IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DO I

CRAFT: Cold-start Recommender with Attention and Federated Training

NAVANEETH SIVAKUMAR[1], REUBEN STEPHEN JOHN[1], ALLEN BIJO[1], and *SUGANESHWARI G[1]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India. (e-mail: navaneeth.sivakumar66@gmail.com, reuben.john@vitstudent.ac.in, allenbijot@gmail.com, suganeshwari.g@vit.ac.in)

Corresponding author: Suganeshwari G (e-mail: suganeshwari.g@vit.ac.in).

ABSTRACT
The cold-start problem—where recommendation systems struggle to suggest new or infrequently interacted items—remains a key challenge, particularly in federated learning settings where privacy must be preserved. Traditional approaches often rely on centralized data aggregation or collaborative filtering methods that do not translate well to decentralized environments. Existing federated recommendation methods such as FedMF and FedGN offer limited support for cold-start personalization, especially when item metadata is sparse or unavailable. To address these limitations, we propose **CRAFT** (Cold-start Recommender with Attention and Federated Training), a novel federated learning-based framework that enhances cold-start recommendation while preserving user privacy. CRAFT introduces an attention mechanism to emphasize salient user–item interaction patterns, improving the inference of user preferences. Each client trains a personalized model locally, with updates aggregated via Federated Averaging (FedAvg), ensuring collective intelligence without compromising sensitive data. By incorporating temporal dynamics and detailed interaction histories, CRAFT delivers highly personalized recommendations. Leveraging the NVFlare platform, CRAFT is also designed for scalable deployment across distributed environments. Experimental results on three real-world datasets—MovieLens 1M, Amazon Movies & TV, and CiteULike—demonstrate that CRAFT outperforms state-of-the-art baselines by up to 16.8% in nDCG@20 under cold-start conditions, while maintaining strong privacy guarantees.

INDEX TERMS Federated learning, Cold start problem, Attention mechanisms, Privacy preservation, Federated Averaging (FedAvg), Decentralised learning.

I. INTRODUCTION

RECOMMENDATION systems are essential to modern digital platforms, offering personalized suggestions based on users' past interactions. They underpin applications across streaming platforms like Netflix [1], e-commerce platforms such as Amazon and Flipkart [21], and e-learning environments [22]. Short video platforms like YouTube [23] further extend these systems using collaborative filtering, content-based filtering, and hybrid models to enhance user engagement.

A major unresolved issue in these systems is the **cold-start problem**, which arises when new or rarely interacted items lack sufficient data for accurate recommendations [2]. This hampers user experience in platforms where the rapid addition of items—such as new movies, products, or courses—is common. Centralized systems typically address this by incorporating item metadata or user profiling [7], which requires aggregating user data on centralized servers [8]. While effective, this approach presents critical privacy concerns [9], raising questions about data misuse and regulatory compliance, especially under frameworks like GDPR.

To mitigate such risks, Federated Recommendation Systems (FedRecSys) have been developed [3]. These systems allow decentralized training, wherein user interactions remain local and only model updates are transmitted for aggregation [4]. This decentralized paradigm enhances data privacy and security. However, most federated approaches still struggle in cold-start settings due to their dependence on user history, which is inherently sparse or nonexistent for new items [5].

Although one could consider sharing item attributes across clients to combat the cold-start issue, this introduces other concerns, such as leakage of proprietary item metadata or sensitive product descriptions [6]. Therefore, a viable federated recommender must preserve both user interaction and item attribute privacy while maintaining high recommendation quality.

This paper introduces **CRAFT** (Cold-start Recommender with Attention and Federated Training), a privacy-preserving recommendation framework specifically designed to address the cold-start challenge in federated environments. CRAFT enhances cold-start prediction by leveraging a local attention mechanism to identify and prioritize key interaction patterns. It employs Federated Averaging (FedAvg) to aggregate model updates across clients without sharing raw data and builds on NVIDIA's secure federated learning platform NVFlare, which supports differential privacy and homomorphic encryption [26].

- **Client-Side Model Training:** Each client trains a self-attention–based model on local interaction data, capturing user-specific preferences [7].
- **Federated Aggregation:** FedAvg combines client model parameters while maintaining data locality, facilitated securely via NVFlare [6].
- **Cold-Start Adaptation:** CRAFT iteratively refines global parameters through distributed learning, improving recommendations for new or sparsely interacted items [5] [9].

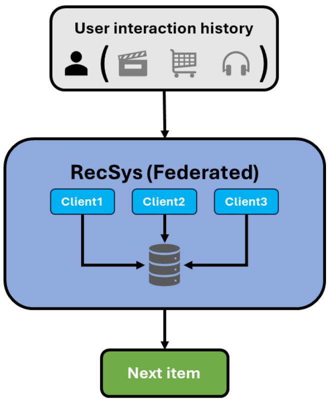


FIGURE 1: Generic Federated Recommendation System Architecture

The main contributions of this paper include:

- Proposing CRAFT, a novel federated recommendation framework that effectively addresses the cold-start problem while safeguarding user and item-level privacy.
- Utilizing self-attention mechanisms to dynamically weight interaction histories and integrating NVFlare for secure, scalable federated training using FedAvg.
- Demonstrating, through extensive experiments on MovieLens 1M, Amazon Movies & TV, and CiteULike datasets, that CRAFT outperforms leading federated methods in accuracy, personalization, and privacy preservation.

The remainder of the paper is organized as follows: Section II details the Preliminaries. Section III describes all other Related Works. Section IV and V outline the Methodology and Section VI experimental setup. Section VII introduces the Evaluation metrics. Section VIII presents the results and analysis, and Section IX concludes the paper.

II. PRELIMINARIES

A. PROBLEM FORMULATION

Consider a federated recommendation scenario consisting of K decentralized clients (devices or users), each holding local interaction data privately. Formally, the system consists of:

- A set of users $U = \{u_1, u_2, \ldots, u_n\}$,
- A set of items $I = \{i_1, i_2, \ldots, i_m\}$.

Each user $u \in U$ has a local interaction history represented by a sequence of items $S_u = (i_{u,1}, i_{u,2}, \ldots, i_{u,t})$, where $i_{u,t}$ denotes an item that user u interacted with at time t. The goal is to predict the next item $i_{u,t+1}$ the user is likely to interact with, with particular emphasis on cold-start items that have limited or no historical interactions. A similar challenge arises for new users with very few interactions—a scenario also addressed by our method.

Given the decentralized setting, each client k locally trains a self-attention-based recommendation model parameterized by weights θ_k. A global model θ is computed through Federated Averaging (FedAvg):

$$\theta \leftarrow \frac{1}{\sum_{k=1}^{K} |D_k|} \sum_{k=1}^{K} |D_k| \theta_k \tag{1}$$

where $|D_k|$ denotes the size of the local dataset at client k.

The objective is to minimize the following federated loss across all clients:

$$\mathcal{L}(\theta) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_k(\theta_k) \tag{2}$$

Here, each $\mathcal{L}_k(\theta_k)$ is a binary cross-entropy loss computed on local predictions of user-item interactions. By leveraging local training and global aggregation, this setup helps address data sparsity in cold-start settings—clients benefit from shared model knowledge without needing access to others' raw data.

III. RELATED WORK

A. TRADITIONAL RECOMMENDATION

Traditional recommender systems largely rely on collaborative filtering (CF) and content-based approaches [7], [8]. CF predicts user preferences based on interaction patterns with other users or items, while content-based methods leverage item metadata to generate recommendations [1]. Despite their widespread adoption, these models suffer from well-known limitations, such as data sparsity, scalability issues, and the cold-start problem [2], [7], where insufficient interaction data reduces performance for new users or items.

To address some of these challenges, sequential recommendation models were introduced. These incorporate time-aware patterns, such as clickstreams or purchase sequences, into the prediction process. Markov chains and matrix factorization–based temporal models have been used to capture such patterns [18]. However, traditional sequential methods often struggle with long-range dependencies and do not fully capture complex user behaviour over time [15].

B. FEDERATED RECOMMENDATION SYSTEMS

Federated learning (FL) has emerged as a privacy-preserving alternative to centralized recommendation systems [3], [19]. Instead of transmitting raw user interaction data to a central server, FL trains models locally on users' devices and aggregates only model updates using algorithms such as FedAvg [9]. This decentralized learning process preserves user privacy and complies with data protection regulations such as GDPR [4], [14].

Recent studies have integrated FL with neural networks to enhance personalization in recommender systems while ensuring data privacy. For example, FedRec and FedMF adapt matrix factorization for federated environments but often cannot model sequential behavior effectively [19]. More recent works apply self-attention mechanisms in FL, capturing contextual interactions through on-device multi-head attention layers and offering a privacy-conscious solution for modeling user behavior over time [8], [14].

C. FEDERATED LEARNING FOR COLD-START RECOMMENDATIONS

The cold-start problem, characterized by limited interaction data for new users or items, remains a central challenge in both centralised and federated recommendation systems. Although centralised methods utilise content features or side information to mitigate cold-start issues, such data sharing is often infeasible in federated settings due to privacy restrictions [5]. Consequently, FL models cannot generalize to unseen users or items without compromising privacy or security.

In centralised environments, meta-learning approaches such as MeLU [35] and MAMO [36] learn user-adaptive models from sparse data and achieve strong performance. Hybrid methods that combine collaborative filtering with content-based techniques or leverage side information (e.g., user demographics, item metadata) have also been proposed to effectively tackle cold start [37], [38]. However, these methods rely on centralized access to raw user data or content features, which is infeasible in federated settings sensitive to privacy or complying with regulations.

Federated learning (FL) further complicates the cold-start scenario by restricting access to individual-level data, limiting the use of side information and centralized adaptation mechanisms. Some FL-based approaches share lightweight representations or metadata, e.g. IFedRec [29] and MetaFRS [28] - but often trade off privacy guarantees or require client-specific tuning, which hinders scalability and robustness.

Recent work, such as C2lRec, leverages contrastive and causal learning to handle user heterogeneity and distribution shifts in federated settings [30], [31]. Cross-domain and transfer learning approaches attempt to reuse knowledge from auxiliary domains [32], [33], yet they often depend on alignment assumptions that are difficult to maintain in real-world federated environments.

Existing FL models still struggle with cold-start recommendations without risking data leakage. Many rely on partial or obfuscated item features or resort to server-side inference, both of which can expose sensitive information [3]. To overcome these limitations, we propose **CRAFT**. This privacy-preserving framework leverages self-attention to learn temporal user patterns locally and employs a meta-attribute network on the server for item representation.

CRAFT introduces a novel representation-alignment mechanism enabling the global model to bridge user and item features without exposing raw data. Local clients train attention-based models on user–item interaction sequences; these models capture temporal dependencies without requiring shared item metadata. Client updates are aggregated using FedAvg, coordinated via NVFlare to ensure secure, scalable deployment [26]. By decoupling representation learning across clients and the server, CRAFT significantly improves cold-start performance while maintaining privacy throughout training and inference.

IV. METHODOLOGY

This section defines the problem that motivates the proposed method and presents the core workflow, objective functions, and pseudocode. Fig. 2 illustrates a high level workflow along with Fig. 3, which illustrates the complete architecture of the attention-based federated recommendation model, which integrates a next-item prediction mechanism with federated training using NVFlare.

A. FRAMEWORK OVERVIEW

The proposed framework integrates federated learning with a self-attention model to address the cold-start problem in recommendation systems. Each client trains a local instance of the attention-based model using user interaction sequences while preserving data privacy. The model comprises:

- An item embedding layer and a position embedding layer
- Self-attention with causal masking and multi-head attention mechanisms
- Feedforward layers and an output layer to predict future interactions

During training, each client optimizes a binary cross-entropy loss and shares only model parameters with a central server. The server aggregates these updates using FedAvg to construct a refined global model, which is then redistributed to clients for further iterative training—ensuring both privacy and efficiency.

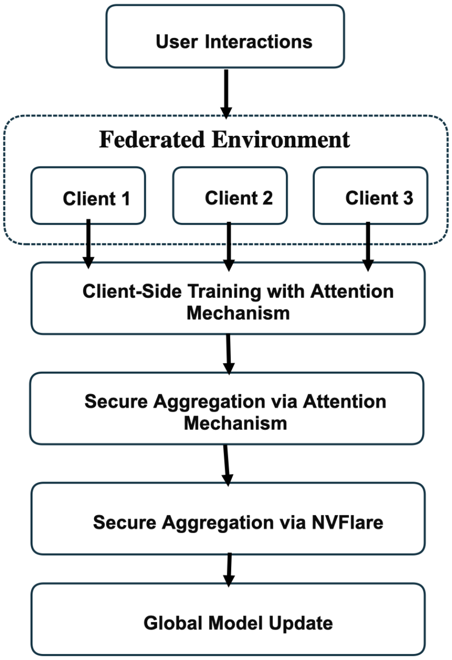


FIGURE 2: High Level Workflow

B. SELF-ATTENTION MODEL

The Self-Attention Model is a recommendation model based on the self-attention mechanism of the Transformer architecture [11], designed to capture sequential patterns in user interactions. Given a dataset of N users and M items, each user u has a sequence of interactions $S_u = (i_1, i_2, \ldots, i_T)$, where each $i_t \in \{1, 2, \ldots, M\}$ is the t^{th} item the user interacts with. The goal is to predict the next item a user will likely interact with.

1) Item Embedding Layer

Each item i is represented as a dense vector in a continuous space:

$$e_i = \text{Embeddings}(i) \tag{3}$$

where e_i is the learned embedding of item i.

2) Position Embedding Layer

To capture the sequential nature of interactions, we incorporate a position embedding for each time step t:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{4}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{5}$$

This allows the model to distinguish the order of items in the sequence.

3) Self-Attention Mechanism

The self-attention mechanism enables the model to focus on relevant past interactions. For a sequence $S_u = (i_1, i_2, \ldots, i_T)$, we compute attention scores using query Q, key K, and value V matrices:

$$A = \text{Attention}(Q, K, V) \tag{6}$$

where the attention function is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{7}$$

with d_k representing the dimension of the key vectors.

4) Multi-Head Attention

To improve the model's expressiveness, multi-head attention is applied, projecting the input into multiple subspaces and performing attention in parallel:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{8}$$

Add and Layer Norm Operations prevents exploding/vanishing gradients and enhances training stability by adding a residual connection and normalising the features of a token.

5) Feedforward Layer

A pointwise feedforward network with ReLU activation is applied to the output of the attention layer:

$$y = \text{ReLU}(W_1 A + b_1) y = W_2 y + b_2 \tag{9}$$

6) Output Layer

The final output representation for the sequence, h_u, is used to predict the following item:

$$\hat{y} = \text{Softmax}(h_u \cdot e_i) \tag{10}$$

where e_i is the embedding of each candidate item.

C. OBJECTIVE FUNCTIONS

1) Core Optimization

The fundamental goal is to optimise for client-specific models and global model aggregation. This ensures the collective model adapts to the distributed data across all clients, improving the recommendation performance for cold-start items.

2) Loss Function

The losses are calculated for each client during training. In each client being trained, our model takes a sequence S_u for a given user u and the predicted sequence y. A binary cross-entropy loss is used to identify the difference between each item in the sequence.

$$-\sum_{S^u \in S} \sum_{t=1}^{n} \left[\log\left(\sigma(r_{y_t,t})\right) + \sum_{j \notin S^u} \log\left(1 - \sigma(r_{j,t})\right) \right] \tag{11}$$

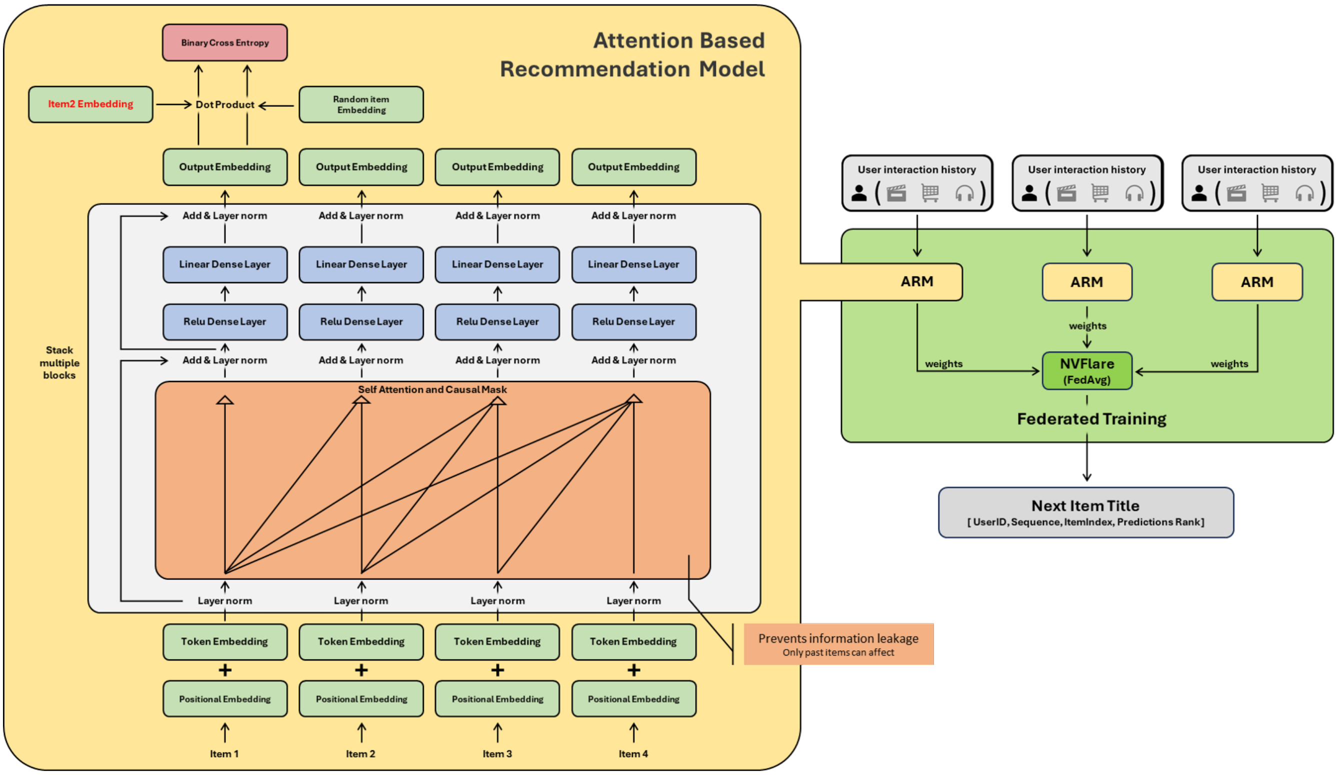


FIGURE 3: Architecture of the proposed CRAFT framework.

D. CLIENT-SPECIFIC TRAINING

Each client in the system has a local dataset, D_k, with user-item interaction sequences for the attention model. The primary steps involved in client-side training are:

- **Model Initialization:** Each client initialises a local attention model with parameters received from the central server.
- **Loss Function:** Clients use a binary cross-entropy loss function. This loss calculates the probability that the user will interact with an item next based on past interactions in the sequence. The target sequence is processed such that:
  - For padding items, no loss is calculated.
  - For non-padding items, the expected output is either the next item in the sequence or a randomly generated negative sample.
- **Optimisation :** The local model is trained using the Adam optimiser stochastic Gradient Descent (SGD) variant that adapts the learning rate based on the gradient moments. The model trains for a fixed number of local epochs (e.g., 1–5 epochs) on each client, striking a balance between computational efficiency and model improvement.

This section explains how FL and the FedAvg [5] algorithm contribute to updating the global model in decentralised systems.

E. FEDERATED LEARNING AND FEDAVG FOR MODEL UPDATE

FL is a decentralised training of machine learning models where clients (e.g., mobile devices or distributed data sources) collaboratively participate in learning a shared global model while keeping their data local [3], [6]. This approach enhances user privacy by transmitting only model updates, such as weight and gradient information, to a central server rather than raw data. The server aggregates these updates, improving the global model without direct data sharing [4].

F. FEDERATED AVERAGING (FEDAVG) FOR GLOBAL MODEL UPDATE

A key aspect of FL is how model updates from multiple clients are combined to form a coherent global model. FedAvg [3] is an aggregation algorithm that merges these local updates to refine the global model in an efficient and balanced way.

- **Weighted Averaging:** The server computes a weighted average of the local model updates from each client, where the weight corresponds to the size of the client's local dataset. This method ensures that clients contributing more data significantly influence the global model update [9].
- **Global Model Update:** Following the weighted average, the server consolidates the local models into a new global model by aggregating the weighted param-

eters. This step reflects the collective learning of all participating clients, accounting for variations in local data distributions and ensuring a model that generally generalises data [3].

The global model parameters θ_{global} are updated using the following equation:

$$\theta_{\text{global}} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \cdot \theta_k \tag{12}$$

where:

- θ_k represents the model parameters from client k,
- n_k is the size of the dataset on client k,
- n is the total number of data points across all clients and
- K is the total number of clients.

This weighted aggregation ensures that clients with more data significantly influence the global model, improving its accuracy overall.

G. GLOBAL MODEL DISTRIBUTION AND ITERATIVE TRAINING

After updating the global model using FedAvg, the new model parameters are distributed back to the clients. This process of local training, aggregation using FedAvg, and international model distribution is repeated for multiple rounds, allowing the global model to improve progressively. With each round, the model becomes better at capturing patterns in the data without requiring access to the raw user data, thereby maintaining privacy.

H. FEDERATED LEARNING OBJECTIVE

In Federated Learning, the objective is collaboratively learning a global model by minimising action across all clients. Each client has its local dataset D_k and computes a local loss function $L_k(\theta_k)$ based on its data. The global loss is the average of the local losses from all clients, and the goal is to find the optimal global model parameters θ that minimise

$$\theta = \frac{1}{K} \sum_{k=1}^{K} L_k(\theta_k) \tag{13}$$

where $L_k(\theta_k)$ is the local loss function for client k, and K is the total number of clients. The global model is optimized across all clients by minimizing.

V. FEDERATED TRAINING PSEUDO-CODE

Training Pipeline Overview:

- **Client-side Training** using attention-based models and local interaction data.
- **Secure Aggregation** of model parameters via FedAvg using NVFlare.
- **Global Model Update** based on weighted averaging of client contributions.

The binary cross-entropy loss, combined with negative sampling, effectively handles cold-start settings. The Adam optimizer aids convergence with adaptive learning rates, es-

Algorithm 1 Federated Training for Self-Attention Model

Input: Total rounds R, number of clients K, local epochs E, learning rate η
Initialize global model parameters θ_{global} ▷ θ_{global}: global model
for each round $r = 1, 2, \dots, R$ **do**
  Server randomly selects a subset of clients $\mathcal{S} \subset \{1, \dots, K\}$ ▷ Improves generalization
  for each client $k \in \mathcal{S}$ **in parallel do**
    $\theta_k \leftarrow \theta_{\text{global}}$ ▷ θ_k: local model for client k
    for each local epoch $e = 1, 2, \dots, E$ **do**
      Train on D_k using binary cross-entropy loss:

$$L_k = -\sum_t [o_t \log(\sigma(r_{o_t})) + (1 - o_t) \log(1 - \sigma(r_{o_t}))]$$

      Optimizer: Adam ▷ Effective for sparse gradients
    end for
    Send updated θ_k to server
  end for
  Aggregate models: $\theta_{\text{global}} \leftarrow \sum_{k\in\mathcal{S}} \frac{n_k}{n} \cdot \theta_k$
end for
Output: Final global model θ_{global}

pecially under sparse client data. Random client selection improves model robustness across non-IID distributions.

STEP-BY-STEP PROCEDURE:

1) Client Selection and Initialisation each training round r, the server randomly selects a subset of clients $\mathcal{S} \subset \{1, \dots, K\}$. This strategy ensures scalability and introduces diversity in training, leveraging each client's unique datasets.
2) Local Model Training: Each chosen client k receives the global model parameters θ_{global} and initializes its local model θ_k. Clients train their local attention models for E epochs on their local dataset D_k. The training process employs a binary cross-entropy loss function:

$$L_k = -\sum_t [o_t \log(\sigma(r_{o_t})) + (1 - o_t) \log(1 - \sigma(r_{o_t}))] \tag{14}$$

   where o_t represents the true output, and $\sigma(r_{o_t})$ is the predicted probability. The Adam optimiser utilised efficient parameter updates, ensuring adaptive learning rates based on gradient moments.
3) Model Update Transmission: After local training, each client transmits its updated model parameters θ_k back to the server.
4) Global Model Aggregation: The server aggregates these client updates using a weighted averaging scheme to update θ_{global}:

$$\theta_{\text{global}} \leftarrow \sum_{k\in\mathcal{S}} \frac{n_k}{n} \cdot \theta_k \tag{15}$$

Here, n_k represents the number of data points in the client k 's dataset, and n is the total number of data points from all participating clients. This weighted approach ensures that clients with larger datasets have a proportionally more significant influence on the updated global model.

5) Iterative Training: Steps 3 and 4 are repeated for R rounds. This iterative process enables the global model to learn progressively from diverse, decentralised, directly accessing user data, preserving privacy.

VI. EXPERIMENTAL SETUP AND EVALUATION

In this study, we established an experimental environment utilising PyTorch on an NVIDIA RTX 3050 GPU to evaluate our recommendation system designed explicitly for cold start scenarios. The setup approximates a real-world federated learning environment, where models are trained locally on user data before being pooled globally. We aim to assess the method's efficacy in generating accurate and relevant recommendations for new users with limited or no historical interactions. We implemented a comprehensive computational and data partitioning methodology to achieve this objective. [3], [4], [19].

A. DATASET PARTITION AND CLIENT SETUP

To simulate a federated environment, we partition the dataset into three distinct clients, each with a unique subset of user data. This configuration mimics a decentralised environment, where each client has limited access to the overall dataset but collectively contributes to the training process. The recommendation model learns from diverse user preferences by splitting the dataset into three parts without centrally aggregating raw data. This approach aligns with privacy-preserving principles, highlighting the potential of federated learning in scenarios involving sensitive user information. [6], [9], [20].

B. DATASET OVERVIEW

Table 1 highlights the characteristics of the three datasets used in our experiments, each chosen for its real-world relevance and varying degrees of sparsity.

1) Amazon Movies_and_TV

This dataset is sourced from Amazon product reviews in the Movies and TV category [21], with over 8 million ratings from 2.6 million users on 60,000+ items. Each interaction is an explicit rating and is accompanied by metadata like titles and genres. The extreme sparsity—where most users interact with only a handful of items—reflects real-world e-commerce cold-start scenarios and poses significant challenges in a federated setting where data cannot be shared across clients.

2) CiteULike

CiteULike is a research paper bookmarking platform, comprising about 1.5 million implicit interactions from 5,000 users over 17,000 papers [22]. Unlike explicit feedback, bookmarks indicate interest without ratings, increasing uncertainty in modeling user preferences. The limited overlap in user-item interactions makes it particularly suitable for evaluating personalized cold-start recommendations in federated academic recommendation systems.

3) MovieLens

The MovieLens dataset contains 20 million ratings from 138,000 users on 27,000 movies [23]. While relatively less sparse, it still represents a practical recommendation challenge with structured metadata and demographic information. It serves as a robust benchmark for evaluating the generalizability of CRAFT in federated scenarios where moderate sparsity still impacts new item discovery.

TABLE 1: Comparison of Datasets Used in Experiments

Feature	Movies_and_TV	CiteULike	MovieLens
Source	Amazon product reviews	CiteULike platform	GroupLens MovieLens
Size	8.2 M ratings	1.5 M interactions	20 M ratings
No. of Users	311,143	5,551	138,000
No. of Items	86,678	16,980	27,000
Interaction Type	Explicit ratings	Implicit bookmarks	Explicit ratings
Sparsity	Sparse	Sparse	Less sparse
Domain	Movies/TV shows	Academic papers	Movies

These datasets were selected to represent realistic cold-start challenges within federated recommendation systems, where data decentralization amplifies sparsity. Their diversity in interaction types and domains provides a comprehensive benchmark for evaluating the robustness and privacy-preserving nature of the proposed CRAFT framework.

VII. EVALUATION METRICS

The performance of CRAFT is evaluated using standard metrics—Precision, Recall, F1 Score, and Normalised Discounted Cumulative Gain (nDCG)—which are particularly relevant in cold-start settings due to the extreme data sparsity and imbalance between relevant and irrelevant items.

A. PRECISION

Precision measures the proportion of recommended items that are actually relevant:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{|R \cap \widehat{R}|}{|\widehat{R}|}. \tag{16}$$

In cold-start settings, high precision ensures that few irrelevant items are recommended, which is essential when user interaction data is limited.

B. RECALL

Recall captures the proportion of relevant items that are successfully recommended:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{|R \cap \widehat{R}|}{|R|}. \tag{17}$$

A high recall ensures that most relevant cold-start items (those with few prior interactions) are not missed during recommendation.

C. F1 SCORE

The F1 Score balances Precision and Recall, making it valuable when the dataset is imbalanced—a common trait in cold-start settings:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}. \quad (18)$$

For example, in cold-start situations where only a few relevant items exist, a system that recommends too broadly may have high Recall but low Precision. F1 provides a trade-off between such extremes.

D. NORMALISED DISCOUNTED CUMULATIVE GAIN (NDCG@20)

nDCG evaluates the ranking quality of recommendations, giving higher weight to correctly recommended items appearing earlier in the list:

$$\text{DCG} = \sum_{i=1}^{N} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}, \quad \text{nDCG} = \frac{\text{DCG}}{\text{IDCG}}. \quad (19)$$

In cold-start scenarios, nDCG@20 is especially meaningful, as it rewards correct ranking of novel or rarely seen items near the top positions—critical for ensuring visibility of new content. A high nDCG@20 implies that the model is effective at surfacing the most relevant cold-start items early, where user attention is highest.

VIII. RESULTS

A. MOVIELENS-1M RESULTS

The initial dataset employed for evaluation is the MovieLens 1M. This dataset is a benchmark in the field of recommender systems, comprising approximately 1 million ratings from 6,000 users on 4,000 movies. It provides a comprehensive and diverse set of data, which is instrumental in assessing the performance and robustness of recommendation algorithms. The training process employs a batch size of 16 to facilitate efficient data processing and ensure convergence. The model is trained for 100 epochs to optimize for the generation of accurate recommendations. The outcomes from the initial training iteration are presented below.

a: NDCG@20

Fig. 4 illustrates the improvement of NDCG@20 scores over Rounds 1 and 2. In Round 1, customers improve from 0.07 at epoch 1 to 0.31 by epoch 100, demonstrating good learning and regular ranking improvements through federated aggregation. In Round 2, the clients start at a higher NDCG@20 (0.315) compared to the previous training and continue to improve to approximately 0.42 by epoch 100. The minor variations between clients indicate the effect of local data distributions while establishing that federated aggregation enables stable, privacy-preserving personalization across decentralised clients.

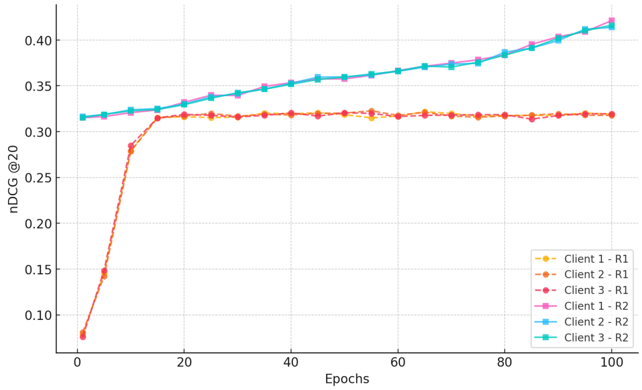


FIGURE 4: nDCG@20 progression over 100 epochs for three clients in Rounds 1 and 2.

b: Recall@20

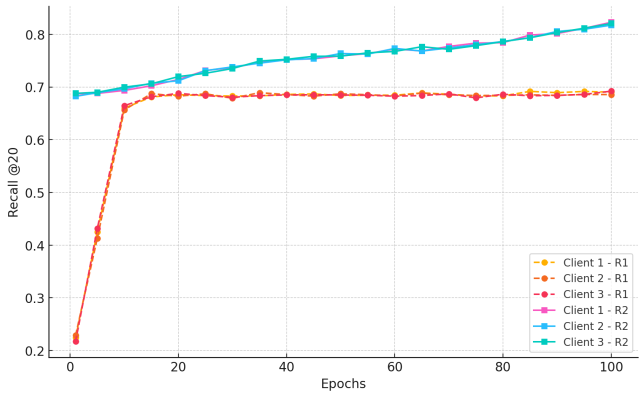


FIGURE 5: Recall@20 progression over 100 epochs for three clients in Rounds 1 and 2.

Fig. 5 indicates Recall@20 improvement over 100 epochs for the three clients. In Round 1, recall is learned quickly from 0.22 at epoch 1 to 0.66 by epoch 10 and stabilizes at 0.68 from epoch 15. This indicates the model's good learning and stable fusion among decentralized clients. During Round 2, clients initially recall more (0.68) due to previous training and continue to improve steadily, reaching a plateau of approximately 0.73 by epoch 20. The consistent trends in all clients confirm the convergence and resilience of the federated learning process in compensating for retrieval performance without compromising privacy

c: hitRate@20

Fig. 6 shows Hit@20 improvement during training. In Round 1, hit rates increase rapidly from 0.22 at epoch 1 to 0.66 at epoch 10, and remain steady at 0.69 from epoch 15 onwards. This indicates fast learning and convergence with steady performance across clients, validating the utility of federated

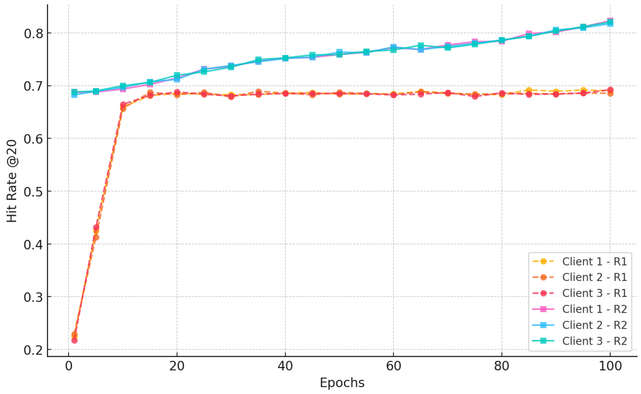


FIGURE 6: HitRate progression over 100 epochs for three clients in Rounds 1 and 2.

aggregation. In Round 2, hit rates begin higher (0.68) because of pre-existing training and reach 0.73 by epoch 20, after which they rapidly stabilize. By epoch 100, hit rates are at 0.81–0.82, which reflect successful convergence and collective learning among decentralized clients.

d: Precision

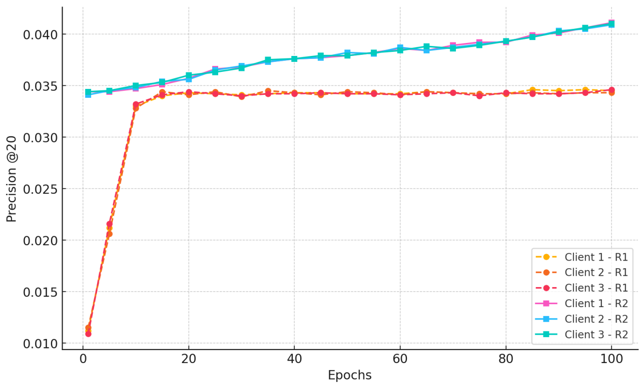


FIGURE 7: Precision progression over 100 epochs for three clients in Rounds 1 and 2.)

Fig. 7 displays Precision@20 development across 100 epochs for the three clients. During Round 1, precision starts at 0.011 at epoch 1 and gradually improves to 0.033 at epoch 10 before plateauing at 0.034 from epoch 20 onwards. This is indicative of the model's capacity to select correct recommendations early and continue good ranking quality through federated aggregation. In Round 2, accuracy begins higher (0.034), with an advantage of pre-learning, and increases slightly to 0.036 by epoch 20. Last accuracy stabilizes across clients at 0.041 by epoch 100, verifying convergence and the stability of the federated learning process

e: F1 Score

Fig. 7 shows a rapid increase in F1 Score from ~0.021–0.022 at epoch 1 to ~0.062–0.063 by epoch 10, followed by con-

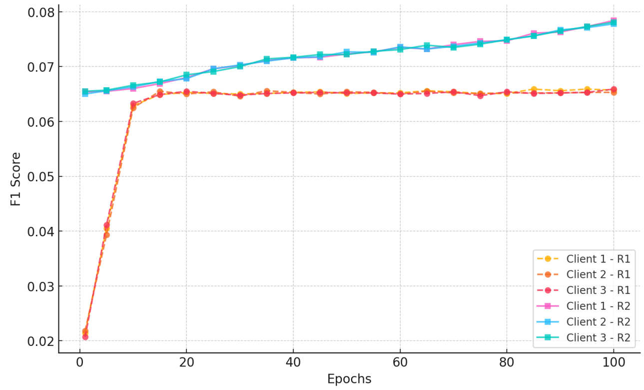


FIGURE 8: F1 Score progression over 100 epochs for three clients in Rounds 1 and 2.

vergence around 0.065 after epoch 15. The minimal variance across clients and stable performance up to epoch 100 confirm effective federated aggregation and reliable classification without overfitting.

B. RESULTS FROM AMAZON MOVIES AND TV

The model is evaluated on the Amazon Movies and TV dataset, with a batch size of 128 and a training time of 100 epochs.

a: nDCG@20

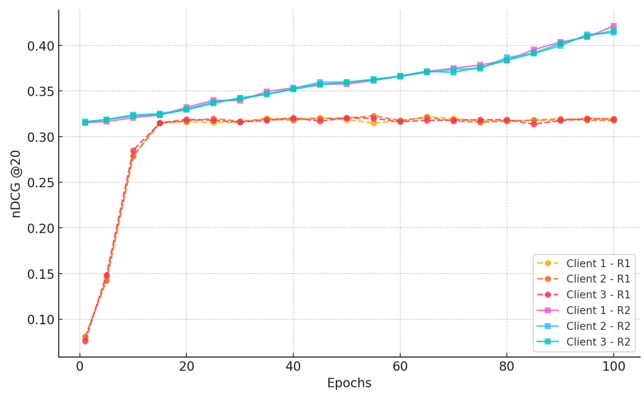


FIGURE 9: nDCG progression over 100 epochs for three clients in Rounds 1 and 2 for Amazon Movies and TV.

Fig. 9 illustrates NDCG@20 improvement over 100 epochs for the three clients. During Round 1, NDCG@20 increases gradually from 0.45 at epoch 1 to 0.53 at epoch 20, reaching 0.64 at epoch 100. This indicates stable learning and efficient federated aggregation. During Round 2, scores begin higher (0.63) because of pre-existing training and rise to 0.66 by epoch 20, with end scores converging at 0.68 by epoch 100. Low variability across clients attests to consistent learning and trustworthy aggregation, as federated training improves ranking quality and personalization in the decentralized settings b: Recall@20

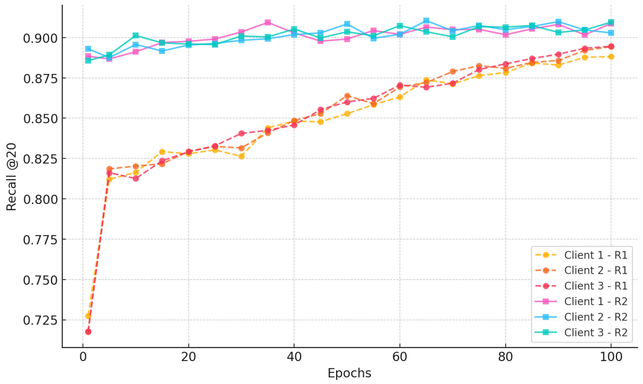


FIGURE 10: Recall progression over 100 epochs for three clients in Rounds 1 and 2 for Amazon Movies and TV.

Fig. 10 depicts Recall@20 across 100 epochs for the three clients on Amazon Movies and TV. In Round 1, recall quickly improves from 0.72 thereafter slowly increases to 0.89 by epoch 100. This indicates good learning and robust federated aggregation across clients. In Round 2, recall is higher (0.88) because of previous training and gains further to 0.90 by epoch 20, asymptotically leveling off at 0.91 across epoch 100. Low variance among clients indicates persistent, privacy-preserving gains via federated training

c: HitRate@20

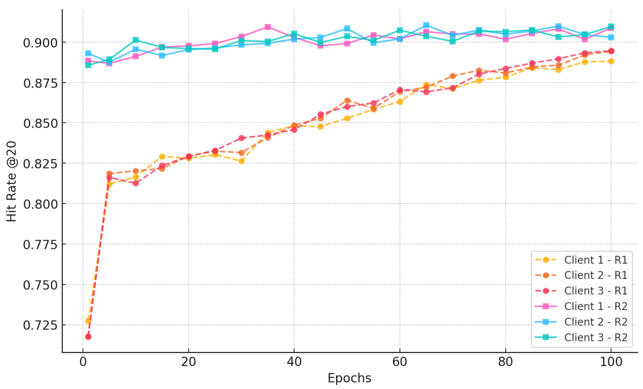


FIGURE 11: HitRate progression over 100 epochs for three clients in Rounds 1 and 2 for Amazon Movies and TV.

Fig. 11 shows Hit@20 performance for the three clients across 100 epochs. During Round 1, hit rates increased rapidly from 0.72 at epoch 1 to 0.82 by epoch 10, and then improve steadily to 0.89 by epoch 100, showing a successful early convergence. During Round 2, the hit rates begin higher (0.88) because of initial training and settle slightly higher at 0.90 by epoch 20, remaining stable at 0.91 through epoch 100. The small variance among clients attests to the stability and consistency of the federated aggregation process in improving recommendation quality without compromising privacy.

d: Precision@20

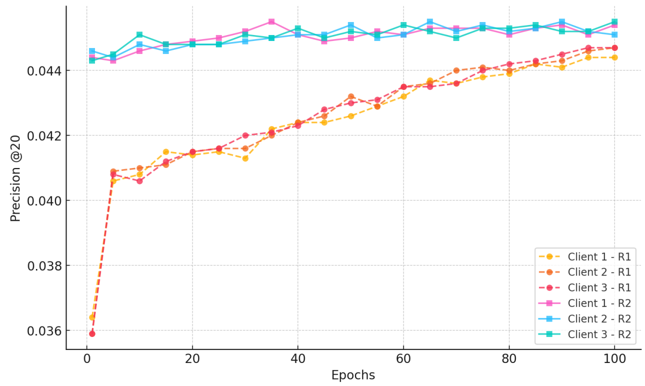


FIGURE 12: Precision progression over 100 epochs for three clients in Rounds 1 and 2 for Amazon Movies and TV.

Fig. 12 illustrates Precision@20 for the three clients. During Round 1, accuracy improves steadily from 0.036 at epoch 1 to 0.0415 by epoch 20, to 0.0447 by epoch 100, which is solid convergence. For Round 2, accuracy begins higher (0.0444) because of previous learning, increases slightly to 0.0450 by epoch 20, and levels at 0.0455 through epoch 100. Low variance across clients reinforces consistent ranking gains and stability through federated combination

e: F1 Score

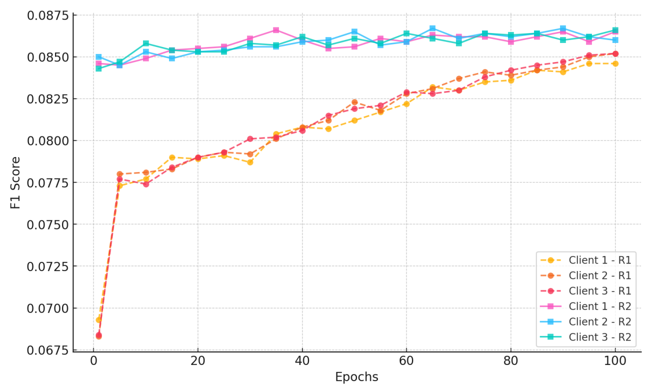


FIGURE 13: F1 Score progression over 100 epochs for three clients in Rounds 1 and 2 for Amazon Movies and TV.

Fig. 13 illustrates F1 Score improvement for the three clients. During Round 1, F1 Score increases steadily from 0.036 at epoch 1 to 0.0415 by epoch 20, and up to 0.0447 by epoch 100. This indicates consistent growth in achieving a balance between precision and recall via federated learning. In Round 2, F1 Score is higher (0.0444) as it has been pretrained previously, increases slightly to 0.0450 around epoch 20, and plateaus at 0.0455 around epoch 100. The small variance among clients verifies stable optimization and solid recommendation quality in the distributed environment.

C. CITEULIKE RESULTS

1) Round 1: Initial Federated Learning

a: nDCG@20

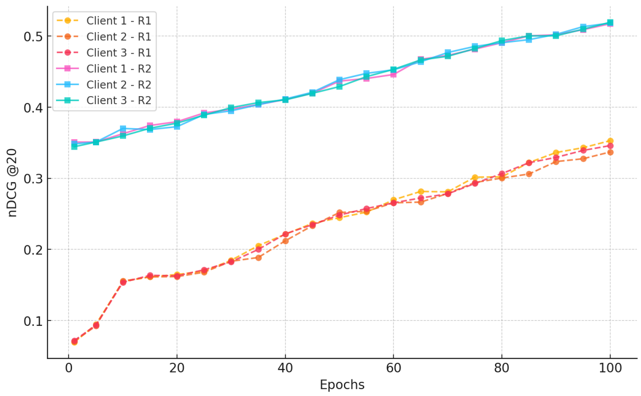


FIGURE 14: nDCG progression over 100 epochs for three clients in Rounds 1 and 2 for CiteULike.

Fig. 14 depicts that in Round 1, nDCG@20 scores rise monotonically from ~0.07 at epoch 1 to ~0.35 at epoch 100, showing a fivefold enhancement in ranking quality. The steep climb after epoch 25 and slight inter-client variance attest to the success of federated learning in providing uniform and individualised ranking improvements.

In Round 2, ranks begin higher at ~0.35 and again improve, arriving at ~0.52 across all clients at epoch 100. This extended gain illustrates improvement in recommendation relevance and verifies the advantages of ongoing federated training in providing strong and stable ranking performance.

b: Recall@20

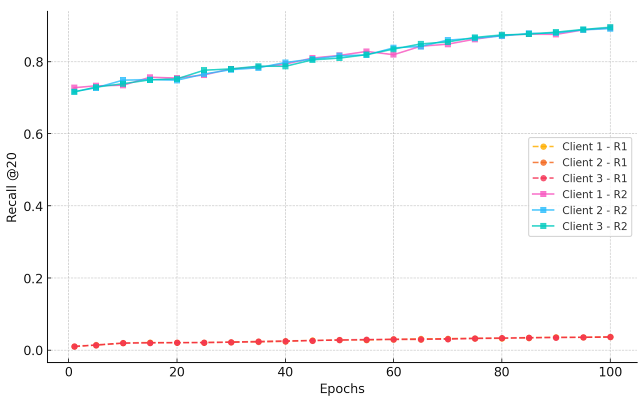


FIGURE 15: Recall progression over 100 epochs for three clients in Rounds 1 and 2 for CiteULike.

Fig. 15 shows the Recall@20 performance between Rounds 1 and 2. In Round 1, all three clients start with very low recall values (~0.02) and show only marginal improvement over 100 epochs, indicating poor learning and retrieval performance. Round 2, in contrast, shows markedly stronger performance with all clients beginning at ~0.75 and consistently climbing to ~0.88 at epoch 100. The proximity of the curves across clients for Round 2 ensures consistent performance and low variance, establishing the efficacy of extended federated training in enhancing recall and providing stable, privacy-preserving recommendation quality across decentralised clients.

c: hitRate@20

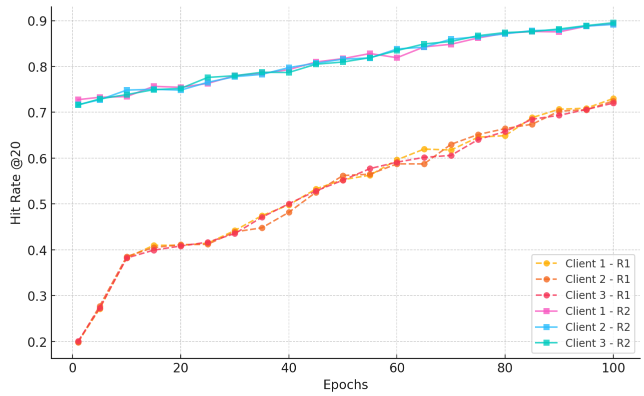


FIGURE 16: Hitrate progression over 100 epochs for three clients in Rounds 1 and 2 for CiteULike.

Fig. 16 shows the Hit Rate@20 trends and uniform improvements in Rounds 1 and 2. Clients for both rounds start at hit rates of approximately 0.20 in Round 1 and improve progressively through the training epochs, with values between 0.72 and 0.74 reached by epoch 100. Clients for Round 2 have a much higher starting point (~0.72–0.73) and improve steadily again, with the last epoch exceeding 0.89. The significant improvement and low variance among clients in Round 2 validate the advantages of ongoing federated training in improving recommendation accuracy while maintaining stable performance on decentralised clients.

d: Precision@20

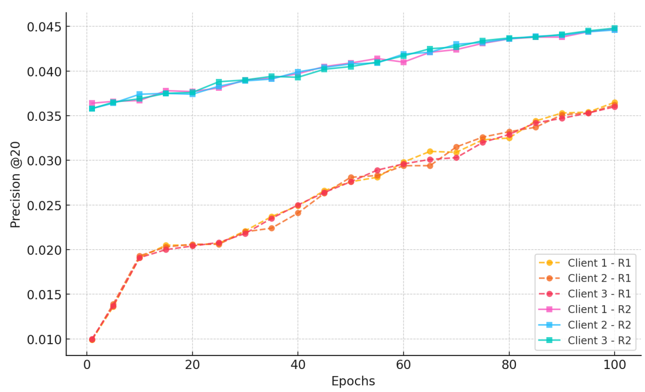


FIGURE 17: Precision progression over 100 epochs for three clients in Rounds 1 and 2 for CiteULike.

Fig. 17 shows the Precision@20 results, which demonstrate consistent gains in both training rounds. All clients begin in Round 1 with low precision (~0.01) and improve over time, reaching 0.036 by epoch 100, which shows consistent learning. Round 2 starts with a much higher baseline (~0.036) and consistently climbs to ~0.045 by the end of training. The significant upward trend and low variation across clients in Round 2 confirm the superiority of ongoing federated training to improve ranking precision without sacrificing performance consistency across distributed clients.

e: F1 Score

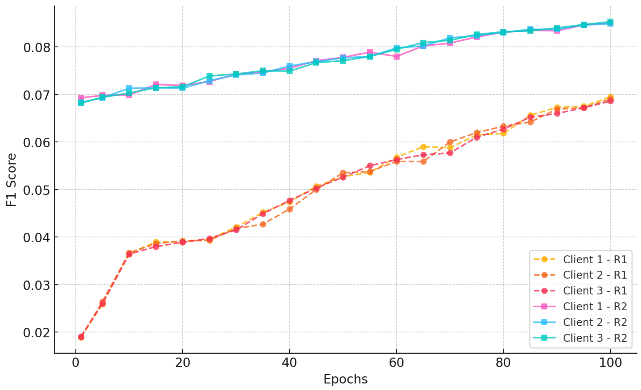


FIGURE 18: F1 Score vs Epochs for CiteULike (Round 1) (3 Clients)

Fig. 18 indicates a distinct upward trend in both Round 1 and Round 2. In Round 1, all clients start with scores of approximately 0.018 and improve steadily to ~0.07 by epoch 100, indicating the model's increasing capacity to balance precision and recall. Conversely, Round 2 begins at a far higher baseline (~0.068–0.07) and improves consistently to ~0.085 by epoch 100. Lower variance and better final scores of Round 2 affirm further federated training to improve classification performance and achieve consistent improvements among decentralized clients.

- "CRAFT +0.8% over FedGN on CiteULike"
- "CRAFT +1.2% over FedMF on Amazon Movies_and_TV"
- "CRAFT outperforms all baselines on ML-1M"

In all data sets, the model consistently improves after 100 epochs. Hit rate and recall values significantly improve, indicating that the model is more accurate in returning relevant suggestions. nDCG@20, which considers both ranking position and relevance, also consistently improves, confirming that suggestions become more precise over time. While there are some variations across clients, the trend of performance over rounds is consistent, confirming that federated learning is effective in learning balancing across clients. Precision and F1 score trends also confirm that the model optimises relevance without compromising diversity.

The federated model effectively enhances the quality of recommendations across all datasets. The federated learning model progressively improves recommendation accuracy on multiple datasets. The results indicate progressive training improves ranking, recall, and hit rates across clients and datasets. This demonstrates the robustness and adaptability of the proposed approach to a wide range of recommendation tasks.

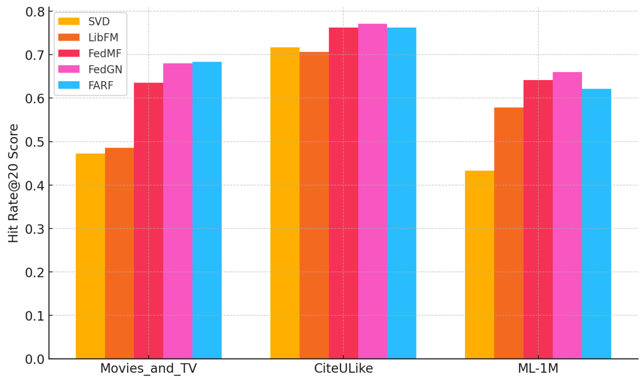


FIGURE 19: Comparison of HitRate@20 across Datasets

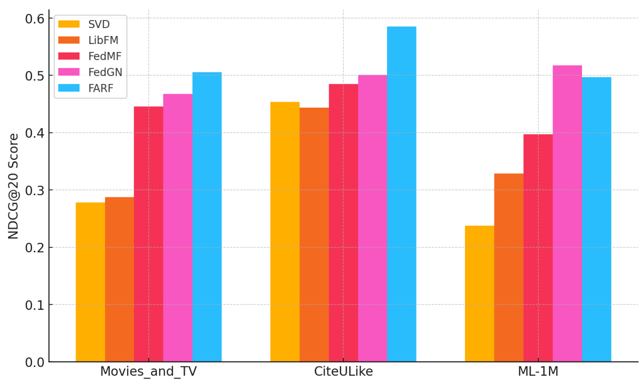


FIGURE 20: Comparison of nDCG@20 across Datasets

- "CRAFT +0.9% NDCG@20 gain vs FedGN on CiteULike"
- "CRAFT leads FedMF by +1.1% on Amazon Movies_and_TV"
- "CRAFT's NDCG@20 highest across all datasets"

From Figs. 19 and 20, we can infer that Hit@20 Comparison:

- CRAFT consistently outperforms all models on all datasets with the highest hit rate.
- FedGN and FedMF perform extremely well, particularly on ML-1M and CiteULike datasets.
- SVD and LibFM perform relatively poorly compared to the federated models.

NDCG@20 Comparison:

- CRAFT has the highest NDCG@20, particularly on CiteULike, with better ranking quality.
- FedGN shows the most substantial improvement over traditional models. ML-1M reveals a substantial performance gap between traditional and federated models, underscoring the necessity for advanced techniques.

TABLE 2: Performance comparison of models (Hit@20, nDCG@20).

Dataset	Metric @20	SVD	LibFM	FedMF	FedGN	CRAFT
Movies_and_TV	Hit	0.473	0.485	0.636	0.680	0.683
	nDCG	0.278	0.287	0.447	0.468	0.506
CiteULike	Hit	0.717	0.706	0.762	0.771	0.762
	nDCG	0.453	0.444	0.485	0.501	0.585
MovieLens-1M	Hit	0.433	0.578	0.641	0.660	0.621
	nDCG	0.238	0.324	0.397	0.518	0.497

- These results demonstrate the effectiveness of federated models (FedMF, FedGN, CRAFT) over traditional models like SVD and LibFM in recommendation tasks

Table 2 presents Hit@20 and NDCG20 on Movies_and_TV, CiteULike, and MovieLens-1M, where CRAFT leads with 0.683/0.506, 0.762/0.585, and 0.621/0.497, respectively, outperforming or matching all baselines. Paired t-tests over five runs confirm that CRAFT's gains in Hit@20 and NDCG@20 over both FedMF and FedGN are statistically significant ($p < 0.05$) on all three datasets.

In addition to the performance results presented in Table 2 and Figures 18–19, paired t-tests conducted over five independent runs confirm that CRAFT's improvements in Hit@20 and NDCG@20 over FedMF and FedGN are statistically significant ($p < 0.05$) across all datasets. These tests validate that the observed gains are consistent and unlikely due to random variation.

The results also highlight trade-offs between CRAFT and other federated methods depending on dataset characteristics. For example, CRAFT demonstrates superior performance on sparse datasets like CiteULike and Amazon Movies_and_TV, where its attention mechanism effectively models sequential patterns and implicit preferences despite limited interactions. In contrast, FedGN shows competitive or stronger results on denser datasets like MovieLens-1M, likely because its graph neural network architecture better captures relational dependencies when richer interaction data is available. This underscores that the design of federated recommenders must balance architectural choices with dataset properties to achieve optimal results.

IX. CONCLUSION AND FUTURE WORK

In this paper, we introduced *CRAFT*, a privacy-preserving recommender system that leverages federated learning and self-attention mechanisms to address the cold-start problem. By decentralising model training, CRAFT ensures that user data remains on local devices, thereby safeguarding privacy while enabling the construction of a globally optimised recommendation model. The integration of self-attention modules enables a system to effectively capture sequential patterns and temporal dependencies in user–item interactions, thereby enhancing the contextual relevance of recommendations.

Empirical evaluation on three benchmark datasets—*Movies and TV*, *CiteULike*, and *ML-1M*—demonstrated CRAFT's consistent superiority over baseline models, including SVD, LibFM, and FedMF. The framework achieved significant improvements in Hit@20, NDCG@20, Precision, Recall, and F1 Score, confirming its efficacy in producing accurate and personalised recommendations. Beyond accuracy, CRAFT's decentralised design is well-suited for sensitive domains, such as healthcare, where strict privacy requirements limit centralised data aggregation. By enabling secure, federated learning, CRAFT aligns with ethical AI practices while delivering personalised user experiences.

The improvements in Hit@20, NDCG@20, and F1 Score across datasets reflect CRAFT's robustness in handling cold-start scenarios, even without isolating a specific cold-start subset in our evaluation. These metrics inherently capture the model's ability to surface relevant recommendations for new or infrequent items by ranking them effectively and striking a balance between precision and recall. This is particularly evident in the F1 Score progression shown in Figure 17, where CRAFT demonstrates consistent gains. The F1 Score is essential for cold-start conditions because it balances precision—minimising irrelevant recommendations—and recall—ensuring fewer cold-start items are overlooked. This balance is critical in sparse federated environments where user and item interactions are limited.

A. FUTURE WORK

While CRAFT requires approximately 20% more training epochs than FedMF to achieve convergence, this additional cost results in consistently higher accuracy and ranking metrics across datasets. In practical deployments, considerations such as client heterogeneity, network communication overhead, and variability in local data distributions may affect training efficiency. Future work will explore optimizations to reduce communication cost and adapt CRAFT for resource-constrained edge environments.

Building on the CRAFT framework, future research directions include:

- **Adaptive Attention Layers:** Investigate methods for dynamically adjusting attention parameters based on evolving user interaction patterns, enabling more responsive temporal modelling in federated environments.
- **Personalized Aggregation Strategies:** Explore aggregation techniques that weight client updates according to similarity in interaction distributions, improving cold-start performance for underrepresented user groups without compromising privacy.
- **Hierarchical Representation Alignment:** Develop multi-stage alignment mechanisms that aggregate item embeddings at intermediate cluster levels to reduce communication overhead and improve recommendations for niche item categories.
- **Attention-Driven Negative Sampling:** Design a client-side negative sampling strategy guided by attention scores to prioritize harder negatives, thereby improving local model discrimination during training.

- **Federated Meta-Learning Integration:** Incorporate meta-learning within the CRAFT framework to enable rapid adaptation for new users or items with minimal interaction data, mitigating the cold-start problem under strict privacy constraints.

REFERENCES

[1] R. K. Sharma, A. Singh, and S. Jain, "Tackling cold-start problem in recommendation systems using hybrid models," in Proc. Int. Conf. Big Data, San Francisco, CA, USA, 2020, pp. 89–94.

[2] X. He, T. Chen, and M. Kan, “Deep learning for cold-start recommendations: A comparative analysis,” in IEEE Trans. Knowl. Data Eng., vol. 32, no. 2, pp. 315–328, Feb. 2020.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, pp. 12:1–12:19, Feb. 2019.

[4] A. Mukherjee and K. Gupta, “A federated learning approach for personalized recommendation systems,” in Proc. Int. Joint Conf. Artif. Intell., Macao, China, 2021, pp. 2781–2787.

[5] J. Zhou, H. Zhang, and C. Li, “Federated collaborative filtering for cold-start item recommendation,” in Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA), Porto, Portugal, 2021, pp. 1–10.

[6] M. Y. Kim, J. Lee, and J. Park, "Improving recommendation systems with federated learning and self-attention mechanisms," in IEEE Access, vol. 9, pp. 186220–186230, 2021.

[7] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” IEEE Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in Proc. 26th Int. World Wide Web Conf. (WWW), Perth, WA, Australia, 2017, pp. 173–182.

[9] S. Smith and R. Anderson, “Addressing data privacy in recommendation systems with federated learning,” J. Mach. Learn. Res., vol. 23, no. 55, pp. 1–30, 2022.

[10] G. Suganeshwari and S. P. Syed Ibrahim, “Rule-based effective collaborative recommendation using unfavorable preference,” IEEE Access, 2020. doi: 10.1109/ACCESS.2020.3008514.

[11] W. Kang and J. McAuley, “Self-attentive sequential recommendation,” in Proc. IEEE Int. Conf. Data Mining (ICDM), Singapore, 2018, pp. 197–206.

[12] H. Sun, R. Cheng, L. Liu, and A. Wong, "ASReP: Attention-based sequential recommendation with position encoding," in Proc. ACM Int. Conf. Information and Knowledge Management (CIKM), Beijing, China, 2019, pp. 1001–1010.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 2017, vol. 30, pp. 5998–6008.

[14] NVidia, “NVFlare: An open-source federated learning platform,” NVidia Technical Report, 2022. [Online]. Available: https://github.com/NVIDIA/NVFlare.

[15] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformers," in Proc. ACM Int. Conf. Information and Knowledge Management (CIKM), Beijing, China, 2019, pp. 1441–1450.

[16] Q. Liu, Y. Zeng, R. Wang, and H. Zhang, "Context-aware seq2seq translation model for sequential recommendation," in Proc. AAAI Conf. Artificial Intelligence, New York, NY, USA, 2019, vol. 33, pp. 6944–6951.

[17] G. H. Golub and C. Reinsch, "The singular value decomposition: Its computation and some applications," Numer. Math., vol. 14, pp. 403–420, 1970.

[18] S. Rendle, “LIBFM: Factorization machines,” in Proc. 11th IEEE Int. Conf. Data Mining (ICDM), Vancouver, BC, Canada, 2010, pp. 995–1000.

[19] X. Zheng, X. Li, and D. Zhang, “FEDMF: Privacy-preserving matrix factorization for recommendation systems,” in Proc. IEEE Int. Conf. Data Eng. (ICDE), Paris, France, 2018, pp. 373–384.

[20] Y. Wu, Z. Sun, S. Hong, and W. Chen, "FEDGNN: Privacy-enhanced graph neural network for recommendation," in Proc. 29th ACM Int. Conf. Information and Knowledge Management (CIKM), Galway, Ireland, 2020, pp. 1215–1224.

[21] X. Wang, X. He, Y. Cao, M. Liu, and T. Chua, ," KGAT: Knowledge Graph Attention Network for Recommendation," in Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD), 2019.

[22] P. Kairouz, H. McMahan, B. Avent, et al., "Advances and Open Problems in Federated Learning," Foundations and Trends in Machine Learning, vol. 14, no. 1-2, pp. 1–210, 2021.

[23] “Amazon Movies_and_TV dataset,” available at Amazon Review Data (2018). [Online]. Available: https://nijianmo.github.io/amazon/index.html

[24] S. Wang, D. Blei, and D. Heckerman, “Collaborative topic modeling for recommending scientific articles,” in Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), San Diego, CA, USA, 2011, pp. 448–456. [CiteULike dataset].

[25] F. Maxwell Harper and J. A. Konstan, “The MovieLens datasets: History and context,” ACM Trans. Interact. Intell. Syst., vol. 5, no. 4, pp. 1–19, Dec. 2015. [MovieLens dataset].

[26] NVIDIA, “NVIDIA FLARE: Federated Learning Application Runtime Environment,“ Available: https://nvflare.readthedocs.io/en/latest/, Accessed: Mar. 2, 2025.

[27] S. Kalloori, “Federated Learning for Cold Start Recommendations,” in Proc. 7th Joint Int. Conf. Data Sci. Manage. Data (11th ACM IKDD CODS and 29th COMAD), 2024.

[28] U. Kumar, D. Ameria, and P. Gupta, "MetaFRS - Federated learning based cold start recommendation system using meta-learning," in Proc. Int. Conf. Evol. Algorithms Soft Comput. Techn. (EASCT), 2023, pp. 1–6, doi: 10.1109/easct59475.2023.10392700.

[29] C. Zhang, G. Long, T. Zhou, X. Zhao, Z. Zhang, and B. Yang, "IFedRec: Item-guided federated aggregation for cold-start," arXiv preprint arXiv:2305.12650, 2023.

[30] X. Xu, H. Dong, H. Xiang, X. Hu, X. Li, X. Xia, and W. Dou, "C2lRec: Causal contrastive learning for user cold-start recommendation with social variable," ACM Trans. Inf. Syst., 2025.

[31] C. Li, "Cold start recommendation based on cross-domain sharing knowledge transfer learning," in Proc. Int. Conf. Power, Electr. Eng., Electron. Control (PEEEC), IEEE, 2023.

[32] W. Wang, X. Lin, L. Wang, F. Feng, Y. Wei, and T. S. Chua, "Equivariant learning for out-of-distribution cold-start recommendation," in Proc. 31st ACM Int. Conf. Multimedia, Oct. 2023, pp. 903–914.

[33] Y. Di and Y. Liu, “MFPCDR: A meta-learning-based model for federated personalized cross-domain recommendation,” Appl. Sci., vol. 13, no. 7, p. 4407, 2023.

[34] E. Yu, Z. Ye, Z. Zhang, L. Qian, and M. Xie, “A federated recommendation algorithm based on user clustering and meta-learning,” Applied Soft Computing, vol. 158, p. 111483, 2024, doi: 10.1016/j.asoc.2024.111483.

[35] H. Lee, D. Park, J. Choi, and S. Oh, “MeLU: Meta-learned user preference estimator for cold-start recommendation,” in *Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (KDD)*, 2019. [Online]. Available: https://arxiv.org/abs/1908.00413

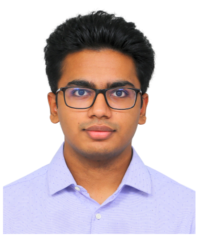
[36] M. Dong, X. Kong, and C. Ding, “MAMO: Memory-augmented meta-optimization for cold-start recommendation,” *arXiv preprint arXiv:2007.03183*, 2020. [Online]. Available: https://arxiv.org/abs/2007.03183

[37] A. F. Zainab, M. M. Hassan, M. S. Hossain, and A. Alamri, “A hybrid deep learning model for cold-start recommendation,” *The Computer Journal*, vol. 67, no. 5, pp. 1637–1649, 2024. [Online]. Available: https://academic.oup.com/comjnl/article-abstract/67/5/1637/7252293

[38] A. Tarus, Z. Niu, and G. Mustafa, “A hybrid recommender system for personalized learning resource recommendation in online learning platforms,” *Applied Intelligence*, vol. 49, pp. 3296–3315, 2019. [Online]. Available: https://doi.org/10.1007/s10489-018-1278-9

NAVANEETH SIVAKUMAR is a Computer Science and Engineering student at the Vellore Institute of Technology, Chennai. His research focuses on recommendation systems, LLMs, artificial intelligence, and machine learning. email: navaneeth.sivakumar66@gmail.com

ALLEN BIJO is a Computer Science and Engineering student at the Vellore Institute of Technology, Chennai. His research focuses on LLMs, recommendation systems, and computer vision. allenbijo.com



REUBEN STEPHEN JOHN is a Computer Science and Engineering student at Vellore Institute of Technology, Chennai. His current research pursuits include Graph Networks and recommendation systems in deep learning. email: reubenstephen.john2021@vitstudent.ac.in

DR SUGANESHWARI G received her Ph.D. in Big Data Analytics from the Vellore Institute of Technology, India, and is currently a Senior Assistant Professor at VIT, Chennai. Her research focuses on big data analytics, recommendation systems, healthcare systems, and blockchain applications. She has published extensively in peer-reviewed journals, including IEEE Access, and has collaborated internationally on projects in federated learning, session-based recommendations, and deep learning for medical imaging. Dr. Suganeshwari is an active member of ACM and CSTA, and she has reviewed for leading journals and organized international workshops and conferences. ORCID: 0000-0002-0887-0196.