised hardware. Researchers can partition the dataset into synthetic client shards of any size to explore how data heterogeneity affects convergence.

V. RESULTS

A. Dataset Characteristics

After preprocessing, the final dataset contained approximately 20,000 movie records covering multiple genres, release periods, and production categories. Entries with missing plot overviews and duplicate titles were removed to improve data quality. The cleaned dataset was then distributed across the simulated client nodes in a consistent manner. Among all genres, drama represented the largest portion of the dataset, followed by comedy and action. An initial examination of the overview text using word-cloud visualization showed that genre-specific terms were clearly present, indicating that the TF-IDF method could effectively capture meaningful textual patterns for recommendation.

B. Federated Learning Output

The federated learning process was carried out using several simulated client nodes, where each client stored a separate portion of the movie dataset. During each communication round, clients trained the model locally on their own data and sent encrypted parameter updates to the central server. These updates were combined using the FedAvg algorithm to produce an improved global model, which was then redistributed to all clients for the next round of training. Throughout the process, raw user data remained within the client environment and was never shared directly. Experimental results showed that communication costs remained reasonable, while the global model improved steadily over multiple training rounds.

C. Web Application and Recommendation Output

The recommendation system was integrated into a Django-based web application. When a user enters a movie title or keyword, the system converts the query into TF-IDF vector form and calculates cosine similarity with all movie records in the database. Based on these scores, the top-N most relevant movies are returned. Each recommendation is displayed with the movie title, poster image, and similarity score. The interface was intentionally kept simple so that users could focus on the recommended results and the generated SHAP explanations.

D. Similarity Score Analysis

The cosine similarity scores ranged from 0 to 1 for all evaluated queries. Movies belonging to similar genres or sharing common themes and descriptive terms generally produced scores above 0.7, indicating strong similarity. Lower scores, typically between 0.2 and 0.4, were observed when movies had only limited keyword overlap. The similarity matrix also enabled fast nearest-neighbour search, and the system maintained low response time even when the dataset size approached 20,000 titles.

E. Explainability Output

SHAP analysis was applied to selected recommendation results to examine the interpretability of the model. In most cases, the features with the highest positive contribution were genre-related keywords, character names, directors, or franchise terms that were logically associated with the input query. Negative SHAP values highlighted terms that reduced similarity, such as horror-related keywords appearing in comparisons with comedy movies. The close agreement between SHAP explanations and human expectations indicates that the recommendation model learns meaningful relationships from content features rather than relying on random correlations.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

F. System Execution

The pipeline ran end-to-end without critical failures across preprocessing, federated training, recommendation generation, and SHAP analysis. The modular design made it straightforward to adjust the number of federated rounds, the client partition sizes, and the value of K in KNN without restructuring the codebase. Overall, the system produced accurate, relevant, and explainable recommendations in a fully decentralised training setting.

VI. Discussion and Limitations

The experimental results are broadly consistent with trends reported in recent federated recommendation literature [1], [3], [15]: decentralised training can match centralised baselines on content-based tasks while materially reducing privacy risk. A few findings are worth unpacking.

The privacy story is straightforward. Because raw movie interaction logs never leave the simulated client devices, the server-side attack surface shrinks to the parameter space alone. Differential privacy noise on the update vectors adds a further layer of protection. The combination brings the system into alignment with data minimisation principles advocated in regulatory frameworks like GDPR, without sacrificing recommendation quality.

On the accuracy side, TF-IDF combined with cosine similarity turned out to be a reasonable fit for this content domain. Movie overviews are rich enough in discriminative vocabulary that a bag-of-words representation captures most of the genre and thematic signal needed for useful similarity rankings. The KNN retrieval step then translates similarity scores directly into ranked lists, keeping the pipeline transparent and deterministic.

The inclusion of SHAP is one of the key contributions of the proposed system. Most earlier studies on federated recommendation systems mainly focused on improving prediction performance and protecting user privacy, while model interpretability received comparatively less attention [13]. In the present work, SHAP explanations are integrated directly with recommendation results to reduce the black-box nature of machine learning models. This allows users to see the factors that influenced each recommendation and helps them better understand or evaluate the suggested movies. During testing, the features highlighted by SHAP were largely similar to those that a domain expert would consider important, which indicates that the model makes recommendations based on meaningful content relationships.

Despite the positive results, some limitations should be acknowledged. The federated learning process was tested in a simulated environment on a single machine instead of real distributed devices. Therefore, practical issues such as network latency, interrupted communication, and hardware variation were not fully analysed. In addition, the recommendation engine uses only content-based filtering. User behaviour information, such as ratings or interests shared by users with similar preferences, was not included. As a result, the system may have limited ability to generate highly diverse or unexpected recommendations.

Another limitation concerns non-IID data distribution across clients. Although the proposed framework handled moderate differences in local datasets during experiments, larger federated systems with highly unbalanced or significantly different client data may reduce training efficiency and model consistency. Communication overhead may also increase as more clients participate in the learning process. At present, techniques such as model compression, selective client participation, and asynchronous updates were not applied.

A. Future Scope

There are several directions for future improvement. Implementing the system on real distributed devices would make it possible to study performance under practical conditions such as communication delay, unstable connectivity, and different hardware capabilities. This would provide a more realistic assessment of federated recommendation systems in large-scale environments.

A hybrid recommendation model that combines content-based features with privacy-preserving collaborative signals could further improve recommendation accuracy and diversity. Such a system would consider both item characteristics and shared user interests while maintaining data confidentiality.

To address differences in client data distributions, advanced federated optimisation methods such as FedProx and FedNova may be explored. These methods are designed to improve training stability and reduce the effect of client drift in heterogeneous environments.

The explainability module may also be enhanced through interactive dashboards that allow users to adjust the influence of selected features according to their personal preferences. This would provide greater user control and support a more personalised recommendation experience.

VII. Conclusion

This paper introduced a movie recommendation system built on federated learning, where user data remains on local devices while model improvement is achieved through collaborative training. In this approach, users benefit from a shared recommendation model without the need to transfer personal data to a central server. The proposed framework consists of three major parts: a content-based recommendation module using TF-IDF and cosine similarity, a federated training mechanism based on the FedAvg algorithm, and a SHAP explainability layer that helps users understand the reasons behind each recommendation.

The experimental study was carried out on a dataset of nearly 20,000 movie titles. Results showed that the system was able to provide relevant and consistent recommendations even when client datasets followed non-IID distributions. During the complete training process, raw user information remained on local clients and was not exchanged with other participants, which supported privacy preservation. In addition, SHAP explanations were found to be meaningful and aligned with

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

human understanding, showing that the model relied on useful content features when generating recommendations.

Although the framework was developed for movie recommendations, the same concept can be applied in many other domains where personalisation and privacy are equally important. Examples include online shopping platforms, digital learning environments, news recommendation systems, and healthcare information services. Because the system was implemented using open-source tools and a reproducible methodology, it can be extended or adapted by future researchers for different applications.

Several directions remain open for future improvement. Real-world deployment on distributed devices would allow testing under practical network and hardware conditions. The inclusion of collaborative filtering methods with differential privacy could further improve recommendation quality. In addition, more advanced interactive explanation interfaces may enhance user trust and engagement. Overall, the results of this work show that federated learning and explainable artificial intelligence can be effectively combined to develop recommendation systems that are reliable, privacy-aware, and transparent.

REFERENCES

[1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Personalized federated recommendation via joint representation learning, user clustering, and model adaptation," in *Proc. 31st ACM Int. Conf. Information and Knowledge Management (CIKM)*, Aug. 2022.

[2] R. Sharma, P. Verma, and S. Gupta, "Federated learning for personalized financial recommendations," *Int. J. Sci. Technol. Eng.*, vol. 10, no. 3, pp. 112–118, 2024.

[3] Y. Zhang, J. Liu, M. Chen, and Q. Wang, "PerFedRec++: Enhancing personalized federated recommendation with self-supervised pre-training," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 2, pp. 1–21, 2024.

[4] A. Kumar, S. Mehta, and R. Patel, "Progressive search personalization and privacy protection using federated learning," *Expert Syst.*, vol. 40, no. 4, e13021, 2023.

[5] J. Wang, H. Liu, Z. Zhang, and Q. Chen, "Federated deep recommendation system based on multi-view feature embedding," in *Proc. IEEE 9th Int. Conf. Data Science and Advanced Analytics (DSAA)*, Porto, Portugal, Oct. 2022.

[6] M. Al-Khatib, R. Nair, and S. Thomas, "Preserving privacy in next keyword prediction using federated learning," in *Proc. 2nd Int. Conf. Advances in Computational Intelligence and Communication (ICACIC)*, Coimbatore, India, Dec. 2023.

[7] J. Lin, W. Chen, and X. Zhao, "Federated recommendation with additive personalization (FedRAP)," *arXiv preprint*, 2023.

[8] H. Liang, Z. Li, and Y. Wang, "Personalized federated recommender systems with private and partially federated autoencoders," in *Proc. 56th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, USA, Dec. 2022.

[9] S. Park, J. Kim, and H. Lee, "Cluster-driven personalized federated recommendation with interest-aware graph convolution network for multimedia," in *Proc. 32nd ACM Int. Conf. Multimedia*, Melbourne, Australia, Oct. 2024.

[10] K. Zhou, L. Huang, and M. Sun, "A federated recommendation algorithm based on user clustering and meta-learning," *Appl. Soft Comput.*, vol. 145, p. 110456, 2024.

[11] Z. Feng, Y. Xu, and H. Zhang, "ReFRS: Resource-efficient federated recommender system for dynamic and diversified user preferences," *arXiv preprint*, 2022.

[12] T. Wu, J. Ren, and Y. Zhao, "Towards fair and personalized federated recommendation," *Pattern Recognit.*, vol. 141, p. 109595, 2023.

[13] P. Li, Q. Zhang, and Y. Chen, "A tutorial on personalized federated recommender systems: Recent advances and future directions," *arXiv preprint*, 2024.

[14] S. Gupta, R. Mehra, and A. Singh, "Horizontal federated learning and secure distributed training for recommendation systems with Intel SGX," *arXiv preprint*, 2022.

[15] J. Liu, H. Wang, and X. Zhou, "Recent advances and future challenges in federated recommender systems," *Int. J. Data Sci. Anal.*, vol. 15, no. 2, pp. 87–104, 2023.

[16] M. Chen, Y. Zhao, and K. Xu, "Transformer-based federated learning models for recommendation systems," *ACM Trans. Recommender Syst.*, vol. 2, no. 1, pp. 1–24, 2024.

[17] A. Verma, S. Kulkarni, and P. Patil, "A privacy-preserving system for movie recommendations using federated learning," *ACM Trans. Recommender Syst.*, vol. 2, no. 2, p. 15, 2024.

[18] R. Nair, S. Iyer, and M. Joshi, "A three-tier architecture of federated learning for recommendation systems," in *Proc. 7th Int. Conf. Computing Methodologies and Communication (ICCMC)*, Erode, India, Feb. 2023.

[19] Kaggle, "Movie metadata dataset (movies_metadata.csv)," 2025. [Online]. Available: https://www.kaggle.com. Accessed: Jan. 2025.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)